Altova XMLSpy 2023 Enterprise Edition



Manuel de l'utilisateur et de référence

Altova XMLSpy 2023 Enterprise Edition
Manuel de l'utilisateur et de référence

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Published: 2022

© 2016-2022 Altova GmbH

Table des matières

1 À propos de XMLSpy et de cette documentation 28

1.1 Nouvelles fonctions 2022 ... 29
1.1.1 Version 2022 ... 29
1.1.2 Version 2021 ... 30
1.1.3 Version 2020 ... 31
1.1.4 Versions 2019 ... 32
1.2 Windows File Paths ... 34
1.3 About RaptorXML Server ... 35

2 Tutoriel XMLSpy 36

2.1 Interface XMLSpy ... 37
2.1.1 Les modes ... 38
2.1.2 Les fenêtres ... 39
2.1.3 Menus et barres d'outils ... 42
2.1.4 Paramètres du Mode Texte ... 44
2.1.5 Options d'application ... 46
2.2 Schémas XML : fonctions de base ... 49
2.2.1 Créer un nouveau fichier de Schéma XML ... 49
2.2.2 Définir les espaces de noms ... 51
2.2.3 Définir un modèle de contenu ... 52
2.2.4 Ajouter des éléments par glisser-déposer ... 57
2.2.5 Configurer le Mode Modèle de contenu ... 58
2.2.6 Terminer le Schéma de base ... 60
2.3 Schémas XML : Avancé ... 64
2.3.1 Travailler avec des types complexes et des types simples ... 64
2.3.2 Référencer des éléments globaux ... 72
2.3.3 Attributs et énumérations d'attributs ... 74
2.4 Schémas XML : fonctions XMLSpy ... 77
2.4.1 Navigation de schéma ... 77

2.4.2 Documentation de schéma ... 79
2.5 Documents XML ... 84
2.5.1 Créer un nouveau fichier XML ... 84
2.5.2 Spécifier le type d'un élément ... 86
2.5.3 Saisir les données dans le Mode Grille ... 88
2.5.4 Saisir les données dans le Mode Texte ... 89
2.5.5 Valider le document ... 94
2.5.6 Ajouter des éléments et des attributs ... 98
2.5.7 Édition dans le Mode Base de données/Table ... 100
2.5.8 Modifier le Schéma ... 103
2.6 Transformations XSLT ... 105
2.6.1 Attribuer un fichier XSLT ... 105
2.6.2 Transformer le fichier XML ... 106
2.6.3 Modifier le fichier XSL ... 107
2.7 Gestion de projet ... 110
2.7.1 Les avantages des projets ... 110
2.7.2 Construire un projet ... 110
2.8 Et voilà ! ... 113

3 GUI et environnement 114

3.1 L'Interface graphique utilisateur (GUI) ... 115
3.1.1 Fenêtre principale ... 116
3.1.2 Fenêtre Projet ... 118
3.1.3 Fenêtre Info ... 120
3.1.4 Assistants de saisie ... 121
3.1.5 Fenêtre de sortie : Messages ... 122
3.1.6 Fenêtre de sortie : XPath/XQuery ... 124
3.1.7 Fenêtre de sortie : XSL Outline ... 125
3.1.8 Fenêtre de sortie : HTTP ... 126
3.1.9 Fenêtre de sortie : Trouver dans fichiers ... 126
3.1.10 Fenêtre de sortie : Trouver dans schémas ... 128
3.1.11 Fenêtre de sortie : Trouver dans XBRL ... 129
3.1.12 Fenêtre de sortie : Graphiques ... 129
3.1.13 Fenêtre de sortie : XULE ... 131

3.1.14 Menu Bar, Toolbars, Status Bar (2) ... 132
3.2 L'environnement d'application ... 134
3.2.1 Paramètres et personnalisation ... 134
3.2.2 Tutoriels, Projets, Exemples ... 137
3.2.3 Fonctions et Aide Authentic Desktop et les produits Altova ... 137

4 Modes d'édition 139

4.1 Sauvegarde automatique des fichiers ... 141
4.2 Mode Texte ... 143
4.2.1 Formatage dans le Mode Texte ... 144
4.2.2 Afficher le document ... 146
4.2.3 Édition dans le Mode Texte ... 149
4.2.4 Parcourir le document ... 152
4.2.5 Chercher et remplacer ... 155
4.2.6 Valider un document XML ... 162
4.2.7 Assistants de saisie dans le Mode Texte ... 164
4.2.8 Mode Écran divisé ... 165
4.2.9 Raccourcis Mode texte ... 167
4.3 Mode Grille ... 168
4.4 Mode Schéma ... 170
4.4.1 Mode XSD : XSD 1.0 ou 1.1 ... 172
4.4.2 Aperçu de schéma ... 176
4.4.3 Mode de Modèle de contenu ... 188
4.4.4 Attributs, Assertions et Contraintes d'identité ... 210
4.4.5 Assistants à la saisie dans le Mode Schéma ... 226
4.4.6 Validation et Smart Fixes ... 236
4.4.7 Messages d'assertion ... 238
4.4.8 Modification de Type de Base ... 240
4.4.9 Smart Restrictions ... 242
4.4.10 xml:base, xml:id, xml:lang, xml:space ... 246
4.4.11 Avant et Arrière : se déplacer d'une position à une autre ... 248
4.5 Mode WSDL ... 250
4.5.1 Fenêtre principale ... 251
4.5.2 Assistant à la saisie Vue d'ensemble ... 255

4.5.3 Assistant à la saisie Détails ... 262
4.6 Mode XBRL ... 263
4.6.1 Fenêtre principale : onglet Éléments ... 263
4.6.2 Fenêtre principale : onglets Définitions, Présentation, Calcul, Formule, Table ... 268
4.6.3 Assistants à la saisie dans le Mode XBRL ... 271
4.6.4 Paramètres du mode XBRL ... 277
4.7 Mode Authentic ... 278
4.8 Mode Navigateur ... 279
4.9 Mode Archive ... 281
4.10 Raccourcis communs ... 283

5 XML 285

5.1 Créer, ouvrir et enregistrer des documents XML ... 286
5.2 Attribuer des schémas et valider ... 288
5.3 XML dans le Mode Texte ... 290
5.4 XML dans le Mode Grid ... 293
5.4.1 Affichage du document ... 295
5.4.2 Structure du document ... 299
5.4.3 Document Content ... 302
5.4.4 Glisser-déposer dans le Mode Grille ... 306
5.4.5 Affichage de la table ... 309
5.4.6 Filtres ... 314
5.4.7 Formules ... 316
5.4.8 Images ... 319
5.4.9 Graphiques ... 320
5.4.10 Menu contextuel dans le Mode Grille ... 323
5.4.11 Paramètres du Mode Grilleiew Settings ... 326
5.5 XML dans le Mode Authentic ... 332
5.6 Assistants à la saisie (Mode Texte, Mode Authentic) ... 334
5.7 Valider un document XML ... 336
5.8 Whitespace ... 338
5.9 Insérer des fragments XML ... 340
5.10 Traiter un document avec XSLT et XQuery ... 342

5.11 Polices PDF ... 344
5.12 Graphiques ... 347
5.12.1 Créer un graphique ... 350
5.12.2 XPath de source ... 354
5.12.3 Sélection de l'Axe X ... 358
5.12.4 Sélection de l'Axe Y ... 363
5.12.5 Données de graphique ... 367
5.12.6 Superpositions ... 369
5.12.7 Configurations du graphique : référence rapide ... 370
5.12.8 Paramètres et apparence des graphiques ... 374
5.12.9 Exporter ... 399
5.12.10 Exemple de graphique : simple ... 400
5.12.11 Exemple de graphique : avancé ... 402
5.12.12 Exemple de graphique : chandelier ... 408
5.13 Signatures XML ... 412
5.13.1 Créer des signatures XML ... 414
5.13.2 Vérifier des Signatures XML ... 417
5.13.3 Travailler avec des certificats ... 420
5.14 Fonctions supplémentaires ... 425

6 DTD et Schémas XML 427

6.1 Gestionnaire de schéma ... 428
6.1.1 Exécuter Schema Manager ... 431
6.1.2 Catégories de statut ... 434
6.1.3 Retoucher ou Installer un schéma ... 436
6.1.4 Désinstaller un schéma, Réinitialiser ... 437
6.1.5 Interface de ligne de commande (CLI) ... 438
6.2 DTD ... 445
6.3 Schémas XML ... 447
6.4 Sous-ensembles de Schéma ... 448
6.5 Règles de schéma ... 452
6.5.1 Gérer les Ensembles de règles ... 452
6.5.2 Définir un Ensemble de règles ... 454
6.6 Catalogues dans XMLSpy ... 459

6.6.1 Comment fonctionnent les catalogues ... 459
6.6.2 Structure du catalogue dans XMLSpy ... 460
6.6.3 Personnaliser vos catalogues ... 462
6.6.4 Variables d'Environnement ... 463
6.7 Travailler avec SchemaAgent ... 465
6.7.1 Se connecter à SchemaAgent Server ... 466
6.7.2 Ouvrir des Schémas trouvés dans le Chemin de recherche ... 469
6.7.3 Utiliser des IIRs ... 469
6.7.4 Consulter des schémas dans SchemaAgent ... 474
6.7.5 Validation de SchemaAgent ... 474
6.8 Chercher dans les schémas ... 477
6.8.1 Terme de recherche ... 479
6.8.2 Composants ... 480
6.8.3 Propriétés ... 482
6.8.4 Étendue ... 485
6.8.5 Commandes Recherche et Remplacer ... 486
6.8.6 Résultats et informations ... 488
6.8.7 Chercher et renommer les Globaux ... 489

7 XSLT 491

7.1 Documents XSLT ... 492
7.2 Traitement XSLT ... 494
7.3 XSL Outline ... 497
7.3.1 Fenêtre XSL Outline ... 498
7.3.2 Fenêtre Info ... 502
7.4 Optimiseur vitesse XSL ... 505

8 XQuery 507

8.1 Éditer les documents XQuery ... 509
8.1.1 Documents XQuery ... 510
8.1.2 Assistants à la saisie XQuery ... 511
8.1.3 Coloration de la syntaxe XQuery ... 511
8.1.4 Édition intelligente XQuery ... 513

8.2 Évaluation XQuery ... 516
8.3 Validation XQuery ... 517
8.4 Exécution XQuery/Update ... 518
8.5 XQuery Update Facility ... 521
8.5.1 Prévisualiser et appliquer XQuery Updates ... 521
8.5.2 Mise à jour des opérations et de la syntaxe ... 524
8.6 XQuery et bases de données XML ... 528

9 XSLT/XQuery débogueur et profileur 532

9.1 Débogueur XSLT et XQuery ... 533
9.1.1 Mécanisme et interface ... 534
9.1.2 Commandes et icônes de barre d'outils ... 536
9.1.3 Paramètres du débogueur ... 540
9.1.4 Démarrer une session de débogage ... 541
9.1.5 Fenêtres d'Information ... 543
9.1.6 Points d'arrêt ... 550
9.1.7 Points de trace ... 553
9.1.8 Raccourcis débogueur ... 559
9.2 Profileur XSLT et XQuery ... 560
9.2.1 Profilage XSLT ... 565
9.2.2 Profilage XQuery ... 567
9.2.3 Résultats du profileur : exportations et graphiques ... 571

10 Expressions XPath/XQuery 574

10.1 À propos de la fenêtre XPath/XQuery ... 575
10.2 Évaluer l'expression ... 578
10.3 Déboguer l'expression ... 584
10.4 Générateur d'expression ... 592
10.5 Expressions XQuery pour JSON ... 595
10.6 Points importants ... 598

11 Authentic 600

11.1 Tutoriel Mode Authentic ... 602

11.1.1 Ouvrir un document XML dans le Mode Authentic ... 603
11.1.2 L'interface du Mode Authentic ... 604
11.1.3 Opérations de nœud ... 607
11.1.4 Saisir des données dans le Mode Authentic ... 610
11.1.5 Saisir des valeurs d'attribut ... 612
11.1.6 Ajouter des entités ... 613
11.1.7 Imprimer le document ... 614
11.2 Interface Mode Authentic ... 616
11.2.1 Aperçu de la GUI ... 616
11.2.2 Icônes de la barre outils du Mode Authentic ... 618
11.2.3 Fenêtre principale du Mode Authentic ... 620
11.2.4 Assistants à la saisie du Mode Authentic ... 623
11.2.5 Menus contextuels Mode Authentic ... 627
11.3 Éditer dans le Mode Authentic ... 630
11.3.1 Édition de base ... 630
11.3.2 Tables dans le Mode Authentic ... 635
11.3.3 Éditer une BD ... 643
11.3.4 Travailler avec des dates ... 650
11.3.5 Définir les entités ... 652
11.3.6 Signatures XML ... 654
11.3.7 Images dans le Mode Authentic ... 655
11.3.8 Séquences de touche dans le Mode Authentic ... 656
11.4 Authentic Scripting ... 657

12 HTML et CSS 659

12.1 HTML ... 660
12.2 CSS ... 662

13 JSON, Schéma JSON 668

13.1 Données JSON ... 671
13.2 Schéma JSON ... 674
13.3 JSON Lines and JSON Comments ... 676
13.4 Mode Texte JSON ... 677

13.5 Mode JSON Grille ... 682
13.5.1 Affichage du document dans le Mode Grille ... 685
13.5.2 Éditer la structure du document ... 689
13.5.3 Éditer le contenu du document ... 690
13.5.4 Glisser-déposer dans le Mode Grille ... 693
13.5.5 Affichage table ... 696
13.5.6 Filtres ... 701
13.5.7 Formules ... 702
13.5.8 Images ... 706
13.5.9 Graphiques ... 707
13.5.10 Menu contextuel dans le Mode Grille ... 710
13.5.11 Paramètres du Mode Grille ... 713
13.6 Mode Schéma JSON ... 718
13.6.1 Version de Schéma JSON ... 719
13.6.2 Ajouter des définitions globales ... 721
13.6.3 Assistants à la saisie : Aperçu, Détails, Contraintes ... 722
13.6.4 Définitions Globales et Locales ... 725
13.6.5 Mode Design ... 727
13.6.6 Objets et propriétés ... 729
13.6.7 Propriétés non spécifiées ... 734
13.6.8 Objets et dépendances ... 737
13.6.9 Tableaux ... 741
13.6.10 Types Atomiques ... 743
13.6.11 Sélecteurs de type (Any, Multiple) ... 745
13.6.12 BSON (Binary JSON) for MongoDB ... 747
13.6.13 Opérateurs ... 751
13.6.14 Conditionnels ... 753
13.6.15 Configurer l'affichage Design ... 755
13.6.16 Générer une documentation de schéma JSON ... 756
13.7 Valider les documents JSON ... 759
13.8 Inserting JSON Fragments ... 761
13.9 Transformations JSON avec XSLT/XQuery ... 763
13.10 XQuery Expressions for JSON ... 766
13.11 Générer un schéma JSON depuis une instance JSON ... 768
13.12 Générer une instance JSON depuis un schéma JSON ... 771

13.13 Convertir entre JSON et XML ... 772

14 Avro, Schéma Avro 773

14.1 Schéma Avro ... 776
14.2 Données Avro en format JSON ... 779
14.3 Mode Avro : un Mode Grille des binaires Avro ... 780

15 WSDL et SOAP 782

15.1 Tutoriel WSDL ... 783
15.1.1 Créer un nouveau document ... 783
15.1.2 Créer un PortType ... 784
15.1.3 Créer une liaison ... 786
15.1.4 Créer un service et des ports ... 788
15.1.5 Valider le document WSDL ... 789
15.1.6 Se connecter à un service web et ouvrir des fichiers ... 789
15.1.7 Envoyer une requête SOAP depuis le fichier WSDL ... 791
15.1.8 Créer une Documentation WSDL ... 792
15.1.9 Convertir en WSDL 2.0 ... 794
15.2 SOAP ... 795
15.2.1 Validation SOAP ... 795
15.2.2 Débogueur SOAP ... 796

16 HTTP 812

16.1 Envoyer la requête ... 813
16.2 Importer une requête à envoyer ... 819
16.3 Recevoir la réponse ... 821

17 XBRL 825

17.1 Gestionnaire de taxonomie ... 826
17.1.1 Exécuter le Gestionnaire de taxonomie ... 830
17.1.2 Catégories de statut ... 832
17.1.3 Retoucher ou Installer une taxonomie ... 833

17.1.4 Désinstaller une taxonomie, Réinitialiser ... 834
17.1.5 Interface de ligne de commande (CLI) ... 835
17.2 Procédures de base ... 843
17.2.1 Taxonomies : nouvelles et existantes ... 843
17.2.2 Aperçu des fichiers de taxonomie ... 844
17.2.3 Créer une nouvelle taxonomie ... 847
17.2.4 Importer une taxonomie ... 848
17.2.5 Espace de noms ... 852
17.2.6 Fichiers de taxonomie ... 854
17.2.7 Ajouter des éléments à une taxonomie ... 857
17.2.8 Relations et rôles de lien ... 860
17.2.9 Créer des relations : partie 1 ... 862
17.2.10 Créer des relations : partie 2 ... 865
17.3 Procédures supplémentaires ... 868
17.3.1 Libellés préférés ... 868
17.3.2 Typed Domains ... 870
17.3.3 Duplicate Detection and De-Duplication ... 871
17.3.4 Inline XBRL ... 872
17.4 Éditeur Formule XBRL ... 873
17.4.1 Bases de lien et Rôles de lien de formule ... 874
17.4.2 Composants de formule ... 875
17.4.3 Éditer les propriétés et contenus de composant ... 892
17.4.4 Relations de composant de formule ... 894
17.4.5 Paramètres de formule ... 896
17.4.6 Trouver des composants de formule ... 897
17.5 Éditeur de définitions de table XBRL ... 899
17.5.1 Bases de lien et Rôles de lien de table ... 901
17.5.2 Structure de table ... 903
17.5.3 Composants de table ... 917
17.5.4 Éditer les propriétés et le contenu de composant ... 922
17.5.5 Relations de composant de table ... 923
17.5.6 Paramètres de table ... 924
17.5.7 Aperçu de la mise en page de table ... 928
17.5.8 Trouver des composants de table ... 931
17.6 XULE ... 933

17.6.1 Documents XULE ... 934
17.6.2 Fenêtre XULE ... 936
17.6.3 Exécution XULE ... 940
17.7 Chercher dans XBRL ... 941
17.7.1 Terme de Recherche ... 941
17.7.2 Exécution de la commande ... 944
17.7.3 Résultats et informations ... 946
17.8 OIM ... 948
17.9 Valider les instances XBRL et les taxonomies ... 949

18 Office Open XML, ZIP, EPUB 950

18.1 Travailler avec des fichiers OOXML ... 952
18.2 Fichiers d'exemples OOXML ... 954
18.3 Fichiers ZIP ... 956
18.4 Fichiers EPUB ... 958

19 Bases de données 963

19.1 Connexion à une source de données ... 965
19.1.1 Lancer l'Assistant de la connexion ... 966
19.1.2 Aperçu des pilotes de base de données ... 969
19.1.3 Configurer une connexion ADO ... 972
19.1.4 Configurer une connexion ADO.NET ... 979
19.1.5 Configurer une connexion ODBC ... 986
19.1.6 Configurer une connexion JDBC ... 990
19.1.7 Configurer une connexion PostgreSQL ... 995
19.1.8 Configurer une connexion SQLite ... 997
19.1.9 Ressources globales ... 1000
19.1.10 Exemples de connexion à la base de données ... 1001
19.2 Bases de données prises en charge ... 1058

20 Ressources Globales Altova 1060

20.1 Définir les Ressources globales ... 1061
20.1.1 Fichiers ... 1063

20.1.2 Dossiers ... 1068
20.1.3 Bases de données ... 1070
20.2 Utiliser les Ressources globales ... 1073
20.2.1 Attribuer des Fichiers et des Dossiers ... 1073
20.2.2 Attribuer des bases de données ... 1076
20.2.3 Changer la Configuration active ... 1077

21 Projets 1079

21.1 Créer et éditer des projets ... 1080
21.2 Utiliser les projets ... 1085

22 RaptorXML(+XBRL) Server 1087

22.1 Ajouter des serveurs et des configurations de serveur ... 1088
22.2 Valider avec RaptorXML Server ... 1092
22.3 Options de validation ... 1093
22.3.1 Options communes ... 1094
22.3.2 XML avec DTD ... 1094
22.3.3 DTD ... 1094
22.3.4 XML avec Schéma W3C ... 1095
22.3.5 Schéma W3C ... 1095
22.3.6 Instance Inline XBRL ... 1096
22.3.7 Instance XBRL ... 1098
22.3.8 Taxonomie XBRL ... 1099
22.3.9 XBRL Taxonomy Package ... 1100
22.3.10 XBRL Versioning Report ... 1101
22.3.11 XSLT ... 1101
22.3.12 XQuery ... 1102
22.3.13 JSON ... 1103
22.3.14 JSON Schema ... 1103
22.3.15 AVRO ... 1103
22.3.16 AVRO JSON ... 1104
22.3.17 AVRO Schema ... 1104
22.3.18 EDGAR ... 1104

22.4 XSLT et XQuery avec RaptorXML Server ... 1107

23 Comparaisons Fichier/Répertoire 1109

23.1 Comparaisons de fichiers ... 1110
23.2 Comparaisons de répertoires ... 1111

24 Contrôle de code source 1113

24.1 Configurer le contrôle de source ... 1115
24.2 Systèmes de Contrôle de source pris en charge ... 1116
24.3 Dossier Poste de travail local ... 1118
24.4 Projet d'application ... 1119
24.5 Ajouter au contrôle de source ... 1121
24.6 Travailler avec le contrôle de source ... 1124
24.6.1 Ajouter à, Supprimer du contrôle de source ... 1124
24.6.2 Extraire, Archiver ... 1125
24.6.3 Obtenir les fichiers en lecture seule ... 1127
24.6.4 Copier et partager depuis le contrôle de source ... 1129
24.6.5 Modifier le contrôle de source ... 1132
24.7 Contrôle de source avec Git ... 1134
24.7.1 Activer le Contrôle de source avec le plugin Git SCC ... 1135
24.7.2 Ajouter un projet au Contrôle de source avec Git ... 1135
24.7.3 Cloner un projet depuis le Contrôle de source avec Git ... 1137

25 XMLSpy dans Visual Studio 1139

25.1 Installer le plugin XMLSpy ... 1140
25.2 Différences avec XMLSpy Standalone ... 1141
25.3 Débogueurs de XMLSpy dans Visual Studio ... 1144

26 XMLSpy dans Eclipse 1146

26.1 Install the Integration Package for Eclipse (2) ... 1147
26.2 XMLSpy Perspective in Eclipse ... 1149
26.3 Other XMLSpy Entry Points in Eclipse ... 1152

26.4 XMLSpy's Debugger Perspectives ... 1154
26.5 Points de contact XMLSpy dans Eclipse ... 1155
26.6 Perspectives de débogeur de XMLSpy ... 1158

27 Générateur de code 1159

27.1 Les nouveautés ... 1161
27.2 Générer du code depuis des Schémas XML ou des DTDs ... 1163
27.2.1 À propos des Schema Wrapper Libraries (C++) ... 1166
27.2.2 À propos des Schema Wrapper Libraries (C#) ... 1169
27.2.3 À propos des Schema Wrapper Libraries (Java) ... 1171
27.2.4 Intégrer des Bibliothèques Schema Wrapper ... 1173
27.2.5 Exemple: Bibliothèques de livres ... 1176
27.2.6 Exemple: Purchase Order ... 1200
27.3 Référence aux classes générées (C++) ... 1208
27.3.1 altova::DateTime ... 1208
27.3.2 altova::Duration ... 1211
27.3.3 altova::DayTimeDuration ... 1213
27.3.4 altova::YearMonthDuration ... 1214
27.3.5 altova::meta::Attribute ... 1214
27.3.6 altova::meta::ComplexType ... 1215
27.3.7 altova::meta::Element ... 1216
27.3.8 altova::meta::SimpleType ... 1216
27.3.9 [YourSchema]::[CDoc] ... 1218
27.3.10 [YourSchema]::[ElementType] ... 1220
27.3.11 [YourSchema]::MemberAttribute ... 1221
27.3.12 [YourSchema]::MemberElement ... 1222
27.4 Référence aux classes générées (C#) ... 1223
27.4.1 Altova.Types.DateTime ... 1223
27.4.2 Altova.Types.DateTimeFormat ... 1226
27.4.3 Altova.Types.Duration ... 1227
27.4.4 Altova.Xml.Meta.Attribute ... 1230
27.4.5 Altova.Xml.Meta.ComplexType ... 1231
27.4.6 Altova.Xml.Meta.Element ... 1231
27.4.7 Altova.Xml.Meta.SimpleType ... 1232

27.4.8 [YourSchema].[Doc] 1233
27.4.9 [YourSchema].[ElementType] 1235
27.4.10 [YourSchemaType].MemberAttribute 1236
27.4.11 [YourSchemaType].MemberElement 1237
27.5 Référence aux classes générées (Java) 1239
27.5.1 com.altova.types.DateTime 1239
27.5.2 com.altova.types.Duration 1244
27.5.3 com.altova.xml.meta.Attribute 1247
27.5.4 com.altova.xml.meta.ComplexType 1248
27.5.5 com.altova.xml.meta.Element 1248
27.5.6 com.altova.xml.meta.SimpleType 1249
27.5.7 com.[YourSchema].[Doc] 1250
27.5.8 com.[YourSchema].[ElementType] 1251
27.5.9 com.[YourSchema].[YourSchemaType].MemberAttribute 1253
27.5.10 com.[YourSchema].[YourSchemaType].MemberElement 1253
27.6 Astuces de Génération de Code 1255
27.7 Options du Générateur de Code 1256
27.8 Référence SPL 1258
27.8.1 Structure SPL de base 1258
27.8.2 Déclarations 1259
27.8.3 Variables 1261
27.8.4 Variables prédéfinies 1262
27.8.5 Créer des fichiers de sortie 1263
27.8.6 Opérateurs 1265
27.8.7 Conditions 1266
27.8.8 Collections et foreach 1267
27.8.9 Sous-routines 1268
27.8.10 Types intégrés 1271

28 Commandes de menu 1276

28.1 Menu Fichier 1277
28.1.1 Nouveau 1277
28.1.2 Ouvrir 1281
28.1.3 Recharger 1286

28.1.4 Encodage ... 1286
28.1.5 Fermer, Fermer tout, Fermer tout sauf actives ... 1287
28.1.6 Enregistrer, Enregistrer sous, Enregistrer tout ... 1287
28.1.7 Envoyer par mail ... 1293
28.1.8 Imprimer ... 1294
28.1.9 Aperçu d'impression, paramètres d'impression ... 1296
28.1.10 Fichiers récents, Quitter ... 1297
28.2 Menu Édition ... 1298
28.2.1 Annuler, Rétablir ... 1299
28.2.2 Couper, Copier, Coller, Supprimer ... 1299
28.2.3 Copier comme texte XML ... 1300
28.2.4 Copier comme Texte Structuré ... 1301
28.2.5 Copy as Image ... 1301
28.2.6 Copier XPath ... 1302
28.2.7 Copier XPointer ... 1302
28.2.8 Insérer ... 1302
28.2.9 Enregistrer comme image ... 1306
28.2.10 Pretty-Print ... 1307
28.2.11 Supprimer les espaces blancs ... 1307
28.2.12 Sélectionner tout ... 1307
28.2.13 Recherche, Trouver suivant ... 1307
28.2.14 Remplacer ... 1311
28.2.15 Chercher dans fichiers ... 1313
28.2.16 Commandes signet ... 1314
28.2.17 Commenter/Décommenter ... 1315
28.3 Menu Projet ... 1316
28.3.1 Nouveau projet ... 1319
28.3.2 Ouvrir le projet ... 1319
28.3.3 Recharger le projet ... 1319
28.3.4 Fermer projet ... 1319
28.3.5 Enregistrer projet, Enregistrer projet sous ... 1320
28.3.6 Contrôle de source ... 1320
28.3.7 Ajouter des fichiers au projet ... 1334
28.3.8 Ajouter une Ressource globale au Projet ... 1335
28.3.9 Ajouter une URL au projet ... 1335

28.3.10 Ajouter fichier actif au projet ... 1335
28.3.11 Ajouter des fichiers actifs et liés au projet ... 1335
28.3.12 Ajouter un dossier de projet au projet ... 1336
28.3.13 Ajouter un dossier externe au projet ... 1336
28.3.14 Ajouter un dossier web externe au projet ... 1339
28.3.15 Paramètres du script ... 1343
28.3.16 Propriétés ... 1343
28.3.17 Projets les plus récents ... 1346
28.4 Menu XML ... 1347
28.4.1 Type ... 1348
28.4.2 Insérer avant/après ... 1348
28.4.3 Apposer, Ajouter enfant ... 1348
28.4.4 Wrapper dans array ... 1348
28.4.5 Éditer en tant que Texte brut ... 1349
28.4.6 Déplacer vers le Haut/Bas/Gauche/Droite ... 1349
28.4.7 Afficher en tant que Table ... 1349
28.4.8 Tri Montant/Descendant ... 1349
28.4.9 Échanger Lignes/Colonnes ... 1349
28.4.10 Évaluer XPath ... 1350
28.4.11 Vérifier bonne formation ... 1350
28.4.12 Valider XML ... 1352
28.4.13 Valider XML sur serveur (performance élevée) ... 1356
28.4.14 Validation des fichiers WSDL ... 1357
28.4.15 Valider sur Édition ... 1358
28.4.16 Mise à jour Assistants à la saisie ... 1358
28.4.17 Préfixe d'espace de noms ... 1358
28.4.18 Créer Signature XML ... 1359
28.4.19 Vérifier Signature XML ... 1362
28.5 Menu JSON ... 1366
28.5.1 Type ... 1366
28.5.2 Insérer, Apposer, Ajouter enfant ... 1367
28.5.3 Wrapper dans Array/Objet ... 1367
28.5.4 Déplacer ... 1368
28.5.5 Display as Table ... 1368
28.5.6 Tri Montant/Descendant ... 1368

28.5.7 Échanger Lignes/Colonnes ... 1368
28.5.8 Supprimer commentaires, Tout réévaluer ... 1369
28.6 Menu DTD/Schéma ... 1370
28.6.1 Assigner DTD ... 1370
28.6.2 Assigner Schéma ... 1371
28.6.3 Inclure une autre DTD ... 1371
28.6.4 Aller à la DTD ... 1372
28.6.5 Aller au Schéma ... 1372
28.6.6 Aller à la Définition ... 1372
28.6.7 Générer DTD/Schéma ... 1373
28.6.8 Aplatir DTD ... 1375
28.6.9 Convertir DTD en Schéma ... 1375
28.6.10 Aplatir Schéma ... 1377
28.6.11 Convertir Schéma en DTD ... 1378
28.6.12 Convertir en UML ... 1378
28.6.13 Générer XML depuis BD, Excel, EDI avec MapForce ... 1380
28.6.14 Créer sortie HTML/PDF/Word avec StyleVision ... 1380
28.6.15 Générer fichier échantillon XML/JSON ... 1380
28.6.16 Générer code de programme ... 1384
28.6.17 Vider cache mémoire ... 1385
28.7 Menu Conception de schéma ... 1386
28.7.1 Paramètres du schéma ... 1386
28.7.2 Enregistrer diagramme ... 1389
28.7.3 Générer la documentation ... 1389
28.7.4 Configurer mode ... 1394
28.7.5 Zoomer ... 1398
28.7.6 Afficher tous les Globaux ... 1398
28.7.7 Afficher le diagramme ... 1398
28.7.8 Extensions de schéma pour les bases de données ... 1399
28.7.9 Se connecter à SchemaAgent Server ... 1403
28.7.10 Déconnecter de SchemaAgent Server ... 1404
28.7.11 Afficher dans SchemaAgent ... 1405
28.7.12 Validation SchemaAgent ... 1405
28.7.13 Créer un sous-ensemble de schéma ... 1405
28.7.14 Aplatir le schéma ... 1407

28.8 Menu XSL/XQuery ... 1409
28.8.1 Transformation XSL ... 1410
28.8.2 Optimiseur de vitesse XSL ... 1411
28.8.3 Transformation XSL-FO ... 1412
28.8.4 Paramètres XSL / Variables XQuery ... 1413
28.8.5 Exécution XQuery/Mise à jour ... 1417
28.8.6 Activer Back-Mapping ... 1418
28.8.7 Activer profilage XSLT/XQuery ... 1420
28.8.8 Assigner XSL ... 1420
28.8.9 Assigner XSL-FO ... 1421
28.8.10 Assigner fichier échantillon XML ... 1421
28.8.11 Aller au XSL ... 1422
28.8.12 Aller à l'instruction de source ... 1422
28.8.13 Aller au nœud de contexte ... 1422
28.8.14 Démarrer Débogueur / OK ... 1422
28.8.15 Arrêter Débogueur XSLT ... 1423
28.8.16 Redémarrer Débogueur ... 1423
28.8.17 Terminer session débogueur ... 1423
28.8.18 Pas à pas détaillé ... 1423
28.8.19 Pas à pas sortant ... 1424
28.8.20 Pas à pas principal ... 1424
28.8.21 Afficher nœud d'exécution actuel ... 1424
28.8.22 Insérer/Supprimer Point d'arrêt ... 1424
28.8.23 Insérer/Supprimer Point de trace ... 1425
28.8.24 Activer/Désactiver Point d'arrêt ... 1425
28.8.25 Activer/Désactiver Point de trace ... 1425
28.8.26 Points d'arrêt/ Points de trace ... 1426
28.8.27 Déboguer les fenêtres ... 1427
28.8.28 Paramètres de débogage ... 1427
28.9 Menu Authentic ... 1428
28.9.1 Nouveau document ... 1429
28.9.2 Modifier les données de la base de données ... 1429
28.9.3 Assigner une feuille de style StyleVision ... 1430
28.9.4 Éditer une feuille de style StyleVision ... 1431
28.9.5 Sélectionner nouvelle ligne avec données XML à éditer ... 1431

28.9.6 Signature XML ... 1432
28.9.7 Définir les entités XML ... 1434
28.9.8 Afficher balisage ... 1436
28.9.9 RichEdit ... 1437
28.9.10 Ajouter/Insérer/Dupliquer/Supprimer ligne ... 1437
28.9.11 Déplacer Ligne, Supprimer Ligne ... 1438
28.9.12 Générer Document HTML, RTF, PDF, Word 2007+ ... 1438
28.9.13 Emplacements approuvés ... 1438
28.10 Menu DB ... 1440
28.10.1 Requête de base de données ... 1440
28.10.2 IBM DB2 ... 1456
28.10.3 Serveur SQL ... 1462
28.10.4 Oracle XML DB ... 1465
28.11 Menu Conversion ... 1470
28.11.1 Importer fichier texte ... 1470
28.11.2 Importer données de base de données ... 1473
28.11.3 Importer Document Microsoft Word ... 1478
28.11.4 Créer le Schéma XML depuis la structure BD ... 1478
28.11.5 Importation BD basée sur un Schéma XML ... 1483
28.11.6 Créer une structure BD depuis le Schéma XML ... 1484
28.11.7 Exporter vers les fichiers de texte ... 1488
28.11.8 Exporter vers une base de données ... 1491
28.11.9 Convertir instance XML en/de JSON ... 1494
28.11.10 Convertir Schéma XML en/de Schéma JSON ... 1497
28.11.11 Convert to OIM xBRL-XML ... 1500
28.11.12 Convert to OIM xBRL-JSON ... 1500
28.11.13 Convert to OIM xBRL-CSV ... 1500
28.12 Menu Mode ... 1501
28.12.1 Mode Texte ... 1501
28.12.2 Mode Grille avancé ... 1502
28.12.3 Mode Design Schéma ... 1502
28.12.4 Mode Design WSDL ... 1502
28.12.5 Mode Taxonomie XBRL ... 1503
28.12.6 Mode Authentic ... 1503
28.12.7 Mode Navigateur ... 1503

28.12.8 Agrandir ... 1504
28.12.9 Réduire ... 1504
28.12.10 Agrandir complètement ... 1504
28.12.11 Réduire non sélectionnés ... 1504
28.12.12 Largeurs optimales ... 1505
28.12.13 Retour à la ligne ... 1505
28.12.14 Aller à la ligne/au caractère ... 1505
28.12.15 Aller au fichier ... 1506
28.12.16 Paramètres mode Texte ... 1506
28.13 Menu Navigateur ... 1508
28.14 Menu WSDL ... 1509
28.14.1 Composants WSDL 1.1 ... 1509
28.14.2 Composants WSDL 2.0 ... 1512
28.14.3 Types, Enregistrer diagramme ... 1516
28.14.4 Générer documentation ... 1517
28.14.5 Reparser le Document WSDL ... 1521
28.14.6 Convertir en WSDL 2.0 ... 1521
28.14.7 Générer le code de programme WSDL avec MapForce ... 1521
28.15 Menu SOAP ... 1522
28.15.1 Créer une nouvelle requête SOAP ... 1522
28.15.2 Envoyer la requête au serveur ... 1524
28.15.3 Paramètres de requête SOAP ... 1525
28.15.4 Session de débogage Soap ... 1529
28.15.5 Aller à ... 1530
28.15.6 Pas à pas ... 1531
28.15.7 Arrêter à la prochaine requête ... 1531
28.15.8 Arrêter à la prochaine réponse ... 1531
28.15.9 Arrêter le serveur proxy ... 1531
28.15.10 Options de débogage SOAP ... 1532
28.16 Menu XBRL ... 1533
28.16.1 Rôles d'arc ... 1533
28.16.2 Rôles de lien ... 1535
28.16.3 Préfixes d'espace de noms ... 1537
28.16.4 Définir l'espace de noms cible ... 1538
28.16.5 Valeurs de paramètres ... 1538

28.16.6 Importer/Référence ... 1539
28.16.7 Trouver le composant par ID ... 1541
28.16.8 Générer documentation ... 1541
28.16.9 Paramètres de mode ... 1545
28.16.10 Générer XBRL depuis BD, Excel, CSV avec MapForce ... 1546
28.16.11 Présenter XBRL en tant que HTML/PDF/Word avec StyleVision ... 1546
28.16.12 Exécuter Formule (sur Serveur) ... 1546
28.16.13 Générer Table (sur Serveur) ... 1549
28.16.14 Détecter les duplicatas (sur Serveur) ... 1551
28.16.15 Execute XULE ... 1552
28.16.16 Transformer Inline XBRL ... 1553
28.16.17 Valider EDGAR sur serveur ... 1554
28.16.18 Options de traitement ... 1555
28.17 Menu Outils ... 1557
28.17.1 Orthographe ... 1558
28.17.2 Options du vérificateur orthographique ... 1562
28.17.3 Éditeur de script ... 1566
28.17.4 Macros ... 1567
28.17.5 Comparaisons ... 1567
28.17.6 Outils définis par l'utilisateur ... 1577
28.17.7 Ressources globales ... 1577
28.17.8 Configuration Active ... 1578
28.17.9 Gérer les serveurs Raptor ... 1579
28.17.10 Serveurs Raptor et Configurations ... 1582
28.17.11 Gestionnaire de taxonomie ... 1582
28.17.12 XML Gestionnaire de schéma ... 1583
28.17.13 Personnaliser ... 1583
28.17.14 Restaurer barres d'outils et fenêtres ... 1601
28.17.15 Options ... 1601
28.18 Menu Fenêtre ... 1647
28.19 Menu Aide ... 1649
28.19.1 Table des matières, Index, Rechercher ... 1649
28.19.2 Mappage clavier ... 1650
28.19.3 Activation, Bon de commande, Inscription, Mises à jour ... 1651
28.19.4 Autres commandes ... 1654

28.20 Ligne de commande ... 1656

29 Programmers' Reference 1657

29.1 Scripting Editor ... 1659
29.1.1 Creating a Scripting Project ... 1660
29.1.2 Built-in Commands ... 1673
29.1.3 Enabling Scripts and Macros ... 1683
29.2 IDE Plugins ... 1686
29.2.1 Registration of IDE PlugIns ... 1686
29.2.2 ActiveX Controls ... 1687
29.2.3 Configuration XML ... 1687
29.2.4 ATL sample files ... 1690
29.2.5 IXMLSpyPlugIn ... 1695
29.3 Application API ... 1702
29.3.1 Overview ... 1703
29.3.2 Interfaces ... 1739
29.3.3 Interfaces (obsolete) ... 1963
29.3.4 Enumerations ... 2000
29.3.5 Application API for Java (obsolete) ... 2014
29.4 ActiveX Integration ... 2065
29.4.1 Prerequisites ... 2065
29.4.2 Adding the ActiveX Controls to the Toolbox ... 2066
29.4.3 Integration at Application Level ... 2068
29.4.4 Integration at Document Level ... 2070
29.4.5 ActiveX Integration Examples ... 2073
29.4.6 Command Reference ... 2092
29.4.7 Object Reference ... 2111

30 Annexes 2133

30.1 Informations concernant le moteur XSLT et XQuery ... 2134
30.1.1 XSLT 1.0 ... 2134
30.1.2 XSLT 2.0 ... 2134
30.1.3 XSLT 3.0 ... 2137

30.1.4 XQuery 1.0 2137
30.1.5 XQuery 3.1 2140
30.2 Fonctions XSLT et XPath/XQuery 2141
30.2.1 Fonctions d'extension Altova 2142
30.2.2 Fonctions d'extension diverses 2230
30.3 Types de données dans les schémas XML générés par BD 2234
30.3.1 ADO 2234
30.3.2 MS Access 2235
30.3.3 MS SQL Server 2236
30.3.4 MySQL 2237
30.3.5 ODBC 2237
30.3.6 Oracle 2238
30.3.7 Sybase 2239
30.4 Types de données dans les BD générés depuis des schémas XML 2241
30.4.1 MS Access 2241
30.4.2 MS SQL Server 2242
30.4.3 MySQL 2244
30.4.4 Oracle 2246
30.5 Données techniques 2249
30.5.1 SE et exigences de mémoire 2249
30.5.2 Moteurs Altova 2249
30.5.3 Prise en charge Unicode 2250
30.5.4 Utilisation Internet 2250
30.6 Informations de licence 2252
30.6.1 Distribution électronique de logiciel 2252
30.6.2 Activation de logiciel et le license metering 2253
30.6.3 Altova Contrat de licence de l'utilisateur final 2254

Index **2255**

1 À propos de XMLSpy et de cette documentation

Altova XMLSpy 2023 Enterprise Edition est l'éditeur le plus avancé pour XML et JSON disponible pour la conception, l'édition et le débogage d'applications de classe professionnelle comprenant JSON, XML, Schéma XML, XSLT, XQuery, SOAP, WSDL, Web services, OOXML et les technologies XBRL. XMLSpy est exécuté sur Windows 7 SP1 avec mise à jour de la plateforme, Windows 8, Windows 10, Windows 11 et Windows Server 2008 R2 SP1 avec mise à jour de la plateforme ou plus récent. XMLSpy est disponible dans les versions 64-bit et 32-bit et fait partie de la suite d'outils Altova MissionKit au rapport qualité/prix excellent.



Cette documentation est articulée autour des sections principales suivantes :

- Un tutoriel pour vous aider à vous lancer
- Des descriptions des fonctions XMLSpy, classées par technologie ou par fonction spécifique à XMLSpy
- Des descriptions des commandes de menu
- Une références des programmeurs
- Des annexes, qui contiennent des informations concernant les moteurs (i) XSLT et XQuery de XMLSpy, et les fonctions d'extensions (ii) XPath/XQuery d'Altova

Dernière mise à jour : 13.10.2022

© 2016-2022 Altova GmbH

1.1 Nouvelles fonctions 2022

- Vous pouvez sélectionner un thème classique, clair ou sombre pour l'application. Lorsqu'un thème est actif, son affichage texte peut être formaté séparément pour le type de document et le mode dans le dialogue des Options.
- Un thème XMLSpy peut aussi être sélectionné pour les intégrations XMLSpy dans Eclipse et Visual Studio.
- Le nouveau composant Altova SchemaManager vous permet d'installer communément des DTD et Schémas XML et de les intégrer pour utilisation avec XMLSpy et d'autres produits Altova. Vous pouvez accéder au dialogue Gestionnaire de schéma par le biais du menu de commande **Outils | Gestionnaire de schéma XML**.
- Vous pouvez choisir Microsoft Edge WebView2 comme navigateur à utiliser dans le Mode Navigateur. Ceci est l'alternative plus récente utilisant Internet Explorer comme moteur de navigateur dans le Mode Navigateur.
- Si vous utilisez le moteur de navigation Microsoft Edge WebView2 pour le Mode Navigateur, vous pouvez alors utiliser les outils de développeurs du moteur de navigation pour déboguer et tester votre code HTML.
- La validation Inline XBRL a une option pour vérifier la conformité du document avec le Manuel de rapport du European Single Electronic Format (ESEF). L'option est un paramètre Inline XBRL dans le dialogue des Options.
- La fenêtre de sortie HTTP a été améliorée avec un volet de journal et une information sur les paramètres proxy.
- La prise en charge de base de données a été élargie à : SQLite 3.38.5.
- La prise en charge Eclipse a été mise à jour pour couvrir les versions suivantes : 2022-09; 2022-06; 2022-03; 2021-12.

1.1.1 Version 2022

Version 2022r2

- Un paramètre valide pour l'application entière permet d'activer le pretty-printing de documents XML et JSON automatiquement lorsque ceux-ci sont chargés dans le mode Texte.
- Dans le dialogue des Options, vous pouvez spécifier le traitement par défaut de fichiers lorsqu'ils sont ouverts dans XMLSpy. Ce traitement s'appliquera aux fichiers qui ont une extension qui n'est pas celle pour laquelle le traitement a été défini.
- Lorsque vous générez des échantillons de documents XML depuis un schéma, vous pouvez spécifier jusqu'à quel niveau d'imbrication pour lequel des éléments non obligatoires seront générés..
- La Conversion d'un schéma XML en schéma JSON prend désormais en charge la définition de types au sein d'un projet JSON. Il s'agit ici d'une alternative à la définition du type en la référençant.
- Les images stockées comme strings Base64 encodés dans les documents XML et JSON peuvent être enregistrés comme images. Cette fonction est disponible dans le mode Texte (*voir for XML, for JSON*) et le mode Grille (*voir ces documents pour XML, pour JSON*).
- La fenêtre de projet a désormais une barre d'outils pour accéder rapidement des commandes de projet utilisées fréquemment.
- Lorsque vous passez au-dessus du fichier de l'image qui a été placée dans un dossier de projet (de la fenêtre de projet), une prévisualisation de l'image est affichée.

- La prise en charge d'Eclipse a été mise à jour sur les versions les plus récentes : 2021-12, 2021-09 ; 2021-06 ; 2021-03.
- La prise en charge de Visual Studio a été étendue à la version 2022 (64-bit).
- La prise en charge de la génération de code C++ et C# a été étendue à Visual Studio 2022 et .NET 6.
- La prise en charge de base de données a été élargie à : PostgreSQL 14,1, SQLite 3.37.2, MariaDB 10.6.5, MySQL 8.0.28.

Version 2022

- Prise en charge de l'édition du schéma BSON.
- Prise en charge de la conversion de données XBRL en formats OIM et validation de documents OIM xBRL en tant que documents XBRL.
- Pour les Schémas JSON de la version draft-2019-09 et plus élevée, un composant de schéma peut en même temps référencer une autre définition de schéma et contenir ses propres définitions locales. Ceci est différent de ce que c'était dans des définitions précédentes, où soit une référence, soit des définitions locales étaient permises, mais pas les deux. JSON Schema View a été mis à jour pour gérer de telles références élargies.
- La gestion de l'espace blanc a été améliorée.
- Le Mode Grille a été amélioré de telle façon à pouvoir scroller vers le bas d'un document, une barre d'en-tête est affichée qui contiendra des nœuds ancêtre du nœud actuellement affiché en haut du Mode Grille.
- Limite le nombre de messages de validation à afficher peut être défini séparément pour des erreurs, avertissements et inconsistances XBRL.
- Quitter les options de mode vous permet de choisir comment des changements non enregistrés et documents ouverts sont gérés sont gérés quand XMLSpy quitte.
- Dans la fenêtre XPath/XQuery, vous pouvez charger/enregistrer dans les snippets XPath/XQuery depuis/vers un fichier XQuery. Ceci vous permet d'être flexible si vous voulez réutiliser des expressions et snippets XPath/XQuery.
- Nouvelles Fonctions d'extension d'Altova pour accéder à l'information de schéma : fonctions Unix Time (ou Epoch Time).
- La prise en charge de base de données a été élargie à : PostGreSQL 13, IBM DB2 11.5 et MySQL 8.0.25.
- La prise en charge Eclipse a été actualisée pour désormais couvrir les versions suivantes : 2021-09; 2020-06; 2020-03; 2020-12.

1.1.2 Version 2021

Version 2021r3

- La prise en charge de Schémas JSON a été étendue aux versions 2020-12 et 2019-09.
- La prise en charge Eclipse a été étendue pour couvrir maintenant les versions suivantes : 2021-03, 2020-12; 2020-09; 2020-06.
- Le mode Grille des documents XML et documents JSON a été élargi et amélioré.

Version 2021r2

- Le Mode Grille des documents XML a été amélioré avec un nombre de nouvelles fonctions.

© 2016-2022 Altova GmbH

- Les commandes et paramètres liés au Mode Grille de documents XML ont été réorganisés pour faciliter l'utilisation : voir les Paramètre du Mode Grille et le dialogue des Options.
- Les paramètres des documents JSON du Mode Grille ont été réorganisés pour plus d'aisance d'utilisation : voir les Paramètres du Mode Grille et le dialogue Options.
- Le Mode Grille des documents DTD a été amélioré avec de nouvelles fonctions.
- La nouvelle fonction de superposition de glisser dans le Mode Grille XML et Mode Grille JSON fournit des informations précieuses qui facilitent l'utilisation de glisser-déplacer dans le Mode Grille.
- Nouvelles fonctionnalités pour ajouter facilement et rapidement des fragments XML et des fragments JSON de sources externes.
- Dans le Mode Grille, des composants supplémentaires (tels que des commentaires XML) peuvent être dotés d'un formatage spécifique au composant. Chaque type de compostant sera ensuite affiché dans le Mode Grille avec le formatage y assigné. Donc, les commentaires, par exemple, peuvent être affichés en surbrillance jaune dans le Mode Grille.
- Une nouvelle fonction pour valider les documents pendant que vous éditez peut être activée/désactivée.
- La prise en charge Eclipse a été étendue pour couvrir maintenant les versions suivantes : 2020-12 ; 2020-09 ; 2020-06 ; 2020-03.

Version 2021

- Un outil XBRL Gestionnaire de taxonomie, qui vous permet d'installer, de mettre à niveau et de gérer aisément des taxonomies pour une utilisation avec XMLSpy.
- Le menu JSON propose les améliorations du Mode JSON Grid suivantes : sélectionner un composant de Mode Grille et insérer un item avant ou après ; déplacer le composant de Mode Grille vers le haut/bas ou droite/gauche ; supprimer tous les commentaires du document..
- Nouvelles fonctions Mode JSON Grid : (i) le menu contextuel a été simplifié ; (ii) la grille peut être zoomée; (iii) plusieurs composants frères peuvent être agrandis/réduits ensemble; (iv) le sens de lecture peut être modifié en lecture de droite à gauche pour des cellules contenant des strings (utile pour des langages se lisant de droite à gauche, comme l'arabe et l'hébreu) ; la commande Copier a été améliorée.
- La validation de Schéma JSON prend la propriété `$id` en considération.
- Nouvelles Fonctions d'extension Altova pour accéder à l'information de schéma.
- La prise en charge Eclipse a été étendue pour couvrir maintenant : 2020.06; 2020.03; 2019.12; 2019.09.

1.1.3 Version 2020

Version 2020 Release 2

- Les fichiers qui sont édités sont sauvegardés automatiquement et peuvent être restaurés dans le cas d'une cessation imprévue du programme. Vous pouvez définir si vous souhaitez exécuter des sauvegardes automatiques et à quelle fréquence.
- La fonction Valider sur Édition marque les erreurs de validation et de bonne formation pendant votre saisie dans le Mode Texte et le Mode JSON Grid. Cette fonction peut être activée/désactivée dans les options de programme, et via une icône de la barre d'outils.
- Dans le Mode Schéma, la commande Aller à la Définition de type contenue dans plusieurs menus contextuels vous permet de passer à une définition de type de composant (simple ou complexe).
- De nouvelles options de Mode JSON Grid vous permettent de : (i) déterminer si les filtres et les formules sont enregistrées automatiquement dans un fichier de métadonnées JSON à l'échelle de

toute l'application, su demande, ou jamais , et (ii) de spécifier l'emplacement de ce fichier de métadonnées JSON. Le fichier de métadonnées peut être utilisé par la suite pour appliquer les filtres et formules stockées dans les fichiers JSON afférents sur plusieurs instances de XMLSpy, permettant ainsi une portabilité des filtres et des formules. (*Note: Supprimé dans la version 2021r2.)*

- Mode JSON Grid: Le menu contextuel contient des commandes (i) pour réduire tous les composants non sélectionnés, et (ii) pour supprimer tous les commentaires du document, y compris, en option, des formules (qui sont stockées en tant que commentaires).
- Mode JSON Grid: Amélioration des fonctions de Recherche.
- Paramètres pour spécifier la gestion des espaces blancs dans in Inline XBRL.
- Nouvelles méthodes pour la génération de code.

Version 2020

- Mode JSON Grid propose des fonctions d'édition : (i) détection de type automatique, (ii) commandes dans les cellules, (iii) filtres XQuery pour modifier le mode et (iv) des formules XQuery pour générer une sortie supplémentaire depuis des données JSON.
- Validation et prise en charge de JSON Lines et JSON Comments.
- Dans le Mode JSON Grid, les images peuvent être affichées et les graphiques peuvent être créés et affiché.
- Le dialogue Options (**Outils | Options**) fournit des paramètres pour le dans le Mode TexteMode JSON Grid et pour des documents JSON pretty-print dans le Mode Texte. (*Note : déplacé vers le dialogue Paramètres du Mode Grid et la section Pretty-print du dialogue Options.*
- Un valideur XULE et un processeur XULE ont été ajoutés. XULE est un langage de requête pour les documents d'instance XBRL. Voir la section XBRL > XULE pour un aperçu des nouvelles fonctions XULE.
- Une fenêtre XULE vous permet de requêter interactivement des documents d'instance XBRL.
- Aide d'édition pour des documents XULE.
- Plusieurs documents XBRL Inline peuvent être traités en un ensemble, contrairement au traitement précédent de chaque document séparément.
- L'Évaluateur et le Créateur d'expression de la fenêtre XPath/XQuery ont été simplifiés et optimisés.
- La fonction de débogage XPath de la fenêtre XPath/XQuery s'est vue enrichie de plusieurs nouvelles fonctions : (i) Expressions Watch, (ii) amélioration de l'interface pour un meilleur aperçu, et (iii) des analyses plus puissantes. Ainsi, il est plus simple de tester et de déboguer les expressions XPath/XQuery par rapport aux fichiers XML et JSON.
- Auparavant, le mode d'édition par défaut des fichiers avec des types de fichier différents a été choisi dans le dialogue Options. Le mode d'édition par défaut peut maintenant, en outre être lui-même choisi directement dans le mode d'édition lui-même.
- La prise en charge de l'intégration de XMLSpy dans Visual Studio a été étendue à Visual Studio 2019.
- La prise en charge de l'intégration de XMLSpy dans Eclipse a été étendue à Eclipse 4.11 et 4.12.

1.1.4 Versions 2019

Version 2019 Release 3

Les nouvelles fonctions de XMLSpy **Version 2019r3** sont regroupées ci-dessous.

- Le Mode Texte a été élargi par un Mode Écran divisé. Cette option vous offre deux affichages d'un document en cours d'édition, ce qui vous permet de voir plusieurs parties d'un long document côte à côte ; au lieu de devoir faire défiler un seul affichage pour référer à une autre partie du document.

© 2016-2022 Altova GmbH

- Prise en charge des nouvelles versions Schéma JSON ; draft-06 et draft-07. Voir la section JSON de la documentation.
- Le dialogue des Options (**Outils | Options**) propose une option pour définir le chemin Java Virtual Machine qui contourne le chemin détecté automatiquement.
- Prise en charge de Windows Server 2019.
- La prise en charge d'Eclipse a été étendue à Eclipse 4.9 et 4.10.

Version 2019

Les nouvelles fonctions de XMLSpy **Version 2019** sont regroupées ci-dessous.

- La prise en charge des bases de données est étendue à : Firebird 3.0, IBM Informix 12.10, MariaDB 10.3, Microsoft SQL Server 2017, MySQL 8.0, PostgreSQL 10
- Intégration avec Eclipse 4.8
- Une prise en charge de la recommandation Generic Preferred Label 1.0 permet de spécifier des libellés préférés pour des ressources XBRL, comme des relations de calcul.
- Les définitions des domaines typés XBRL sont affichés dans un onglet Domaines Typé séparé de l'assistant de saisie Détails.
- Nouvelles commandes de menu XBRL pour détecter des faits dupliqués dans les documents d'instance XBRL.
- Un dialogue d'options de traitement XBRL contient des options pour : (i) ignorer les faits dupliqués et (ii) la génération de table.
- La section XBRL du dialogue Options (**Outils | Options**) propose un onglet contenant des options de validation pour les documents d'instance XBRL. Vous pouvez décider si vous souhaitez contrôler des documents pour (i) respecter la conformité Dimensions 1.0 et Units Registry 1.0, (ii) dupliquer des faits, et (iii) les incohérences résultant des calculs.
- Les attributs personnalisés des éléments XBRL sont aussi affichés dans l'Assistant à la saisie des Détails.

1.2 Windows File Paths

Chemins de fichiers dans Windows

Les chemins de fichiers dans cette documentation ne seront pas les mêmes pour tous les systèmes d'exploitation. Veuillez les correspondances suivantes :

- (Mon) dossier de documents : Situé par défaut aux emplacements suivants. Les fichiers d'exemple sont situés dans un sous-dossier de ce dossier.

Windows 7/8/10/11	`C:\Users\<username>\Documents`

- *Dossier d'application :* Le dossier d'application est le dossier où votre application d'Altova est situé. Le chemin du dossier d'application est, par défaut, le suivant.

Windows 7/8/10/11	`C:\Program Files\Altova\`
version 32-bit sur 64-bit OS	`C:\Program Files (x86)\Altova\`

Note : XMLSpy est aussi pris en charge sur Windows Server 2008 R2 SP1 avec mise à jour de la plateforme ou plus récent.

© 2016-2022 Altova GmbH

1.3 About RaptorXML Server

Altova RaptorXML(+XBRL) Server (aussi appelé Raptor ou RaptorXML) est le processeur XML (et XBRL) de troisième génération d'Altova extrêmement rapide. Il a été conçu pour utiliser de manière optimale les derniers standards et environnements de calcul parallèles. Il peut être utilisé sur de nombreuses plateformes, de plus il profite de l'omniprésence des ordinateurs multicœurs actuels pour fournir un traitement extrêmement rapide des données XML et XBRL.

RaptorXML est disponible dans deux éditions :

- RaptorXML Server, qui peut être accédé depuis un réseau et peut transformer plusieurs fichiers simultanément.
- RaptorXML+XBRL Server edition, qui peut être accédé depuis un réseau, peut transformer plusieurs fichiers simultanément, et prend en charge la validation XBRL.

RaptorXML peut être exécuté depuis la ligne de commande et possède des interfaces pour COM, Java, .NET et Python. Un serveur Raptor peut aussi être exécuté à partir de l'interface XMLSpy.

Site web Altova : Serveur de validation XML, Valideur XML

2 Tutoriel XMLSpy

Ce tutoriel propose un aperçu de XML et vous présente un certain nombre de tâches XML importantes. Dans le cadre de ce processus, vous apprendrez à utiliser certaines des fonctions XMLSpy les plus importantes.

Le tutoriel est divisé dans les parties suivantes :

- Interface XMLSpy, qui vous aide à vous familiariser avec l'interface graphique utilisateur de l'application (GUI).
- Créer un schéma XML. Vous apprendrez à créer un schéma XML dans le Mode Schéma intuitif de XMLSpy, comment créer des modèles de contenus complexes par simple glisser-déposer et comment configurer le Mode Schéma.
- Utiliser les fonctions Mode Schéma pour créer des types complexes et simples, des références d'élément global et des énumérations d'attribut.
- Apprendre comment parcourir les schémas dans le Mode Schéma et comment générer une documentation de schémas.
- Créer un document XML. Vous apprendrez comment attribuer un schéma pour un document XML, éditer un document XML dans le Mode Grille et le Mode Texte et valider des documents XML à l'aide du validateur intégré XMLSpy.
- Transformer un fichier XML à l'aide de feuilles de style XSLT. Cela implique l'attribution d'un fichier XSLT et l'exécution de la transformation à l'aide des moteurs XSLT intégrés de XMLSpy.
- Travailler avec des projets XMLSpy, qui vous permettent d'organiser facilement vos documents XML.

Installation et configuration

Ce tutoriel présume que vous avez installé avec succès XMLSpy sur votre ordinateur et que vous avez obtenu un code d'évaluation gratuit ou bien que vous êtes un utilisateur enregistré. La version d'évaluation de XMLSpy est entièrement fonctionnelle mais limitée à une période de 30 jours. Vous pouvez demander une licence ordinaire depuis notre serveur web sécurisé ou par le biais d'un de nos revendeurs.

Fichiers d'exemples de tutoriel

Les fichiers de tutoriel sont disponibles dans le dossier de l'application :

```
C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial
```

Le dossier **Examples** contient plusieurs fichiers XML qui vous permettront d'expérimenter avec les fonctions, le dossier **Tutorial** contient tous les fichiers utilisés dans ce tutoriel.

Le dossier **Template** contenu dans le dossier d'application (généralement dans **C:\Program Files\Altova**) contient tous les fichiers de modèle XML qui sont utilisés à chaque fois que vous sélectionnez l'option de menu **Fichier | Nouveau**. Ces fichiers fournissent les données nécessaires (espaces de noms et déclarations XML) pour que puissiez commencer à travailler immédiatement avec le document XML respectif.

© 2016-2022 Altova GmbH

2.1 Interface XMLSpy

Dans cette section du tutoriel, vous allez lancer XMLSpy et apprendre à vous familiariser avec l'interface.

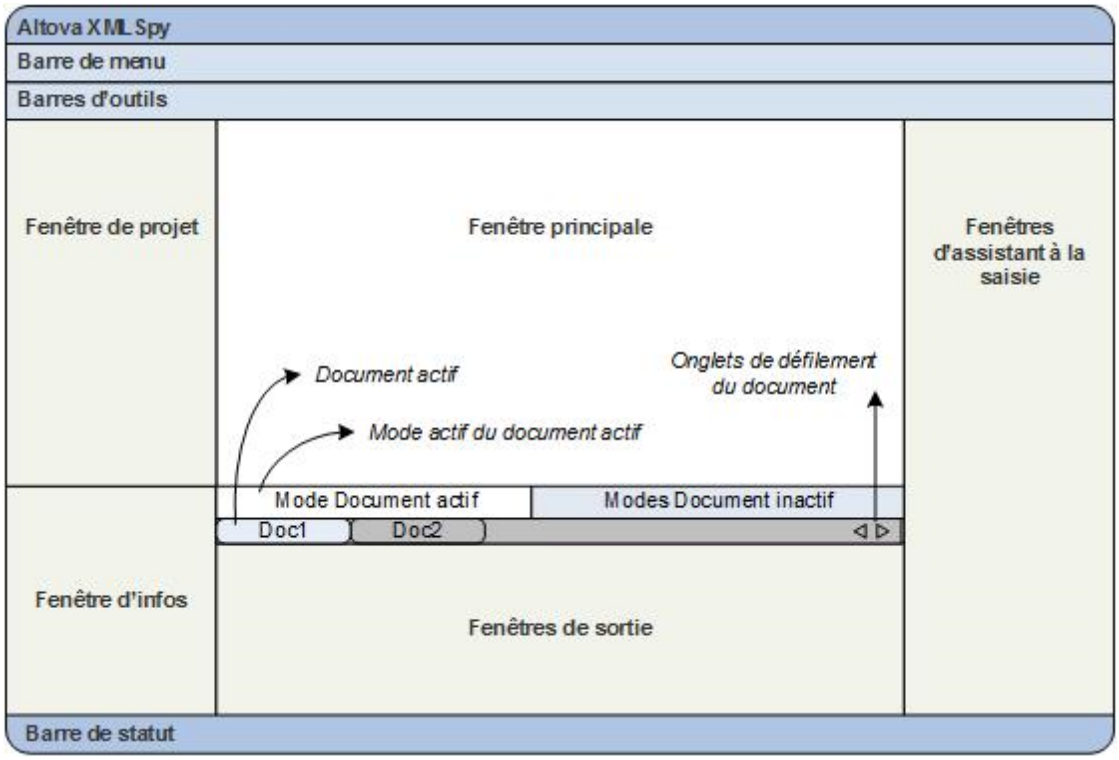
Lancer XMLSpy

Pour lancer XMLSpy, double-cliquez sur l'icône XMLSpy de votre bureau ou bien utilisez le menu **Démarrer | Tous les programmes** pour accéder au programme XMLSpy. Aucun document ne s'ouvre sur l'interface lors du lancement de XMLSpy. Ouvrir XMLSpy maintenant.

Aperçu de l'interface

Le mode d'affichage par défaut de l'interface XMLSpy est structuré dans trois zones verticales (*figure ci-dessous*). Ces trois zones contiennent, de gauche à droite : (i) les fenêtres Projet et Info ; (ii) les fenêtres Principale et Sortie et (iii) les fenêtres d'Aide à la saisie. Jetez un coup d'œil à la fenêtre Projet. Elle contiendra le projet Exemples, qui s'ouvre par défaut lorsque vous lancez XMLSpy pour la première fois.

Ci-dessous, vous pouvez voir les points les plus importants qui vous aideront à comprendre la mise en page de l'interface et le fonctionnement de ses composants divers. Les sous-sections de cette première partie du tutoriel vous aideront à vous familiariser avec l'interface.



Barre Document dans la Fenêtre principale : lorsque plusieurs documents sont ouverts, chaque document est affiché dans un onglet dans la barre de document de la Fenêtre principale (*voir graphique*). Cliquer sur un onglet permet d'activer ce document. Vous pouvez faire défiler les onglets du document en cliquant sur les flèches du côtés droits de la barre de document. Ouvrir deux ou plus de fichiers (par exemple, depuis le projet Examples), et vérifiez comment les onglets fonctionnent.

Modes d'édition du document : le document actif peut être affiché dans un des différents modes d'édition applicables. Par exemple :

- Un document XML (`.xml`) peut être affiché dans le Mode Texte, le Mode Grille, le Mode Authentic et le Mode Navigateur, mais pas dans d'autres modes, comme par exemple le Mode Schéma.
- Un document Schéma (`.xsd`), en revanche, peut être affiché dans le Mode Texte, le Mode Grille, le Mode Schéma et le Mode Navigateur, mais pas dans le Mode Authentic.

Les modes suivants sont disponibles : Mode Texte, Mode Grille, Mode Schéma, Mode WSDL, Mode XBRL, Authentic View, Mode Archive, and Mode Navigateur.

Assistants de saisie : les fenêtres d'aide à la saisie changent conformément au type du document actif (par exemple XML ou XSD ou CSS ou WSDL) et conformément au mode de document actif actuellement (par exemple, Mode Texte ou Mode Schéma). Les assistants de saisie vous permettent d'éditer rapidement et correctement le document actif en apportant une prise en charge d'édition sensible au contexte.

2.1.1 Les modes

Dans cette section du tutoriel, vous allez apprendre : (i) à naviguer entre les différents modes d'édition de document et (ii) à changer le mode d'édition par défaut d'un type de document particulier.

Naviguer entre les modes de document

Lorsque vous ouvrez un document, il s'ouvrira dans le mode qui a été configuré en tant que mode par défaut pour ce type de document. Pour ouvrir un document, procéder comme suit :

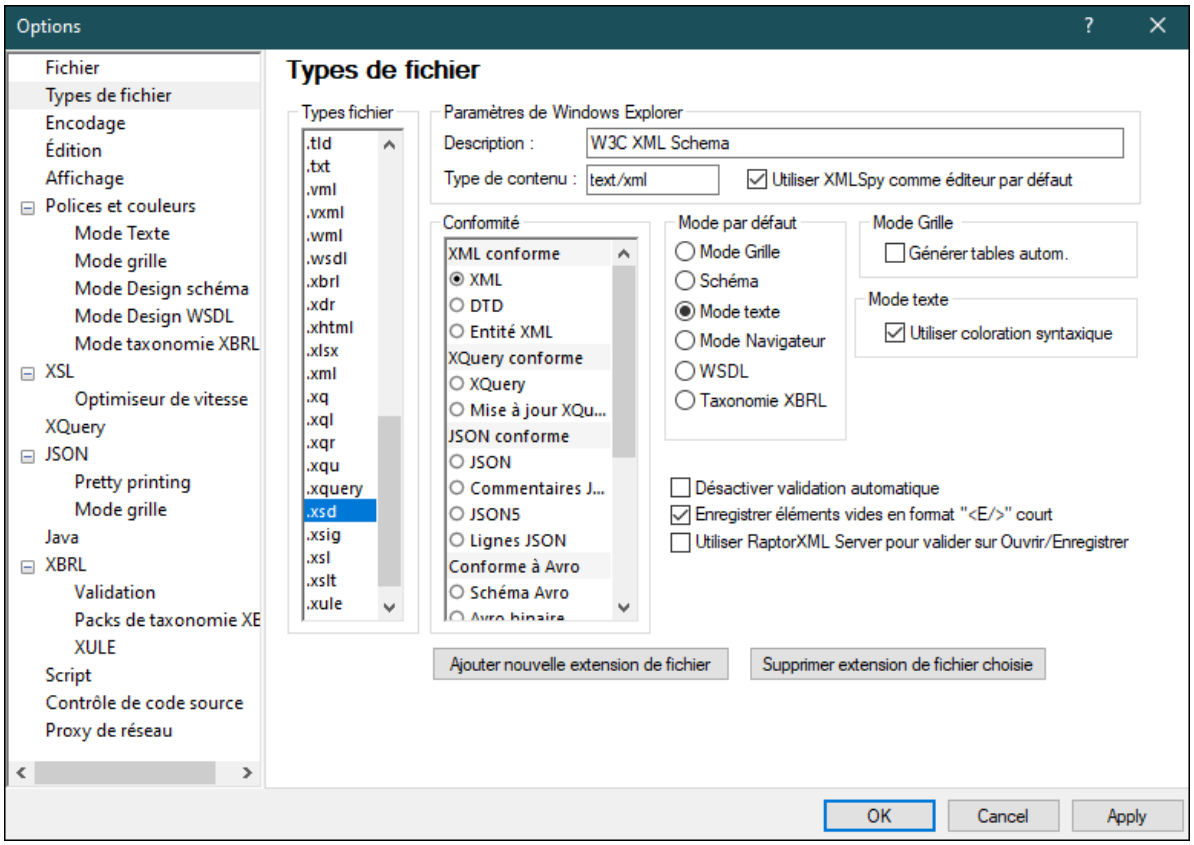
1. Cliquer sur la commande **Fichier | Ouvrir**.
2. Chercher le fichier `AddressFirst.xsd`, qui est situé dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`, le sélectionner, et cliquer sur **Ouvrir**. Le fichier s'ouvre dans le Mode Schéma.
3. Passer d'un mode à l'autre en cliquant sur les onglets d'affichage situés en bas de la fenêtre Principale (Mode Texte, Mode Grille, etc). Vous pourrez voir le document de Schéma XML dans le Mode Texte, Mode Grille, Mode Schéma et Mode Navigateur.
4. Vous pouvez également changer de mode en sélectionnant le mode que vous souhaitez depuis les options disponibles dans le menu **Modes**. Essayez de changer le mode du document `AddressFirst.xsd` en utilisant les commandes de menu **Mode**.
5. Fermer le document (par le biais de **Fichier | Fermer**).

Changer le mode par défaut d'un type de document

Tous les documents présentant une extension `.xsd` s'ouvriront par défaut dans le Mode Schéma. Vous pouvez changer le mode d'ouverture par défaut de n'importe quel document dans le dialogue Options. Procédons à cette étape pour les documents `.xsd`.

1. Cliquer sur la commande **Outils | Options** et rendez-vous dans la section *Types de fichier* (*capture d'écran ci-dessous*).
2. Dans le panneau *Types de fichier*, descendre jusqu'à `.xsd` et sélectionner cette extension (*marquée dans la capture d'écran*).
3. Dans le panneau *Mode par défaut*, sélectionner Mode Texte.

© 2016-2022 Altova GmbH



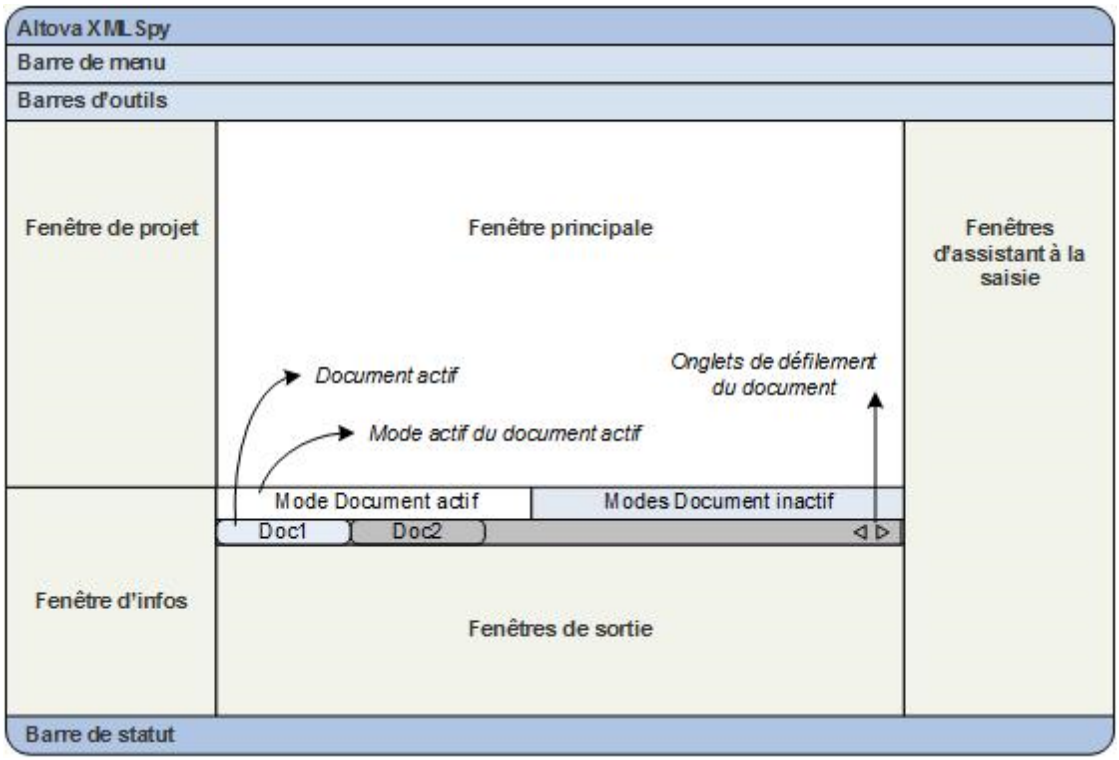
4. Cliquer sur **OK**.
5. Cliquer sur la commande **Fichier | Ouvrir** et ouvrir le fichier `AddressFirst.xsd`. Le fichier s'ouvre dans le Mode Texte.
6. Passer au Mode Schéma pour afficher le fichier dans ce mode puis fermer le fichier (**Fichier | Fermer**).
7. Retourner au dialogue Options (**Outils | Options**), et, dans la section *Types de fichier*, changer le mode par défaut des fichiers `.xsd` pour retourner au Mode Schéma.

Note : Dans la section *Types de fichier* du dialogue Options (*capture d'écran ci-dessus*), vous pouvez changer le mode par défaut de n'importe quelle extension de fichier recensée dans la liste. Il est possible d'ajouter une nouvelle extension de fichier à la liste par le biais du bouton **Ajouter nouvelle extension de fichier**.

2.1.2 Les fenêtres

Par défaut, les différentes fenêtres sont situées autour de la fenêtre principale (*voir capture d'écran ci-dessous*) et sont organisées dans les groupes de fenêtres suivants :

- Fenêtre Projet
- Fenêtre Info
- Aides à la saisie (variées, selon le type de document actif actuellement)
- Fenêtre de sortie : Messages, Graphiques, XPath, XSL Outline, Rechercher dans fichiers, Rechercher dans Schémas, Rechercher dans XBRL



Dans cette section, vous allez apprendre à allumer et éteindre l'affichage de groupes de fenêtres et à déplacer les fenêtres autour de l'écran. Il est utile de savoir déplacer les fenêtres lorsque vous nécessitez plus d'espace sur l'interface.

Allumer et éteindre l'affichage des groupes de fenêtres

Les groupes de fenêtres (Fenêtre Projet, Fenêtre Info, Aides à la saisie, Fenêtre de sortie) peuvent être déplacés ou dissimulés en les allumant ou en les éteignant à l'aide des commandes dans le menu **Fenêtre**. Un groupe de fenêtre affiché peut aussi être dissimulé en cliquant avec la touche de droite sur sa barre de titre et en sélectionnant la commande **Masquer**. Une fenêtre masquée peut uniquement être affichée par le biais du menu **Fenêtre**.

Ouvrir un des fichiers XML contenus dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial` et entraînez-vous à effectuer ces commandes de base jusqu'à ce qu'elles vous soient familières. Pour plus d'informations concernant l'affichage et le masquage de groupes de fenêtres, voir la section, Interface XMLSpy.

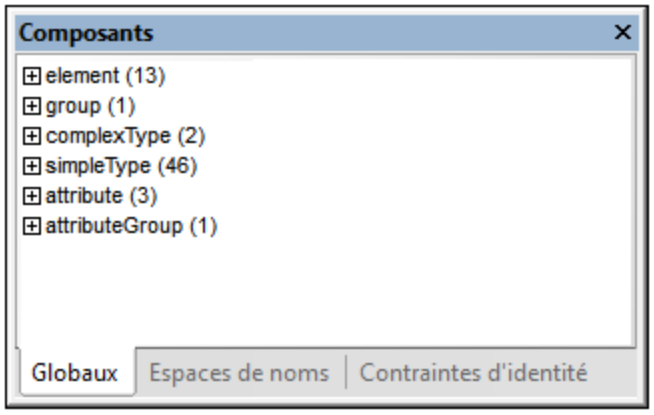
Statut Enregistrer et statut Sauvegarder

Par défaut, XMLSpy sauvegarde des documents non-enregistrés à des intervalles de 5 secondes. Chaque onglet de fichier en bas de la Fenêtre principale fournit des informations par le biais de symboles d'indicateur concernant le statut enregistré / non-enregistré du fichier et son statut sauvegardé. Vous devriez connaître la signification de ces indicateurs car ils apparaîtront sans cesse pendant votre travail. Voir la section Sauvegarde automatique des fichiers pour plus d'information concernant ces indicateurs.

© 2016-2022 Altova GmbH

Déplacer les fenêtres autour de l'écran

Une fenêtre individuelle peut flotter librement sur l'interface ou peut y être ancrée. Une fenêtre peut aussi être ancrée en tant qu'onglet dans un groupe de fenêtres (*les groupes de fenêtres sont expliqués ci-dessus*). Par exemple, la capture d'écran ci-dessous affiche l'assistant à la saisie Composants dans le Mode Schéma qui possède trois fenêtres à onglet : la fenêtre Globaux, la fenêtre Espaces de noms et la fenêtre Contraintes d'identité.



Vous pouvez faire flotter une fenêtre ou l'ancrer en utilisant une des méthodes suivante dans n'importe quel mode :

- Double-cliquer la barre de titre de la fenêtre. Si elle était ancrée, la fenêtre flotte à présent. Si elle flotte, elle s'ancrera sur la position à laquelle elle était fixée précédemment.
- Cliquer avec le bouton de droite sur la barre de titre d'une fenêtre puis choisir la commande requise (**Flottant** ou **Ancrage**).
- Glisser la fenêtre (en utilisant sa barre de titre en guise de poignée) hors de sa position d'ancrage pour la faire flotter. Faire glisser une fenêtre flottante (par sa barre de titre) vers l'emplacement d'ancrage. Deux groupes de flèches bleues apparaissent. Le groupe extérieur de quatre flèches permet un ancrage par rapport à la fenêtre d'application (le long du rebord supérieur, droit, bas ou gauche de la GUI). Le groupe intérieur de flèches permet un ancrage par rapport à la fenêtre au-dessus de laquelle le curseur est placé actuellement. Si vous déposez une fenêtre glissée sur le bouton au centre du groupe intérieur de flèches (ou sur la barre de titre d'une fenêtre) permet d'ancrer la fenêtre glissée en tant que fenêtre à onglet dans le cadre de la fenêtre dans laquelle elle a été déposée.

Pour détacher une fenêtre à onglet, double-cliquer sur son onglet. Pour faire glisser une fenêtre à onglet hors d'un groupe de fenêtres à onglets, faites-la glisser par son onglet.

Afin de vous entraîner, vous pouvez ouvrir n'importe quel fichier de Schéma XML depuis le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial` et, tant que vous vous trouvez dans le mode Schéma, essayez les méthodes décrites ci-dessus jusqu'à ce que vous soyez en mesure de déplacer aisément les fenêtres sur l'interface.

2.1.3 Menus et barres d'outils

Dans cette section du tutoriel, vous apprendrez très rapidement à utiliser les fonctions principales des menus et barres d'outils de XMLSpy.

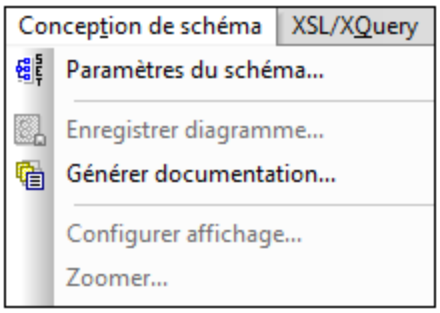
Menus

Il existe deux barres de menu : (i) un menu par défaut qui est affiché lorsqu'aucun document n'est ouvert, et (ii) le menu d'application complet de XMLSpy, qui s'affiche dès qu'un document est ouvert. Procéder comme suit :

1. Fermer tous les documents ouverts avec la commande de menu **Fichier | Tout fermer**. Vous voyez apparaître le menu par défaut.
2. Ouvrir le fichier `AddressFirst.xsd` en cliquant sur son nom dans la liste des fichiers les plus récemment ouverts situés en bas du menu **Fichier**. Lorsque le fichier s'ouvre dans le mode Schéma, le menu passera au plein menu d'application XMLSpy.

Les menus sont organisés principalement conformément à la fonction et une commande dans un menu est uniquement activée quand elle peut être exécutée sur le point de curseur ou pour une sélection dans le mode actuel du document actif. Pour comprendre les facteurs qui déterminent si une commande de menu est activée ou pas, procéder comme suit :

1. Cliquer sur le menu **Conception de schéma**. Veuillez noter que les commandes **Enregistrer le diagramme**, **Configurer le Mode** et **Zoom** sont désactivés (*capture d'écran ci-dessous*).



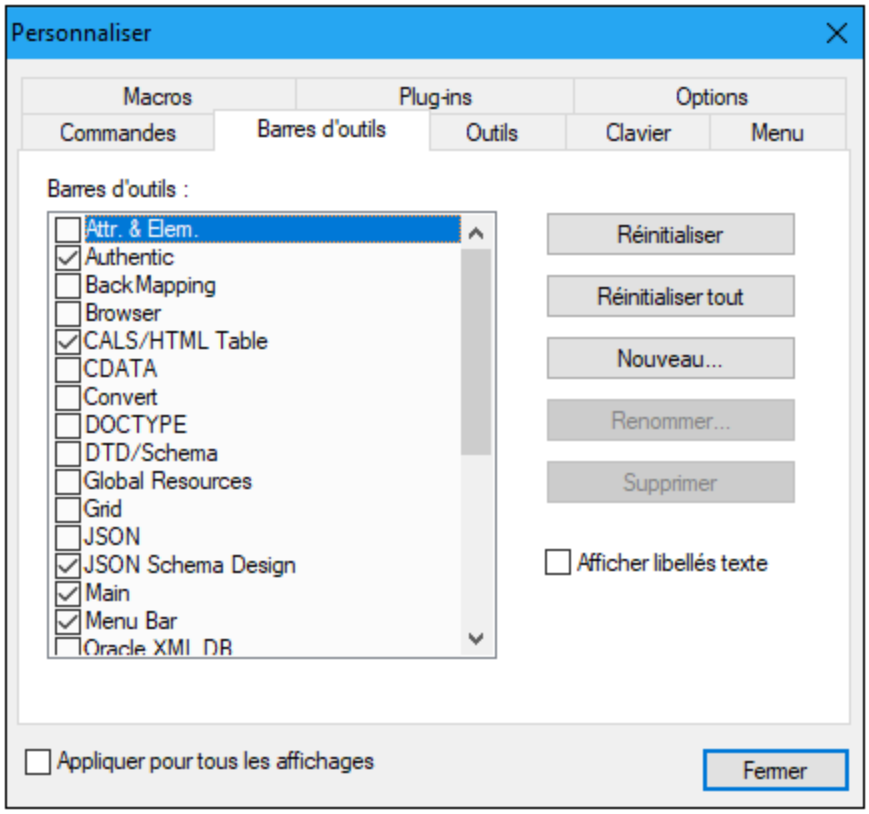
2. Cliquer dans un espace vierge situé en dehors du menu pour le faire disparaître. Puis cliquer sur l'icône **Afficher le diagramme**  située à la gauche de composant d'élément. Cela vous amène au mode de Modèle de contenu du mode Schéma (le deuxième des deux modes du Mode de Schéma ; le premier est Aperçu de Schéma). Si vous cochez maintenant le menu de Design de Schéma, vous verrez que les commandes **Enregistrer le diagramme**, **Configurer le Mode** et **Zoom** ont été activées. Elles sont activées uniquement dans le Mode Modèle de contenu du Mode de Schéma et ni dans l'Aperçu de Schéma du Mode de Schéma, ni dans un autre mode. Veuillez noter également que seuls les fichiers de Schéma XML peuvent être ouverts dans le Mode de Schéma.
3. Un fichier de Schéma XML est aussi un fichier XML, il est donc affiché sous la forme d'un fichier XML dans le Mode Texte et le Mode Grille et toutes les commandes de menu qui s'appliquent aux fichiers XML seront activées dans ces modes. Comparer les commandes dans le menu **Éditer** (qu'elles soient activées ou pas) dans le Mode de schéma et le Mode Texte.
4. Ensuite, comparer les commandes dans le menu **XML | Insérer** (activé ou désactivé) dans le Mode Texte et le Mode Grille. Les commandes dans ce menu sont uniquement activées dans le Mode Grille.

© 2016-2022 Altova GmbH

Pour des descriptions de toutes les commandes de menu, voir la section Références d'utilisateur de la documentation utilisateur.

Barres d'outils

L'affichage de barres d'outils varie conformément au mode actuel. Les paramètres par défaut de l'application livre les barres d'outils correctes pour chaque mode et seront différentes pour chaque mode. Néanmoins, vous pouvez personnaliser des barres d'outils dans l'onglet *Barres d'outils* du dialogue Personnaliser (**Outils | Personnaliser | Barres d'outils**, *capture d'écran ci-dessous*).



À présent, entraînez-vous à déplacer les barres d'outils sur la GUI. Cliquer sur la poignée d'une barre d'outils et la tirer où vous voulez dans la GUI. (La poignée de la barre d'outils est indiqué par la ligne en pointillé verticale à gauche de chaque barre d'outils ; *voir capture d'écran ci-dessous*.)

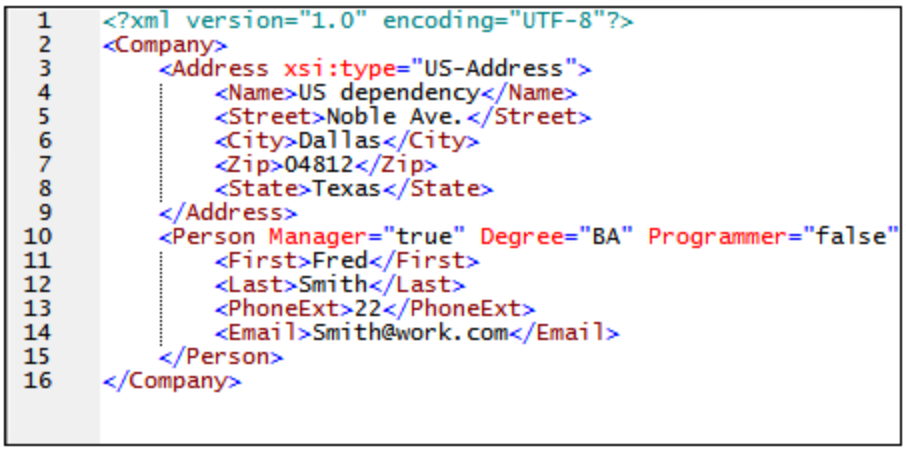


Essayez de glisser une barre d'outils à l'emplacement suivant : (i) une autre ligne dans la zone de la barre d'outils ; (ii) à gauche ou à droite de la barre d'outils ; (iii) le centre de la fenêtre principale ; (iv) ancrée au côté gauche ou droite de la fenêtre d'application (pour ce faire, la poignée de saisie doit être placée au-dessus du rebord gauche ou droite de la fenêtre d'application).

Une fois avoir terminé, fermer le fichier `AddressFirst.xsd`.

2.1.4 Paramètres du Mode Texte

Dans cette section, vous apprendrez comment configurer un document « pretty-print » et comment utiliser des signets pendant l'édition. Lorsqu'un document est imprimé en pretty-print, il est affiché dans le Mode Texte de manière à ce que chaque niveau XML hiérarchique inférieur soit un peu plus en retrait que le niveau précédent (*voir la capture d'écran ci-dessous*). Les signets vous permettent de marquer les positions des documents auxquelles vous souhaitez retourner rapidement.

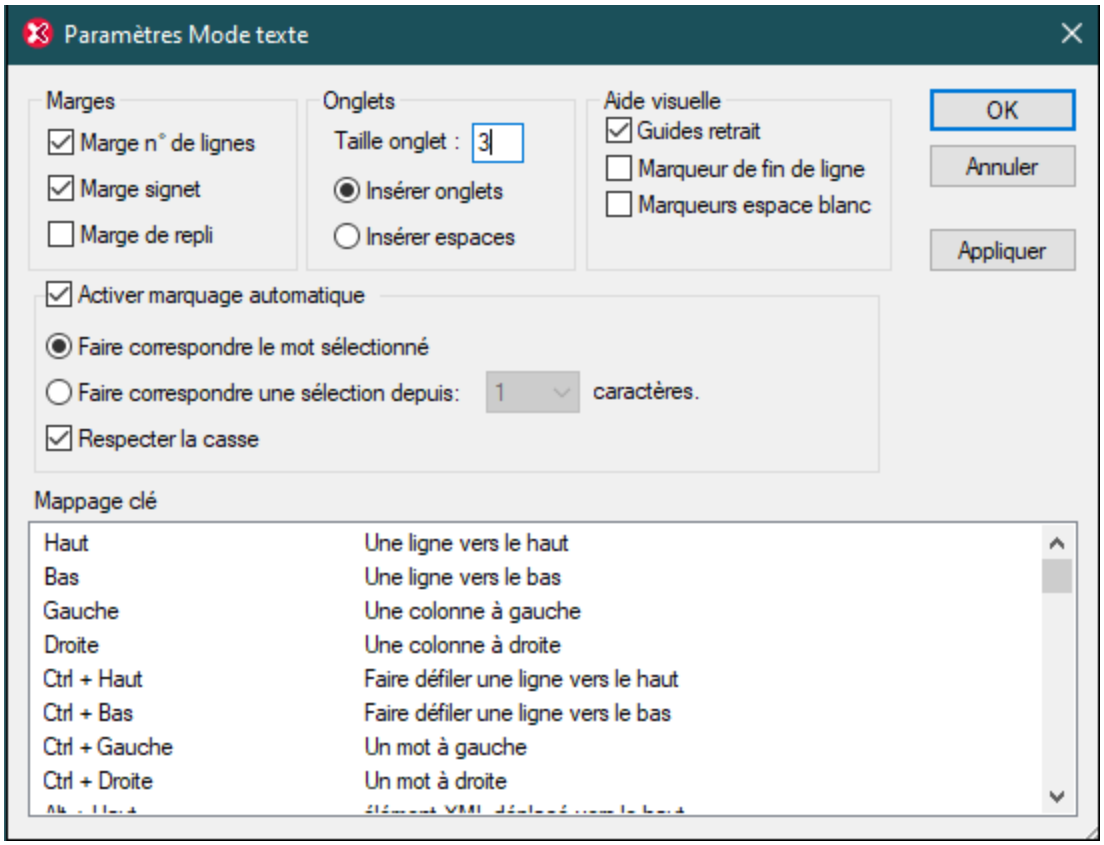


Pretty-print

L'impression en pretty-print implique deux étapes : (i) activer la fonction de pretty-print et spécifier l'ampleur du retrait, et (ii) appliquer le pretty-print.

1. Ouvrir le fichier **`CompanyFirst.xml`**, qui se trouve dans le dossier **`C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`**.
2. Passez au Mode Texte si celui-ci n'est pas le mode par défaut de démarrage pour les documents XML.
3. Sélectionnez la commande de menu **Mode | Paramètres Mode Texte** pour ouvrir le dialogue des Paramètres Mode Texte (*voir la capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



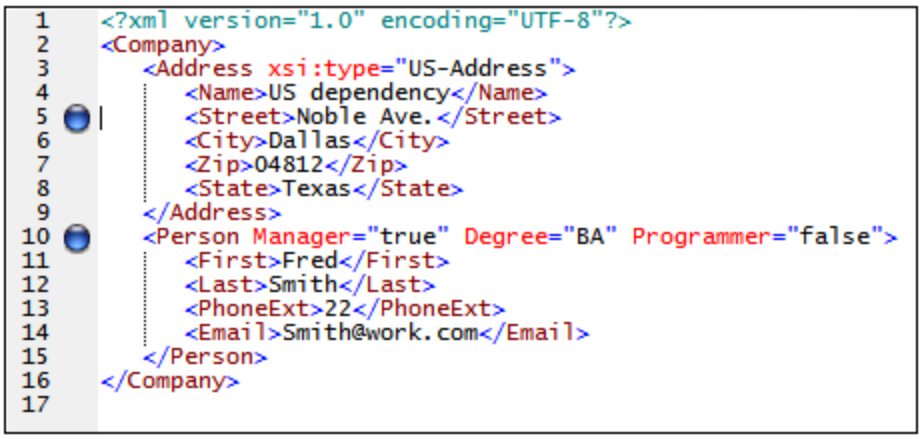
4. Dans le panneau Onglets, réduire la taille de l'onglet à 3. Laisser la sélection du bouton radio *Insérer onglets* inchangé. Cela aura pour conséquence que le retrait en pretty-printing prendra la forme d'un onglet (au lieu d'espaces) et que chaque onglet aura une largeur équivalent à trois espaces. Cliquer sur **OK** une fois terminé.
5. Cliquez sur la commande de menu **Édition | Pretty-Print**. Ceci applique pretty print. Le document apparaîtra en pretty-print avec les nouvelles valeurs d'onglet.
6. Ouvrir le dialogue de Paramètres Mode Texte (**Mode | Paramètres Mode Texte**) et, dans le panneau *Aide visuelle*, passez aux marqueurs de fin de ligne.
7. Dans le Mode Texte, allez jusqu'à la fin de n'importe quelle ligne et supprimez le marqueur de fin de ligne, la ligne suivante sautera d'une ligne vers le haut.
8. Passez au Mode Grille puis retournez au Mode Texte
9. Cliquez sur la commande de menu **Édition | Pretty-Print**. Le document apparaîtra en pretty-print et le marqueur de fin de ligne que vous avez supprimé sera rétabli.

Note : si, dans la section Pretty-printing du dialogue des options (**Tools | Options | Pretty-printing**), vous décochez la case *Utiliser Indentations* et pretty-print, alors toutes les lignes commenceront sans aucune indentation.

Signets

Les signets sont placés dans une marge à signets à gauche des lignes que vous souhaitez marquer. Les signets vous permettront de vous déplacer rapidement dans le document.

1. Dans le dialogue de Paramètres Mode Texte (**Mode | Paramètres Mode Texte**, *voir la capture d'écran ci-dessus*) assurer que l'option Marge signet dans le panneau *Marges* a été sélectionné. Cliquez sur **OK** une fois terminé.
2. Dans le Mode Texte du fichier `CompanyFirst.xml`, placez le curseur à un endroit quelconque de la ligne que vous souhaitez marquer, puis choisissez la commande de menu **Éditer | Insérer/Supprimer signet**. La ligne sera marquée d'un signet et celui-ci sera indiqué par un signet bleu dans la marge de signet (*voir la capture d'écran ci-dessous*).
3. Créer un signet sur une autre ligne de la même manière que décrit dans l'étape 2.

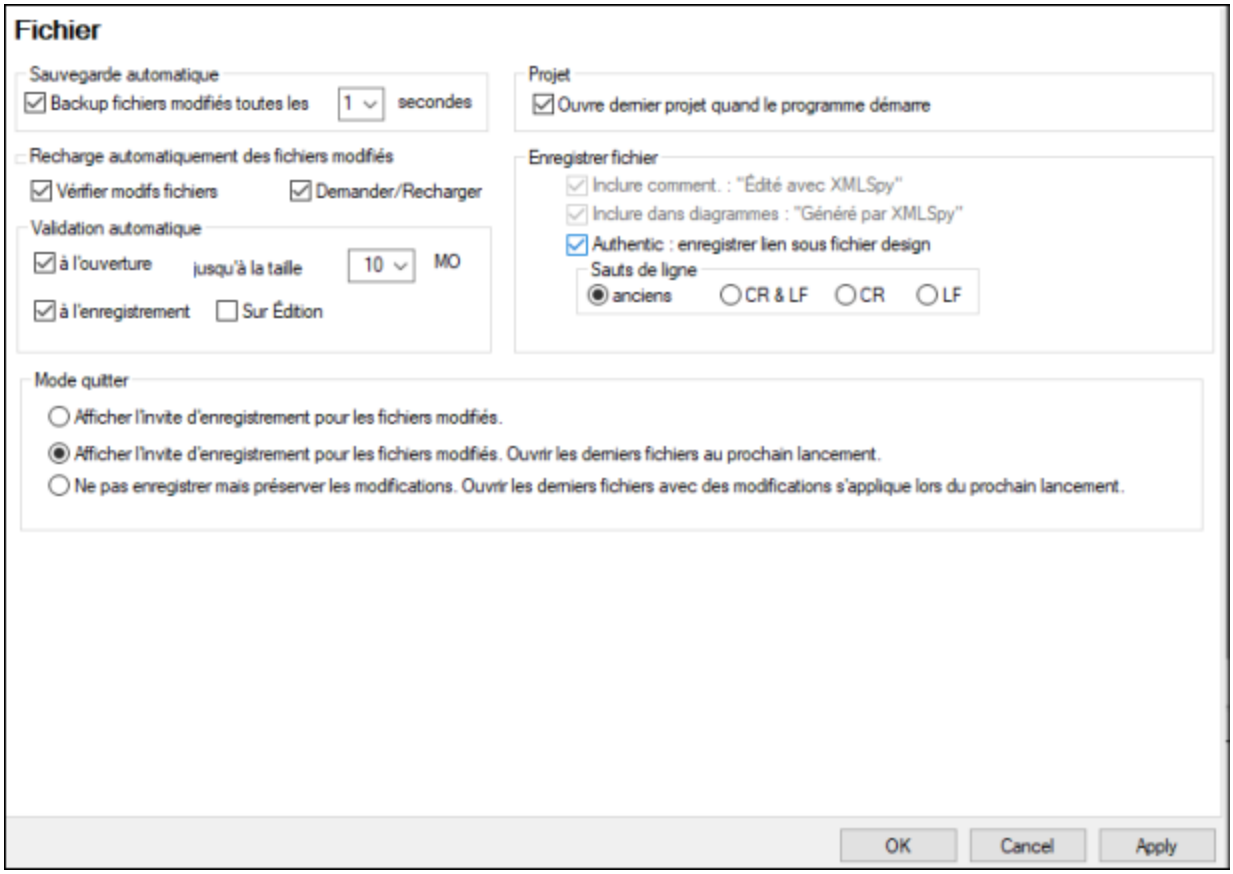


4. Appuyez sur **F2** (ou sur la commande **Éditer | Aller au signet suivant**) pour descendre dans le document vers le signet suivant. Appuyez sur **Shift+F2** (ou la commande **Éditer | Aller au signet suivant**) pour remonter dans le document vers le signet précédent. Renouveler l'opération autant de fois que vous le souhaitez.
5. Placer le curseur dans une des lignes marquées et sélectionner la commande de menu **Éditer | Insérer/Supprimer signet**. Le signet est supprimé.
6. Enregistrer et fermer le fichier. Aucune information de signet n'est enregistrée avec le fichier. Rouvrir le fichier pour vérifier cela.

2.1.5 Options d'application

Étant donné que XMLSpy contient une pléthore de fonctions, vous avez la possibilité de définir toute une série d'options et qui peuvent influer considérablement sur les différents aspects de votre travail. Il serait donc avantageux pour votre travail au long terme d'être sensible aux paramètres d'application que vous pouvez modifier pour l'adapter à vos exigences de travail et style de travail. Ces paramètres sont accessibles dans le dialogue Options *(capture d'écran ci-dessous)*, que vous pouvez ouvrir par le biais de la commande de menu **Outils | Options**.

© 2016-2022 Altova GmbH



Les paramètres sont organisés en section, et recensées dans le panneau de gauche. Nous suggérons que vous consultiez chaque section pour obtenir une idée du type des paramètres disponibles. Dans la liste ci-dessous, nous avons attiré votre attention aux paramètres qui touchent certaines des fonctions utilisées communément. Pour des descriptions de tous les paramètres, rendez-vous dans la documentation du dialogue Options.

Lorsque vous modifiez un paramètre, cliquer sur **OK** pour enregistrer la modification dans le registre et pour fermer le dialogue. La touche **Appliquer** entraîne l'affichage des modifications dans les documents ouverts actuellement.

Fichier

- La sauvegarde automatique (des fichiers que vous éditez) peut être allumée/éteinte.
- La validation sur modification réalise des contrôles de la bonne formation et des contrôles de validation pendant votre saisie. Si cela vous gêne, vous pouvez éteindre cette fonction et effectuer des contrôles de bonne formation et des contrôles de validation à tout moment selon votre décision.

Types de fichier

- Définir le mode par défaut par types de fichier. Vous pouvez sélectionner le mode avec lequel vous vous sentez le plus à l'aise pour chaque type de document.

Affichage

- Dans la barre de titre XMLSpy, montrer soit uniquement le nom de fichier ou le chemin de fichier complet. Veuillez noter que vous pouvez voir le chemin de fichier si vous planez sur le nom du fichier dans l'onglet de fichier en bas de la fenêtre Principale.

Polices et couleurs

- Vous pouvez configurer les polices, leurs tailles et les couleurs du texte et des autres composants. Il existe différents paramètres pour chaque mode.

© 2016-2022 Altova GmbH

2.2 Schémas XML : fonctions de base

Un Schéma XML décrit la structure d'un document XML. Un document XML peut être validé par rapport à un Schéma XML pour contrôler si celui-ci est conforme aux exigences spécifiées dans le schéma. Si cela est le cas, il est désigné comme **valide** ; dans le cas contraire, il est **invalide**. Les Schémas XML permettent aux concepteurs du document de spécifier la structure autorisée et le contenu d'un document XML et de contrôler si un document XML est valide.

La structure et la syntaxe d'un document de Schéma XML est complexe et étant lui-même un document XML, un Schéma XML doit être valide conformément aux règles de la spécification XML Dans XMLSpy, le Mode Schéma vous permet de créer aisément des Schémas XML valides en utilisant des techniques de glisser-déposer graphiques. Le document de Schéma XML que vous construisez est également éditable dans le Mode Texte et le Mode Grille, mais il est plus facile de le créer et de le modifier dans le Mode Schéma.

Objectifs

Dans cette section du tutoriel, vous apprendrez à éditer des schémas XML dans le Mode Schéma. En particulier, vous apprendrez à faire les opérations suivantes :

- Créer un nouveau fichier de schéma
- Définir des espaces de noms pour le schéma
- Définir un modèle de contenu de base
- Ajouter des éléments au modèle de contenu à l'aide des menus de contexte et glisser/déposer
- Configurer le Mode Modèle de contenu

Une fois après avoir achevé la création du schéma de base, vous pouvez passer à la section suivante du tutoriel, qui vous apprendra à travailler avec les fonctions les plus avancées du Schéma XML de XMLSpy. Cette section avancée est suivie par une section concernant la navigation et documentation de schéma dans XMLSpy.

Commandes utilisées dans cette section

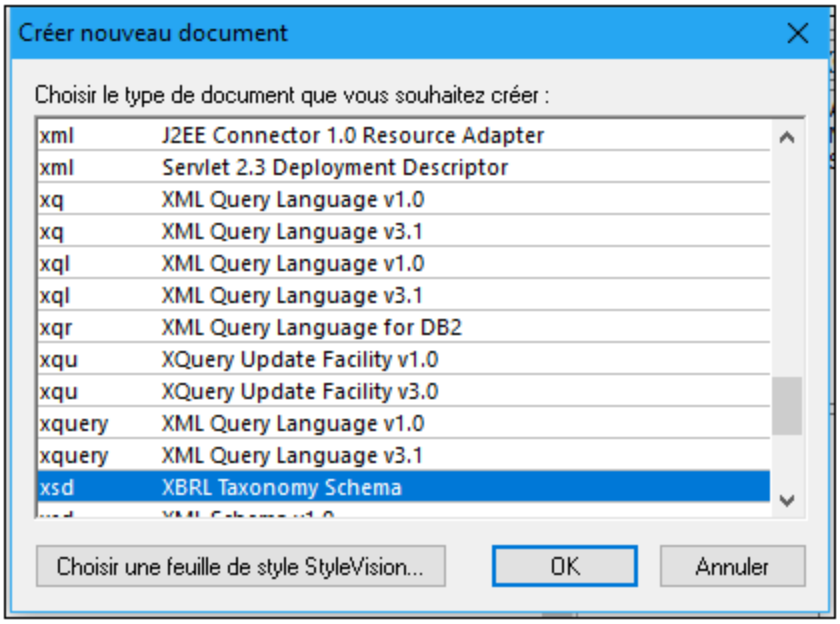
Dans cette section du tutoriel, vous utiliserez le Mode Schéma en exclusivité. Les commandes suivantes sont utilisées :

	Diagramme d'affichage (ou Mode d'Affichage du modèle de contenu). Cette icône est située à gauche de tous les composants globaux dans l'Aperçu de Schéma. Un clic sur l'icône permet l'affichage du modèle de contenu du composant global associé.

2.2.1 Créer un nouveau fichier de Schéma XML

Pour créer un nouveau fichier de schéma XML :

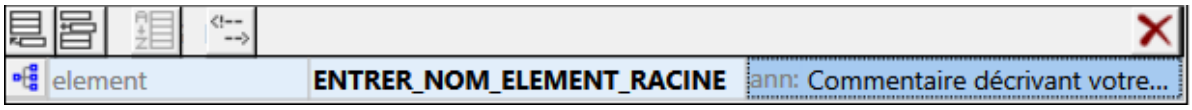
1. Sélectionner l'option de menu **Fichier | Nouveau**. Le dialogue Créer nouveau document s'ouvre.



2. Dans le dialogue, sélectionner l'entrée `XSD (Schéma XML v1.0)` (la description du document et la liste dans la fenêtre peuvent varier de celle représentée dans la capture d'écran) et confirmer avec **OK**. Un fichier de schéma vide apparaît dans la Fenêtre principale dans le Mode Schéma.
3. Dans la barre d'outils du Design de schéma, cliquer sur le bouton du mode **XSD 1.0** (*voir capture d'écran ci-dessous*) afin que le mode Schéma se trouve dans le mode d'édition XSD 1.0.

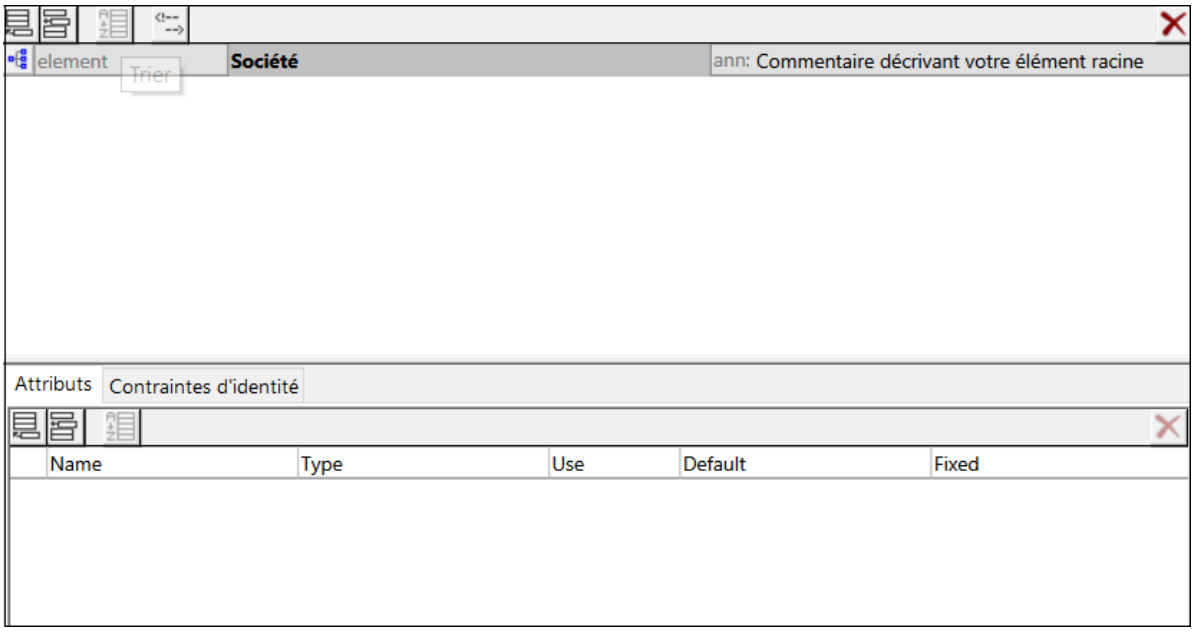


4. Vous êtes invité à saisir le nom de l'élément racine.



5. Double-cliquer dans le champ marqué et saisir `Company`. Confirmer avec **Entrée**. `Company` est désormais l'élément racine de ce schéma et est créé en tant qu'un élément global. Le mode que vous voyez dans la fenêtre principale *(capture d'écran ci-dessous)* est appelé Aperçu de schéma. Il fournit un aperçu du schéma en affichant une liste de tous les composants globaux dans le panneau supérieur de la fenêtre principale ; le panneau inférieur affiche les attributs et les contraintes d'identité du composant global sélectionné. (Vous pouvez voir et éditer le modèle de contenu des composants globaux individuels en cliquant sur l'icône Afficher le diagramme à gauche de ce composant global.)

© 2016-2022 Altova GmbH



6. Dans le champ Annotations (`ann`) de l'élément `Company`, saisir la description de l'élément, dans ce cas, `Root element`.
7. Cliquer sur l'option de menu **Fichier | Enregistrer**, et enregistrez votre Schéma XML avec le nom que vous souhaitez (`AddressFirst.xsd`, par exemple).

Les symboles de cercles de couleur dans l'onglet du fichier indiquent le statut de sauvegarde du fichier. Voir Sauvegarde automatique des fichiers pour une description de ces indicateurs.

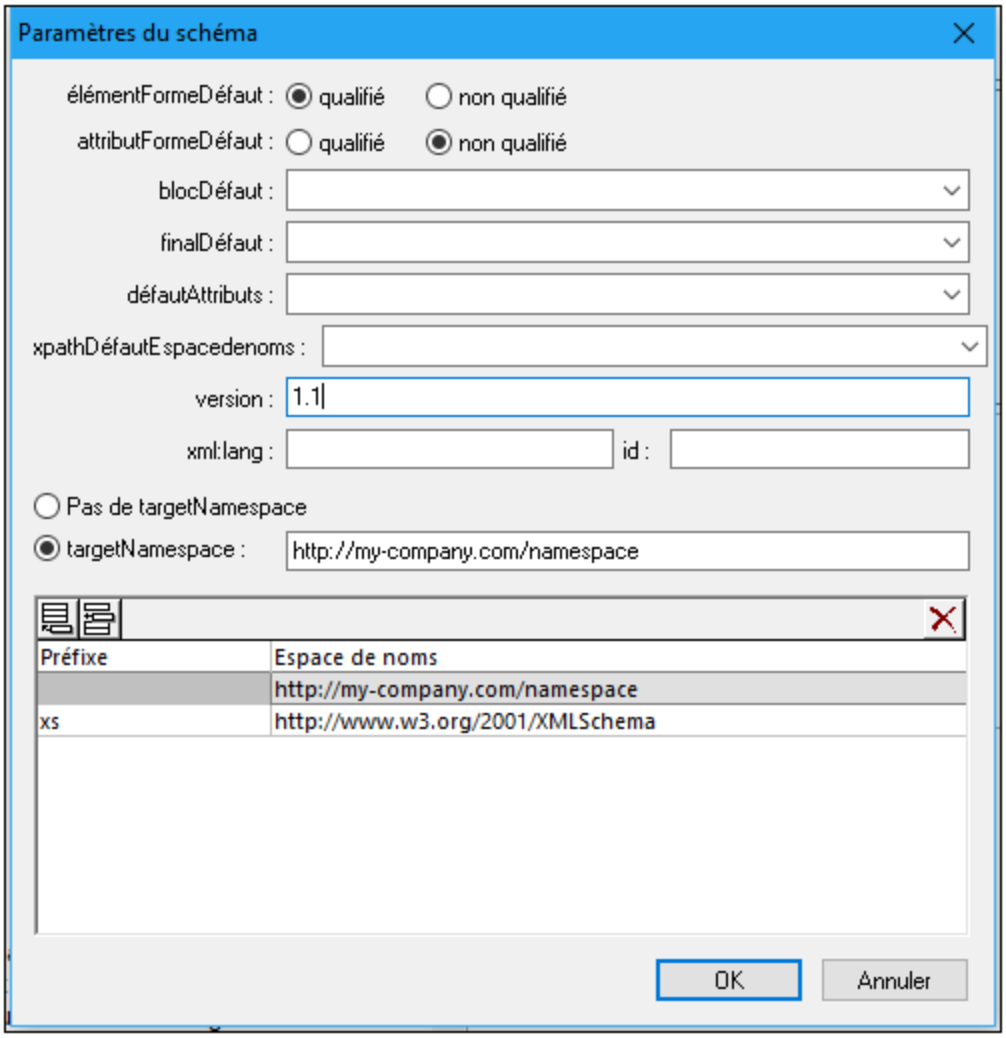
2.2.2 Définir les espaces de noms

Les espaces de noms XML jouent un rôle important dans le domaine des Schémas XML et des documents XML. Un document de Schéma XML doit référencer un espace de noms de Schéma XML et, en option, il peut définir un espace de noms cible pour l'instance de document XML. En tant que concepteur de schéma, c'est à vous de décider comment définir ces deux espaces de noms (en substance, avec quels préfixes.)

Dans le Schéma XML que vous créez, vous allez définir un espace de noms cible pour les instances de document XML. (La référence à l'espace de nom de Schéma XML requise est créée automatiquement par XMLSpy lorsque vous créez un nouveau document de Schéma XML.)

Pour créer un espace de noms cible :

1. Sélectionnez l'option de menu **Conception de schéma | Paramètres du Schéma**. Cela permet d'ouvrir le dialogue Paramètres du schéma. (*voir la capture d'écran ci-dessous*).



2. Cliquez sur le bouton radio *Target Namespace*, et saisissez `http://my-company.com/namespace`. Dans XMLSpy, l'espace de noms que vous saisissez en tant que l'espace de noms cible est créé en tant que l'espace de nom cible du document de Schéma XML et affiché dans la liste des espaces de noms dans le panneau inférieur du dialogue.
3. Confirmez avec le bouton **OK**.

Veuillez noter les points suivants :

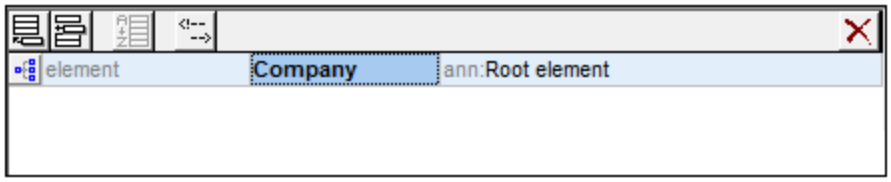
- L'espace de noms de Schéma XML est créé automatiquement par XMLSpy et reçoit le préfixe `xs:`.
- Lorsque l'instance de document XML est créée, elle doit avoir l'espace de noms cible définie dans le Schéma XML pour que le document XML soit valide.

2.2.3 Définir un modèle de contenu

Dans l'Aperçu de Schéma, vous avez déjà créé un élément global appelé `Company`. Cet élément doit contenir un élément `Address` et un nombre illimité d'éléments `Person`. Ceci devient alors le modèle de contenu de

 © 2016-2022 Altova GmbH

l'élément Company. Les composants globaux qui peuvent présenter des modèles de contenu sont des éléments, des complexTypes et des groupes d'élément. Dans XMLSpy, le modèle de contenu d'un composant global est affiché dans le Mode de Modèle de contenu du Mode de schéma. Pour consulter et éditer le modèle de contenu d'un composant global, cliquer sur l'icône Afficher le diagramme situé à gauche du composant global.

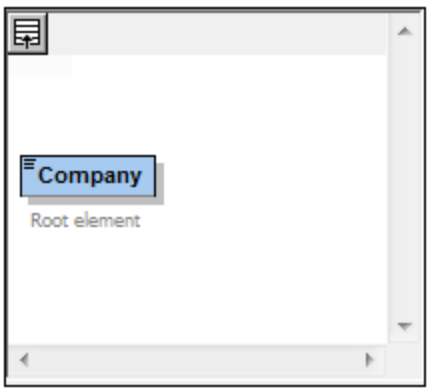


Dans cette section, vous allez créer le modèle de contenu de l'élément **Company**.

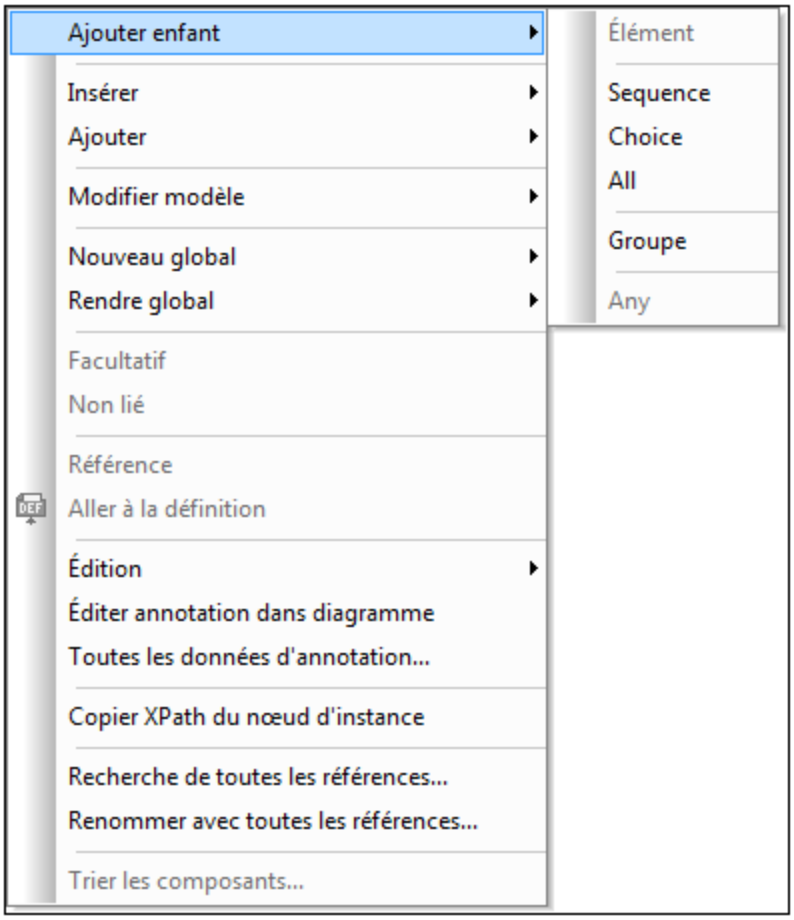
Créer un modèle de contenu de base

Pour créer le modèle de contenu de l'élément **Company** :

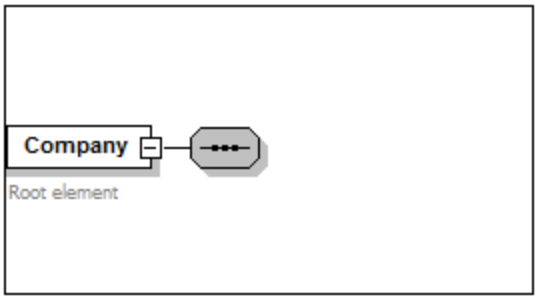
1. Dans l'Aperçu de schéma, cliquer sur l'icône Afficher le diagramme de l'élément **Company**. Cela permet d'afficher le modèle de contenu de l'élément **Company** (*voir la capture d'écran ci-dessous*), actuellement vide. En alternative, vous pouvez double-cliquer l'entrée **Company** dans l'assistant d'entrée Composants pour afficher son modèle de contenu.



2. Un modèle de contenu consiste en des **compositeurs** et **composants**. Les compositeurs spécifient la relation entre deux composants. À ce point du modèle de contenu de **Company**, vous devez ajouter un compositeur enfant à l'élément **Company** pour ajouter un élément enfant. Afin d'ajouter un compositeur, cliquez avec la touche de droite sur l'élément **Company**. À partir du menu de contexte qui apparaît, choisissez **Ajouter enfant | Séquence**. (Séquence, Choix et Tous sont les trois compositeurs qui peuvent être utilisés dans un modèle de contenu.)

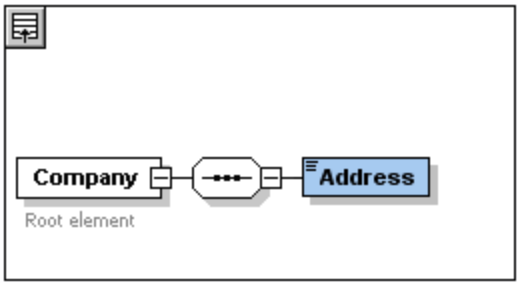


Cela permet d'insérer le compositeur de Séquence, qui définit que les composants ultérieurs doivent apparaître dans la séquence spécifiée.

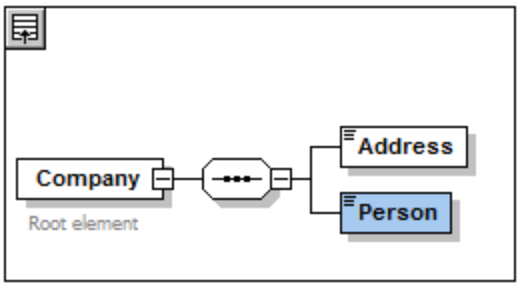


3. Cliquez avec la touche de droite sur le compositeur de séquence et choisissez **Ajouter enfant | Élément**. Un composant d'élément sans nom est ajouté.
4. Saisir **`Address`** en tant que le nom de l'élément et confirmer avec **Entrée**.

 © 2016-2022 Altova GmbH

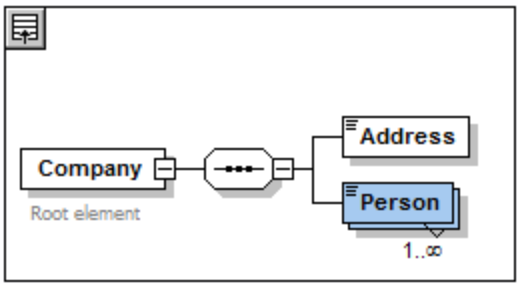


5. Cliquez à nouveau avec la touche de droite sur le compositeur de séquence, **Ajouter enfant | Élément**. Nommez le nouveau composant d'élément créé `Person`.



Jusqu'à présent, vous avez défini un schéma qui permet d'afficher une adresse et une personne par entreprise. Nous souhaitons augmenter le nombre d'éléments **`Person`**.

6. Cliquer avec la touche de droite sur l'élément **`Person`** et sélectionnez **Non lié** depuis le menu contexte. L'élément **`Person`** dans le diagramme montre maintenant le nombres d'occurrences autorisées : 1 à infini.



En alternative, dans l'assistant de saisie Détails, vous pouvez éditer les champs **`minOcc`** et **`maxOcc`** pour spécifier le nombre autorisé d'occurrences, dans ce cas 1 et non lié, respectivement.

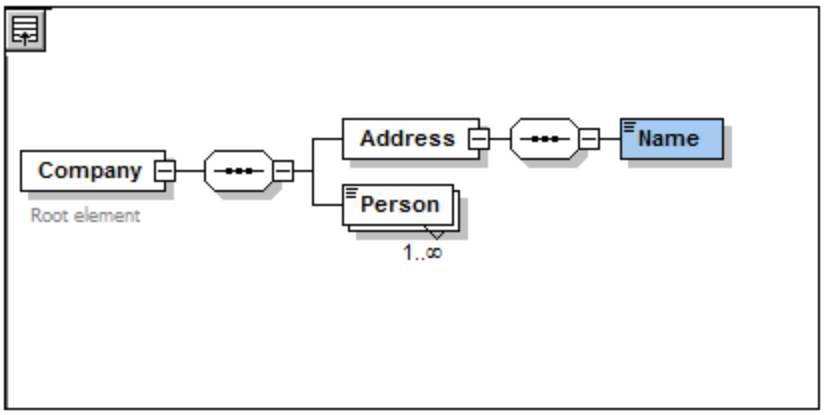
Ajouter des niveaux supplémentaires à la structure de modèle de contenu

Le modèle de contenu de base que vous avez créé contient un niveau : un niveau enfant pour l'élément **`company`** qui contient les éléments **`Address`** et **`Person`**. À présent, nous allons définir le contenu de l'élément **`Address`** afin qu'il contienne les éléments **`Name`**, **`Street`**, et **`City`**. Ceci est un deuxième niveau. Nous devons une nouvelle fois ajouter un compositeur enfant à l'élément **`Address`**, puis les composants d'éléments eux-mêmes.

Pour ce faire, procédez comme suit :

1. Cliquer avec la touche de droite sur l'élément **`Address`** pour ouvrir le menu contextuel et sélectionnez **Ajouter enfant | Séquence**. Cela permet d'ajouter le compositeur de Séquence.

2. Cliquez avec la touche de droite sur le compositeur de séquence et sélectionnez **Ajouter enfant | Élément**. Nommez le composant d'élément récemment créé `Name`.



Types complexes, types simples et types de données Schéma XML

Pour le moment, nous n'avons pas explicitement défini de type d'élément. Cliquez sur l'onglet **Text** pour afficher le Mode Texte de votre schéma (*voir la liste ci-dessous*). Vous constaterez qu'à chaque fois qu'un compositeur de Séquence a été inséré, l'élément `xs:sequence` a été inséré à l'intérieur de l'élément **`xs:complexType`**. En bref, les éléments **`Company`** et **`Address`** sont des types complexes puisqu'ils contiennent des éléments enfant. Un élément de type complexe est un élément qui contient des attributs ou éléments.

```
<xs:element name="Company">
   <xs:annotation>
     <xs:documentation>Root element</xs:documentation>
   </xs:annotation>
   <xs:complexType>
     <xs:sequence>
       <xs:element name="Address">
         <xs:complexType>
           <xs:sequence>
             <xs:element name="Name"/>
           </xs:sequence>
         </xs:complexType>
       </xs:element>
       <xs:element name="Person"/>
     </xs:sequence>
   </xs:complexType>
</xs:element>
```

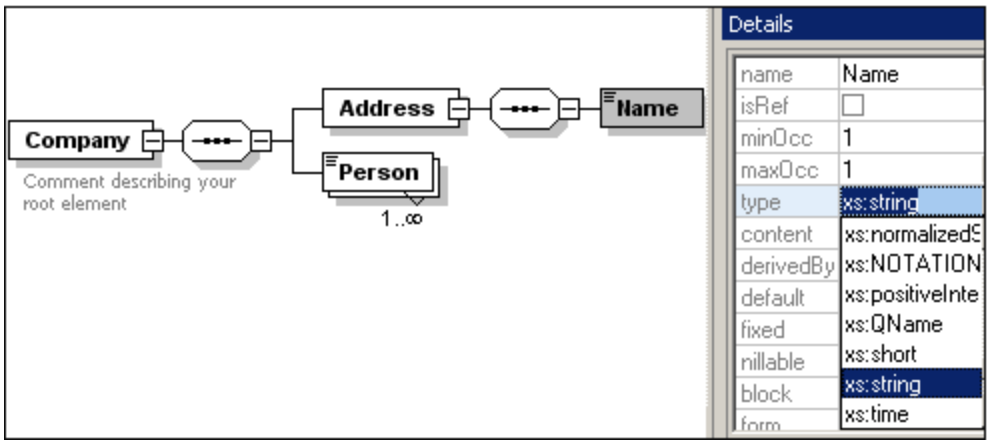
Les éléments de type simple, en revanche, ne contiennent que du texte et ne possèdent pas d'attributs. Le texte peut être des chaînes, des dates, des nombres, etc. Nous souhaitons que l'enfant **`Name`** de **`Address`** ne contienne que du texte. Il s'agit d'un type simple, et nous souhaitons que le contenu sous forme de texte soit limité à un string. Pour ce faire, nous pouvons utiliser le type de données de Schéma XML **`xs:string`**.

Pour définir que l'élément **`Name`** soit de ce type de données :

1. Cliquez sur l'onglet **Schéma** pour retourner au Mode Schéma.
2. Cliquez sur l'élément **`Name`** pour le sélectionner.

© 2016-2022 Altova GmbH

3. Dans l'assistant de saisie Détails, à partir du menu déroulant de la zone de liste modifiable `type`, sélectionnez l'entrée **`xs:string`** .



Veuillez noter que **`minOcc`** et **`maxOcc`** ont tous deux une valeur de 1, montrant que cet élément ne se produit qu'une fois.

La représentation textuelle de l'élément `Name` est comme suit :

```
<xs:element name="Name" type="xs:string"/>
```

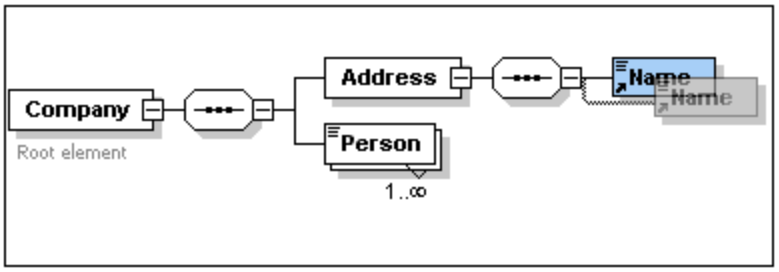
Note : un élément de type simple peut présenter n'importe quel type de données de Schéma XML. Dans tous ces cas, l'icône indiquant le contexte textuel apparaît dans la fenêtre d'éléments.

2.2.4 Ajouter des éléments par glisser-déposer

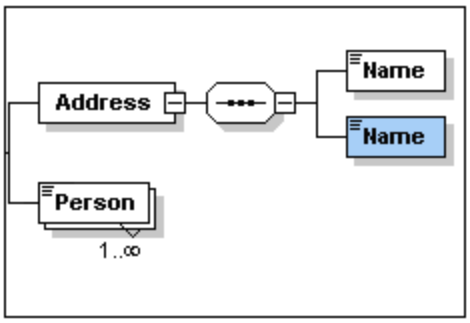
Vous avez ajouté des éléments en utilisant le menu contextuel qui apparaît lorsque vous cliquez avec la touche de droite sur un élément ou un compositeur. Vous pouvez aussi créer des éléments à l'aide d'un glisser-déposer ce qui est plus rapide que d'utiliser des commandes de menu. Dans cette section, vous ajouterez plus d'éléments à la définition de l'élément `Address` via glisser-déposer pour compléter cette définition.

Afin de compléter la définition de l'élément `Address` en utilisant un glisser-déposer :

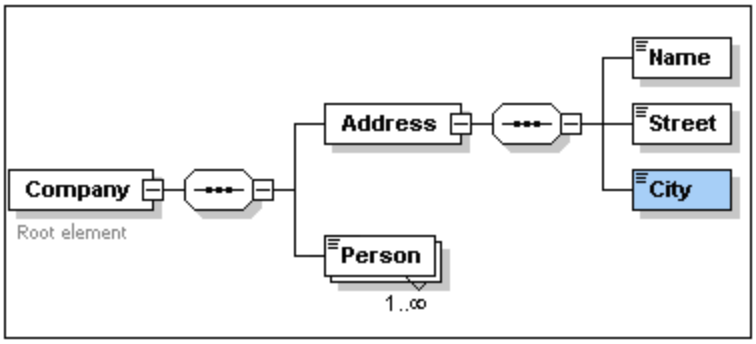
1. Cliquer sur l'élément `Name` de l'élément `Address`, maintenir enfoncée la touche **Ctrl** et glisser la fenêtre d'élément avec la souris. Un petit "plus" apparaît la fenêtre d'élément, indiquant le fait que vous êtes sur le point de copier l'élément. Une copie de l'élément apparaît, ainsi qu'une ligne de connexion, indiquant l'endroit où l'élément sera créé.



2. Relâcher le bouton de la souris pour créer le nouvel élément dans la séquence `Address`. Si le nouvel élément apparaît à un endroit incorrect, il suffit de le glisser à un endroit en-dessous de l'élément `Name`.



3. Double-cliquer dans la fenêtre d'élément et saisir le mot `Street` pour changer le nom de l'élément.
4. Utiliser la même méthode pour créer un troisième élément nommé `City`. Le modèle de contenu devrait maintenant ressembler à l'image ci-dessous :



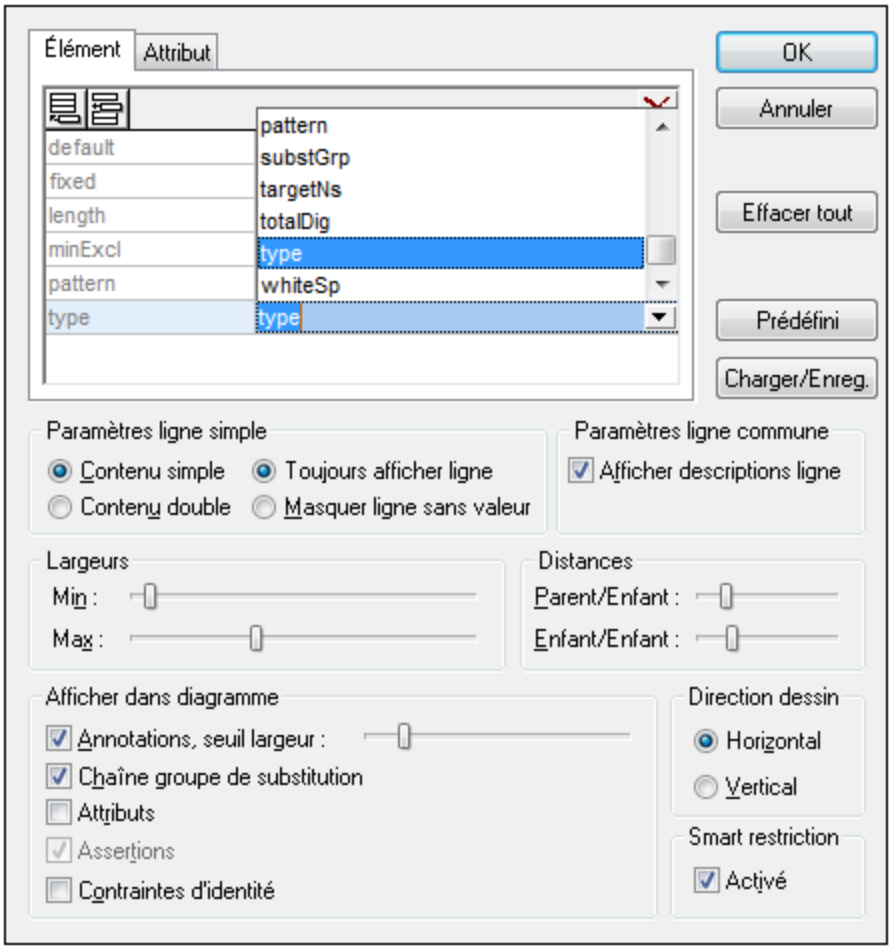
L'élément `Address` dispose maintenant d'une séquence d'un élément `Name`, `Street` et `City`, dans cet ordre.

2.2.5 Configurer le Mode Modèle de contenu

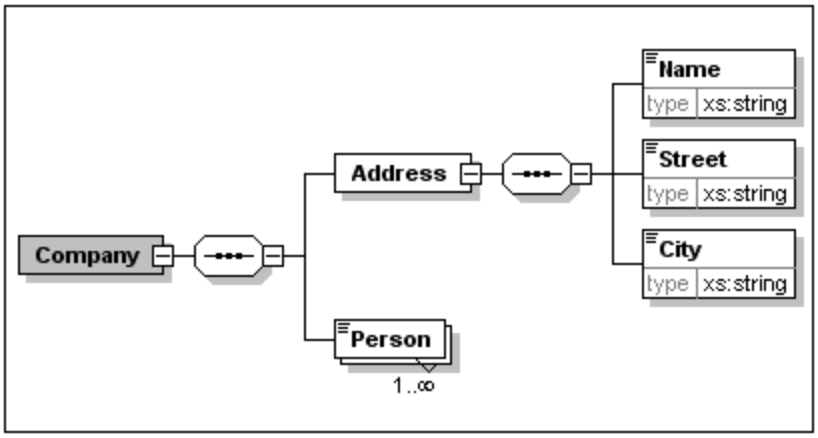
Le moment est maintenant venu de configurer le Mode de Modèle de contenu. Nous allons configurer le mode de manière à ce que le type de l'élément soit affiché. Pour ce faire, procéder comme suit :

1. Sélectionner le Mode de Modèle de contenu (cliquer l'icône du Mode de Modèle de contenu ).
2. Lorsque vous êtes dans le Mode de Modèle de contenu, la commande de menu **Conception de schéma | Configurer Affichage** est activée. Sélectionnez la commande pour afficher lle dialogue de configuration de l'Affichage Schéma (*voir la capture d'écran ci-dessous*).

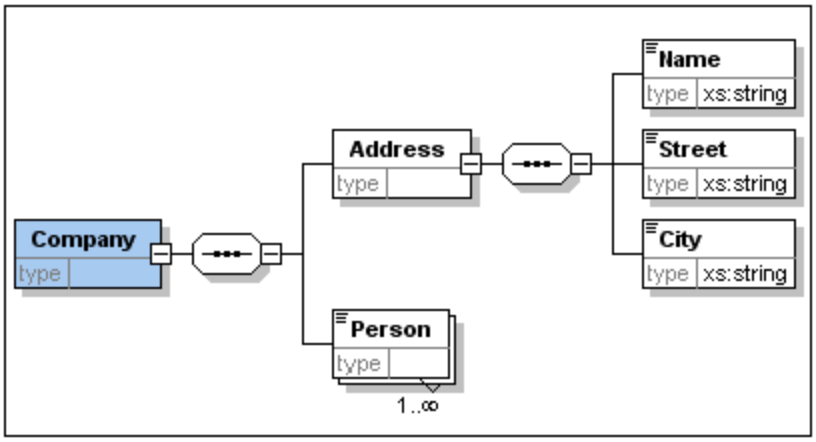
© 2016-2022 Altova GmbH



3. Dans l'onglet *Élément* (*voir la capture d'écran ci-dessus*), cliquez sur l'icône **Apposer** , et sélectionnez *Type (voir la capture d'écran)* pour ajouter la ligne de description de la propriété aux boîtes d'éléments.
4. Dans le panneau *Paramètres ligne simple*, sélectionnez *Masquer ligne sans valeur*. Cela permet de dissimuler la description du type de données dans la fenêtre d'élément si l'élément ne possède pas de type de données (par exemple, si l'élément est un type complexe). Dans la capture d'écran ci-dessous, notez que la ligne de description de type apparaît pour le Nom, la Rue et les éléments de Ville, qui sont des types simples du type xs:string, mais pas pour les éléments de type complexe. Ceci est dû au fait que la bascule Masquer Ligne Si Aucune Valeur est sélectionnée.



5. Dans le panneau *Paramètres ligne simple*, sélectionnez le bouton radio *Toujours Afficher Ligne*.
6. Cliquez sur **OK** pour confirmer les modifications.



Veuillez noter que la ligne de description pour le type de données est toujours affichée — même dans les fenêtres d'élément de types complexes, où ils apparaissent sans valeur.

Veuillez noter les points suivants :

- Les lignes de description de la propriété sont éditables, les valeurs que vous saisissez font donc partie de la définition de l'élément.
- Les paramètres que vous définissez dans le dialogue de configuration d'affichage du Schéma s'appliquent à la sortie de documentation du schéma ainsi qu'à la sortie d'impression

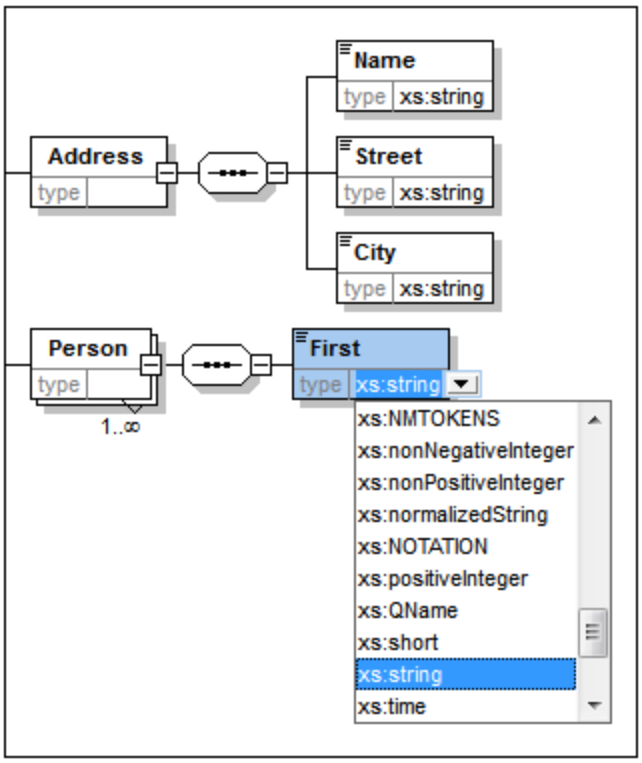
2.2.6 Terminer le Schéma de base

Vous avez défini le contenu de l'élément `Address`. À présent, vous devez définir le contenu de l'élément `Person` qui doit contenir les éléments enfants suivants, qui sont tous des types simples : `First`, `Last`, `Title`, `PhoneExt` et `Email`. Tous ces éléments doivent être obligatoires sauf pour `Title` et ils doivent se produire dans l'ordre précisé. Ils doivent tous être de type `xs:string` sauf `PhoneExt` qui doit être de type `xs:integer` et limité à deux chiffres.

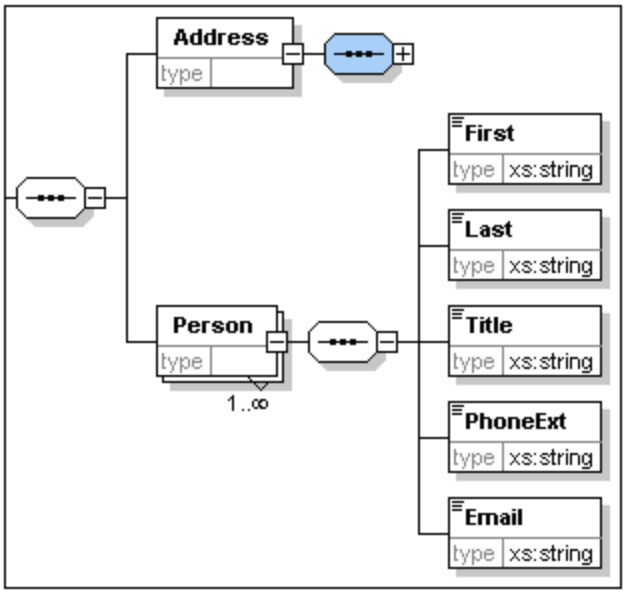
© 2016-2022 Altova GmbH

Afin de créer le modèle de contenu pour `Person`, procédez comme suit :

1. Cliquer avec la touche de droite sur l'élément `Person` pour ouvrir le menu contextuel, et sélectionner **Ajouter enfant | Séquence**. Cela permet d'insérer le compositeur de séquence.
2. Cliquer avec la touche de droite sur le compositeur de séquence et sélectionner **Ajouter enfant | Élément**.
3. Saisir `First` en tant que le nom de l'élément et appuyer sur la touche **Tab**. Cela permet de placer automatiquement le curseur dans le champ `type`.



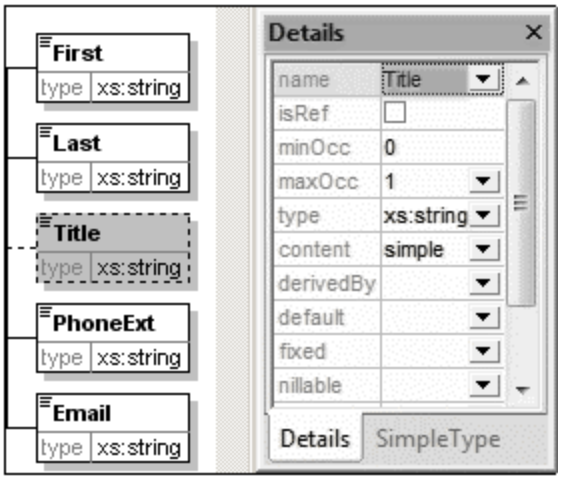
4. Sélectionner l'entrée `xs:string` depuis la liste déroulante ou la saisir dans le champ de valeur `type`.
5. Utiliser la méthode par glisser-déposer pour créer quatre éléments supplémentaires. Les nommer `Last`, `Title`, `PhoneExt` et `Email`, respectivement.



Veuillez noter : Vous pouvez sélectionner plusieurs éléments en maintenant la touche **Ctrl** appuyée et en cliquant sur chacun des éléments nécessaires. Cela permet, par ex. de copier plusieurs éléments à la fois.

Rendre un élément optionnel

Cliquer avec la touche de droite sur l'élément `Title` et sélectionner **Facultatif** depuis le menu contextuel. Le cadre de la fenêtre d'élément passe d'une ligne à des tirets ; cela représente l'indication visuelle qu'un élément est facultatif.



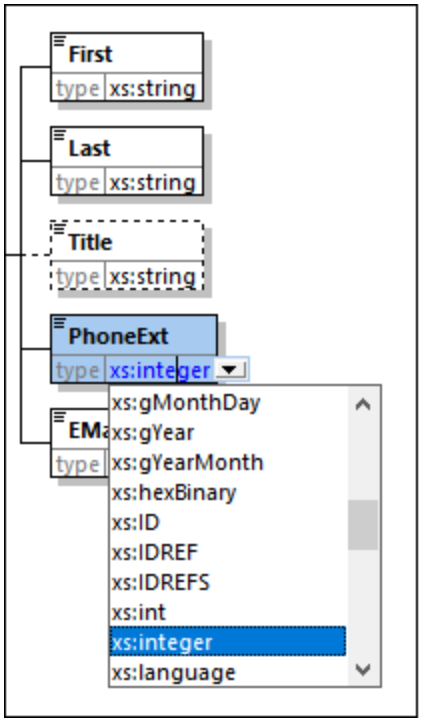
Dans l'assistant à la saisie des Détails, vous verrez `minOcc=0` et `maxOcc=1`, indiquant que l'élément est optionnel. En alternative, vous pouvez définir `minOcc=0` pour rendre l'élément optionnel au lieu d'utiliser le menu contextuel.

© 2016-2022 Altova GmbH

Limiter le contenu d'un élément

Pour définir que l'élément `PhoneExt` soit de type `xs:integer` et ne présente pas plus de deux chiffres, procédez comme suit :

1. Cliquer avec la touche de droite dans le champ `type` de l'élément `PhoneExt` et sélectionner (ou saisir) l'entrée `xs:integer` depuis la liste déroulante (*voir la capture d'écran ci-dessous*).



2. Les items contenus dans l'Assistant à la saisie Facettes changeront à ce moment. Dans l'Assistant à la saisie Facettes, double-cliquer le champ `maxIncl` et saisir `99`. Confirmer avec **Entrée**. Cela définit que les extensions téléphoniques peuvent avoir des valeurs de `0` à `99`.
3. Sélectionner le **Fichier | Enregistrer** pour enregistrer les changements apportés au schéma.

Veuillez noter :

- Si vous sélectionnez un type de données de Schéma XML qui est un type simple (par exemple, `xs:string` ou `xs:date`), le modèle de contenu passera automatiquement à simple dans l'Assistant à la saisie Détails (`content = simple`).
- Le fait d'ajouter un compositeur à un élément (`sequence`, `choice`, or `all`), change automatiquement le modèle de contenu en complexe dans l'Assistant à la saisie Détails (`content = complex`).
- Le schéma décrit ci-dessus est disponible sous `AddressFirst.xsd` dans le dossier `Tutorial` de votre dossier d'application XMLSpy.

2.3 Schémas XML : Avancé

Maintenant que vous avez créé un schéma de base, nous pouvons passer à certains des aspects avancés du développement de schémas.

Objectif

Dans cette section, vous allez apprendre à :

- Travailler avec des types complexes et des types simples qui peuvent ensuite être utilisés en tant que des types d'éléments de schéma.
- Créer des éléments globaux les référencer depuis d'autres éléments.
- Créer des attributs et leurs propriétés, y compris les valeurs énumérées.

Vous allez commencer cette section avec le schéma de base `AddressFirst.xsd` que vous avez créé dans la première partie de ce tutoriel.

Commandes utilisées dans cette section

Dans cette section du tutoriel, vous allez utiliser uniquement le Mode Schéma. Les commandes suivantes sont utilisées :

	Afficher le diagramme (ou le Mode Afficher le modèle de contenu). Cette icône est située à gauche de toutes les composantes globales dans l'Aperçu de schéma. Cliquer sur l'icône pour afficher le modèle de contenu de la composante globale associée.
	Afficher tous les globaux. Cette icône est située en haut à gauche du Mode Modèle de contenu. Cliquer sur l'icône pour passer du mode à l'Aperçu de schéma qui affiche tous les composants globaux.
	Ajouter. L'icône Ajouter est située en haut à gauche de l'Aperçu de schéma. Cliquer sur l'icône pour ajouter un composant global.

2.3.1 Travailler avec des types complexes et des types simples

Une fois que vous avez défini le modèle de contenu d'un élément, vous pouvez décider de vouloir le réutiliser ailleurs dans votre schéma. Ceci peut être le cas, par exemple, si vous voulez définir un modèle de contenu pour les adresses aux États-Unis d'Amérique et au Royaume-Uni. Quelques composants des deux formats d'adresse sont communs, par exemple, les composants rue et ville. D'autres composants, par contre, sont différents. Une stratégie judicieuse serait donc de réutiliser les composants communs dans chaque modèle de contenu (US et UK), et compléter chaque modèle de contenu avec les composants spécifiques y afférents (tels que le code postal ZIP aux États-Unis d'Amérique et le code postal au Royaume-Uni). Pour ce faire, nous pouvons créer des composants communs tel qu'un type complexe global (ou, en alternative, chaque composant commun en tant qu'élément global), et réutiliser le type complexe global (ou les éléments globaux) dans le modèle de contenu de chaque type d'adresse.

Dans cette section, vous allez travailler avec des types complexes globaux. Une définition complexe est une définition de type pour des éléments qui contiennent d'autres éléments et/ou des attributs. Vous allez d'abord créer un type complexe au niveau global puis l'importer dans un modèle de contenu et l'étendre. Vous apprendrez à gérer les éléments globaux plus loin dans ce tutoriel.

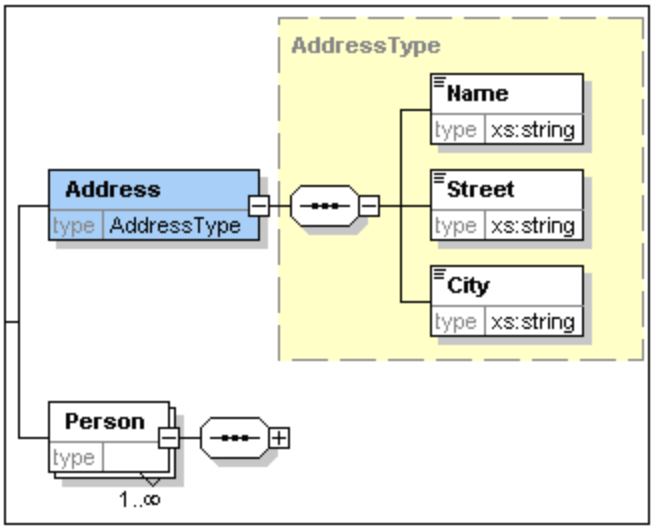
© 2016-2022 Altova GmbH

Créer un type complexe global

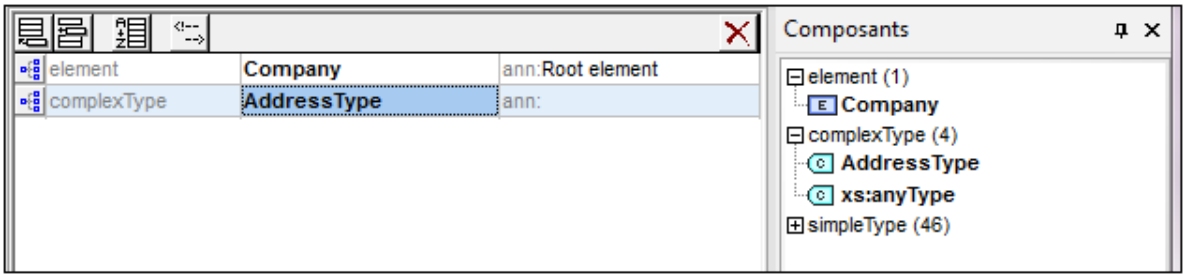
L'élément de base **Address** que nous avons défini (contenant les éléments **Name**, **Street**, et **City**) peut être réutilisé dans des formats d'adresse variés. C'est pourquoi nous allons créer cette définition d'élément en tant que type complexe, qui pourra être réutilisé.

Pour créer un type complexe global :

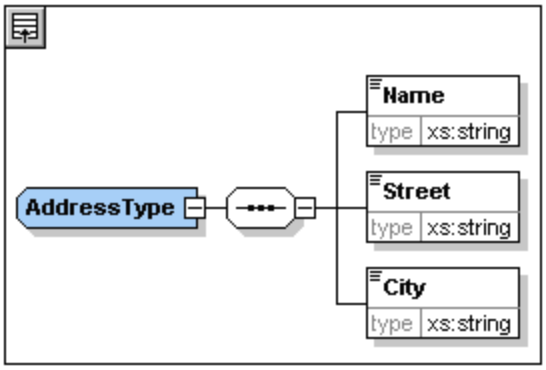
1. Dans le Mode de Modèle de contenu, cliquez avec la touche de droite sur l'élément **Address**.
2. Dans le menu contextuel qui apparaît maintenant, choisir **Rendre Global | Type complexe**. Un type complexe global appelé **AddressType** est créé et l'élément **Address** dans le modèle contenu **Company** est attribué à ce type. Le contenu de l'élément **Address** est le modèle de contenu **AddressType**, qui est affiché dans un encadre jaune. Notez que le type de données de l'élément **Address** est désormais **AddressType**.



3. Cliquez sur l'icône Afficher tous les globaux . Cela vous permet de consulter l'Aperçu de schéma dans lequel vous pouvez consulter tous les composants globaux du schéma.
4. Cliquez sur les icônes d'agrandissement pour les entrées *Expand*, **element** et **complexType** dans l'Assistant de saisie Composants, pour consulter les structures de schéma respectifs. L'Aperçu de schéma affiche maintenant deux composants globaux : l'élément **Company** et le type complexe **AddressType**. L'Assistant de saisie Composants affiche aussi le type complexe **AddressType**.



5. Cliquer sur l'icône du Mode Modèle de contenu  de **AddressType** pour voir son modèle de contenu (*voir la capture d'écran ci-dessous*). Veuillez noter la forme du conteneur du type complexe.



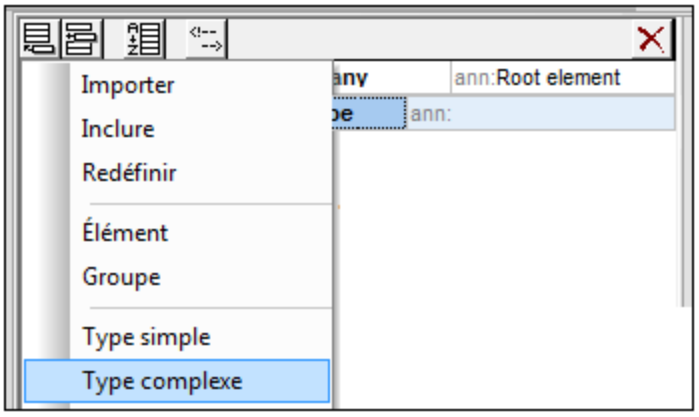
6. Cliquer sur l'icône Afficher tous les globaux pour retourner à l'Aperçu de schéma.

Étendre une définition de type complexe

Nous allons maintenant utiliser le composant global **`AddressType`** pour créer deux types d'adresses spécifiques au pays. À cette fin, nous allons définir un nouveau type complexe basé sur le composant de base **`AddressType`**, puis nous allons étendre la définition.

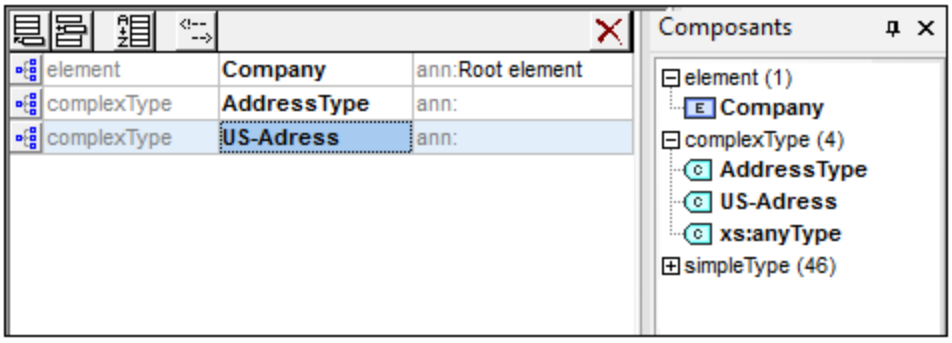
Pour ce faire, procédez comme suit :

1. Passer à l'Aperçu de schéma. (Si vous vous trouvez dans le mode Modèle de contenu, cliquez sur l'icône Afficher tous les globaux .)
2. Cliquez sur l'icône **Apposer** située en haut à gauche de la fenêtre de composant. Le menu suivant s'ouvre :

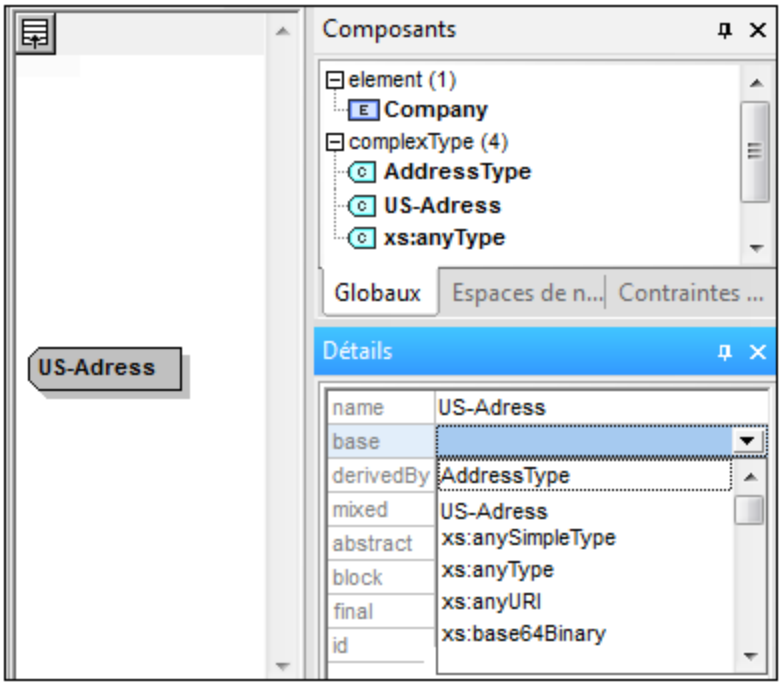


3. Sélectionnez **ComplexType** depuis le menu. Une nouvelle ligne apparaît dans la liste de composants et le curseur est réglé de manière à ce que vous puissiez saisir le nom de composant.
4. Saisissez **`US-Address`** et confirmez avec **Enter**. (Si vous oubliez de saisir le signe **`"-"`** et que vous saisissez un espace, le nom d'élément apparaîtra en rouge, signalisant un caractère invalide.)

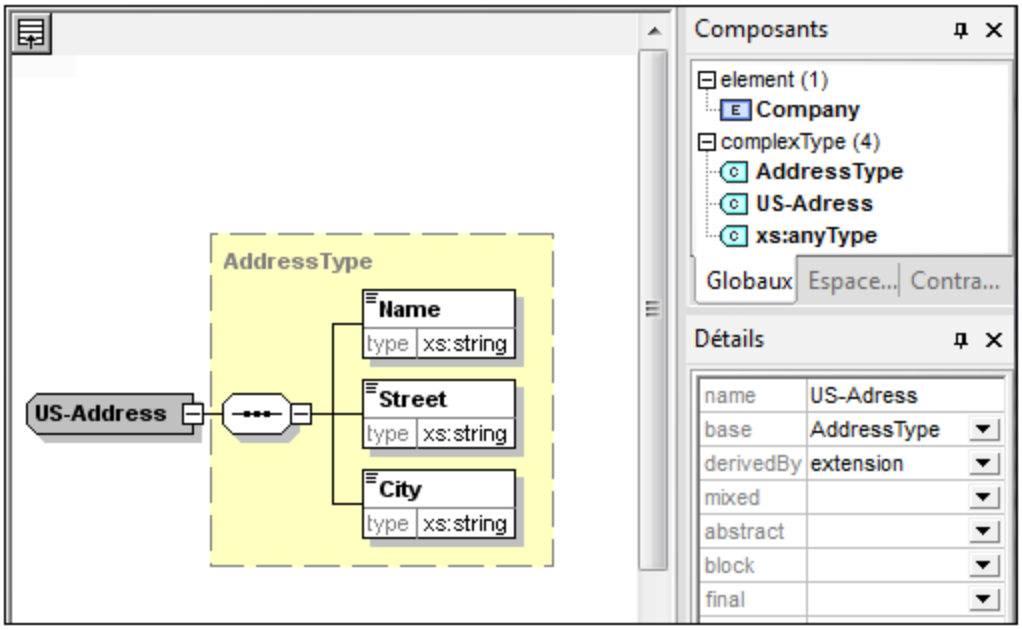
© 2016-2022 Altova GmbH



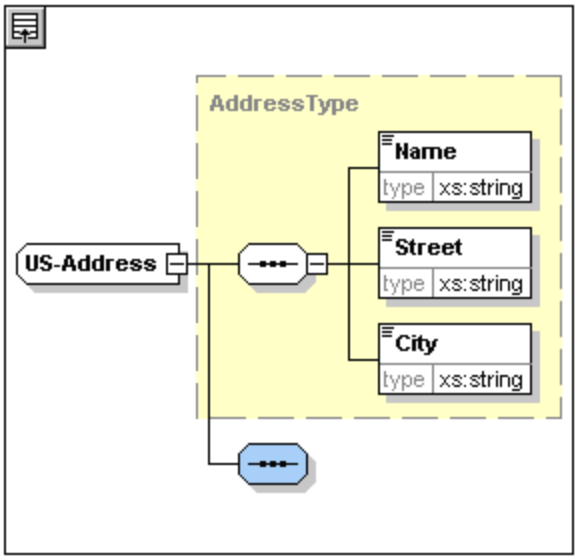
5. Cliquer l''icône de Mode de Modèle de contenu de **US-Address** pour voir le modèle de contenu du nouveau type complexe. Le modèle de contenu est vide (*voir la capture d'écran ci-dessous*).
6. Dans l'assistant de saisie Détails, cliquez sur la liste de choix **base** et sélectionnez l'entrée **AddressType**.



Le Mode de Modèle de contenu affiche maintenant le modèle de contenu **AddressType** en tant que le modèle de contenu des adresses US (*voir la capture d'écran ci-dessous*).

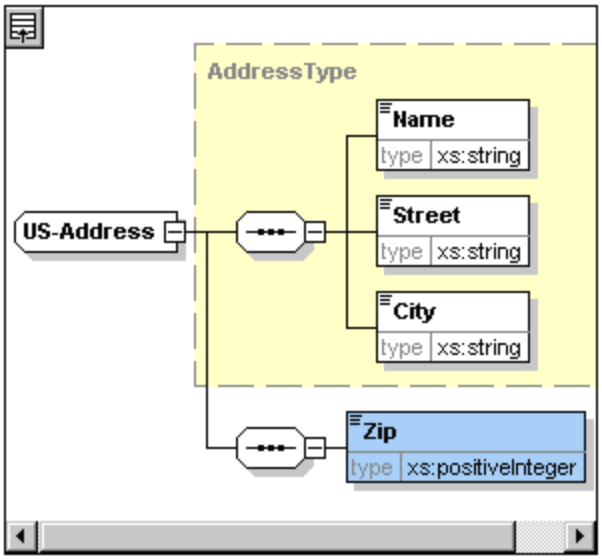


7. Maintenant, nous pouvons étendre le modèle du type complexe de **`US-Address`** pour prendre un élément de code postal (ZIP Code). Pour ce faire, cliquez avec la touche de droite sur le composant **`US-Address`** et, à partir du menu contextuel qui apparaît, choisissez **Ajouter enfant | Séquence**. Un nouveau compositeur de séquence est affiché à l'extérieur de la fenêtre **`AddressType`** (*voir la capture d'écran ci-dessous*). Il s'agit d'une indication visuelle pour indiquer qu'il s'agit de l'extension de l'élément.



8. Cliquez avec la touche de droite sur le nouveau compositeur de séquence et sélectionnez **Ajouter enfant | Élément**.
9. Nommez le nouvel élément **`Zip`**, puis appuyez sur la touche **Tab**. Cela place le curseur dans le champ de valeur de la ligne de descripteur de type.
10. Choisissez **`xs:positiveInteger`** depuis le menu déroulant qui apparaît et confirmer avec **Enter**.

 © 2016-2022 Altova GmbH



Vous disposez maintenant d'un type complexe appelé **US-Address**, qui est basé sur le type complexe **AddressType** et s'étend pour pouvoir contenir un code postal.

Types simples globaux

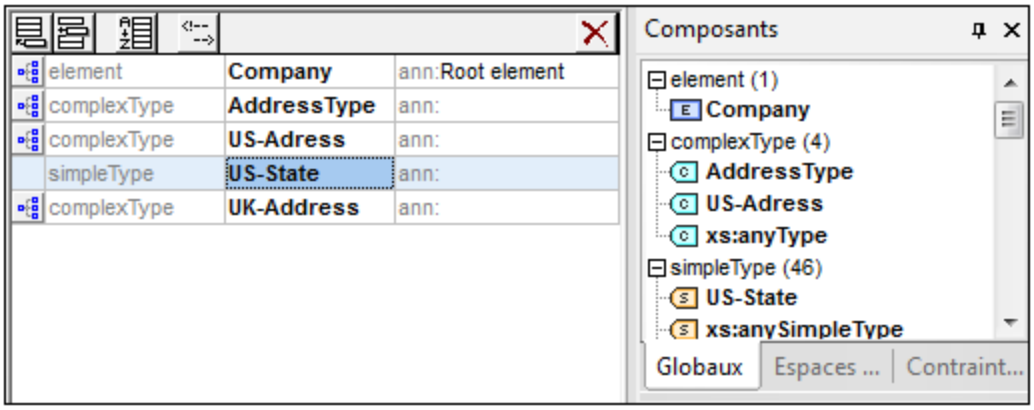
De même que le type complexe **US-Address** est basé sur le type complexe **AddressType**, un élément peut aussi être basé dur un type simple. L'avantage est le même que pour les types complexes globaux : le type simple peut être réutilisé. Pour pouvoir réutiliser un type simple, celui-ci doit être défini globalement. Dans ce tutoriel, vous allez définir un modèle de contenu pour les États US en tant que type simple. Celui-ci sera utilisé en tant que base pour un autre élément

Créer un type simple global

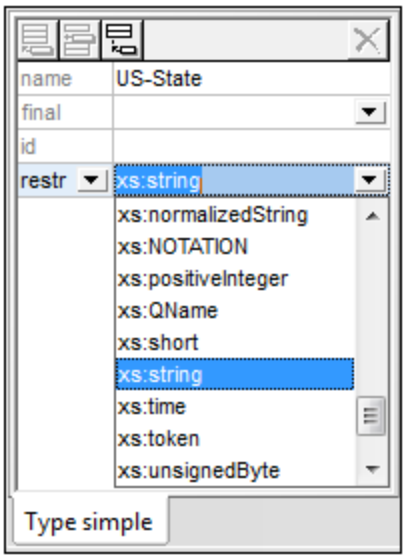
Créer un type simple global consiste à ajouter un nouveau type simple à la liste des composants globaux, à le nommer et à définir son type de données.

Pour créer un type simple global :

1. Passer à l'Aperçu de schéma. (Si vous vous trouvez dans le mode Modèle de contenu, cliquez sur l'icône Afficher tous les globaux .)
2. Cliquez sur l'icône **Apposer**, et dans le menu contextuel qui apparaît, sélectionnez **SimpleType**.
3. Saisissez **US-State** en tant que le nom du simpleType récemment créé.
4. Appuyez sur **Enter** pour confirmer. Le type simple **US-State** est créé et apparaît dans la liste des types simples dans l'Assistant de saisie des composants (cliquez sur l'icône d'agrandissement de l'entrée simpleType pour le voir).



5. Dans l'Assistant de saisie Détails (*voir la capture d'écran ci-dessous*), placez le curseur dans le champ de valeur de **restr** et entrez **xs:string**, ou sélectionnez **xs:string** depuis la liste déroulante dans le champ de valeur **restr**.



Cela crée un type simple appelé **US-State**, qui est de type de données **xs:string**. Ce composant global peut maintenant être utilisé dans le modèle de contenu de **US-Address**.

Utiliser un type simple global dans un modèle de contenu

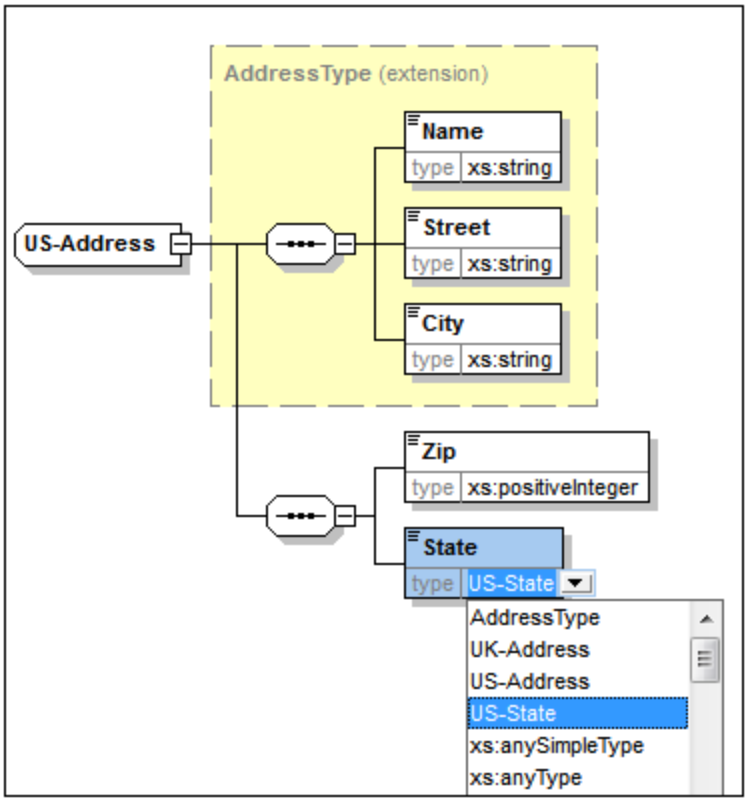
Un type simple global peut être utilisé dans un modèle de contenu pour définir le type d'un composant. Nous allons utiliser **US-State** pour définir un élément appelé **State** dans le modèle de contenu de **US-Address**.

Pour ce faire, procédez comme suit :

1. Dans l'Aperçu de schéma, cliquer sur l'icône Mode Modèle de contenu de **US-Address**.
2. Cliquez avec la touche de droite sur le compositeur de séquence inférieur et sélectionnez **Ajouter enfant | Élément**.
3. Saisissez **State** pour le nom d'élément.
4. Appuyer sur la touche **Tab** pour placer le curseur dans le champ de valeur de la ligne de descripteur de type.

 © 2016-2022 Altova GmbH

5. À partir du menu déroulant de cette liste de choix, choisissez **US-State**.



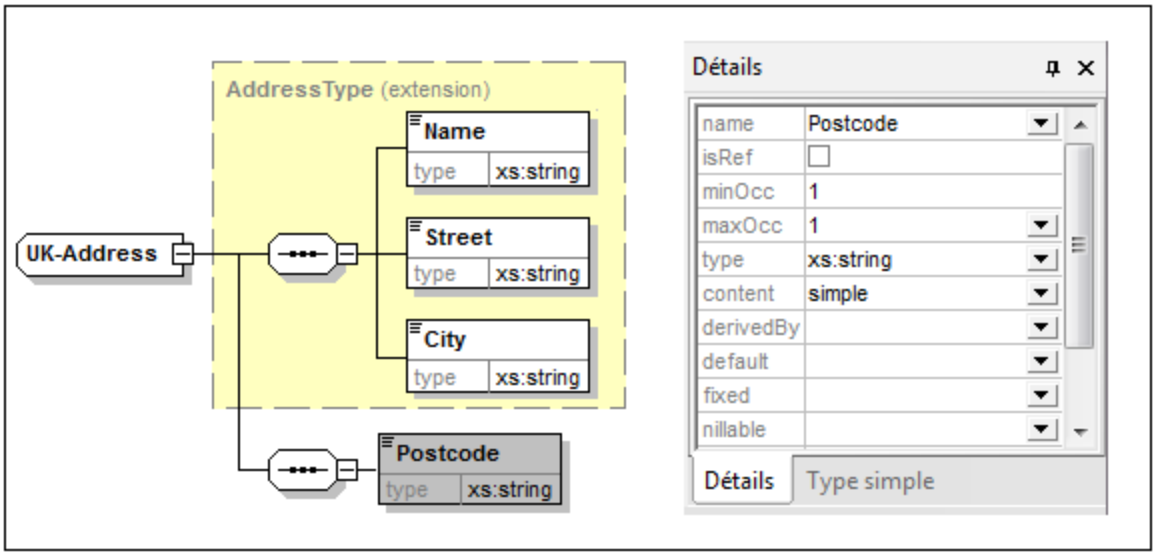
L'élément `State` est maintenant basé sur le type simple **US-State**.

Créer un second type complexe basé sur AddressType

Nous allons maintenant créer un type complexe global pour contenir des adresses UK (du Royaume-Uni). Le type complexe est basé sur **AddressType** et est étendu pour correspondre au format d'adresses UK.

Pour ce faire, procédez comme suit :

1. Dans l'Aperçu de schéma, créer un type complexe global appelé **UK-Address** et basez-le sur **AddressType** (**base=AddressType**).
2. Dans le Mode Modèle de contenu de **UK-Address**, ajoutez un élément **Code postal** donnez-lui un type de **xs:string**. Votre modèle de contenu **UK-Address** devrait ressembler à ceci :



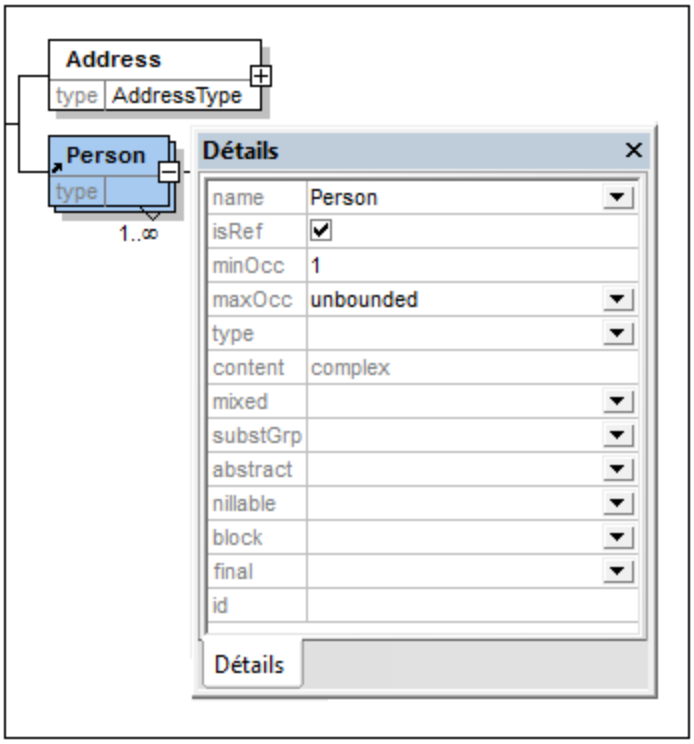
Note : dans cette section, vous créez des types globaux simples et complexes que vous utilisez ensuite dans des définitions de modèle de contenu. L'avantage des types globaux est qu'ils peuvent être réutilisés dans plusieurs définitions.

2.3.2 Référencer des éléments globaux

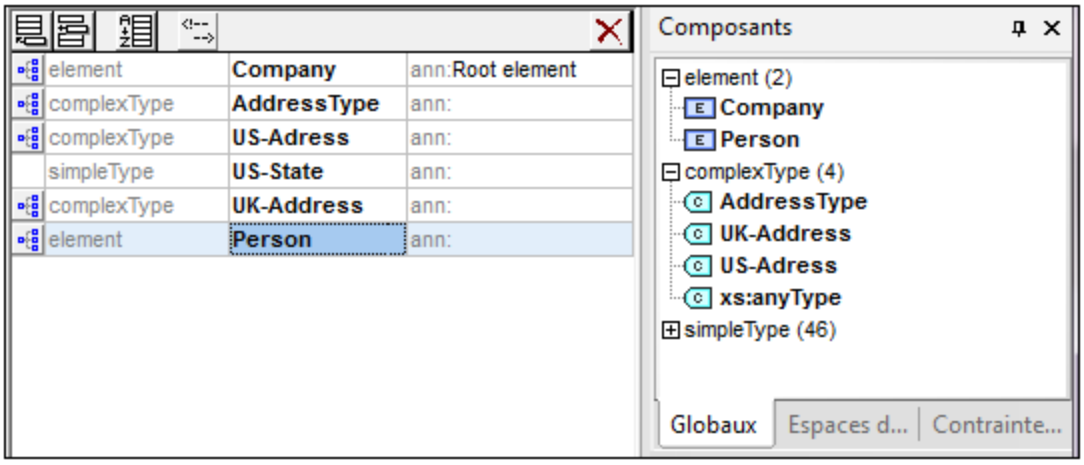
Dans cette section, nous allons convertir l'élément `Person` défini localement en un élément global puis référencer cet élément global depuis l'élément `Company`.

1. Cliquer sur (Afficher Tous les globaux) pour passer à l'Aperçu de schéma.
2. Cliquer sur l'icône Afficher le diagramme de l'élément `Company`.
3. Cliquer avec la touche de droite sur l'élément `Person`, et sélectionner **Rendre global | Élément**. Une petite flèche apparaît dans l'élément `Person`, montrant que cet élément référence à présent l'élément `Person` déclaré globalement. Dans l'Assistant à la saisie Détails, la liste de choix `isRef` est maintenant activée.

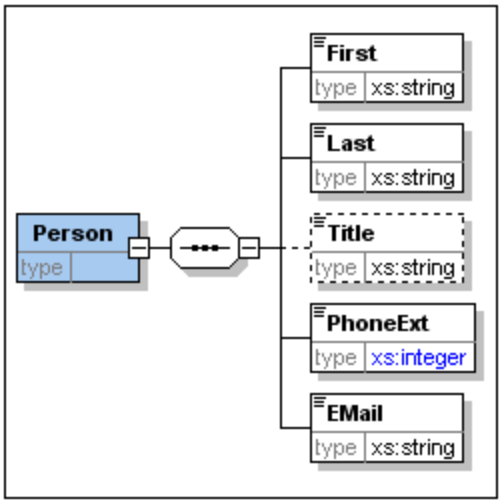
© 2016-2022 Altova GmbH



4. Cliquer sur l'icône Afficher Tous les Globaux pour retourner à l'Aperçu de schéma. L'élément `Person` est maintenant affiché en tant qu'un élément global. Il se trouve également dans l'Assistant à la saisie des Composants.



5. Dans l'Assistant à la saisie des Composants, double-cliquer l'élément `Person` pour voir le modèle de contenu de l'élément global `Person`.



Veuillez noter que la fenêtre de l'élément global ne possède **pas** un lien sous forme de symbole de flèche. Cela est dû au fait qu'il s'agit de l'élément référencé et non de l'élément de référencement. C'est en effet l'élément de référencement qui est représenté avec un lien en forme de flèche.

Veuillez noter :

- Un élément qui référence un élément global doit avoir le même nom que l'élément global qu'il référence.
- Une déclaration globale ne décrit pas l'endroit auquel un composant doit être utilisé dans un document XML. Elle décrit uniquement un modèle de contenu. Ce n'est que lorsqu'une déclaration globale est référencée depuis un autre composant que son emplacement dans le document XML est spécifié.

  Un élément déclaré globalement peut être réutilisé sur plusieurs emplacements. Il diffère d'un type complexe déclaré globalement du fait que son modèle de contenu ne peut pas être modifié sans également modifier l'élément global lui-même. Si vous changez le modèle de contenu d'un élément qui référence un élément global, le modèle de contenu de l'élément global changera aussi, ainsi que le modèle de contenu de tous les autres éléments qui référencent cet élément global.

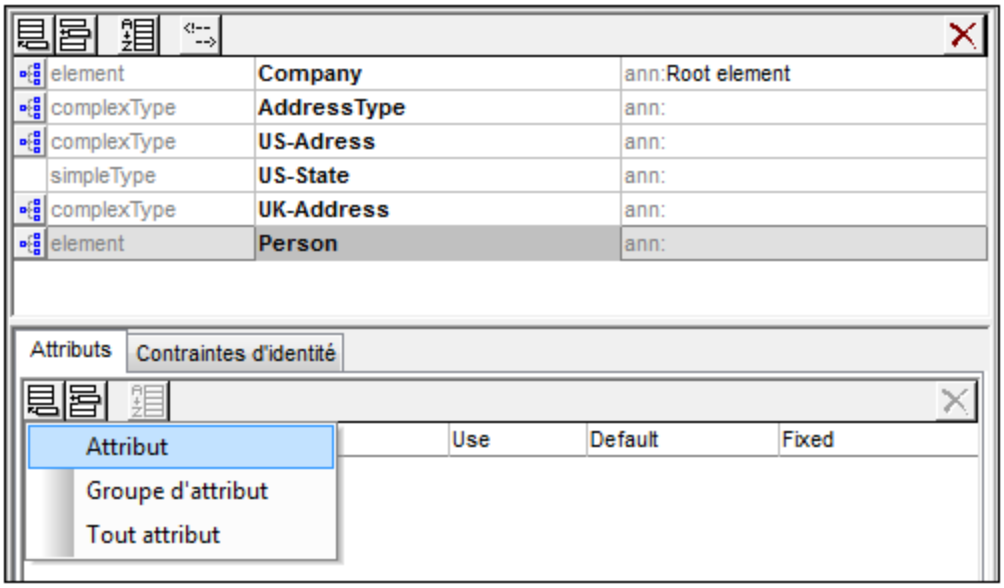
2.3.3 Attributs et énumérations d'attributs

Dans cette section, vous allez apprendre à créer des attributs et des énumérations d'attributs.

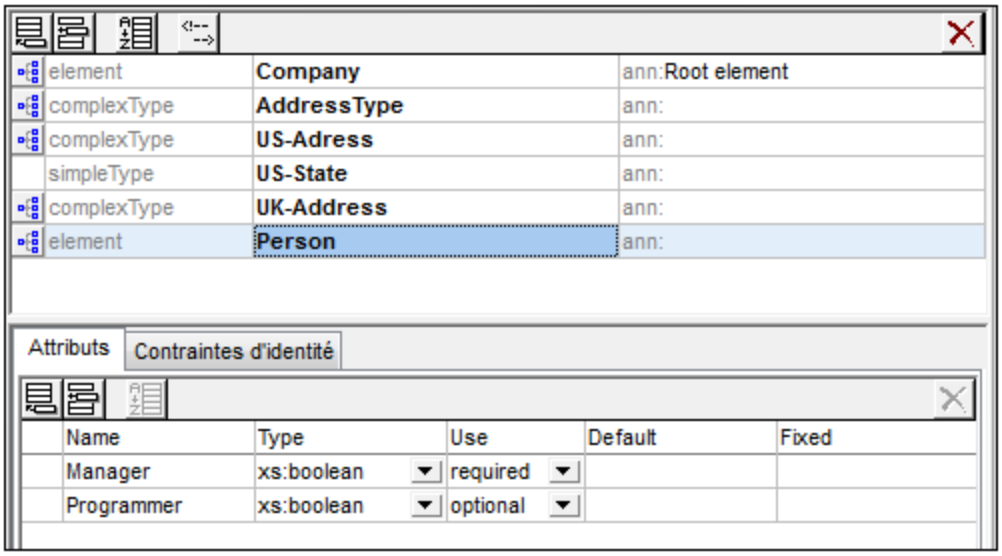
Définir les attributs d'éléments

1. Dans l'Aperçu de schéma, cliquer sur l'élément `Person` pour le rendre actif.
2. Cliquer sur l'icône Ajouter , située en haut à gauche du groupe d'onglets Attributs/Contrainte d'identité (dans la partie inférieure de la fenêtre Aperçu de schéma), et sélectionner l'entrée Attribut.

© 2016-2022 Altova GmbH



3. Saisir `Manager` en tant que le nom d'attribut dans le champ Nom.
4. Utiliser la liste de choix `Type` combo pour sélectionner `xs:boolean`.
5. Utiliser la liste de choix `Use` pour sélectionner `required`.



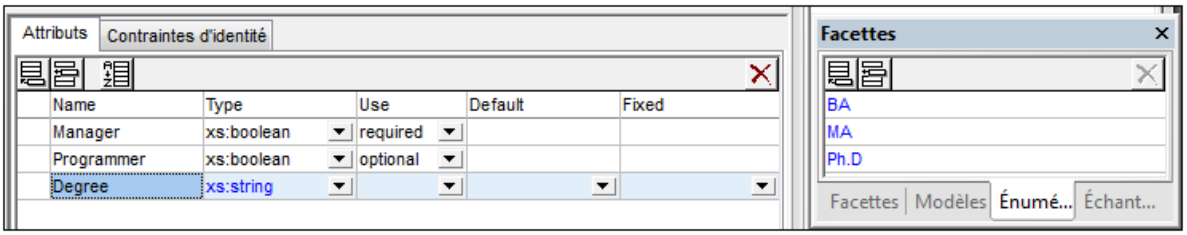
6. Utiliser la même procédure pour créer un attribut `Programmer` avec `Type=xs:boolean` et `Use=optional`.

Définir les énumérations pour les attributs

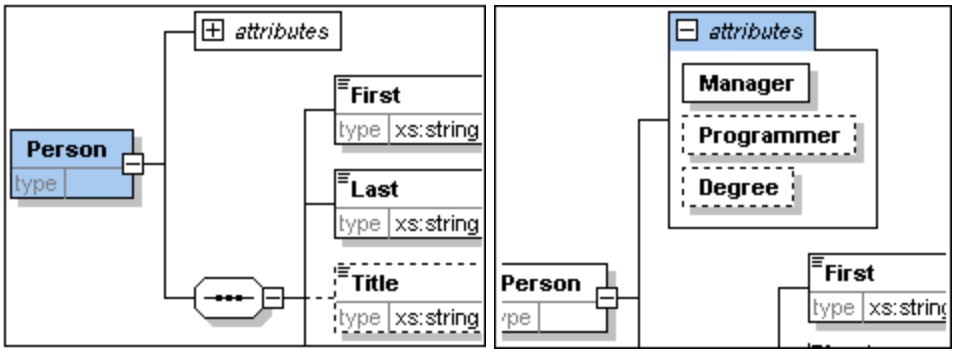
Les énumérations sont des valeurs autorisées pour un certain attribut. Si la valeur de l'attribut dans le document d'instance XML n'est pas une des énumérations spécifiées dans le Schéma XML, alors le document est invalide. Nous allons créer des énumérations pour l'attribut `Degree` de l'élément `Person`.

Procéder comme suit :

1. Dans l'Aperçu de schéma, cliquer sur l'élément `Person` pour le rendre actif.
2. Cliquer sur l'icône Ajouter  située en haut à gauche dans la fenêtre Attributs et sélectionner l'entrée **Attribut**.
3. Saisir `Degree` en tant que le nom d'attribut, et sélectionner `xs:string` en tant que son type.
4. Une fois l'attribut `Degree` sélectionné, dans l'Assistant à la saisie Facettes, cliquer sur l'onglet **Énumérations** (*voir capture d'écran*).



5. Dans l'onglet **Énumérations**, cliquer sur l'icône Ajouter .
6. Saisir `BA`, et confirmer avec **Entrée**.
7. Utiliser la même procédure pour ajouter deux autres énumérations : `MA` et `PhD`.
8. Cliquer sur l'icône Mode Modèle de contenu  de `Person`.



Les attributs définis précédemment sont visibles dans le Mode Modèle de contenu. Cliquer sur l'icône d'agrandissement pour afficher tous les attributs définis pour cet élément. Ce mode d'affichage et l'onglet Attributs peuvent être basculés en sélectionnant l'option de menu **Conception de schéma | Configurer affichage** et en cochant et décochant les cases **Attributs** dans le panneau **Afficher dans le diagramme**.

9. Cliquer sur l'icône Afficher Tous globaux  pour retourner à l'Aperçu de schéma.

Enregistrer le schéma XML achevé

Avant d'enregistrer votre fichier de schéma, renommez le fichier `AddressLast.xsd` qui est fourni avec XMLSpy avec un nom différent (comme par ex. `AddressLast_original.xsd`), pour éviter de l'écraser.

Enregistrer le schéma achevé avec le nom que vous souhaitez (**Fichier | Enregistrer sous**). Nous recommandons de l'enregistrer sous le nom `AddressLast.xsd` puisque le fichier XML que vous avez créé dans la partie suivante de ce tutoriel sera basé sur le schéma `AddressLast.xsd`.

© 2016-2022 Altova GmbH

2.4 Schémas XML : fonctions XMLSpy

Après avoir achevé le Schéma XML, nous vous suggérons de vous familiariser avec quelques-uns des raccourcis de navigation et de consulter la documentation de schéma que vous pouvez générer depuis XMLSpy. Vous les trouverez dans les sous-sections de cette section.

Commandes utilisées dans cette section

Dans cette section du tutoriel, vous utiliserez exclusivement le Mode Schéma. Les commandes suivantes sont utilisées :

	Afficher le diagramme (ou afficher le Mode Modèle de contenu). Cette icône est située à gauche de tous les composants globaux dans l'Aperçu de schéma. Cliquer sur l'icône pour afficher le modèle de contenu du composant global associé.

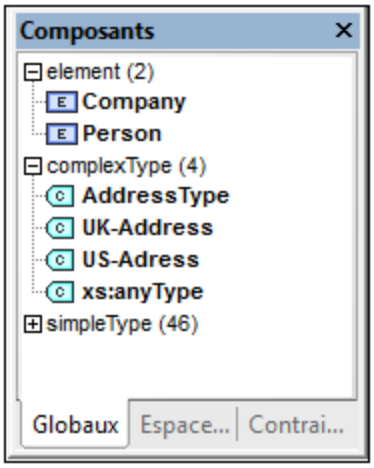
2.4.1 Navigation de schéma

Cette section vous montre comment parcourir efficacement le Mode de Schéma. Nous vous conseillons de tester ces mécanismes de navigation afin de vous familiariser avec eux.

Afficher le modèle de contenu d'un composant global

Les composants globaux qui peuvent avoir des modèles de contenu sont des types complexes, des éléments et des groupes d'élément. Le Mode de Modèle de contenu de ces composants peut s'ouvrir des manières suivantes :

- Dans l'Aperçu de schéma, cliquer sur l'icône Afficher le diagramme  située à gauche du nom de composant.
- Soit dans Aperçu de schéma ou Mode de Modèle de contenu, double-cliquer sur l'élément, le type complexe ou le groupe d'élément dans l'Assistant à la saisie des Composants (*voir capture d'écran ci-dessous*). Cela permet d'afficher le modèle de contenu de ce composant.



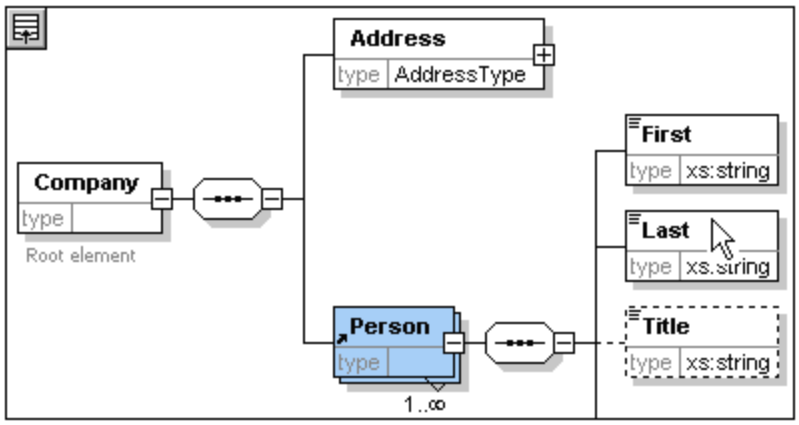
Si vous double-cliquez sur un des autres composants globaux (type simple, attribut, groupe d'attributs) dans l'Assistant à la saisie des Composants, ce composant sera marqué dans l'Aperçu de schéma (puisqu'un tel composant n'aura pas de modèle de contenu).

Dans l'Assistant à la saisie des Composants, le mécanisme de double-clic fonctionne dans les onglets Globaux et Espaces de noms.

Se rendre à la définition d'un élément global depuis un élément de référencement

Si un modèle de contenu contient un élément qui référence un élément global, vous pouvez vous rendre directement au modèle de contenu de cet élément global ou à tout autre de ses composants contenus en maintenant appuyée la touche **Ctrl** et en double-cliquant sur l'élément désiré.

Par exemple, tout en consultant le modèle de contenu `Company`, en maintenant la touche **Ctrl** appuyée tout en double-cliquant `Last` vous permettra d'ouvrir le modèle de contenu `Person` et de marquer l'élément `Last` à l'intérieur.

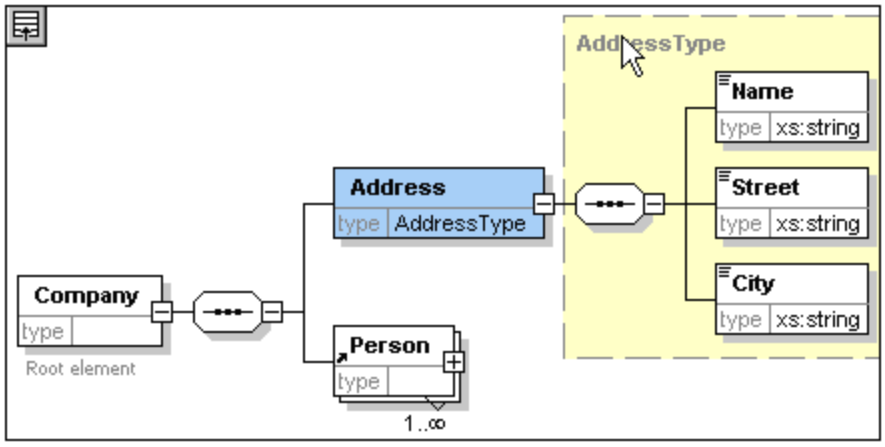


Lorsque l'élément `Last` est marqué, toutes ses propriétés sont immédiatement affichées dans les assistants à la saisie pertinents et dans les fenêtres d'information.

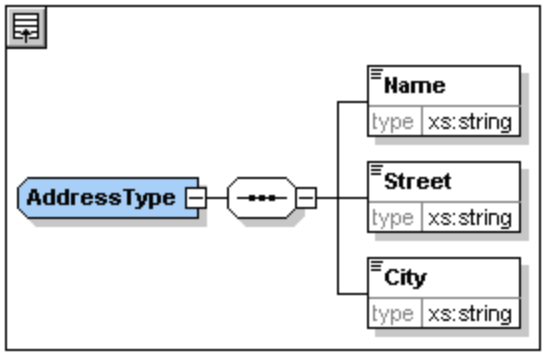
© 2016-2022 Altova GmbH

Se rendre à la définition d'un type complexe

Les types complexes sont souvent utilisés en tant que type de certains éléments à l'intérieur d'un modèle de contenu. Pour vous rendre directement à la définition d'un type complexe depuis l'intérieur d'un modèle de contenu, double-cliquer sur le **nom** du type complexe dans la fenêtre jaune (*voir pointeur de souris dans la capture d'écran ci-dessous*).



Cela vous amène au Mode de Modèle de contenu du type complexe.



Note : Comme avec les éléments globaux référencés, vous pouvez vous rendre directement à un élément situé dans la définition de type complexe en maintenant la touche **Ctrl** appuyée et en double-cliquant l'élément requis dans le modèle de contenu qui contient le type complexe.

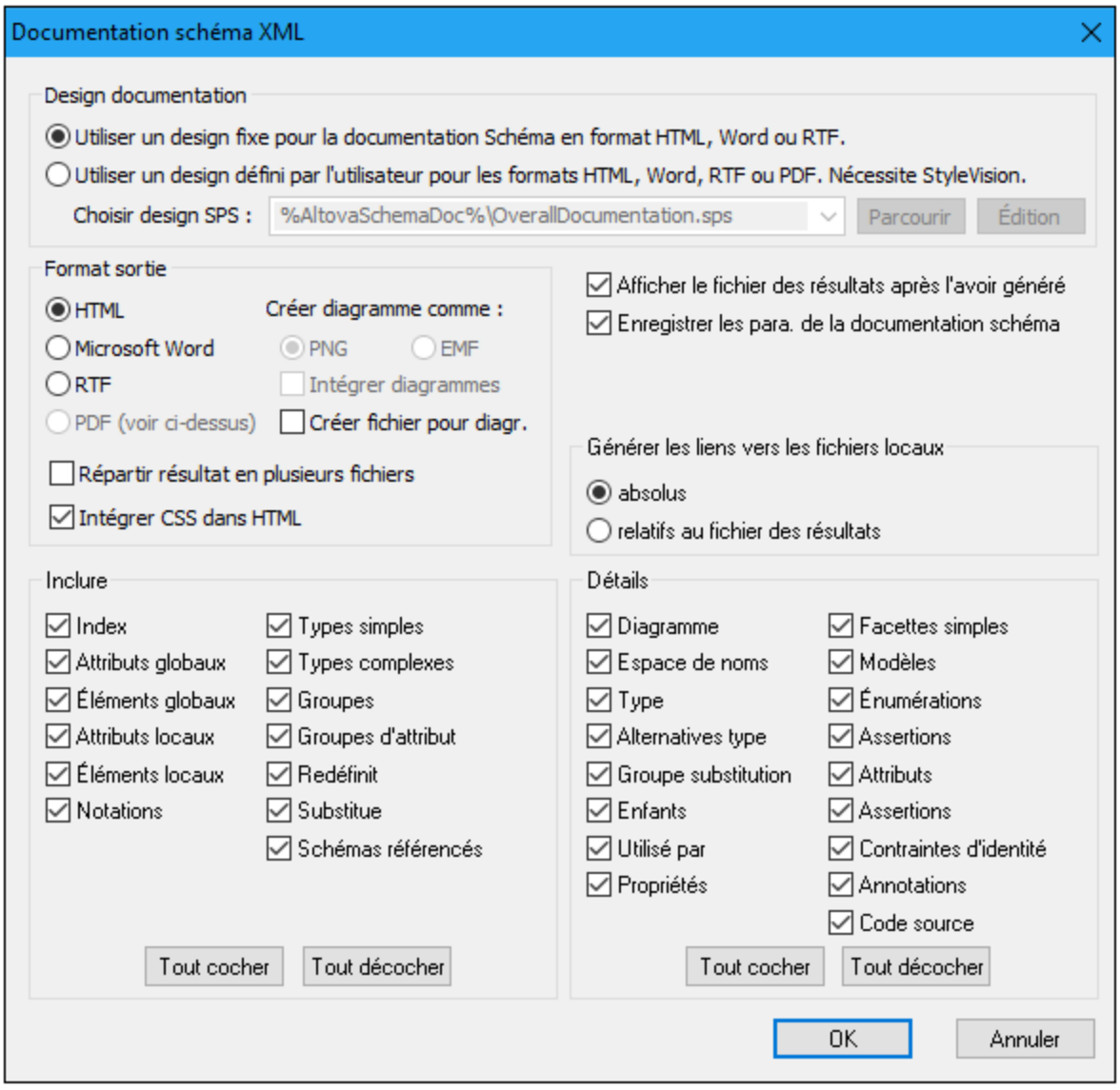
2.4.2 Documentation de schéma

XMLSpy fournit une documentation détaillées des Schémas XML dans des formats HTML et Microsoft Word (MS Word). Vous pouvez sélectionner les composants et le niveau de détails que vous souhaitez documenter. Les composants associés sont liés par hyperliens dans les documents HTML et MS Word. Pour générer une documentation MS Word, vous devez installer MS Word sur votre ordinateur (ou réseau).

Dans cette section, nous allons générer une documentation pour le Schéma XML `AddressLast.xsd`.

Procéder comme suit :

1. Choisir l'option de menu **Conception de schéma | Générer documentation**. Cela ouvre le dialogue Documentation de schéma.



2. Pour l'option Format sortie, sélectionner HTML et cliquer **OK**.
3. Dans le dialogue Enregistrer sous, sélectionner l'emplacement où vous souhaitez sauvegarder le fichier et donnez au fichier un nom approprié (par ex. `AddressLast.html`). Ensuite cliquer sur le bouton **Enregistrer**.

Le document HTML apparaît dans le Mode Navigateur de XMLSpy. Cliquer sur un lien pour vous rendre sur le composant lié correspondant.

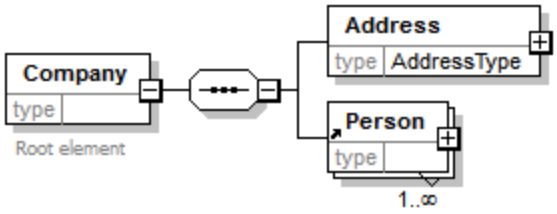
© 2016-2022 Altova GmbH

Schema **AddressLast.xsd**

schema location: C:\Users\alU\Documents\Altova\XMLSpy2013\Examples\Tutorial\AddressLast.xsd
attributeFormDefault: **unqualified**
elementFormDefault: **qualified**
targetNamespace: **http://my-company.com/namespace**

Elements Complex types Simple types
Company **AddressType** **US-State**
Person **UK-Address**
US-Address

element **Company**

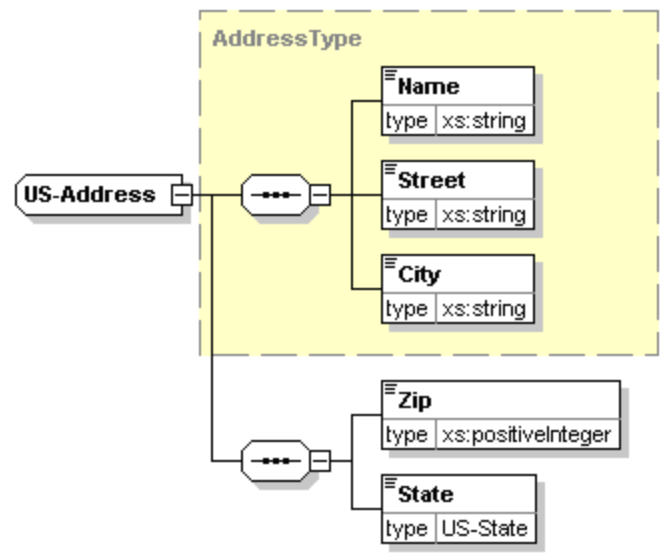
diagram	
namespace	http://my-company.com/namespace
properties	content complex
children	**Address** **Person**
annotation	documentation Root element

source

```
<xs:element name="Company">
  <xs:annotation>
    <xs:documentation>Root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Address" type="AddressType"/>
      <xs:element ref="Person" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Le diagramme ci-dessus montre la **première page** de la documentation de schéma dans le formulaire HTML. Si les composants provenant d'autres schémas ont été inclus, ces schémas sont aussi documentés.

complexType **US-Address**

diagram	
namespace	http://my-company.com/namespace
type	extension of **AddressType**
properties	name US-Address base AddressType
children	**Name Street City Zip State**
source	<xs:complexType name="US-Address"> <xs:complexContent> <xs:extension base="AddressType"> <xs:sequence> <xs:element name="Zip" type="xs:positiveInteger"/> <xs:element name="State" type="US-State"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Le diagramme ci-dessus montre la complexité de documentation des types.

© 2016-2022 Altova GmbH

element **US-Address/Zip**

diagram	**Zip** type xs:positiveInteger
namespace	http://my-company.com/namespace
type	**xs:positiveInteger**
properties	name Zip isRef 0 content simple
source	<xs:element name="Zip" type="xs:positiveInteger"/>

element **US-Address/State**

diagram	**State** type US-State
namespace	http://my-company.com/namespace
type	**US-State**
properties	name State isRef 0 content simple
source	<xs:element name="State" type="US-State"/>

simpleType **US-State**

namespace	http://my-company.com/namespace
type	**xs:string**
properties	name US-State
used by	element **US-Address/State**
source	<xs:simpleType name="US-State"> <xs:restriction base="xs:string"/> </xs:simpleType>

Le diagramme ci-dessus montre comment les éléments et les types simples sont documentés.

Vous pouvez maintenant essayer l'option de sortie MS Word. Le document Word s'ouvrira dans MS Word. Pour utiliser des hyperliens dans le document MS Word, maintenir la touche **Ctrl** appuyée tout en cliquant sur le lien.

2.5 Documents XML

Dans cette section, vous apprendrez comment créer et travailler avec des documents XML dans XMLSpy. Vous apprendrez également comment utiliser les différentes fonctions d'édition intelligentes de XMLSpy.

Objectifs

Les objectifs pour cette section sont d'apprendre à :

- Créer un nouveau document XML basé sur le schéma `AddressLast.xsd`.
- Spécifier le type d'un élément de manière à rendre disponible un modèle de contenu étendu disponible pour cet élément pendant la validation.
- Insérer les éléments et les attributs et saisir un contenu dans le Mode Grille et le Mode Texte en utilisant des assistants à la saisie intelligents.
- Copier les données XML depuis XMLSpy vers Microsoft Excel ; ajouter des données dans MS Excel et copier les données modifiées depuis MS Excel vers XMLSpy. Cette fonction est disponible dans l'Affichage Table du Mode Grille.
- Trier les éléments XML en utilisant la fonction de l'Affichage Table.
- Valider le document XML.
- Modifier le schéma pour permettre les extensions de téléphone à trois chiffres.

Commandes utilisées dans cette section

Dans cette section du tutoriel, vous allez principalement utiliser le Mode Grille et le Mode Texte ainsi que le Mode Schéma. Les commandes suivantes sont utilisées :

	**Fichier | Nouveau**. Créer un nouveau type de fichier XML.
	**Mode | Mode Texte**. Passe en Mode Texte.
	**Mode | Mode Grille**. Passe en Mode Grille améliorée.
	**XML | Afficher en que table**. Affiche les occurrences multiples d'un seul type d'élément dans un niveau hiérarchique simple en tant que table. Ce Mode de l'élément est appelé Affichage Table. L'icône permet de passer entre l'Affichage Table et le Mode Grille normal.
	**F7**. Vérifier la bonne formation.
	**F8**. Valide le document XML par rapport au DTD ou au Schéma associé.
	Ouvrir le fichier DTD ou Schéma XML associé.

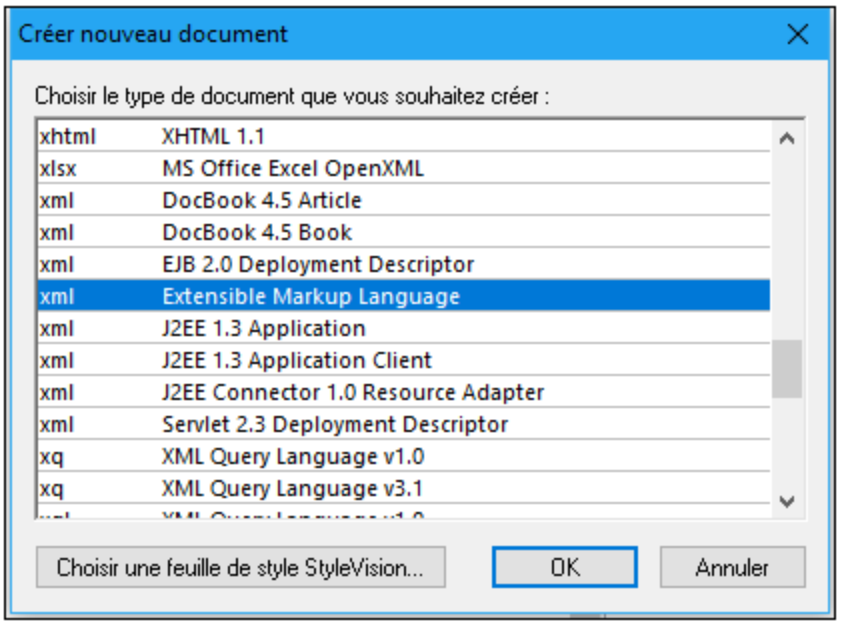
2.5.1 Créer un nouveau fichier XML

Lorsque vous créez un nouveau fichier XML dans XMLSpy, vous avez la possibilité de le baser sur un schéma (DTD ou Schéma XML) ou pas. Dans cette section, vous allez créer un nouveau fichier basé sur le schéma **`AddressLast.xsd`** que vous avez créé précédemment dans le tutoriel.

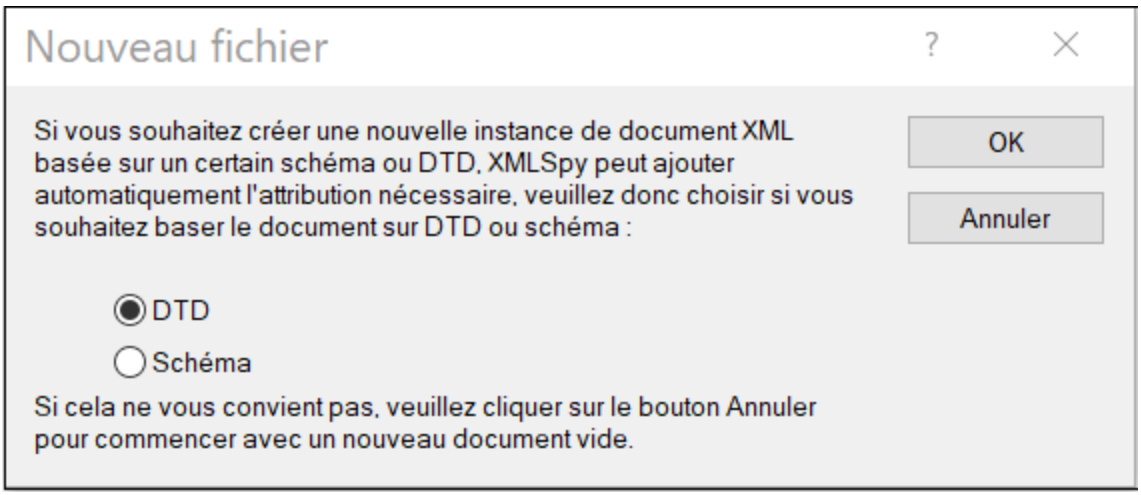
© 2016-2022 Altova GmbH

Pour créer le nouveau fichier XML :

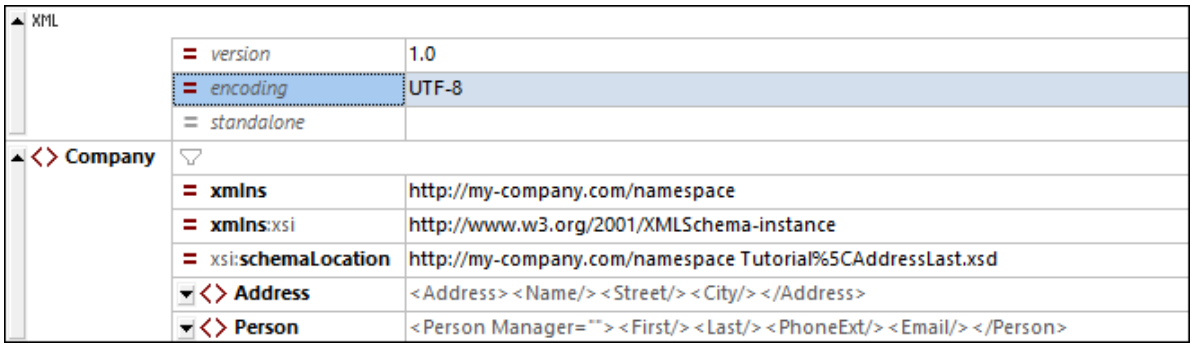
1. Sélectionnez l'option de menu **Fichier | Nouveau**. Le dialogue *Créer nouveau document* s'ouvre.



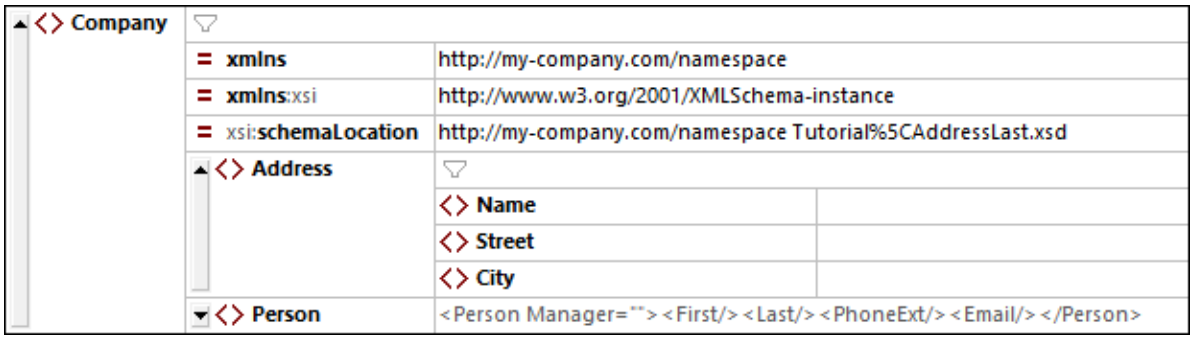
2. Sélectionnez l'entrée *Extensible Markup Language* (ou l'entrée de document XML générique) depuis le dialogue et confirmez avec **OK**. Un message apparaît vous invitant à baser le document XML sur un DTD ou un Schéma.



3. Cliquez sur le bouton radio Schéma et confirmer avec **OK**. Un dialogue supplémentaire apparaît, vous demandant de sélectionner le fichier de schéma sur lequel votre document XML sera basé.
4. Utiliser les boutons **Browse** ou **Window** pour trouver le fichier de schéma. Le bouton **Window** recense tous les fichiers ouverts dans XMLSpy et les projets. Sélectionnez `AddressLast.xsd` (voir introduction du Tutoriel pour l'emplacement), et confirmez avec **OK**. Un document XML contenant les éléments principaux définies par le schéma s'ouvre dans la fenêtre principale.
5. Cliquer l'onglet Grille pour sélectionner le Mode Grille.
6. Dans le Mode Grille, veuillez noter la structure du document. Cliquez sur n'importe quel élément pour réduire la sélection de cet élément. Votre document devrait ressembler à l'exemple ci-dessous :



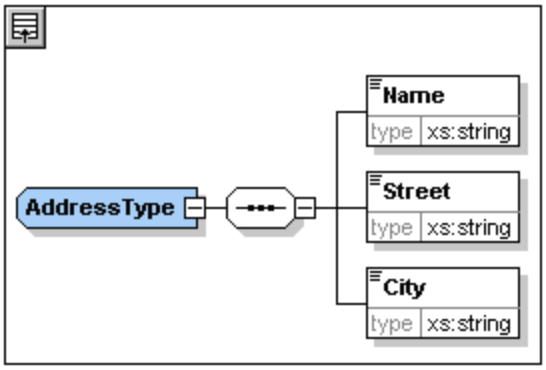
7. Cliquer sur l'icône située à côté de **Address** pour voir les éléments enfant de **Address**. Votre document devrait ressembler à l'exemple ci-dessous :



8. Sélectionnez l'option menu **Fichier | Enregistrer** et enregistrez-le dans le dossier **Tutoriel**. Donnez un nom adapté à votre document XML (par exemple **CompanyFirst.xml**). Notez que le fichier du tutoriel terminé se trouve dans **CompanyFirst.xml** le dossier **Tutoriel**, donc vous allez devoir le renommer avant de donner ce nom au fichier que vous avez créé.

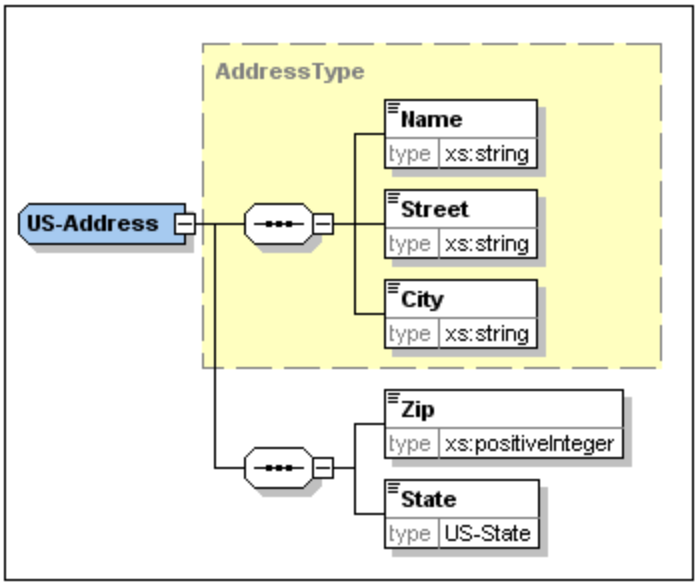
2.5.2 Spécifier le type d'un élément

Les éléments enfant de **Address** sont ceux définis pour le type complexe global **AddressType** (le modèle de contenu défini dans le Schéma XML **AddressLast.xsd** montré dans la capture d'écran Mode Schéma ci-dessous).



 © 2016-2022 Altova GmbH

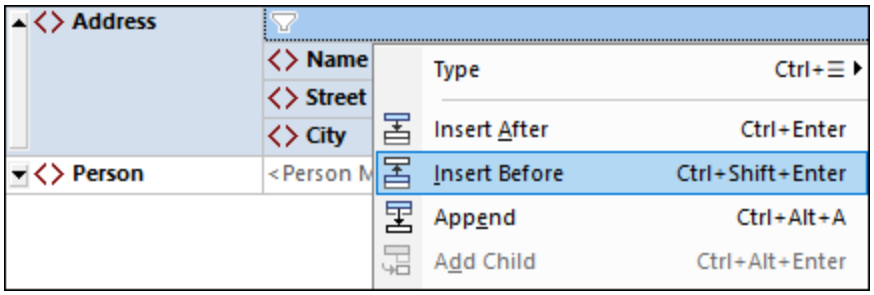
Néanmoins, nous souhaitons utiliser un type d'adresse spécifique pour les US ou le RU à la place du type d'adresse générique. Vous vous rappelez certainement que dans le schéma **AddressLast.xsd**, nous avons créé des types complexes globaux pour **US-Address** et **UK-Address** en étendant le type complexe **AddressType**. Le modèle de contenu de **US-Address** est affiché ci-dessous.



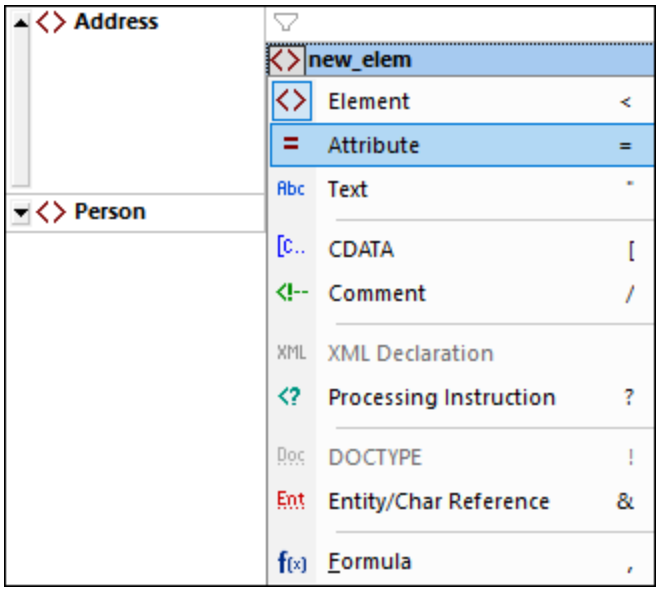
Dans le document XML, afin de spécifier que l'élément **Address** doit être conforme à l'un des types étendus **Address** (**US-Address** ou **UK-Address**) plutôt qu'au type générique **AddressType**, nous devons spécifier le type complexe étendu en tant qu'un attribut de l'élément **Address**.

Nous ajoutons cet attribut de l'élément **Address** comme suit :

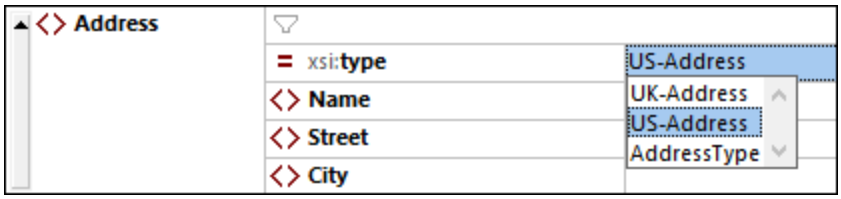
1. Dans le document XML, cliquez avec la touche de droite l'élément `Name` et sélectionnez **Insérer avant** depuis le menu contextuel (*voir la capture d'écran ci-dessous*).



2. Un nouveau nœud d'élément nommé **new_elem** est ajouté avant l'élément **Name** (*voir la capture d'écran ci-dessous*). Cliquez sur le type d'élément à gauche du nom de nœud et, dans le menu qui apparait (*voir la capture d'écran ci-dessous*), sélectionnez le type de nœud *Attribut*. Le type de nœud changera en type de nœud *Attribut*; toutefois, le nom restera **new_elem**.



3. Double-cliquez sur le nom du nœud et, dans le popup de l'assistant de saisie qui apparait, sélectionnez **xsi:type**.
4. Appuyer sur **Tab** pour passer au champ de valeur de l'attribut. Un menu popup apparaît recensant les valeurs disponibles **xsi:type** est affiché (*voir la capture d'écran ci-dessous*). Ces valeurs sont les types complexes qui ont été définis pour l'élément **Address** dans le schéma.



5. Sélectionnez **US-Address** en tant que valeur de l'attribut **xsi:type**.

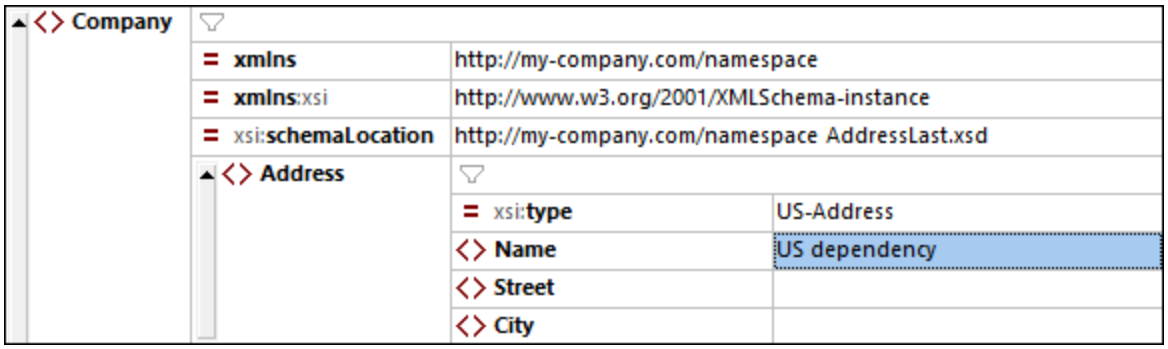
Note : le préfixe **xsi:** vous permet d'utiliser les commandes spéciales liées à Schéma XML dans votre document d'instance XML. Veuillez noter que l'espace de noms pour le préfixe **xsi:** a été ajouté automatiquement à l'élément de document lorsque vous attribuez un schéma à votre fichier XML. Dans le cas ci-dessus, vous avez spécifié un type pour l'élément **Address**. Voir la Spécifications de schéma XML pour plus d'informations.

2.5.3 Saisir les données dans le Mode Grille

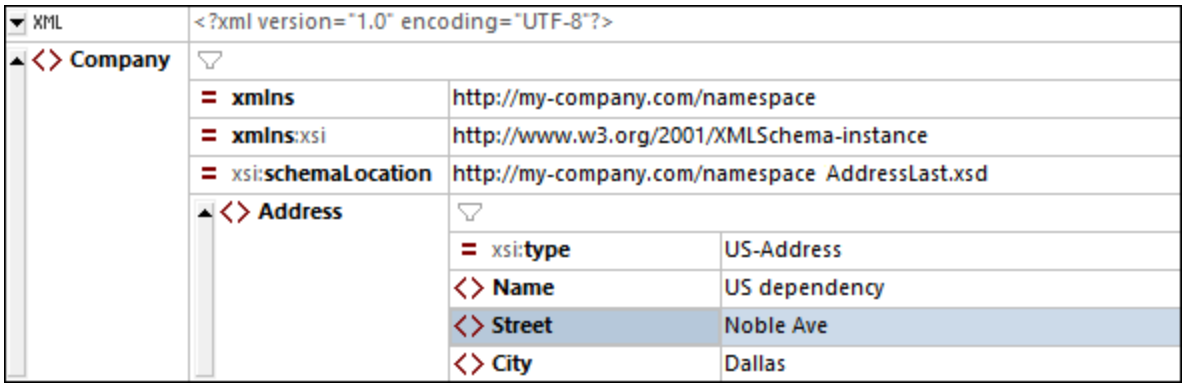
Vous pouvez maintenant saisir les données dans votre document XML. Procéder comme suit :

1. Double-cliquer le champ de valeur `Name` (ou utiliser les touches flèchées) et saisir `US dependency`. Confirmer avec **Entrée**.

© 2016-2022 Altova GmbH



2. Utiliser la même méthode pour saisir un nom de `Street` et `City` (par exemple, `Noble Ave` et `Dallas`).
3. Cliquer sur l'élément `Person` et appuyer sur **Supprimer** pour supprimer l'élément `Person`. (Nous le rajouterons plus tard dans la prochaine section du tutoriel.) Une fois avoir achevé cette opération, tout l'élément `Address` est marqué.
4. Cliquer sur n'importe quel élément enfant de l'élément `Address` pour désélectionner tous les éléments enfants de `Address` sauf l'élément sélectionné. Votre document XML devrait ressembler à l'exemple ci-dessous :



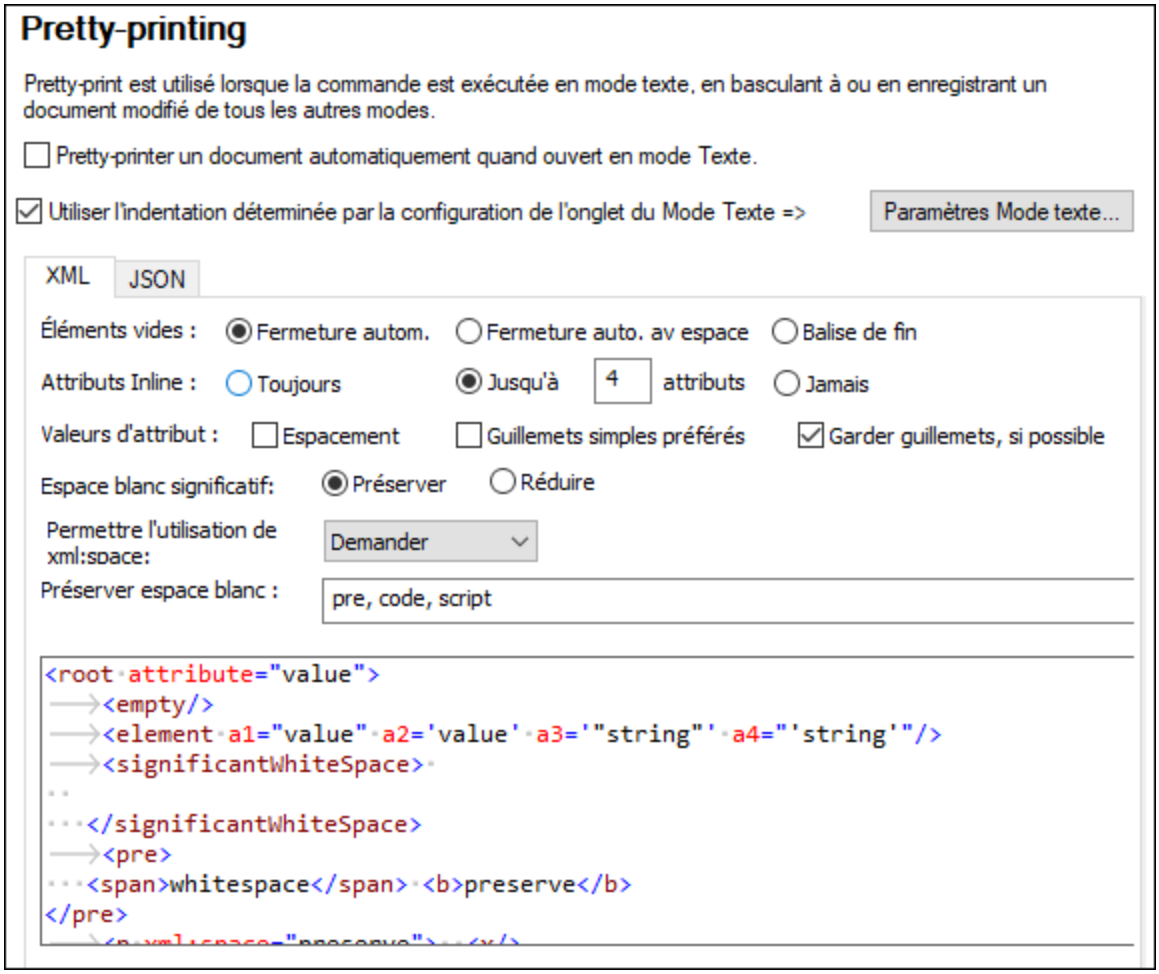
2.5.4 Saisir les données dans le Mode Texte

Le Mode Texte présente les données actuelles et les balises des fichiers XML dans une mise en page structurelle très conviviale, et offre des fonctions d'édition intelligentes liées au schéma

Mise en page du document

La mise en page du document dans le Mode Texte est définie à deux emplacements :

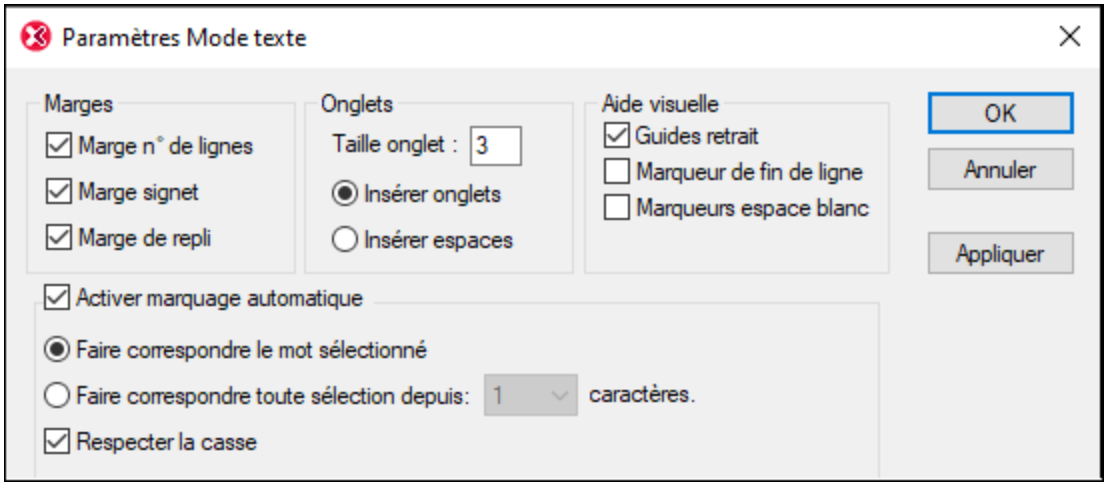
Options de Pretty-print
Ces paramètres sont dans la section Pretty-printing du dialogue des Options (*voir la capture d'écran ci-dessous*). Lorsque vous définissez une option, ses incidences peuvent être vues immédiatement en bas dans le volet de visualisation. Définissez les options de pretty-printing comme bon vous semble. Lorsque vous éditez dans le Mode Texte, vous trouverez éventuellement que la mise en page du document est de moins en moins structurée, en particulier après avoir copier-glisser des blocs de texte. Chaque fois que vous voulez avoir une mise en page propre et hiérarchique, cliquez simplement sur la commande **Éditer | Pretty Print**.



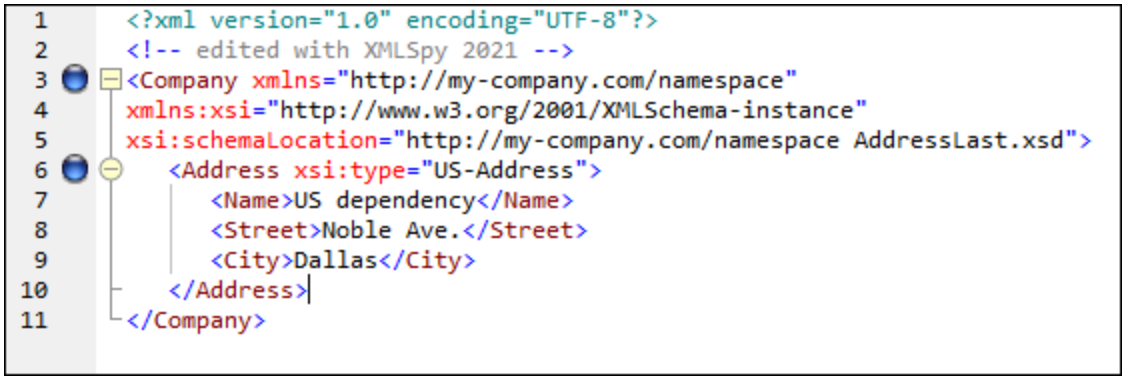
Paramètres du Mode Texte

Le dialogue des Paramètres du Mode Texte (*voir la capture d'écran ci-dessous*) fournit non seulement des options de mise en page additionnelles mais active/désactive des fonctions du Mode Texte, telles que numéroter des lignes et plier des marges. Accédez le dialogue des paramètres du Mode Texte avec la commande **Mode | Paramètres du Mode Texte**.

© 2016-2022 Altova GmbH



La capture d'écran ci-dessous affiche le fichier XML actuel dans le Mode Texte avec des fonctions activées conformément aux paramètres dans le dialogue ci-dessus.



À gauche, vous trouverez trois marges : (i) la marge de numérotation des lignes, (ii) la marge des signets (contenant deux signets bleus), et (iii) la marge de pli de source (qui vous permet d'agrandir et de réduire l'affichage des éléments XML). Les guides d'indentation sont les lignes verticales grises qui montrent l'indentation des onglets au même niveau hiérarchique. De plus, des aides visuelles supplémentaires comme des guides d'indentation, des marqueurs de fin de ligne et des marqueurs d'espaces vides peuvent être activés ou désactivés, en cochant et décochant respectivement les cases dans le panneau *Aide visuelle* (*voir la capture d'écran ci-dessus*).

Note : Les fonctions de pretty-print et de signets liés au Mode Texte ont été abordées dans la section précédente Paramètres du Mode Texte de ce tutoriel.

Éditer dans le Mode Texte

Dans cette section, vous allez saisir et éditer les données dans Mode Texte afin de vous familiariser avec les fonctions du Mode Texte.

Note : Puisque la fonction *Validation sur Éditer* est activée par défaut, toute erreur de validation créée pendant l'édition sera marquée immédiatement, et le message d'erreur sera affiché dans la fenêtre Messages. Ignorez ces erreurs et ces messages pour l'instant. Si vous ne souhaitez pas une validation en arrière-

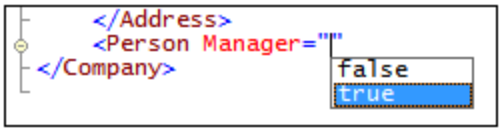
plan, vous pouvez éteindre la *Validation sur Éditer* dans les Paramètres de validation du dialogue Options. Dans le cas où vous suivez cette procédure, veuillez noter que vous pouvez toujours valider votre document à tout moment (décrit dans la prochaine section de ce tutoriel).

Pour ce faire, procédez comme suit :

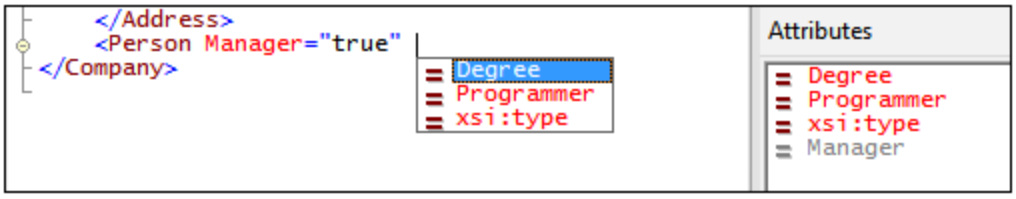
1. Sélectionnez le point de menu **Mode | Mode Texte**, ou cliquez sur l'onglet **Texte**. Vous voyez maintenant le document XML dans sa forme de texte, avec la coloration de syntaxe.
2. Placez le curseur de texte à la fin de la balise de l'élément **Address** et appuyez sur **Enter** pour ajouter une nouvelle ligne.
3. Saisissez le signe « inférieur à » **<** à cet endroit. Une liste déroulante de tous les éléments autorisés à ce point (conformément au schéma) s'affiche. Puisque seul l'élément **Person** est autorisé à cet endroit, ce sera le seul élément affiché dans la liste.



4. Sélectionnez l'entrée **Person**. L'élément **Person** ainsi que son attribut **Manager** sont insérés alors que le curseur se trouve dans le champ de valeur de l'attribut **Manager**.
5. À partir de la liste déroulante qui apparaît pour l'attribut **Manager**, sélectionnez **true**.

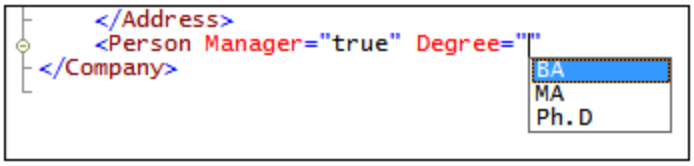


6. Déplacez le curseur à la fin de la ligne (en utilisant la touche **Fin** si vous le souhaitez), et appuyez sur la barre d'espace. Une liste déroulante s'ouvre contenant une liste des attributs autorisés à cet endroit. De même, dans l'assistant à la saisie des Attributs, les attributs disponibles sont recensés en rouge. L'attribut **Manager** est gris parce qu'il a déjà été utilisé.

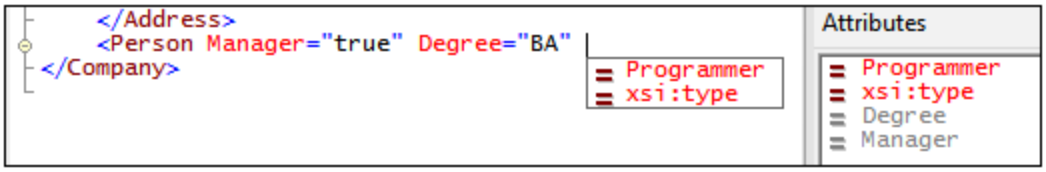


7. Sélectionnez **Degree** avec la touche fléchée vers le bas et appuyez sur **Enter**. Une autre liste déroulante s'ouvre, à partir de laquelle vous pouvez sélectionner une des énumérations prédéfinies (**BA**, **MA**, ou **PhD**). (Les énumérations sont des valeurs qui sont autorisées par le Schéma XML.)

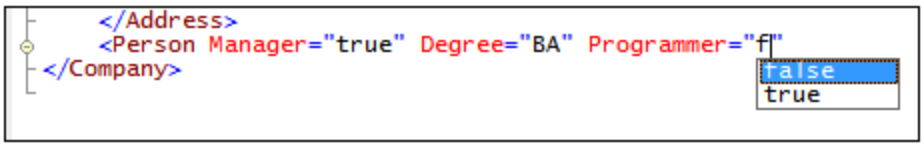
© 2016-2022 Altova GmbH



8. Sélectionnez **BA** avec la touche fléchée vers le bas et confirmez avec **Enter**. Ensuite, déplacez le curseur à la fin de la ligne (avec la touche **Fin**), et appuyez sur la barre d'espace. **Manager** et **Degree** apparaissent maintenant en gris dans l'Assistant de saisie des Attributs.



9. Sélectionnez **Programmer** avec la touche fléchée vers le bas et appuyez sur **Enter**.



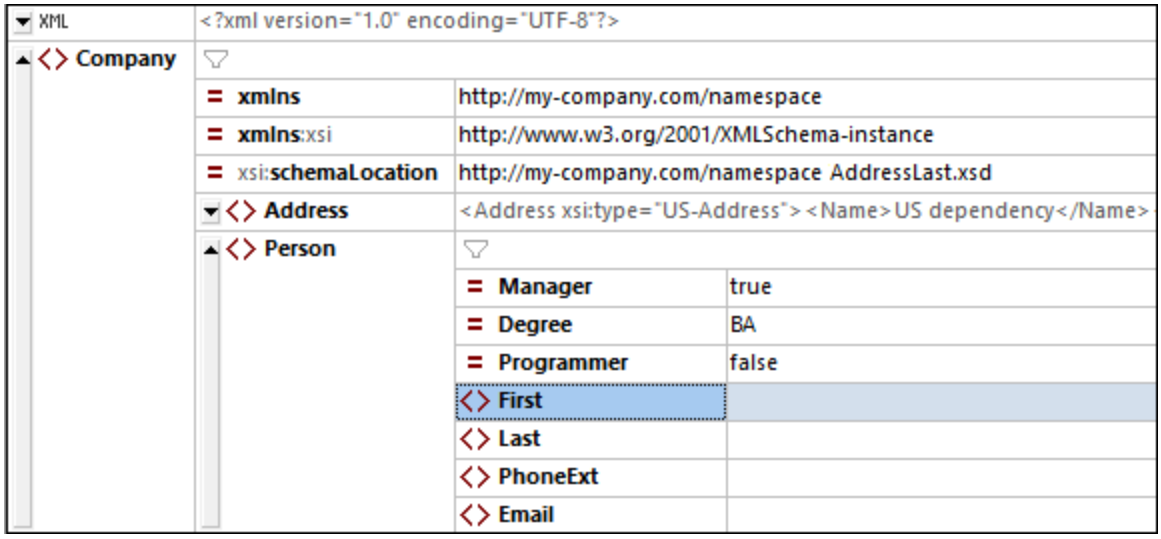
10. Saisissez la lettre « **f** » et appuyez sur **Enter**.
11. Déplacez le curseur à la fin de la ligne (avec la touche **Fin**), et saisissez la barre d'espace >. XMLSpy insère automatiquement tous les éléments enfants nécessaires de **Person**. (Veuillez noter que l'élément optionnel **Title** n'est pas inséré.) Chaque élément présente des balises de début et de fin mais n'a pas de contenu.



Vous pouvez maintenant saisir les données **Person** dans le Mode Texte, mais rendons-nous d'abord au Mode Grille pour voir la flexibilité de déplacement entre les modes lors de l'édition d'un document.

Passer en Mode Grille

Pour passer au Mode Grille, sélectionner l'élément de menu **Mode | Mode Grille,** ou cliquez sur l'onglet **Grille**. Les éléments enfants récemment ajoutés de `Person` sont affichés.



À présent, validons le document et corrigeons toute erreur éventuelle trouvée au cours de la validation.

2.5.5 Valider le document

XMLSpy offre deux évaluations du document XML :

- Une vérification de la bonne formation
- Une vérification de la validation

	Vérifier la bonne formation
	Valider XML

Puisque la fonction *Validation sur Éditer* est activée par défaut, toute erreur de validation créée pendant l'édition sera marquée immédiatement, et le message d'erreur sera affiché dans la fenêtre Messages. Si vous ne souhaitez pas une validation en arrière-plan, vous pouvez éteindre la *Validation sur Éditer* dans les Paramètres de validation du dialogue Options. Dans le cas où vous suivez cette procédure, veuillez noter que vous pouvez toujours effectuer des vérifications de bonne formation et des vérifications de validation séparément à tout moment en invoquant la commande respective dans le menu XML. La partie du tutoriel vous montre comment exécuter ces vérifications.

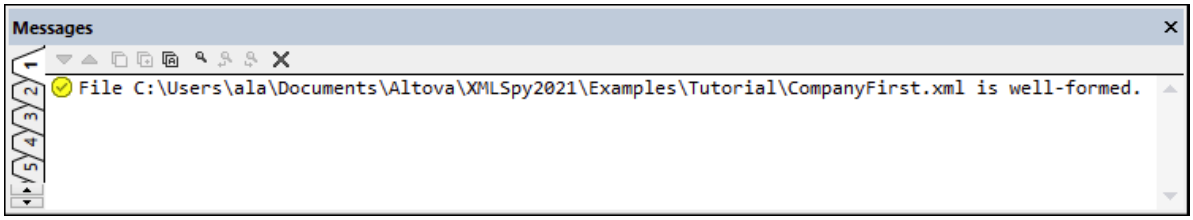
Vérifier la bonne formation

Un document XML est bien formé si les balises de début sont conformes aux balises de fin, si les éléments sont imbriqués correctement, s'il n'y a pas de caractère déplacé ou manquant (comme par exemple une entité sans son délimiteur point-virgule). Vous pouvez effectuer une vérification de la bonne formation dans tous les modes d'édition. Vérifiez votre document comme suit :

1. Choisissez le Mode Texte.
2. Sélectionnez l'option de menu **XML | Vérification de la bonne forme** ou appuyez sur la touche **F7**. (En alternative, vous pouvez cliquer sur l'icône de la commande dans la barre d'outils.) Un message

 © 2016-2022 Altova GmbH

apparaît dans la fenêtre Messages en bas de la Fenêtre principale indiquant que le document est bien formé.



Veuillez noter que la sortie de la fenêtre de Messages a 9 onglets, et la sortie de validation est toujours affichée dans l'onglet actif. Vous pourrez vérifier la bonne formation dans l'onglet 1, et basculer vers l'onglet 2 pour une vérification de la validation. Si vous ne basculez pas vers les onglets, le nouveau résultat écrase le résultat précédent dans l'onglet actif.

Note : cette validation ne vérifie pas la structure du fichier XML pour une conformité avec le schéma. La conformité au schéma est évaluée dans la vérification de la validité.

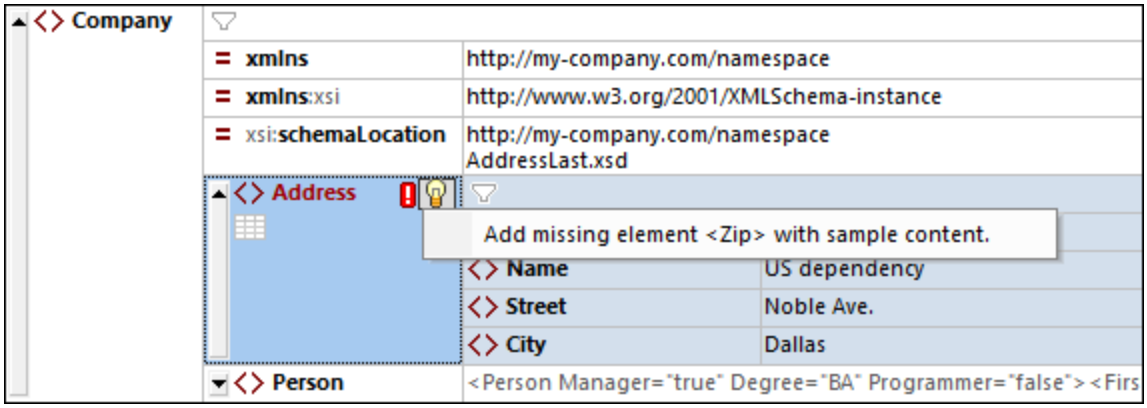
Vérifier la validité

Un document XML est valide conformément à un schéma s'il est conforme à la structure et au contenu spécifié dans ce schéma. Vous pouvez effectuer une vérification de la validité dans tous les modes d'édition. Validez votre document comme suit :

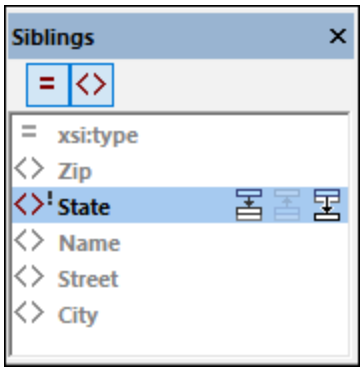
1. Choisissez le Mode Grille.
2. Sélectionnez l'option de menu **XML | Valider** ou appuyez sur la touche **F8**. (En alternative, vous pouvez cliquer sur l'icône de la commande dans la barre d'outils.) Un message d'erreur apparaît dans la fenêtre Messages indiquant que le fichier n'est pas valide. Des éléments obligatoires sont attendus après l'élément **City** dans **Address**. Si vous vérifiez votre schéma, vous verrez que le type complexe **US-Address** (avec lequel vous avez défini cet élément **Address** pour être avec son attribut **xsi:type**) a un modèle de contenu dans lequel l'élément **City** doit être suivi par un élément **Zip** et un élément **State**.

Réparer le document invalide

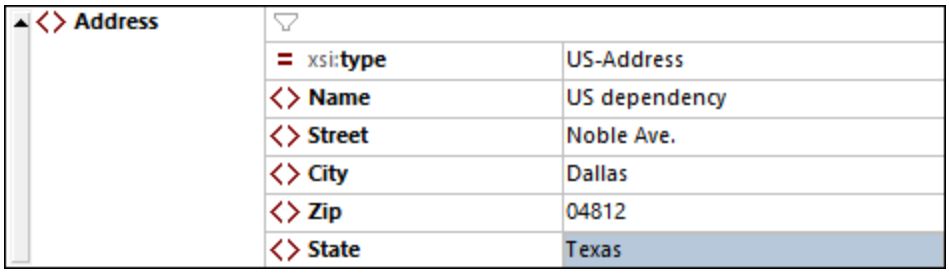
Le point à partir duquel le document devient invalide est marqué en rouge, de pair avec un indicateur d'erreur et un correctif d'erreur. L'élément invalide dans ce cas est l'élément **Address**. Si vous cliquez sur l'icône du correctif d'erreur, vous verrez la fenêtre contextuelle : *Ajouter l'élément manquant <Zip> avec un échantillon de contenu.* Si vous vérifiez le schéma, vous verrez que l'élément **Address/City** doit être suivi d'un élément obligatoire **Zip**. Pour revérifier, sélectionnez l'élément **City** et regardez l'assistant de saisie Siblings. Vous verrez que l'élément **Zip** est préfixé d'un signe d'exclamation, ce qui indique que l'élément est obligatoire dans le contexte actuel.



Cliquez maintenant sur le correctif d'erreur (*voir la capture d'écran ci-dessus*). L'élément **Zip** sera ajouté et contiendra un échantillon de contenu qui rend l'élément valide. Saisissez le code postal **Zip** correct (disons **04812** pour Dallas). Regardez l'assistant de saisie Siblings. Il indique maintenant que l'élément **State** est obligatoire (il est préfixé d'un signe d'exclamation). Si vous sélectionnez l'élément **State**, les options de l'assistant de saisie disponibles deviennent activées (*voir la capture d'écran ci-dessous*). Voici donc les actions à exécuter pour insérer l'élément **State** après l'élément actuellement choisi dans la Fenêtre principale (qui est **City**) ou pour apposer **State** à la suite de tous les éléments sibling de **City**.



Puisque, dans les deux cas, les deux actions ont le même effet, sélectionnez l'une ou l'autre des deux actions. Un élément **State** est ajouté après **City**. Double-cliquez à l'intérieur du champ de contenu de **State** et saisissez nom de l'État, **Texas** (*voir la capture d'écran ci-dessous*). Veuillez noter que l'assistant de saisie Siblings contient maintenant uniquement des élément grisés, indiquant qu'il n'y a plus aucun élément obligatoire à ajouter.



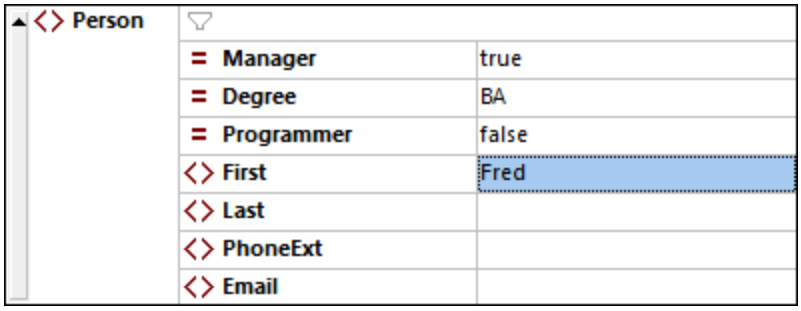
 © 2016-2022 Altova GmbH

Compléter le document et le revalider

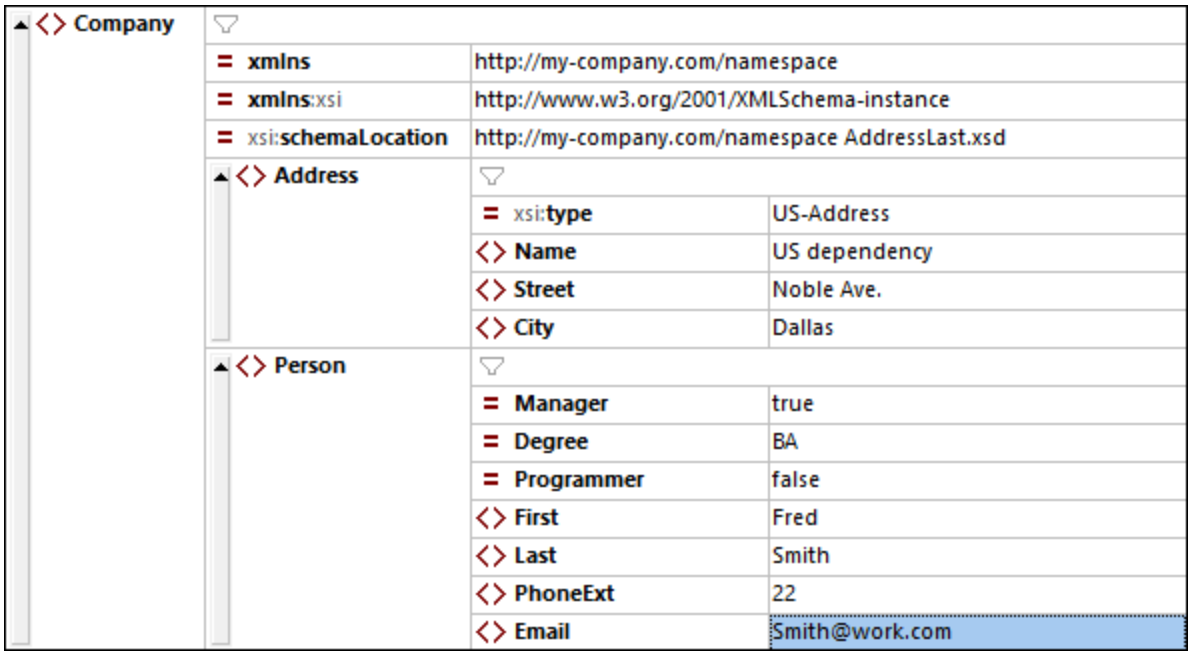
Complétons à présent le document (en saisissant les données restantes du premier élément `Person`) avant de revalider.

Pour ce faire, procédez comme suit :

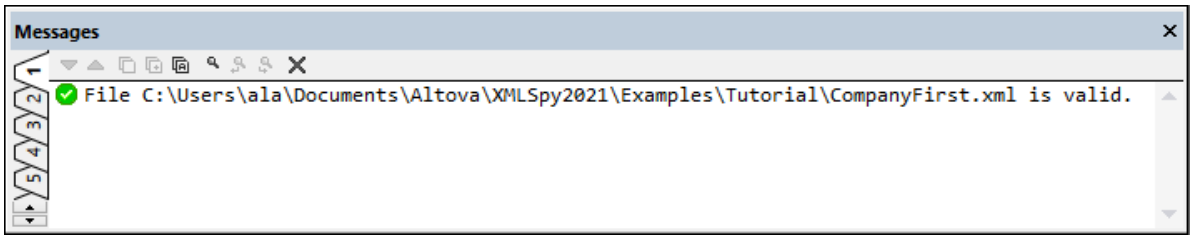
1. Cliquez sur le champ de valeur de l'élément **`First`**, et saisissez un prénom (disons `Fred`). Puis, appuyez sur **Enter**.



2. De la même manière, saisissez les données pour tous les éléments enfants de **`Person`**, à savoir, pour **`Last`**, **`PhoneExt`**, et **`Email`**. Vous pouvez utiliser la touche Tab pour vous déplacez à travers les cellules. Veuillez noter que la valeur de **`PhoneExt`** doit être un entier avec une valeur maximum de **`99`** (puisqu'il s'agit de la plage de valeurs autorisées **`PhoneExt`** que vous avez définie dans votre schéma). Votre document XML devrait ressembler à l'image suivante en Mode Grille :



3. Recliquez  pour vérifier si le document est valide. Un message apparaît dans la fenêtre Messages indiquant que le fichier est valide. Le document XML est maintenant valide par rapport à son schéma.



4. Enregistrez le fichier dans **File | Save**.

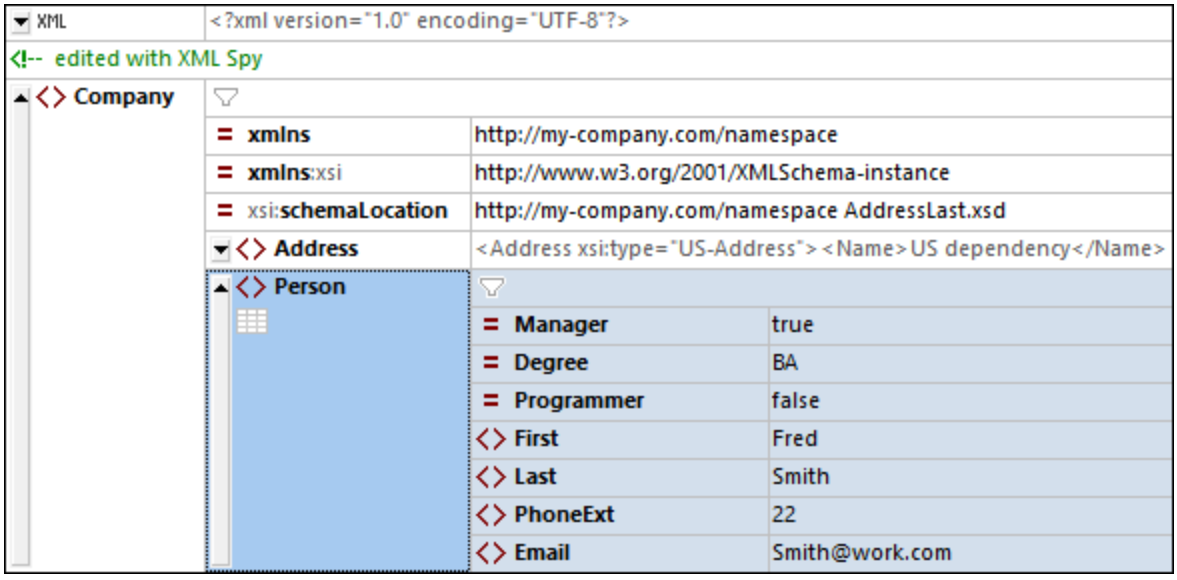
Note : un document XML document ne doit pas être valide afin de pouvoir l'enregistrer. Le fait d'enregistrer un document invalide entraîne l'apparition d'un avertissement vous informant que vous êtes sur le point d'enregistrer un document invalide Vous pouvez sélectionner **Enregistrer tout de même** si vous souhaitez enregistrer le document dans son état invalide actuel.

2.5.6 Ajouter des éléments et des attributs

À ce stade, le document ne recèle qu'un seul élément **`Person`**.

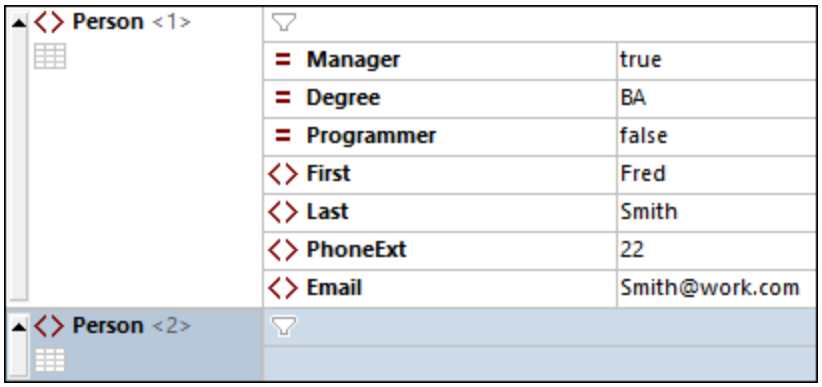
Pour ajouter un nouvel élément **`Person`**, procédez comme suit :

1. Cliquer sur la barre latérale grise située à gauche de l'élément **`Address`** pour réduire l'élément **`Address`**. Cela vous permet de dégager de l'espace dans ce mode.
2. Choisissez tout l'élément **`Person`** en cliquant sur ou sous le nom de l'élément **`Person`** dans le Mode Grid. Veuillez noter que l'élément **`Person`** est désormais disponible dans l'assistant de saisie Siblings.

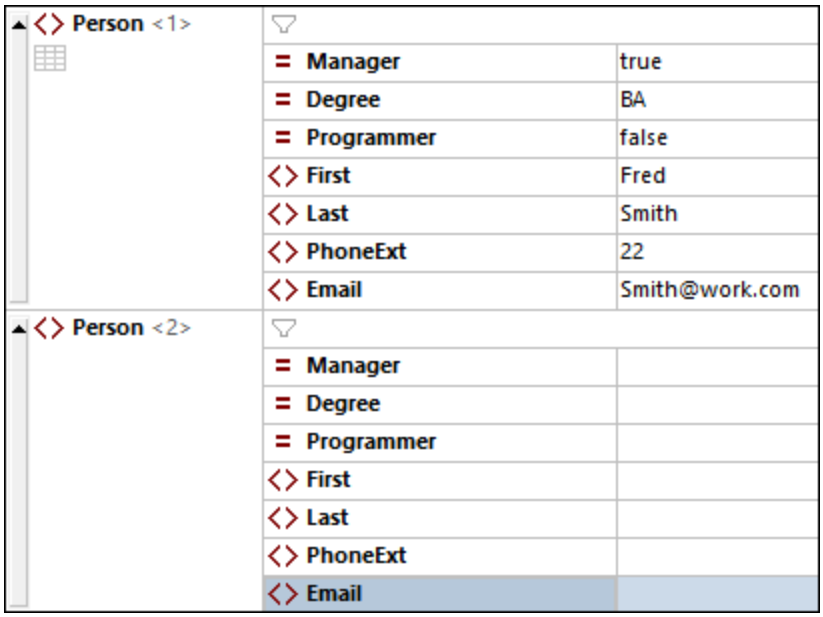


3. Choisissez l'élément **`Person`** dans l'assistant de saisie Siblings et cliquez soit sur **Insérer après** ou **Apposer**. Un nouvel élément **`Person`** element est ajouté (*voir la capture d'écran ci-dessous*).

 © 2016-2022 Altova GmbH



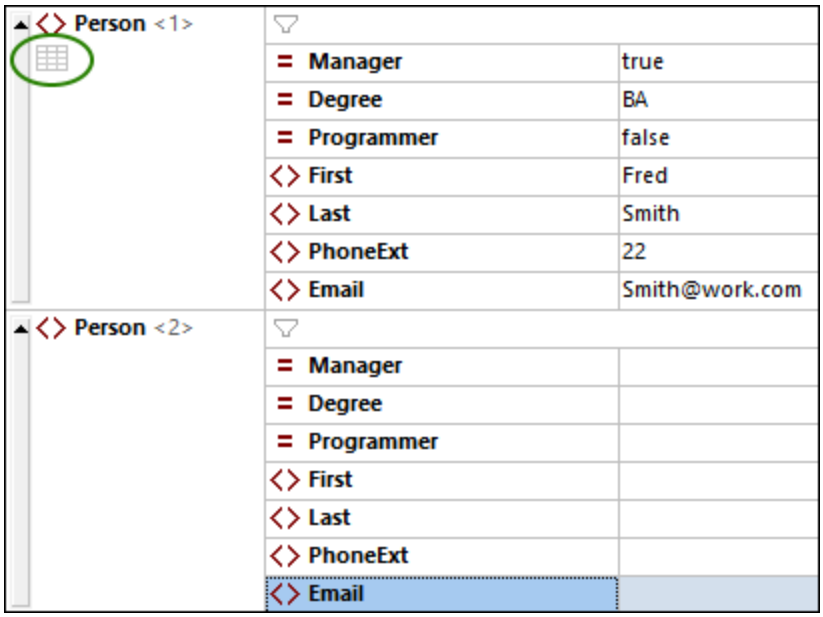
4. Quand l'élément **Person** est choisi, vous verrez les attributs et éléments enfant disponibles de cet élément dans l'Assistant de saisie des Attributs. Double-cliquez sur les attributs et éléments pour ajouter les mêmes nœuds enfant que pour le premier élément **Person**. Quand le focus dans la Fenêtre principale change de l'élément **Person** à un élément enfant ajouté, vous pouvez ajouter des enfants supplémentaires de l'élément **Person** de deux manières : (i) Changez le focus de l'élément **Person** (en le sélectionnant) et ajoutez un nouvel enfant depuis l'assistant de saisie Enfant ; (ii) en mettant le focus sur l'élément enfant ajouté, ajoutez un élément enfant sibling depuis l'assistant de saisie Siblings. Dans les deux assistants de saisie, les nœuds enfant de **Person** qui ont déjà été ajoutés seront grisés.



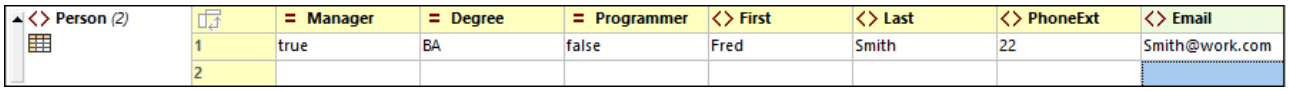
Vous pourriez saisir un contenu pour les nœuds enfant de l'élément **Person** dans le Mode Grille normal, mais passons à l'Affichage Table du Mode Grille puisqu'il est mieux adapté pour l'édition d'une structure avec des occurrences multiples, comme **Person**.

2.5.7 Édition dans le Mode Base de données/Table

Le Mode Grille recèle un mode spécial appelé Affichage Table, qui est pratique pour l'édition d'éléments à occurrences multiples. Par exemple, l'élément **Person** a des occurrences multiples (*voir la capture d'écran ci-dessous*), et peuvent être affichées dans une table. Pour afficher un tel élément en tant que table, cliquez sur l'icône de l'**Affichage Table** de la première occurrence de l'élément. Par exemple, dans la capture d'écran ci-dessous, l'icône de l'**Affichage Table** des éléments **Person** est encerclée en vert. (Alternativement, sélectionnez la commande de menu **XML | Afficher en tant que Table** ou l'icône de barre d'outils de la commande dans barre d'outils du Mode Grille.)



Lorsque vous cliquez sur l'icône Affichage Table, l'élément **Person** sera affichée en tant que table. Dans l'Affichage Table, les nœuds enfant de l'élément (ses attributs et éléments) sont affichés en colonnes, tandis que chaque élément **Person** est affiché en tant que ligne (*voir la capture d'écran ci-dessous*).



Avantages de l'Affichage Table
L'Affichage Table fournit les avantages suivants :

- Vous pouvez glisser-déposer une en-tête de colonne pour repositionner des colonnes entières associées l'une à l'autre. Dans le document XML actuel, ceci convertit en un changement de la position relative de nœuds enfant de toutes les occurrences d'éléments (qui correspondent aux lignes d'une table).
- Les tables—et, respectivement, les occurrences d'élément qu'elles représentent—peuvent être triées (par ordre croissant ou décroissant) conformément aux contenus de chaque colonne. Pour ce faire, utilisez la commande de menu **XML | Trier par ordre croissant** ou **Trier par ordre décroissant**.
- Les lignes additionnelles (à savoir, les occurrences d'éléments) peuvent être apposées ou insérées rapidement en utilisant des commandes dans le menu **XML**. L'avantage est que non seulement un nouvel élément est ajouté mais aussi tous ses enfants représentés par les colonnes de la table.

© 2016-2022 Altova GmbH

- Vous pouvez copier-coller les *données structurées* vers et depuis les produits tiers, tels que Microsoft Excel.
- Les fonctions d'édition intelligentes de XMLSpy sont également disponibles dans l'Affichage Table.

Afficher un élément avec de multiples occurrences en tant que table

Pour afficher le type d'élément **Person** en tant que table, procédez comme suit :

1. Cliquez sur l'icône **Affichage Table** de la première occurrence de l'élément **Person**, tel que décrit ci-dessus.
2. Sélectionnez l'option du menu **Mode | Largeurs optimales** ou l'icône **Largeurs optimales** dans la barre d'outils du Mode Grille.

Note : l' Affichage Table peut être activé/désactivé pour tous les éléments qui ont de multiples occurrences. Toutefois, les éléments nœuds qui ont été affichés en tant que tables continueront de l'être.

Saisir du contenu dans l'Affichage Table

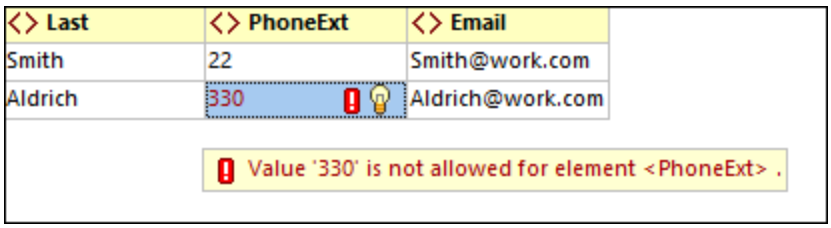
Pour saisir du contenu pour le deuxième élément **Person**, double-cliquez dans chacune des cellules de table dans la deuxième rangée, et saisissez des données. Les fonctions d'édition intelligente sont également actives à l'intérieur des cellules d'une table, donc vous pouvez sélectionnez des options des listes déroulantes, là où elles sont disponibles (par exemple, le contenu booléan et les énumérations des attributs de **Degré**).

	= Manager	= Degree	= Programmer	<> First	<> Last	<> PhoneExt	<> Email
1	true	BA	false	Fred	Smith	22	Smith@work.com
2	false	MA	true	Alfred	Aldrich	33	Aldrich@work.com

Validation dynamique

Notez que, tel que défini dans le schéma, **PhoneExt** doit être un entier de **0** à **99** pour que le fichier soit valide. Vous pouvez activer la fonction de XMLSpy pour valider lors de l'édition. Lorsqu'activé, le fichier est validé à chaque fois que le focus bascule vers un nouveau nœud. Pour la validation dynamique, procédez comme suit :

1. Activez la commande de menu **XML | Valider sur Édition**.
2. Saisissez une valeur invalide **PhoneExt** (chaque valeur supérieure à **99**), tel qu'affiché dans la capture d'écran ci-dessous.
3. Appuyez sur la touche **Tabulation**. Une icône d'erreur et une icône de correctif intelligent apparaissent dans la cellule **PhoneExt** (*voir la capture d'écran ci-dessous*).
4. Pointez sur l'icône d'erreur pour voir le message d'erreur-validation (*voir la capture d'écran ci-dessous*).

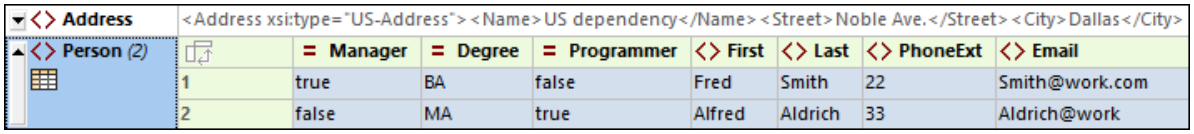


5. Cliquez sur l'icône du correctif intelligent et l'option du correctif intelligent apparaitra. La valeur invalide sera substituée par une valeur valide, et l'indicateur d'erreur disparaitra.

Copier des données XML vers et depuis les applications de feuilles de calcul

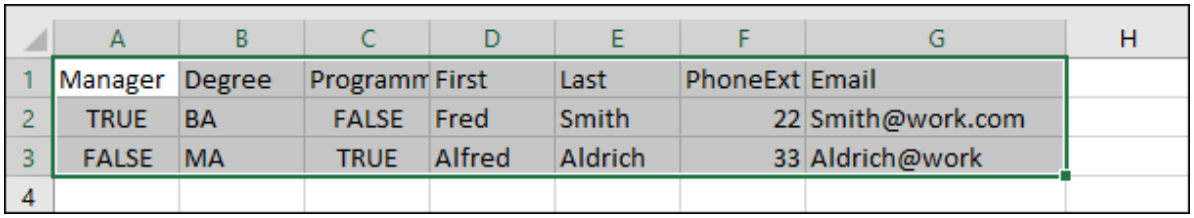
Lorsque vous êtes dans l'Affichage Table, vous pouvez copier les données en tant que texte séparé par une tabulation pour qu'il puisse être changé avec des applications de feuilles de calcul, tel que MS Excel. Pour copier des données dans votre fichier XML, procédez comme suit :

1. Cliquez sur l'élément **Person** (*voir la capture d'écran ci-dessous*). Ceci sélectionne l'en-tête de colonne ainsi que les deux lignes de la table.



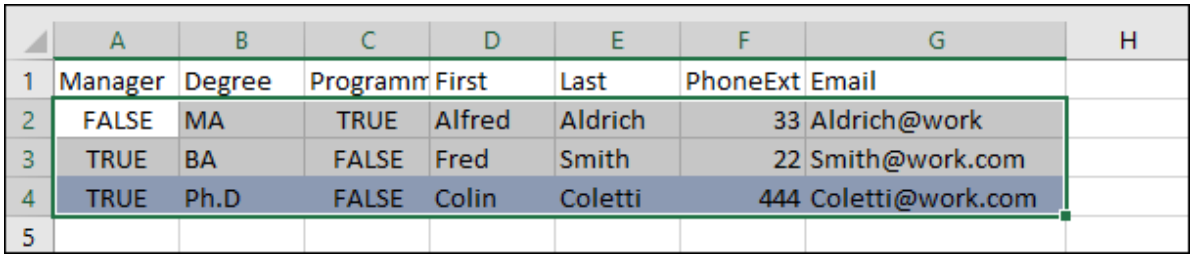
Address	<Address xsi:type="US-Address"><Name>US dependency</Name><Street>Noble Ave.</Street><City>Dallas</City>							
Person (2)		Manager	Degree	Programmer	First	Last	PhoneExt	Email
	1	true	BA	false	Fred	Smith	22	Smith@work.com
	2	false	MA	true	Alfred	Aldrich	33	Aldrich@work

2. Cliquez de la touche droite à l'intérieur de la sélection, dans le menu contextuel qui apparait, sélectionnez la commande **Copier | Copier en tant que texte séparé par tabulation**. Sinon, appuyez sur **Ctrl+C**.
3. Basculez vers une feuille de calcul Excel, sélectionnez la cellule A1 et collez les données XML avec (**Ctrl+V**). Les données seront saisies comme lignes qui correspondent à structure de table dans l'Affichage Table (*voir la capture d'écran ci-dessous*).



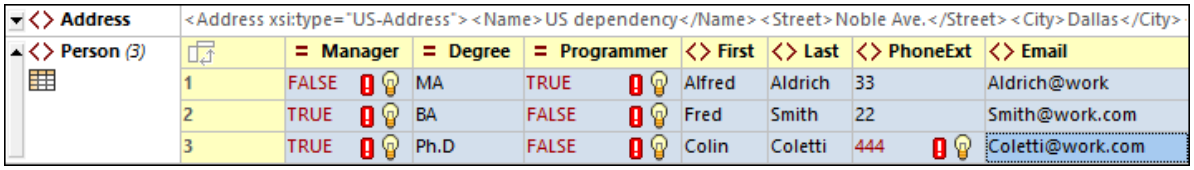
	A	B	C	D	E	F	G	H
1	Manager	Degree	Programm	First	Last	PhoneExt	Email	
2	TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com	
3	FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work	
4								

4. Saisissez une nouvelle ligne de données dans Excel tel qu'indiqué dans la capture d'écran ci-dessous. Veillez à saisir un nombre à trois chiffres pour l'élément **PhoneExt** (disons, **444**).
5. Marquez les données de table dans Excel, excluant les en-têtes de colonne (*voir le cadre vert dans la capture d'écran ci-dessous*), et copiez-les avec **Ctrl+C**.



	A	B	C	D	E	F	G	H
1	Manager	Degree	Programm	First	Last	PhoneExt	Email	
2	FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work	
3	TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com	
4	TRUE	Ph.D	FALSE	Colin	Coletti	444	Coletti@work.com	
5								

6. Dans XMLSpy, veillez à ce que la commande **XML | Valider sur Édition** est allumée.
7. Dans l'Affichage Table de votre document XML dans XMLSpy, sélectionnez la cellule **Manager** de la première ligne et collez les contenus du presse-papiers avec **Ctrl+V**. Votre nouvelle table aura plus ou moins l'aspect de ce qui est affiché dans la capture d'écran ci-dessous.



Address	<Address xsi:type="US-Address"><Name>US dependency</Name><Street>Noble Ave.</Street><City>Dallas</City>							
Person (3)		Manager	Degree	Programmer	First	Last	PhoneExt	Email
	1	FALSE	MA	TRUE	Alfred	Aldrich	33	Aldrich@work
	2	TRUE	BA	FALSE	Fred	Smith	22	Smith@work.com
	3	TRUE	Ph.D	FALSE	Colin	Coletti	444	Coletti@work.com

© 2016-2022 Altova GmbH

8. Les erreurs de validation pour les valeurs booléennes ont été causées par la différence de casse entre XML et Excel. Pour les corriger, appliquez les correctifs intelligents des cellules de table respectives.

Trier la table depuis les contenus d'une colonne

Dans l'Affichage Table, une table peut être triée, par ordre croissant ou décroissant, depuis chacune de ses colonnes. Nous voulons trier la table **Person** depuis le nom de famille. Pour ce faire, procédez comme suit :

1. Sélectionnez une colonne **Dernier** en cliquant sur son en-tête.

Person (3)	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	true	BA	false	Fred	Smith	22	Smith@work.com
2	false	MA	true	Alfred	Aldrich	33	Aldrich@work
3	true	Ph.D	false	Colin	Coletti	444	Coletti@work.com

2. Choisissez l'option du menu **XML | Trier par ordre croissant** ou cliquez sur l'icône **Trier par ordre croissant** dans la barre d'outils du Mode Grille. La colonne et toute la table sont désormais triées par ordre alphabétique. La colonne reste en surbrillance.

Person (3)	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	false	MA	true	Alfred	Aldrich	33	Aldrich@work
2	true	Ph.D	false	Colin	Coletti	444	Coletti@work.com
3	true	BA	false	Fred	Smith	22	Smith@work.com

Puisque l'extension du téléphone **444** est correcte mais invalide, nous devons modifier le Schéma XML pour valider ce nombre. Nous en venons dans la prochaine section.

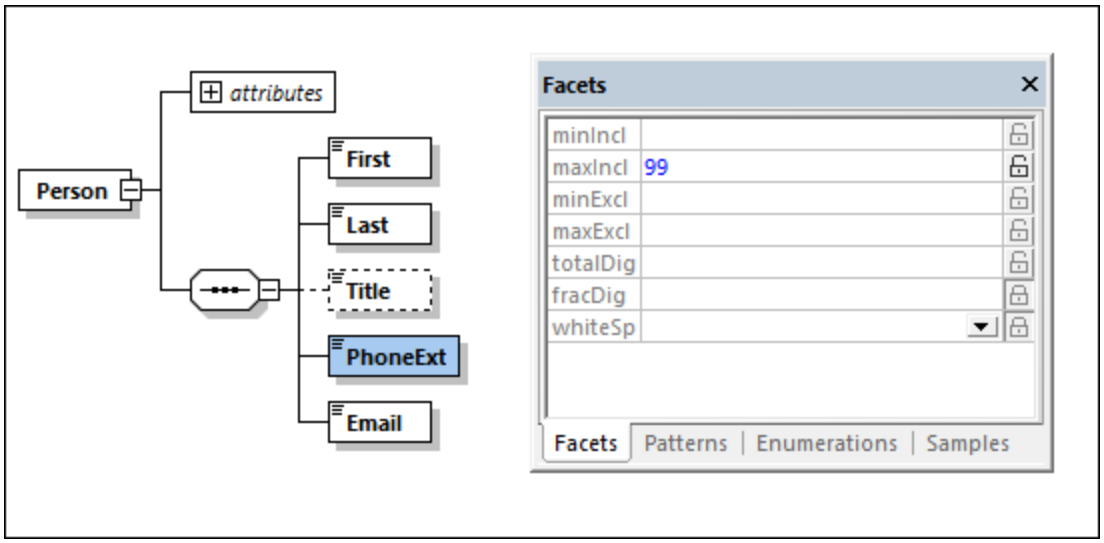
2.5.8 Modifier le Schéma

Puisque les numéros d'extension ne couvrent pas tous les numéros possibles définis dans le Schéma XML allant de (**0** à **99**), nous allons élargir la fourchette à **999**. Procédez comme suit :

1. Dans le Mode Grid, sélectionnez n'importe quelle cellule **PhoneExt** (*voir la capture d'écran ci-dessous*).

Person (3)	Manager	Degree	Programmer	First	Last	PhoneExt	Email
1	false	MA	true	Alfred	Aldrich	33	Aldrich@work
2	true	Ph.D	false	Colin	Coletti	444	Coletti@work.com
3	true	BA	false	Fred	Smith	22	Smith@work.com

2. Choisissez l'option du menu **DTD/Schéma| Aller à la définition** ou cliquez sur l'icône **Aller à la définition** dans la barre d'outils du Mode Grille. Le schéma associé, dans ce cas **AddressLast.xsd**, est ouvert, et la définition **PhoneExt** sera mise en surbrillance (*voir la capture d'écran ci-dessous*).



3. La facette de l'élément **`maxIncl`** est **`99`** (*voir la capture d'écran*). Éditez la valeur à **`999`**, puis enregistrez le schéma.
4. Retournez au document XML et validez-le. Il sera valide.
5. Enregistrez votre fichier en tant que **`CompanyLast.xml`**.

Note : Le dossier du tutoriel XMLSpy contient un fichier intitulé **`CompanyLast.xml`**, qui contient les mêmes données que le fichier que vous aurez enregistré lorsque vous avez terminé ce tutoriel.

© 2016-2022 Altova GmbH

2.6 Transformations XSLT

Objectif

Afin de générer un fichier HTML depuis le fichier XML utilisant une feuille de calcul XSL pour transformer le fichier XML. Veuillez noter qu'une "transformation" ne change pas le fichier XML en un autre format, mais un nouveau fichier de sortie est généré. Le mot "transformation" est une convention.

Méthode

La méthode utilisée pour effectuer la transformation est la suivante :

- Attribuez un fichier XSL prédéfini, `Company.xsl`, au document XML.
- Exécutez la transformation dans le cadre de l'interface XMLSpy en utilisant un des deux moteurs Altova XSLT intégrés. (*voir la note ci-dessous.*)

Commandes utilisées dans cette section

Les commandes XMLSpy suivantes sont utilisées dans cette section :



XSL/XQuery | Assigner XSL, qui assigne un fichier XSL au document XML actif.



XSL/XQuery | Aller au XSL, ouvre le fichier XSL référencé par le document XML actif.



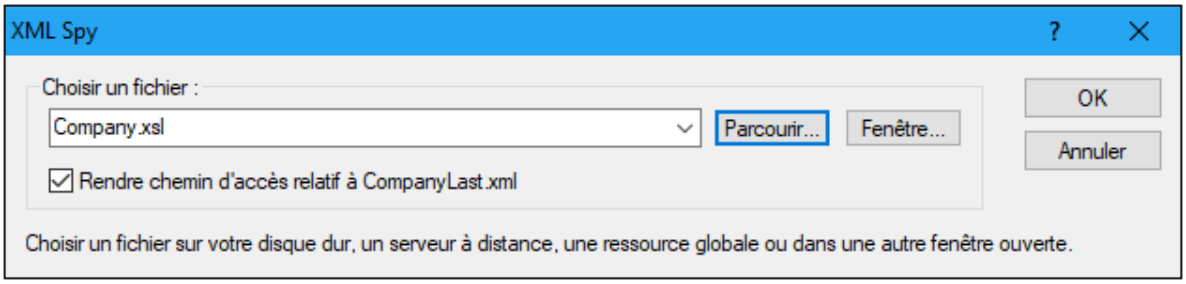
XSL/XQuery | Transformation XSL (F10), ou l'icône de la barre outils, transforme le document XML actif en utilisant la feuille de style XSL assignée au fichier XML. Si un fichier XSL n'a pas été assigné, vous serez invité à le faire lorsque vous sélectionnerez cette commande.

Note : XMLSpy a des moteurs XSLT intégré pour XSLT 1.0, 2.0 et 3.0. Le moteur correct est sélectionné automatiquement par XMLSpy sur la base de l'attribut de version dans l'élément `xsl:stylesheet` ou `xsl:transform`. Dans ce tutoriel, nous utilisons les feuilles de style XSLT 3.0. Le moteur XSLT 1.0 sera sélectionné automatiquement quand la commande **Transformation XML** est invoquée.

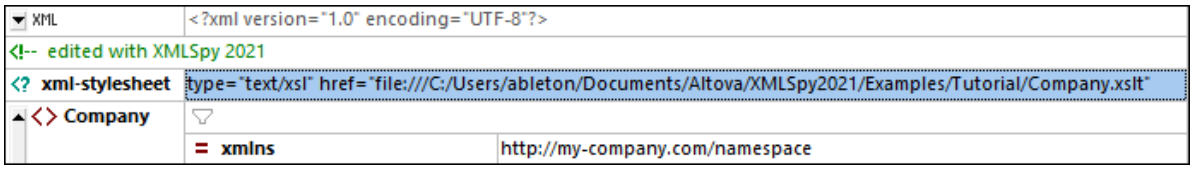
2.6.1 Attribuer un fichier XSLT

Afin d'attribuer un fichier XSLT au fichier `CompanyLast.xml` :

1. Cliquer sur l'onglet `CompanyLast.xml` dans la fenêtre principale afin que `CompanyLast.xml` devienne le document actif puis passer au Mode Texte.
2. Sélectionner la commande de menu **XSL/XQuery | Assigner XSL**.
3. Cliquer sur le bouton **Parcourir** et sélectionner le fichier `Company.xsl` depuis le dossier Tutoriel. Dans le dialogue, vous pouvez activer l'option Rendre le chemin relatif à `CompanyLast.xml` si vous souhaitez rendre le chemin relatif au document XML.



4. Cliquer sur **OK** pour attribuer le fichier XSL au document XML.
5. Passer au Mode Grille pour voir l'attribution (*capture d'écran ci-dessous*). Une instruction de traitement `XML-stylesheet` est insérée dans le document XML qui référence le fichier XSL. Si vous activez la case à cocher *Rendre le chemin relatif à CompanyLast.xml*, alors le chemin sera relatif ; dans le cas contraire, absolu (comme indiqué dans la capture d'écran ci-dessus).

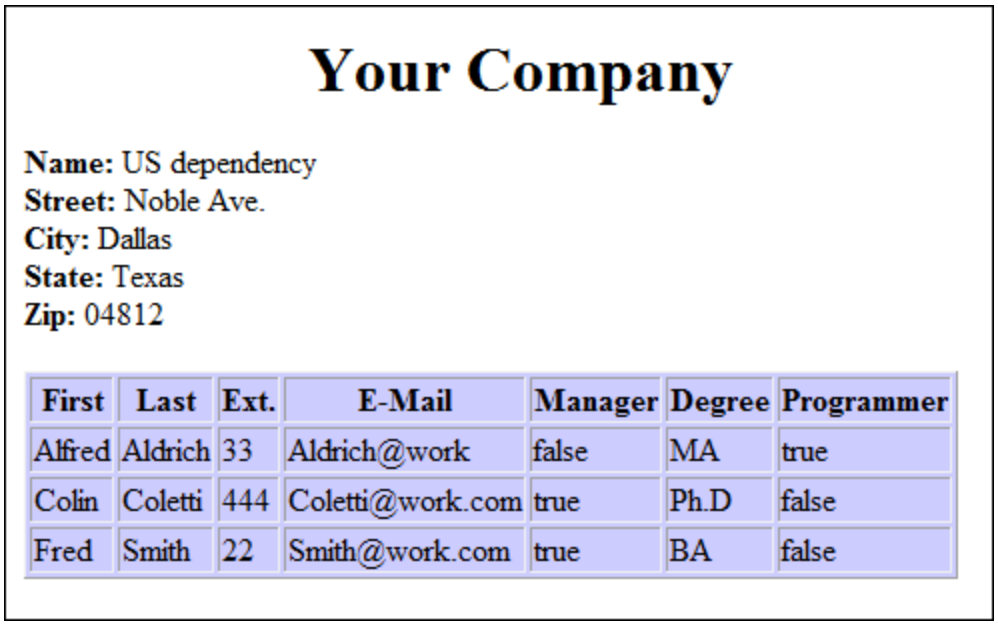


2.6.2 Transformer le fichier XML

Pour transformer le document XML avec le fichier XSL que vous y avez attribué :

1. Veuillez vous assurer que le fichier XML est le document actif.
2. Sélectionnez l'option de menu **XSL/XQuery | XSL Transformation (F10)** ou cliquez sur l'icône de la commande dans la barre d'outils. Cela lance automatiquement la transformation avec la feuille de style XSLT référencée dans le document XML. Le document de sortie est affiché dans le Mode Navigateur ; il porte le nom **`XSL Output.html`**. Le document HTML montre les données **`Company/Address`** en un bloc à gauche, et les données **`Company/Person`** sous forme tabulaire.

© 2016-2022 Altova GmbH



Your Company

Name: US dependency
Street: Noble Ave.
City: Dallas
State: Texas
Zip: 04812

First	Last	Ext.	E-Mail	Manager	Degree	Programmer
Alfred	Aldrich	33	Aldrich@work	false	MA	true
Colin	Coletti	444	Coletti@work.com	true	Ph.D	false
Fred	Smith	22	Smith@work.com	true	BA	false

Note : puisque le fichier `Company.xslt` est un document XSLT 3.0, le Moteur XSLT 3.0 de Altova est automatiquement sélectionné pour la transformation. Si le fichier de sortie HTML n'est pas généré, veillez à ce que, dans la section *XSL* du dialogue des Options (**Outils | Options**), l'extension de fichier par défaut du fichier de sortie a été `.html`. Ceci assure que le navigateur lit le document de sortie correctement en tant que fichier HTML.

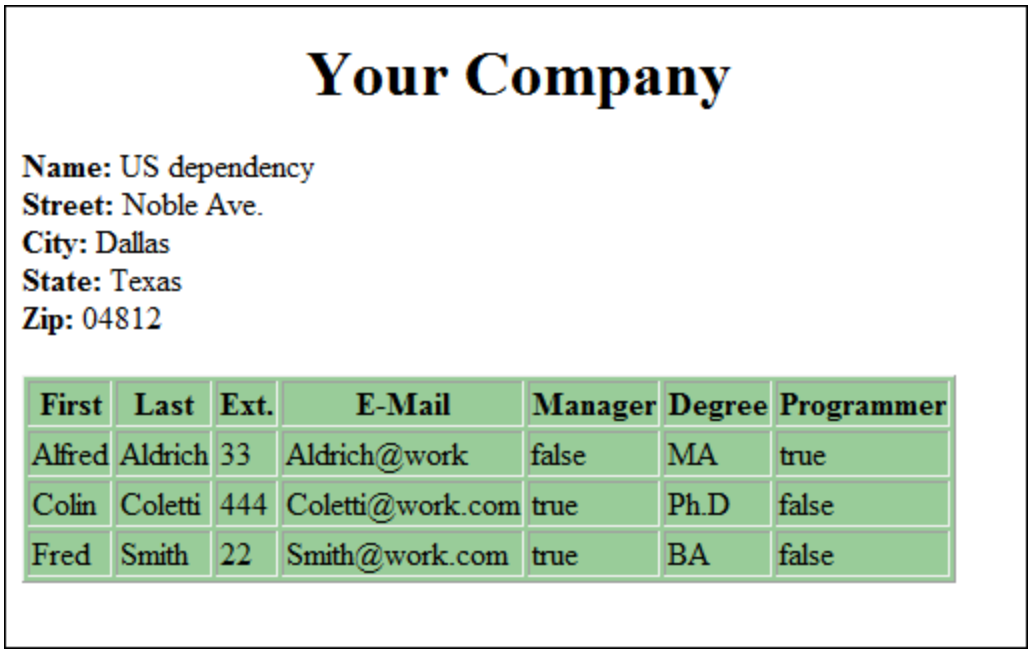
2.6.3 Modifier le fichier XSL

Vous pouvez changer la sortie en modifiant le document XSL. Changeons par exemple, la couleur d'arrière-plan de la table dans la sortie HTML de `#ccccff` à `#99cc99`. Pour ce faire, procédez comme suit :

1. Cliquez sur l'onglet `CompanyLast.xml` pour rendre ce document actif.
2. Sélectionner l'option de menu **XSL/XQuery | Aller à XSL**. La commande ouvre le fichier `Company.xslt` référencé dans le document XML.
3. Trouvez l'onglet de départ de l'élément `table`, ensuite l'attribut de l'élément `bgcolor` (*affiché en surbrillance dans la capture d'écran ci-dessous*). Modifiez la valeur de l'attribut de `#ccccff` à `#99cc99`.

```
<xsl:template match="/">
   <html>
      <head><title>Your company</title></head>
         <body>
            <h1><center>Your Company</center></h1>
            <xsl:apply-templates select="//my:Address"/>
               <table border="1" bgcolor="#ccccff">
                  <thead align="center">
                     <td><strong>First</strong></td>
                     <td><strong>Last</strong></td>
                     <td><strong>Ext.</strong></td>
                     <td><strong>E-Mail</strong></td>
                     <td><strong>Manager</strong></td>
                     <td><strong>Degree</strong></td>
                     <td><strong>Programmer</strong></td>
                  </thead>
                  <xsl:apply-templates select="//my:Person"/>
               </table>
         </body>
   </html>
</xsl:template>
```

4. Sélectionnez l'option de menu **Fichier | Enregistrer** et enregistrez les changements.
5. Cliquez sur l'onglet **`CompanyLast.xml`** pour activer le fichier XML.
6. Exßecutez la commande de menu **XSL/XQuery | XSL Transformation** ; alternativement, appuyez sur **F10**. Un nouveau fichier **`XSL Output.html`** apparait dans le Mode Navigateur, avec une table qui a une nouvelle couleur d'arrière-plan (*voir la capture d'écran ci-dessous*).



Your Company

Name: US dependency
Street: Noble Ave.
City: Dallas
State: Texas
Zip: 04812

First	Last	Ext.	E-Mail	Manager	Degree	Programmer
Alfred	Aldrich	33	Aldrich@work	false	MA	true
Colin	Coletti	444	Coletti@work.com	true	Ph.D	false
Fred	Smith	22	Smith@work.com	true	BA	false

7. Sélectionnez l'option de menu **Fichier | Enregistrer**, et enregistrez le document de sortie en tant que **`Company.html`**.

© 2016-2022 Altova GmbH

2.7 Gestion de projet

Cette section vous initie aux fonctions de gestion de projet de XMLSpy. Une fois avoir découvert quels sont les avantages d'une organisation de vos fichiers XML dans des projets, vous allez organiser les fichiers que vous venez de créer dans un projet simple

2.7.1 Les avantages des projets

Les avantages de l'organisation de vos fichiers XML dans des projets sont recensés ci-dessous.

- Les fichiers peuvent être regroupés dans des dossiers par le biais d'une extension commune. Par exemple, vous pouvez regrouper des fichiers XML et XSD dans des dossiers séparés. Vous pouvez créer toute hiérarchie que vous voulez.
- Chaque dossier a certaines propriétés que vous pouvez définir. Par exemple, un dossier de fichiers XML peut être assigné un schéma pour validation. Tous les fichiers dans ce dossier de projet peuvent ensuite être validés dans un lot par rapport au fichier de schéma du dossier. Si vous modifier l'affectation du schéma du dossier de projet, vous pouvez rapidement exécuter une nouvelle validation de lot. Vous pouvez définir plusieurs autres propriétés de dossier utiles, telles qu'un fichier XSLT pour les transformations de lots avec un seul XSLT.
- Le traitement par batch peut être appliqué à des dossiers spécifiques ou au projet dans son ensemble
- Un DTD ou un Schéma XML peut être attribué à des dossiers spécifiques, permettant la validation des fichiers dans ce dossier.
- Les fichiers XSLT peuvent être attribués à des dossiers spécifiques, permettant les transformations des fichiers XML dans ce dossier en utilisant le XSLT attribué.
- Les fichiers de destination des fichiers de transformation XSL peuvent être spécifiés pour le dossier dans son ensemble

Toutes les configurations de projet mentionnées ci-dessus peuvent être définies en utilisant l'option de menu **Projet | Propriétés**. Les commandes de projets sont également disponibles dans les menus contextuels du projet ou des dossiers de projets individuels. Dans la section suivante, vous allez créer un projet en utilisant le menu **Projet**.

De plus, les fonctions de projet avancées suivantes sont disponibles :

- Les fichiers XML peuvent être placés sous contrôle de source à l'aide de l'option de menu **Projet | Contrôle de source | Ajouter au contrôle de source**. (Voir la Section de contrôle de source pour plus d'informations.)
- Les dossiers externes de votre réseau ainsi que les dossiers web peuvent être ajoutés aux projets. Ceci permet à toutes les fonctions des dossiers de projets, telles la validation et les transformations, d'être appliquées à des dossiers qui sont sur votre réseau ou sur Internet.

2.7.2 Construire un projet

Une fois arrivé(e) à ce niveau du tutoriel, vous vous retrouverez avec un certain nombre de fichiers ouverts liés au tutoriel dans la Fenêtre principale. Vous pouvez regrouper ces fichiers dans un projet de tutoriel. Tout

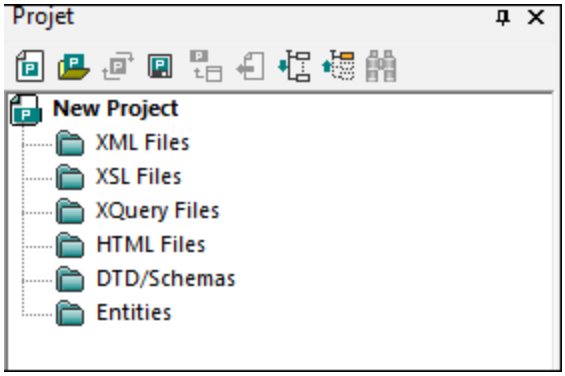
© 2016-2022 Altova GmbH

d'abord, vous créez un nouveau projet puis vous ajoutez les fichiers de tutoriel dans les sous-dossiers appropriés du projet.

Créer un nouveau projet

Créer un nouveau projet comme suit :

1. Sélectionnez l'option de menu **Projet | Nouveau Projet**. Un nouveau dossier de projet, nommé **`New Project`** est créé dans la Fenêtre Projet (*voir la capture d'écran ci-dessous*). Le nouveau projet contient des dossiers vides pour les catégories typiques de fichiers XML.



2. Cliquez sur l'onglet **`CompanyLast.xml`** pour rendre le fichier **`CompanyLast.xml`** actif dans la Fenêtre principale.
3. Sélectionnez l'option de menu **Projet | Ajouter les fichiers actifs et liés au projet**. Deux fichiers sont ajoutés au projet : **`CompanyLast.xml`** and **`AddressLast.xsd`**. Le fichier XML est ajouté au sous-dossier *XML* parce qu'il est le fichier actif. Le fichier de schéma est ajouté au dossier *DTD/Schémas* parce qu'une référence y afférente est contenue dans le fichier XML, le transformant en fichier associé. Veuillez noter que les fichiers référencés avec des instructions de traitement, comme les fichiers XSLT, ne sont pas considérés comme des fichiers liés.
4. Sélectionnez l'option de menu **Projet | Enregistrer le projet** et enregistrer le projet sous le nom **`Tutorial`**.

Note : Chaque dossier (mais pas le projet) a un nom de propriété *File extensions*. Il s'agit d'une liste d'extensions de fichiers séparées par un point-virgule (par exemple, **`xml;svg;wml`**). Cette liste détermine quels fichiers sont ajoutés à quels dossiers quand des fichiers sont ajoutés au projet. Par exemple, quand des fichiers associés et actifs sont ajoutés à un projet, comme nous venons de le faire ci-dessus, les propriétés *File extensions* des dossiers déterminent dans quels dossiers les fichiers ajoutés sont placés.

Propriétés de dossier et de projet

Les propriétés (telles que le schéma pour la validation et XSLT pour la transformation) peuvent être définies non seulement pour le projet entier, mais aussi pour les dossiers individuels. Puis, vous pouvez exécuter des actions, telles que la validation et la transformation, dans l'ensemble du projet ou les dossiers individuels. Pour exécuter une action, cliquez de la touche droite sur le projet ou le dossier, et choisissez l'action que vous voulez réaliser depuis le menu contextuel qui apparait.

Veuillez noter les points suivants :

- Une propriété qui est définie dans un dossier écrase la même propriété du projet.

- Si une propriété est définie dans un projet, elle est appliquée à tous les autres dossiers qui n'ont pas le même ensemble de propriété.
- Si une action est exécutée dans un projet, elle est appliquée à tous les types de fichiers applicables dans tous les dossiers du projet. Par exemple, si une validation est réalisée dans un projet, celle-ci est exécutée sur tous les fichiers XML dans tous les dossiers du projet.. Dans ce cas, le schéma qui a été défini pour le projet est utilisé pour toutes les validations, à l'exception des fichiers XML se trouvant dans des dossiers qui sont dotés de la propriété de validation de schéma définie pour un autre schéma.

Ajouter des fichiers au projet

Vous pouvez également ajouter d'autres fichiers au projet. Pour ce faire, procédez comme suit :

1. Cliquez sur n'importe quel fichier XML ouvert (avec l'extension de fichier `.xml`) autre que `CompanyLast.xml` pour faire de ce fichier XML le fichier actif. (Si aucun autre fichier XML n'est ouvert, en ouvrir un ou créer un nouveau fichier XML.)
2. Sélectionnez l'option de menu **Projet | Ajouter le fichier actif au projet**. Le fichier XML est ajouté au dossier *XML Files* sur la base de son type de fichier `.xml`.
3. De la même manière, ajoutez un fichier HTML et un fichier XSD (par exemple, les fichiers `Company.html` and `AddressFirst.xsd` au projet. Ces fichiers seront ajoutés au dossier *Fichiers HTML* et au dossier *DTD/Schémas*, respectivement.
4. Enregistrez le projet, soit en sélectionnant l'option de menu **Projet | Enregistrer le projet**, ou en sélectionnant tout fichier ou dossier dans la Fenêtre Projet et en cliquant l'icône **Enregistrer** dans la barre d'outils (ou **Fichier | Enregistrer**).

Note : en alternative, vous pouvez cliquer un dossier de projet et sélectionner**Ajouter fichier actif** pour ajouter le fichier actif à ce dossier spécifique.

Autres commandes utiles

Voici quelques-unes des commandes de projet utilisées les plus fréquemment :

- Pour ajouter un nouveau dossier à un projet, sélectionnez **Projet | Ajouter dossier Projet au projet**, et insérer le nom du dossier de projet.
- Pour supprimer un dossier à partir d'un projet, cliquez avec la touche de droite et sélectionnez **Supprimer** depuis le menu contextuel.
- Pour supprimer un fichier depuis un projet, sélectionnez le fichier et appuyez sur la touche **Supprimer**.

© 2016-2022 Altova GmbH

2.8 Et voilà !

Si vous êtes arrivé(e) à la fin du tutoriel, nous vous en félicitons !

Nous espérons que ce tutoriel a été utile en ce sens qu'il vous a introduit aux bases de XMLSpy et que vous serez désormais à même d'effectuer votre travail XML utilisant XMLSpy comme votre éditeur. Si vous avez besoin de plus d'information sur les fonctions spécifiques, veuillez utiliser les fonctions Index ou Recherche dans ce manuel. Veuillez noter que vous pouvez également imprimer la version PDF de ce tutoriel. Elle est disponible comme **`Tutorial.pdf`** dans votre dossier d'application de XMLSpy.

3 GUI et environnement

Cette section décrit :

- La GUI d'application et
- l'environnement d'application.

La section GUI démarre en présentant un aperçu de la GUI, puis décrit chacune des fenêtres de GUI en détail. Elle montre également comment redimensionner, déplacer et travailler avec les fenêtres et la GUI.

Les points de la section d'environnement de l'application détaillent différents paramètres qui contrôlent comment les fichiers sont affichés et comment ils peuvent être édités. Elle explique aussi comment et où vous pouvez personnaliser votre application. Dans cette section, vous apprendrez où les fichiers d'exemples et de tutoriels ont été installés sur votre appareil et, dans la section, vous pourrez suivre un lien vers le site Internet d'Altova, où vous pourrez explorer la matrice des fonctions de votre application, en savoir plus sur les différents formats de votre manuel, découvrir les options de prises en charge variées à votre disposition et découvrir les autres produits contenus dans la gamme Altova.

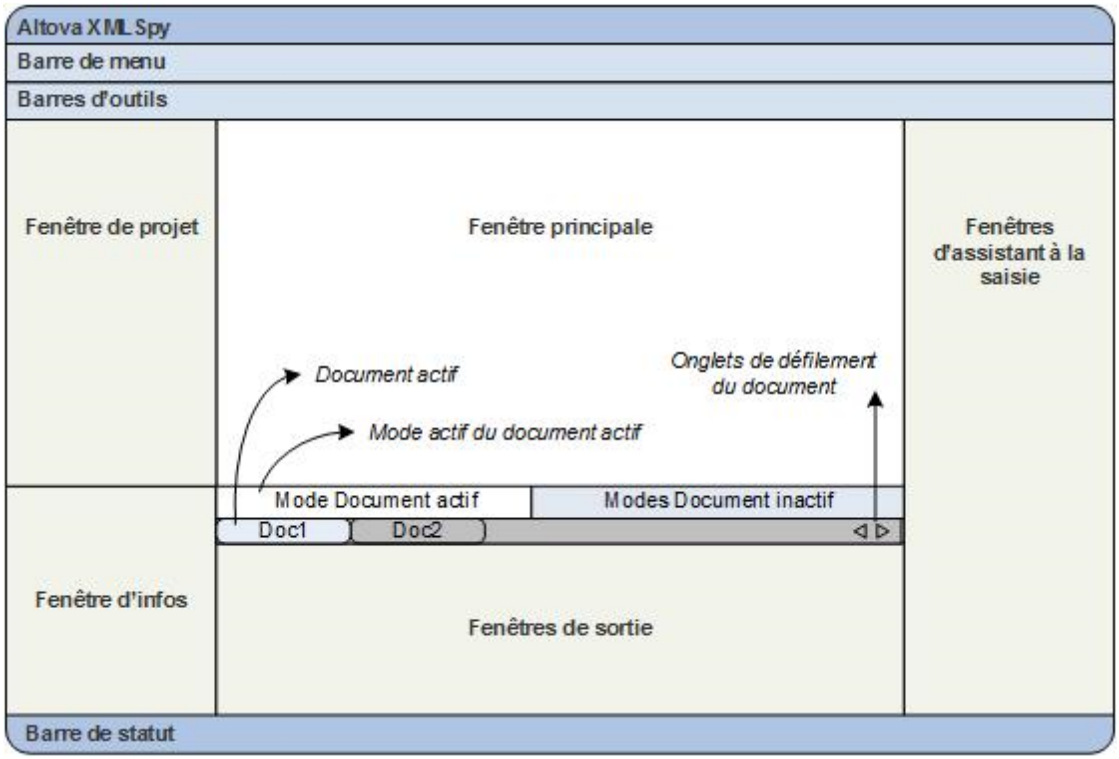
© 2016-2022 Altova GmbH

3.1 L'Interface graphique utilisateur (GUI)

L'interface graphique utilisateur (GUI) consiste en une fenêtre principale et plusieurs barres latérales (*voir l'illustration ci-dessous*). Par défaut, les barres latérales sont situées autour de la fenêtre principale et sont réparties dans les groupes suivants :

- Fenêtre de projet
- Fenêtre d'info
- Assistants à la saisie : Éléments, attributs, entités, etc. (dépendant du type de document actuellement actif)
- Fenêtre de sortie : Messages, XPath/XQuery, XSL Outline, Rechercher dans fichiers, Rechercher dans schémas, Rechercher dans XBRL, Graphiques

Le fenêtre principale et les barres latérales sont décrites dans les sous-sections de cette section.



La GUI contient aussi une barre de menu, une barre de statut et des barres d'outils, dont toutes sont décrites dans la rubrique barre de menu, barres d'outils, barre de statut.

Activer et désactiver l'affichage des barres latérales.

Les groupes latérales fenêtres (Fenêtre Projet, Fenêtre Info, Aides à la saisie, Fenêtre de sortie) peuvent être déplacés ou dissimulés en les allumant ou en les éteignant à l'aide des commandes dans le menu **Fenêtre**. Une barre latérale affichée (ou un groupe de barres latérales à onglets) peut être dissimulée en cliquant avec la touche de droite de la souris sur la barre de titre de la barre latérale affichée (ou du groupe de barres latérales à onglets) et en sélectionnant la commande **Dissimuler**.

Si vous fermez la fenêtre Sortie, vous pouvez y retourner en cliquant sur la commande de menu **Fenêtre | Fenêtre Sortie**.

Faire flotter et ancrer les barres latérales

Une fenêtre de barre latérale individuelle peut soit flotter librement de la GUI, soit être ancrée à l'intérieur de la GUI. Lorsqu'une fenêtre flottante est ancrée, elle vient se placer à sa dernière position d'ancrage. Une fenêtre peut également être ancrée en tant qu'onglet dans une autre fenêtre.

Pour ancrer ou détacher une fenêtre veuillez suivre l'une des méthodes suivantes :

- Cliquer avec le bouton de droite sur la barre de titre d'une fenêtre puis choisir la commande requise (**Flottant** ou **Ancrage**).
- Double-cliquer la barre de titre de la fenêtre. Si elle était ancrée, la fenêtre flotte à présent. Si elle flotte, elle s'ancrera sur la position à laquelle elle était fixée précédemment.
- Glisser la fenêtre (en utilisant sa barre de titre en guise de poignée) hors de sa position d'ancrage pour la faire flotter. Faire glisser une fenêtre flottante (par sa barre de titre) vers l'emplacement d'ancrage. Deux groupes de flèches bleues apparaissent. Le groupe extérieur de quatre flèches permet un ancrage par rapport à la fenêtre d'application (le long du rebord supérieur, droit, bas ou gauche de la GUI). Le groupe intérieur de flèches permet un ancrage par rapport à la fenêtre au-dessus de laquelle le curseur est placé actuellement. Si vous déposez une fenêtre glissée sur le bouton au centre du groupe intérieur de flèches (ou sur la barre de titre d'une fenêtre) permet d'ancrer la fenêtre glissée en tant que fenêtre à onglet dans le cadre de la fenêtre dans laquelle elle a été déposée.

Pour détacher une fenêtre à onglet, double-cliquez sur son onglet. Pour faire glisser une fenêtre à onglet hors d'un groupe de fenêtres à onglets, faites-la glisser par son onglet.

Dissimulation automatique de barres latérales

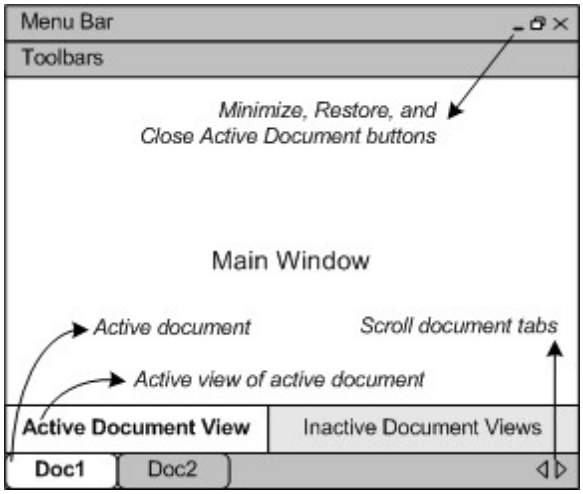
La fonction dissimulation automatique vous permet de minimiser les barres latérales ancrées à des boutons situés le long de la fenêtre d'application. Vous obtenez plus de place d'écran pour la fenêtre principale et les autres barres latérales. Faire défiler cette barre latérale en faisant défiler une barre latérale minimisée.

Pour dissimuler automatiquement et rétablir les barres latérales, cliquer sur l'icône de la punaise dans la barre de titre de la fenêtre de la barre latérale (ou cliquer avec la barre de titre et choisir **Dissimulation automatique**).

3.1.1 Fenêtre principale

La Fenêtre principale (*capture d'écran ci-dessous*) est l'endroit où vous consultez et éditez les documents.

© 2016-2022 Altova GmbH



Fichiers dans la fenêtre principale

- Vous pouvez ouvrir autant de fichiers que vous le souhaitez et les éditer tous à la fois.
- Chaque document ouvert a sa propre fenêtre et un onglet (contenant le nom de fichier du document) en bas de la fenêtre principale. Pour activer un document ouvert, cliquez sur son onglet.
- Si plusieurs fichiers sont ouverts, certains onglets de document peuvent ne pas être visibles ou manquer de place dans la barre des onglets du document. Vous pouvez faire apparaître les onglets en : (i) utilisant les boutons de défilement situés à droite de la barre d'onglet du document, ou (ii) en sélectionnant le document requis depuis la liste en bas du menu Fenêtre.
- Lorsque le document actif est maximisé, ses touches **Minimiser**, **Restaurer**, et **Fermer** sont situés du côté droit de la barre de menu. Lorsqu'un document est ouvert en cascade, en mosaïque ou réduit, les boutons Maximiser, Restaurer et Fermer sont situés dans la barre de titre de la fenêtre du document.
- Lorsque vous maximisez un fichier, tous les fichiers ouverts seront maximisés.
- Les fichiers ouverts peuvent être mis en cascade ou en mosaïque dans le menu Fenêtre.
- Vous pouvez aussi activer les fichiers ouverts dans la séquence dans laquelle ils ont été ouverts en utilisant **Ctrl+Tab** ou **Ctrl+F6**.
- Cliquer de la touche droite de la souris ouvre un menu contextuel avec une sélection de commandes de fichiers, telles que **Imprimer** et **Fermer**.

Modes dans la fenêtre principale

Le document actif peut être affiché et édité dans des modes variés. Les modes disponibles sont affichés dans une barre située au-dessus des onglets de document (*voir illustration ci-dessus*) et le mode actif est marqué. Un mode est sélectionné en cliquant sur le bouton de mode requis ou en utilisant les commandes dans le menu **Mode**.

Les modes disponibles sont soit des modes d'édition, soit de navigateur :

- Mode Texte : Un mode d'édition avec une coloration de la syntaxe pour travailler directement avec un code de document.
- Mode Grille : Pour une édition structurée. Le document est affiché en tant que grille structurée qui peut être manipulée graphiquement. Ce mode contient également un affichage de table, qui affiche des éléments répétitifs dans un format tabulaire.
- Mode de schéma : pour la consultation et l'édition de Schémas XML.

- Mode WSDL : pour la consultation et l'édition de documents WSDL.
- Mode Authentic: Pour éditer des documents XML qui sont basés sur StyleVision Power Stylesheet dans l'interface graphique.
- Mode Navigateur: Un mode de navigateur intégré qui prend en charge les deux feuilles de style CSS et XSL.

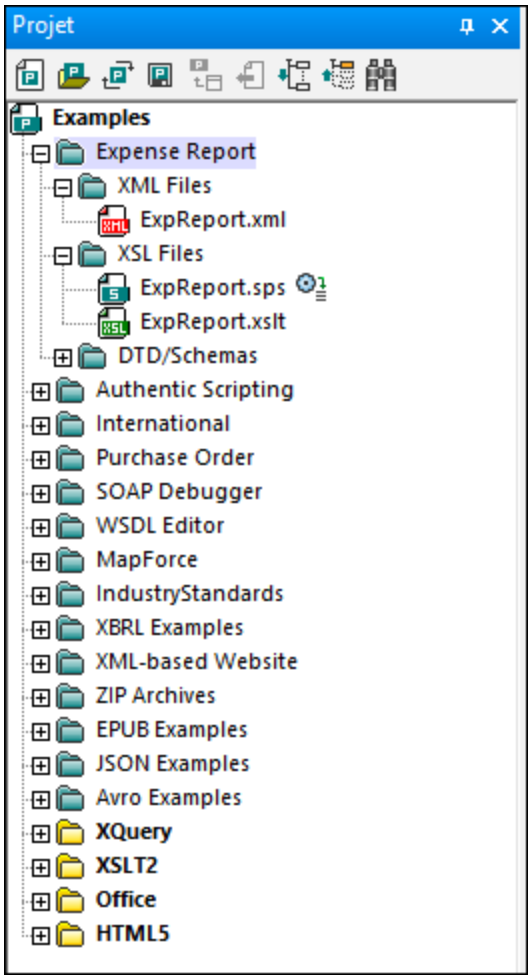
Note : Le mode par défaut pour les extensions de fichiers individuels peut être personnalisé dans le dialogue **Outils | Options** : dans le panneau Mode par défaut de l'onglet Types de fichiers.

3.1.2 Fenêtre Projet

Un projet est une collection de fichiers qui sont liés les uns aux autres d'une manière que vous déterminez. Par exemple, dans la capture d'écran ci-dessous, un projet nommé `Examples` rassemble les fichiers pour les exemples variés dans des dossiers d'échantillons séparés, qui peuvent eux-mêmes être organisés dans des sous-dossiers. Dans le cadre du projet `Examples`, par exemple, le dossier d'exemple `OrgChart` est encore organisé en sous-dossiers pour les fichiers XML, XSL et Schéma.

Note: La fenêtre Projet de XMLSpy contiendra initialement le projet Examples par défaut de l'application. Pour charger le projet par défaut Exemples, rendez-vous dans le dossier **`Examples`** de l'application dans le *dossier (Mes) Documents* , et double-cliquer sur le fichier **`Examples.spp`**.

© 2016-2022 Altova GmbH



Les projets vous permettent donc de rassembler des fichiers qui sont utilisés ensemble et d'y accéder plus rapidement. De plus, vous pouvez définir des schémas et des fichiers XSLT pour des dossiers individuels, permettant ainsi le traitement par lot de fichiers dans un dossier.

Opérations du projet

Les commandes permettant l'exploitation du projet sont disponibles dans le menu **Projet**, et certaines commandes sont disponibles dans les menus contextuels du projet et de ses dossiers (cliquer avec la touche de droite pour y accéder). Un sous-ensemble de commandes du menu **Projet** sont également disponibles dans la barre d'outils de la fenêtre de projet, car elles sont fréquemment utilisées (*capture d'écran ci-dessous*).



Les commandes de la barre d'outils sont, de gauche : *Nouveau projet, Ouvrir projet, Recharger projet, Enregistrer projet, Ajouter fichier actif au projet, Sélectionner fichier actif, Élargir tout, Réduire tout, Chercher*. Les noms de ces commandes sont explicites et expliquées dans le menu de projet.

Les opérations clefs liées à la fenêtre de projet sont recensées ci-dessous.

- Un projet à la fois est ouvert dans la Fenêtre Projet. Lorsqu'un nouveau projet est créé ou qu'un projet existant est ouvert, il remplace le projet actuellement ouvert dans la Fenêtre Projet.
- Une fois que des changements ont été effectués à un projet, celui-ci doit être enregistré (en cliquant sur la commande **Projet | Enregistrer projet**). Un projet avec des changements non enregistrés est indiqué avec un astérisque placé à côté de son nom (*voir capture d'écran ci-dessus*).
- Le projet présente une structure arborescente composée de dossier, de fichiers et d'autres ressources. Les autres ressources peuvent être ajoutées à tous les niveaux et à une profondeur illimitée.
- Les fichiers de projet sont des fichiers *sémantiques* qui représente un regroupement logique de fichiers. Ils **ne doivent pas** nécessairement correspondre à une organisation hiérarchique de fichiers sur votre disque dur.
- Les dossiers peuvent correspondre à et avoir une relation directe à des répertoires physiques dans votre système de fichier. Nous appelons de tels dossiers des *dossiers externes*, et ils sont indiqués dans la Fenêtre de Projet par une icône de dossier jaune (contrairement aux dossiers de projet normaux qui sont verts). Les dossiers de projet externes doivent être synchronisés explicitement en utilisant la commande **Réinitialiser**.
- Un dossier peut contenir un mélange arbitraire de types de fichier. En alternative, vous pouvez définir des extensions de type de fichier pour chaque dossier (dans le dialogue de Propriétés de ce dossier) pour garder les fichiers communs dans un endroit pratique. Lorsqu'un fichier est ajouté au dossier parent, il est automatiquement ajouté au sous-dossier qui a été défini pour contenir des fichiers de cette extension de fichier.
- Lorsque vous passez au-dessus d'un fichier d'image file qui a été placé dans le dossier de projet, une prévisualisation de l'image est affichée (formats `.png`, `.jpeg`, `.gif`, `.bmp`, `.tiff`, et `.ico`). Double-cliquez sur l'image pour l'ouvrir dans l'aperçu de l'image par défaut/du programme d'édition du système.
- Dans la Fenêtre de Projet, un dossier peut être glissé vers un autre dossier ou vers un autre emplacement dans le cadre du même dossier alors qu'un fichier peut être glissé vers un autre dossier mais ne peut pas être déplacé à l'intérieur de ce dossier (dans lequel les fichiers sont classés alphabétiquement). De plus, les fichiers et dossiers peuvent être glissés depuis Windows File Explorer vers la fenêtre de projet.
- Chaque dossier a un ensemble de propriétés qui sont définis dans le dialogue Propriétés de ce dossier. Ces propriétés comprennent des extensions de fichier pour le dossier, le schéma avec lequel valider les fichiers XML, le fichier XSLT avec lequel transformer des fichiers XML, etc.
- Le traitement par lot des fichiers dans un dossier est effectué en cliquant avec la touche de droite sur le dossier et en sélectionnant la commande pertinente depuis le menu contextuel (par exemple, **Valider XML** ou **Vérifier la bonne formation**).

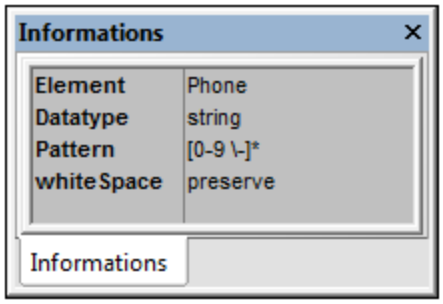
Pour une description plus détaillée des projets, voir la section Projets.

Note : L'affichage de la fenêtre Projet peut être allumé et éteint dans le menu **Fenêtre**.

3.1.3 Fenêtre Info

La Fenêtre Info (*capture d'écran ci-dessous*) montre des informations concernant l'élément ou l'attribut sur lequel le curseur se trouve actuellement. L'information est disponible dans la Fenêtre Info du Mode Texte, le Mode Grille, le Mode XBRL, et le Mode Authentic.

© 2016-2022 Altova GmbH



L'affichage de la Fenêtre Info peut être allumé ou éteint dans le menu **Fenêtre**.

Veuillez prendre note des points suivants :

- Lorsqu'un document XSLT est actif, des informations et commandes supplémentaires relatives à XSLT sont disponibles dans l'onglet XSLT de la fenêtre Info. Vous trouverez dans la section XSLT et XQ | XSLT | XSL Outline une description pour apprendre comment lire les informations et utiliser les commandes dans l'onglet XSLT.
- Lorsqu'un document JSON (instance ou schéma) est actif, des options de validation et d'édition intelligente sont disponibles dans l'onglet JSON de la fenêtre Info. Voir Valider des documents JSON pour plus d'informations.
- Lorsqu'un document de Schéma XML (`.xsd` file) est actif, les options pour activer la validation étendue sont disponibles dans l'onglet Schéma de la fenêtre Info. Vous trouverez la description du réglage de la validation étendue dans la section, Règles de Schéma.
- Lorsqu'un document XULE est le document actif, l'onglet XULE de la fenêtre Info fournit une option pour sélectionner une taxonomie XBRL à utiliser pour l'information sur la structure de l'instance XBRL. L'information de taxonomie est utilisée pour les fonctions de Remplissage automatique de l'éditeur XULE.

3.1.4 Assistants de saisie

Les assistants à la saisie consistent en une fonction intelligente d'édition qui vous aide à créer des documents XML valides rapidement. Lorsque vous éditez un document, les assistants à la saisie affichent des options d'édition structurelles conformément à l'emplacement actuel du curseur. Les assistants à la saisie obtiennent l'information requise depuis le DTD, Schéma XML sous-jacents et/ou StyleVision Power Stylesheet, etc. Si, par exemple, vous éditez un document de données XML, alors les éléments, attributs et entités pouvant être insérés dans l'emplacement actuel du curseur sont affichés dans les fenêtres des assistants de saisie pertinents.

Les assistants de saisie disponibles dépendent de :

1. *Le type de document qui doit être édité*. Par exemple, les documents XML auront des assistants de saisie différents que les documents XQuery : des assistants de saisie éléments, attributs et entités dans le premier cas, mais des assistants de saisie de mots-clés, variables et fonctions XQuery dans le second cas. Les assistants de saisie disponibles pour chaque type de document sont détaillés dans cette documentation, dans la description de ce type de document.
2. *Le mode actuel*. Puisque les mécanismes des différents modes ne sont pas identiques, les assistants de saisie sont conçus pour être compatibles avec le mécanisme d'édition du mode actuel. Par exemple : Dans le mode Texte, un élément ne peut être inséré au point d'emplacement du curseur pour que l'assistant à la saisie soit conçu pour insérer un élément lorsqu'un élément a été double-

cliqué. Mais dans le mode Grille, un élément peut être inséré avant le nœud sélectionné, ajouté après ou ajouté comme nœud enfant, pour que l'assistant de saisie Éléments dans le mode Grille aient trois onglets pour *Insérer*, *Ajouter* et *Ajouter comme enfant*, avec chaque onglet contenant les éléments disponibles pour cette opération particulière.

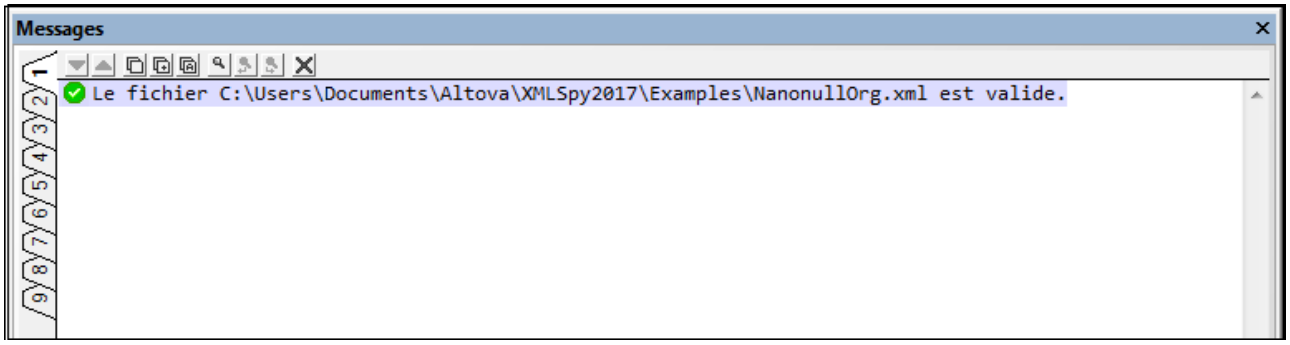
Une description générale des assistants de saisie dans chaque type de Mode est donné dans Modes d'édition. D'autres différences liées au type de document dans le cadre d'un mode sont notées dans la description des types de document individuel, par exemple les assistants de saisie XML et assistants de saisie XQuery.

Veuillez noter les points suivants :

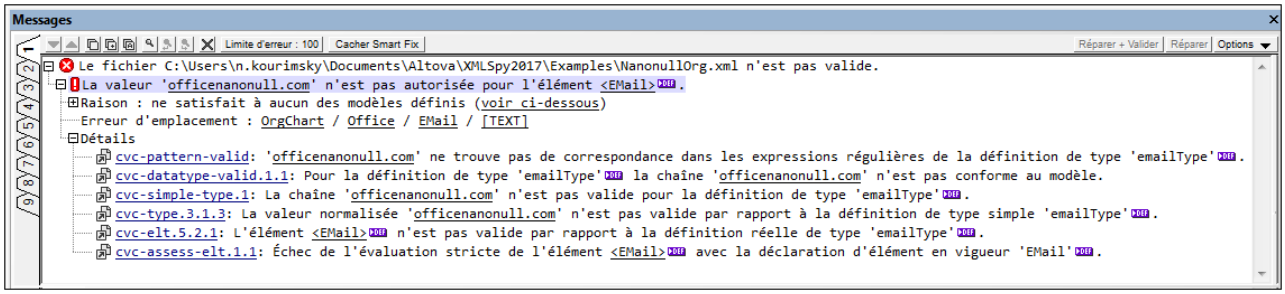
- Vous pouvez allumer ou éteindre l'affichage des assistants de saisie avec l'option de menu **Fenêtre | Assistants de saisie**.
- Dans Visual Studio .NET, les fenêtres de l'assistant de saisie ont un préfixe qui est le nom de l'application.

3.1.5 Fenêtre de sortie : Messages

La Fenêtre Messages affiche des messages concernant les actions exécutées dans <% XMLSPY%> ainsi que les erreurs et autres sorties. Par exemple, si un document XML, XML Schéma, DTD ou XQuery est validé et est valide, un message de réussite de la validation (*voir la capture d'écran ci-dessous*) sera affiché dans la fenêtre Messages.



Dans le cas contraire, un message décrivant l'erreur (*capture d'écran ci-dessous*) sera affiché. Veuillez noter dans la capture d'écran ci-dessous les liens (texte de lien noir) vers les nœuds et le contenu de nœud dans le document XML, ainsi que les liens (texte de lien bleu) menant aux sections dans la spécification pertinente pour la règle en questions sur Internet. Cliquer sur les boutons `Def` mauves pour ouvrir la définition de schéma pertinente dans le Mode Schéma.

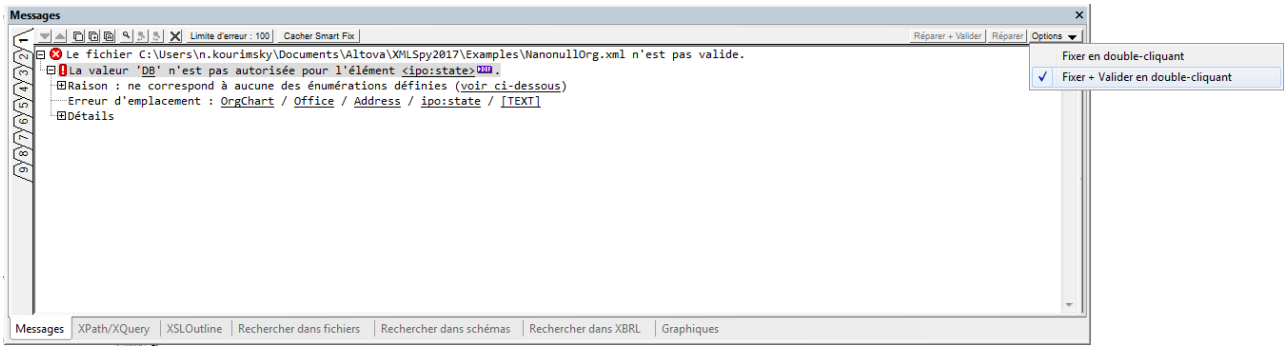


La Fenêtre Messages est activée dans tous les modes mais en cliquant un lien vers le contenu dans un document XML marque ce nœud dans le document XML dans le Mode Texte. Néanmoins, lorsqu'un Schéma XML a été validé dans le Mode Schéma, la vue ne change pas si vous cliquez sur un bouton `Def`.

Smart fixes de Validation XML

Sur la base de l'information dans le schéma, les options pour un smart fix sont aussi suggérées si la validation a été effectuée dans le **Mode Texte ou le Mode Grille**. Pour consulter une liste d'options smart fix, cliquer sur la touche **Afficher Smart Fix** (*voir capture d'écran ci-dessus*). Un panneau contenant les suggestions des options smart fix apparaît dans la fenêtre Messages (*capture d'écran ci-dessous*).

Veuillez noter que les erreurs dans la fenêtre Messages sont affichées une à la fois. De même, les erreurs de bonne formation (comme les balises de début et de fin non correspondantes), le cas échéant, sont affichées avant l'affichage des erreurs de validation. Donc le bouton **Afficher Smart Fix** sera activé uniquement si une erreur de validation est atteinte (une fois que les erreurs de bonne formation ont été corrigées).



Dans le volet Smart Fix, choisir un des smart fix suggérés et cliquez soit sur la touche **Réparer + Valider** ou sur la touche **Réparer** (*voir capture d'écran ci-dessus*). Le texte invalide dans le document XML sera remplacé par le smart fix choisi. En alternative, vous pouvez double-cliquer sur le smart fix que vous souhaitez. Cette action vous permet de réparer ou de réparer et de valider, selon l'option sélectionnée dans la liste déroulante *Options* (*voir* capture d'écran ci-dessus). La commande **Réparer** + Valider est utile parce que lorsqu'une autre validation est effectuée après la réparation, elle détectera d'autres erreurs de validation, le cas échéant.

Pour dissimuler le volet Smart Fix, cliquer sur la touche **Cacher Smart Fix** (*voir capture d'écran ci-dessus*). Pour plus de détails, voir la section Éditer les Modes | Mode Schéma | Validation et Smart Fixes.

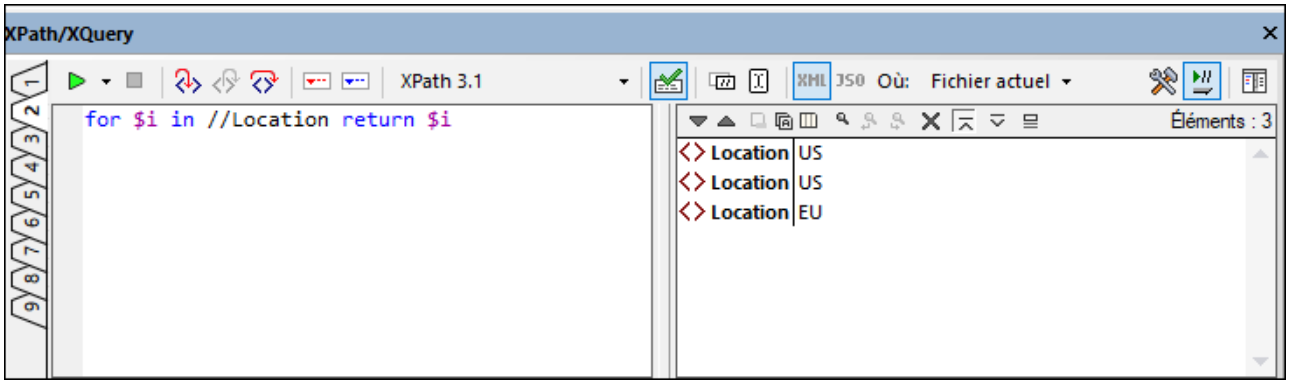
Valider les dossiers et les fichiers dans la fenêtre Projet

La commande **Valider** (dans le menu **XML**) est normalement appliquée au document actif. Mais vous pouvez aussi appliquer la commande à un fichier, dossier ou groupe de fichiers dans le projet actif. Choisir le fichier ou le dossier requis dans la fenêtre Projet (en cliquant dessus), et cliquer sur **XML | Valider XML** ou sur **F8**. Les fichiers invalides dans un projet seront ouverts et rendus actifs dans la Fenêtre principale et le message d'erreur *Le fichier n'est pas valide* sera affiché.

Note : Vous pouvez aussi exécuter le contrôle de bonne formation (**Contrôle de bonne formation** ou **F7**) dans la fenêtre Projet.

3.1.6 Fenêtre de sortie : XPath/XQuery

La fenêtre XPath/XQuery (*capture d'écran ci-dessous*) vous permet de créer, d'évaluer et de déboguer des expressions XPath et XQuery par rapport à des documents XML ou JSON. (Les fonctions qui activent les requêtes JSON ont été introduites dans XPath/XQuery 3.1. *Voir Transformations JSON avec XSLT/XQuery.*)



Cette section fournit un bref aperçu des fonctions principales de la fenêtre XPath/XQuery. Pour une description plus détaillée du travail avec la fenêtre XPath/XQuery, voir la section Expressions XPath/XQuery.

Fonctions principales

La fenêtre XPath/XQuery fournit les fonctions principales suivantes :

- *Mode d'évaluation :* Dans le Mode Évaluation, une expression XPath ou XQuery est évaluée par rapport à un ou plusieurs documents XML/JSON. L'expression est saisie dans le volet Expression et le résultat est affiché dans le volet Résultats annexe. Vous pouvez cliquer sur des nœuds dans le résultat pour vous rendre à ce nœud dans le document XML ou JSON. Voir la section Expressions XPath/XQuery pour les détails.
- *Mode débogage :* Dans le Mode Déboguer, vous pouvez déboguer une expression XPath/XQuery telle qu'elle s'applique au document XML actif actuellement. Vous pouvez définir des points d'arrêt et des points de trace, et passer étape par étape à travers l'évaluation. À chaque étape, vous pouvez voir le contenu des variables ainsi que les expressions Watch personnalisées définies pour contrôler les aspects supplémentaires de l'évaluation. Voir la section Déboguer l'expression pour les détails.
- *Générateur d'expression :* Un Générateur d'expression fournit des assistants de saisie et une information apparaît pour vous aider à construire des expressions correctes au niveau de la syntaxe. Voir la section Générateur d'expression pour plus de détails.
- *Prise en charge pour de multiples langages :* Vous pouvez basculer des versions de langage de XPath 1.0 à XPath 3.1 et XQuery 3.1. L'expression que vous saisissez sera évaluée conformément aux règles du langage sélectionné.
- *Ouvrir et enregistrer des expressions depuis/vers un fichier :* Vous pouvez enregistrer une expression XPath/XQuery ensemble avec les paramètres actuels de la fenêtre, vers un fichier XQuery, et vous pouvez charger des expressions depuis un fichier XQuery.
- *Auto-détection d'un type de fichier (XML/JSON) :* Le type de fichier du document actuel (XML ou JSON) est automatiquement détecté et le mode de document cible correct est défini automatiquement. Si l'étendue est un ensemble de documents, vous pouvez sélectionner manuellement le type de document cible.
- *Étendue flexible de documents cibles :* Dans le champ *Où*, vous pouvez sélectionner si l'expression est testée dans le fichier actif actuel, tous les fichiers ouverts, le projet actuel ou un dossier.

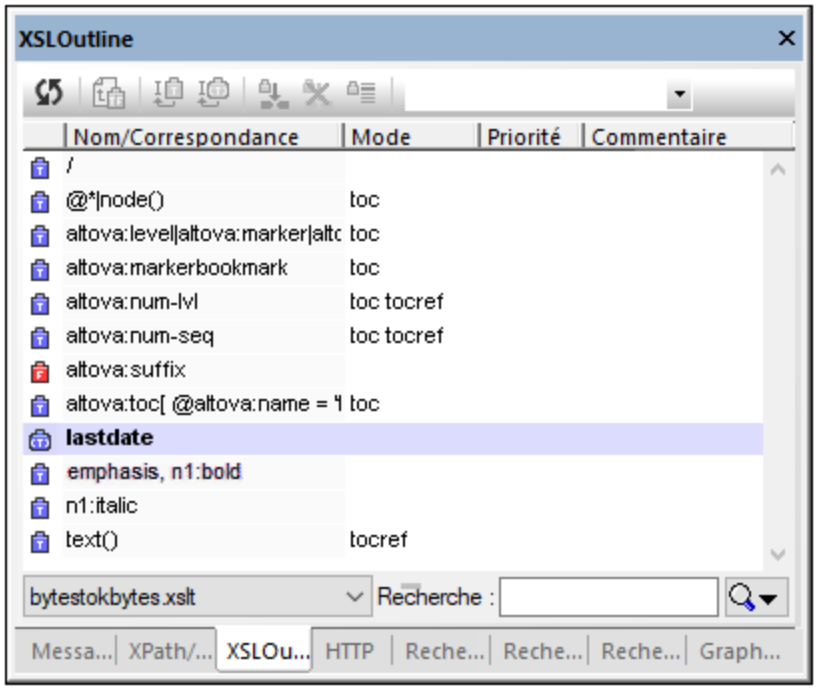
 © 2016-2022 Altova GmbH

- *Options pour la fenêtre de Résultat et fenêtre de Suivi :* Vous pouvez définir comment les items dans la fenêtre de Résultat et ´fenêtre de Suivi des Expressions sont à afficher
- *Les résultats mettent en lien directement avec les documents et les nœuds de document :* Les lignes dans le volet des Résultats contiennent des liens vers les documents pertinents ou les nœuds de document. Cela vous permet de vous rendre directement vers des nœuds spécifiques et d'y vérifier les données.
- *Analytiques de débogage :* Dans le mode de débogage, un large étendue d'information analytique est affichée. De plus, vous pouvez définir des expressions personnalisées pour vérifier les aspects supplémentaires de l'évaluation.

Pour une description détaillée des fonctions de la fenêtre XPath/XQuery, voir la section Expressions XPath/XQuery, dans laquelle tous les modes, icônes et la fonctionnalité disponible dans la barre d'outils sont décrits.

3.1.7 Fenêtre de sortie : XSL Outline

La fenêtre XSL Outline (*capture d'écran ci-dessous*) recense tous les modèles et fonctions dans une feuille de style XSLT et en option, aussi dans toutes les feuilles de style XSLT inclues et importées. La fenêtre XSL Outline est ancrée par défaut avec les Fenêtres de Sortie en bas de la fenêtre XMLSpy. Elle peut être détachée ou ancrée sur un autre bord de la fenêtre XMLSpy.



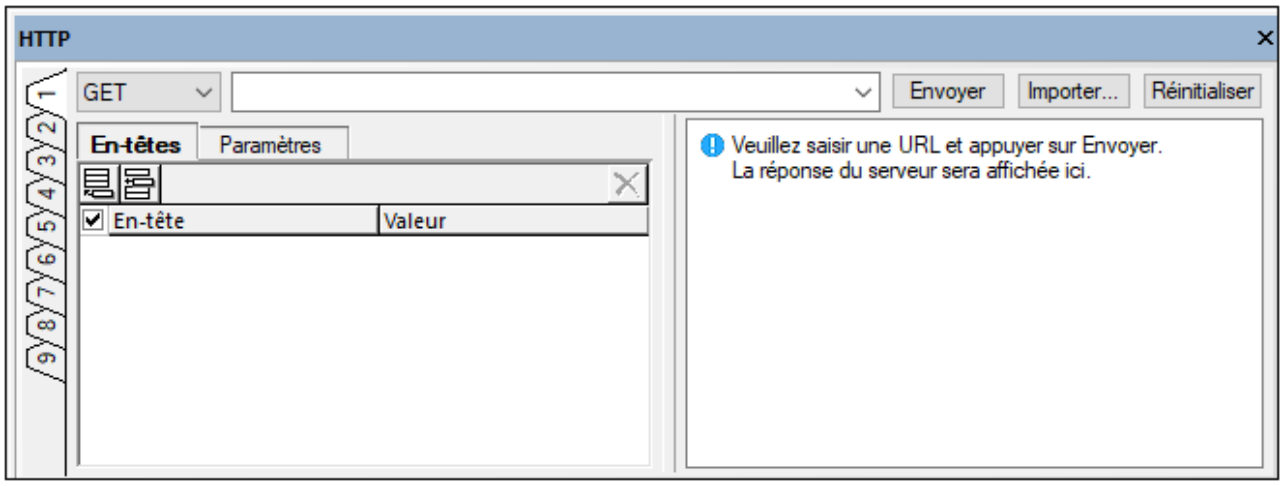
La fenêtre XSL Outline propose des informations concernant les modèles et les fonctions dans la feuille de style. Cette information peut être triée et recherchée, et la barre d'outils de la fenêtre contient des commandes qui vous permettent d'insérer facilement des appels aux modèles nommés et définir des modèles nommés comme point de départ des transformations. La manipulation de la fenêtre XSL Outline est décrite dans la section XSLT et XQuery | XSLT | Sortie XSL | Fenêtre de sortie XSL.

Note : Les informations liées au fichier concernant la feuille de style et les commandes liées au fichier sont disponibles dans l'onglet XSLT de la Fenêtre Info. L'utilisation de ces commandes est décrite dans la section XSLT et XQuery | XSLT | Sortie XSL | Fenêtre Info.

3.1.8 Fenêtre de sortie : HTTP

Dans la fenêtre de sortie HTTP (*capture d'écran ci-dessous*), vous pouvez tester des commandes HTTP : vous pouvez créer et envoyer une requête HTTP au serveur web, et de recevoir et contrôler la réponse.

La fenêtre de sortie HTTP a neuf onglets (*voir la capture d'écran ci-dessous*). Vous pouvez stocker une requête séparée dans chaque onglet et passer d'un onglet à l'autre. Après avoir créé une requête dans la fenêtre, vous pouvez envoyer la requête en cliquant sur le bouton **Envoyer**. La réponse est affichée directement dans la fenêtre.



La fenêtre consiste en les parties suivantes :

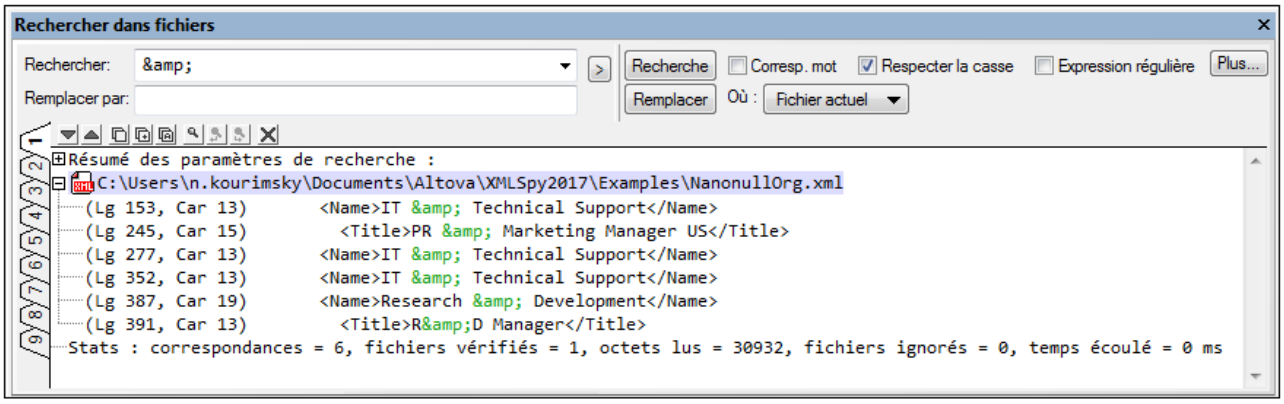
- En haut : (i) une zone de liste dans laquelle vous sélectionnez la méthode HTTP que vous souhaitez utiliser ; (ii) un champ de saisie pour l'URL sur le serveur web ; (iii) des touches liées à l'exécution de requêtes HTTP (**Envoyer**, **Importer** et **Réinitialiser**).
- Un volet de gauche pour créer la requête.
- Un volet de droite pour afficher la réponse.

Pour plus de détails concernant l'utilisation de la fenêtre de sortie HTTP, voir la section HTTP.

3.1.9 Fenêtre de sortie : Trouver dans fichiers

La fenêtre Trouver dans fichier (*voir la capture d'écran ci-dessous*) vous permet de réaliser les opérations trouver-et-remplacer rapidement dans plusieurs documents à la fois, et fournit des mécanismes qui vous aident à naviguer rapidement parmi les instances trouvées. Les résultats de chaque action de recherche et de remplacement sont présentés dans un des onglets numérotés de 1 à 9. En cliquant sur un item trouvé dans les résultats vous pouvez vous rendre sur cet item dans le Mode Textes de ce document.

© 2016-2022 Altova GmbH



Critères de recherche

Il existe deux grands critères de recherche : (i) ce que vous souhaitez chercher, et (ii) l'endroit où chercher.

Ce que vous souhaitez trouver : La ligne recherchée est saisie dans le champ Rechercher. Si cette ligne doit correspondre à un mot entier, alors la case Correspondance du mot entier doit être cliquée. Par exemple, pour le string chercher `fit` avec la Correspondance du mot entier, seul le mot `fit` correspondra au terme recherché ; mais `fit` dans `fitness`, par exemple, ne sera pas retenu. Vous pouvez spécifier que la casse soit respectée en utilisant la case Respecter la casse. Si le texte saisi dans la case « Find What » est une expression régulière, alors la case à cocher Expression régulière doit être cochée. Un assistant à la saisie pour les caractères d'expressions régulières est disponible en cliquant sur le bouton . L'utilisation des expressions régulières pour la recherche est expliquée dans la section concernant la commande Recherche. La touche **Plus** ouvre le dialogue Trouver dans les fichiers, où vous pouvez définir des conditions de recherche avancée et des actions. Pour plus d'informations, voir Éditer | Trouver dans les fichiers.

Où chercher : La recherche peut être menée dans : (i) tous les fichiers qui sont ouverts dans la GUI ; (ii) les fichiers du projet actuel ; et (iii) les fichiers d'un dossier sélectionné. Vous pouvez configurer des conditions dans le Dialogue Rechercher dans les fichiers (accessible en cliquant sur **Plus**).

Remplacer par

Le string avec lequel le string trouvé doit être remplacé est saisi dans le champ de saisie Remplacer par. Veuillez noter que si le champ de saisie Remplacer par est vide et que vous cliquez sur le bouton **Remplacer**, le texte trouvé sera remplacé par un string vide.

Les résultats

Une fois avoir cliqué sur **Rechercher** ou **Remplacer**, les résultats de la recherche ou du remplacement seront affichés dans la fenêtre de sortie Rechercher dans fichiers. Les résultats sont répartis en quatre parties:

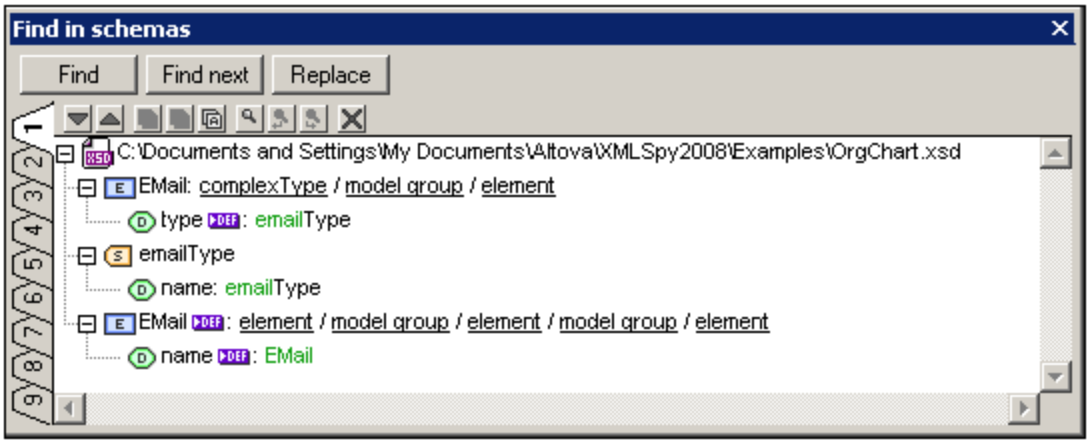
- Un sommaire des paramètres de recherche qui recense le string de recherche et le type de fichier recherché.
- Une liste des strings trouvés ou remplacés (selon que le bouton **Chercher** ou **Remplacer** ait été appuyé). Les items dans cette liste sont des liens vers le texte trouvé/remplacé dans le Mode Texte du document. Si le document n'est pas ouvert, il sera ouvert dans le Mode Texte et le texte trouvé/remplacé sera marqué.

- Une liste des fichiers qui ont été cherchés mais pour lesquels aucune correspondance n'a été trouvée.
- Un sommaire des statistiques pour l'action de recherche, y compris le nombre des correspondances et le nombre des fichiers contrôlés.

Note : Veuillez noter que la fonction Rechercher dans fichiers exécute les commandes **Trouver** et **Remplacer** sur plusieurs fichiers simultanément et affiche les résultats dans la fenêtre des résultats Rechercher dans les fichiers. Pour effectuer une recherche de manière à vous rendre d'un item trouvé à un autre, utiliser la commande Recherche.

3.1.10 Fenêtre de sortie : Trouver dans schémas

Lorsqu'un Schéma XML est actif dans le Mode Schéma, il peut être parcouru de manière intelligente avec la fonction Rechercher et Remplacer dans le Mode Schéma de XMLSpy. Cette fonction peut être accédée par le biais : (i) des commandes **Rechercher** et **Remplacer** dans le menu **Éditer** ; et (ii) les boutons **Rechercher** et **Remplacer** dans la fenêtre Rechercher dans schémas (*capture d'écran ci-dessous*).



Les résultats de la fonction Rechercher et Remplacer dans le Mode Schéma (donc, à chaque fois qu'une commande Rechercher ou remplacer est exécutée) sont affichés dans la fenêtre Rechercher dans schémas. Le terme recherché est affiché en vert ; (dans la capture d'écran ci-dessus, on peut voir que le terme recherché était `e-mail`, sans restriction de casse spécifiée). Veuillez noter que l'emplacement du fichier de schéma est aussi donné.

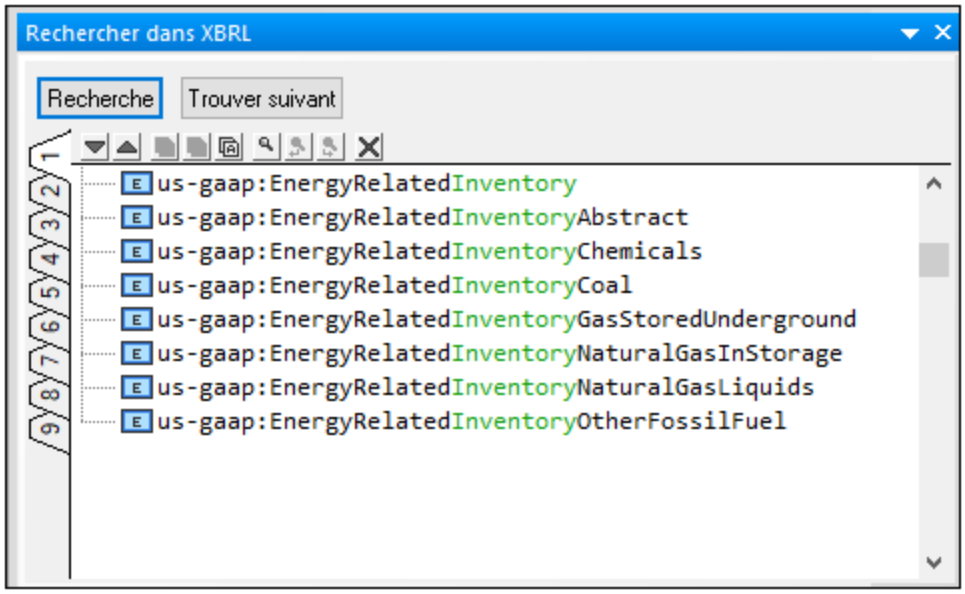
Les résultats sont affichés dans neuf onglets séparés (numérotés de 1 à 9). Ainsi vous pouvez garder les résultats d'une recherche dans un onglet, effectuer une nouvelle recherche dans un nouvel onglet et comparer les résultats. Pour afficher les résultats d'une nouvelle recherche dans un nouvel onglet, choisir un nouvel onglet avant de lancer une recherche. Cliquer sur un résultat dans la fenêtre Rechercher dans schéma pour faire apparaître et marquer le composant pertinent dans la fenêtre principale du Mode Schéma. De cette manière, vous pouvez chercher et parcourir rapidement le document jusqu'au composant désiré et copier des messages sur le presse-papiers. Pour plus de détails, voir la section Résultats et Information dans la description de la fonction Rechercher dans schémas.

© 2016-2022 Altova GmbH

3.1.11 Fenêtre de sortie : Trouver dans XBRL

Trouver dans fenêtre XBRL (*voir la capture d'écran ci-dessous*) affiche les résultats de la recherche dans un document de taxonomie XBRL. La fenêtre présente neuf onglets, les résultats d'un onglet peuvent donc être comparés aux résultats d'un autre onglet.

La Recherche dans XBRL peut être exécutée lorsqu'un document de taxonomie XBRL est ouvert dans le Mode XBRL. Comment réaliser la recherche est décrit dans la section Trouver dans XBRL dans la section XBRL dans le manuel utilisateur.



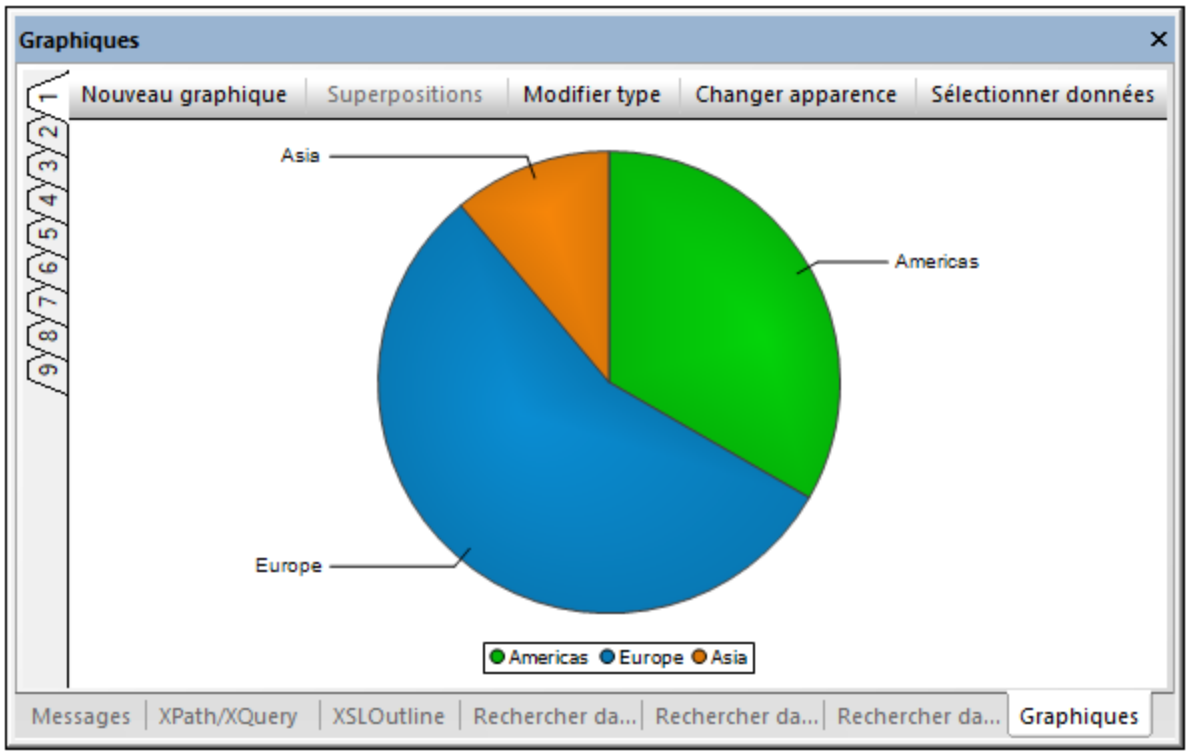
Le bouton **Recherche** s'affiche dans le dialogue de recherche. Le bouton **Trouver suivant** trouvera l'instance suivante du terme recherché dans le document en commençant à partir de la cellule se trouvant juste après celle dans le Mode XBRL dans lequel le curseur est placé actuellement.

Les commandes de la barre d'outils Rechercher dans XBRL suivantes sont disponibles :

- Les icônes ***Suivant*** et ***Précédent*** sélectionnent, respectivement, les résultats de recherche suivant et précédent par rapport au résultat actuellement sélectionné.
- Les commandes ***Copier Messages*** copient, respectivement, le message sélectionné, le message sélectionné et ses messages enfants et tous les messages sur le presse-papiers.
- Les commandes ***Rechercher*** trouvent du texte dans le volet XBRL.
- Les commandes ***Effacer*** effacent tous les messages dans l'onglet actuellement actif.

3.1.12 Fenêtre de sortie : Graphiques

Lorsqu'un document XML est ouvert dans le Mode Texte ou dans le Mode Grille, un graphique (camembert, graphique à barres, etc) représentant des données sélectionnées dans le document XML peut être généré dans la fenêtre Graphiques (*capture d'écran ci-dessous*).



Créer le graphique

Vous trouverez ci-dessous les instructions générales pour la création d'un graphique :

1. Placer le curseur dans le document XML pour sélectionner un nœud contextuel.
2. Cliquez sur la touche **Nouveau Graphique** dans la fenêtre Graphiques (*voir la capture d'écran ci-dessus*) ou cliquez avec la touche de droite dans la fenêtre principale et sélectionnez **Nouveau Graphique** depuis le menu contextuel.
3. Dans le dialogue Choisir colonnes qui apparaît, sélectionnez les données pour la table de données du graphique et cliquez sur **OK**. Le graphique sera créé dans la fenêtre Graphiques (*capture d'écran ci-dessus*).

Pour plus d'informations, voir la section Graphiques dans la section XML du manuel de l'utilisateur.

Modifier et gérer les graphiques

Un graphique peut être créé dans un des neuf onglets de la fenêtre Graphiques (numérotés du côté gauche de la fenêtre). Cela permet de comparer les graphiques parmi les différents onglets. Un graphique créé dans un onglet peut uniquement être écrasé lorsqu'un nouveau graphique est créé dans cet onglet. Il n'est pas possible de supprimer un graphique d'une autre manière. Même lorsque le document XML utilisé pour générer un graphique est fermé, le graphique reste dans l'onglet dans lequel il a été créé.

Les boutons situés en haut de la fenêtre sont les suivants :

- *Nouveau graphique* : Affiche le dialogue Choisir colonnes, dans lequel la table des données de graphique est configurée.

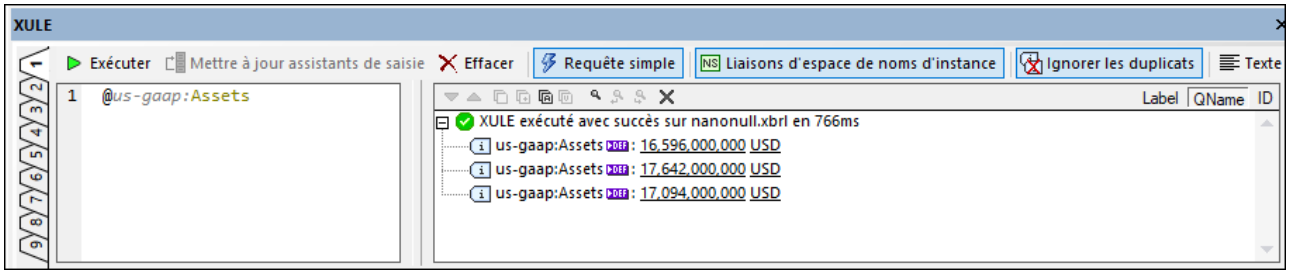
© 2016-2022 Altova GmbH

- *Superpositions :* Vous permet d'ajouter et de supprimer des couches sur le graphique principal. Après avoir créé une nouvelle couche, vous pouvez désormais ajouter un nouveau graphique à cette couche en cliquant sur l'onglet *Nouvelle couche* et spécifier les données à utiliser dans le graphique superposé.
- *Modifier type* : Permet de modifier le type de graphique, par exemple d'un graphique à barres à un camembert.
- *Changer apparence* : Permet la modification de paramètres comme la taille de la police et des schémas de couleur.
- *Sélectionner données* : Affiche le dialogue Sélectionner données qui contient la table des données du graphique et la sélection finale des données qui seront présentées dans le graphique. Les données pour les séries, l'axe X et l'axe Y peuvent être modifiées dans ce dialogue. Les données de l'axe X et de l'axe Y peuvent être sélectionnées graphiquement depuis la table des données du graphique. Cliquer sur **OK** génère le graphique modifié dans la fenêtre Graphiques.
- *Exporter*: le graphique peut être exporté en tant que fichier d'image ou en tant que fragment XSLT ou XQuery sur le presse-papiers. Le fragment XSLT ou XQuery peut être utilisé dans un document XSLT ou XQuery, qui, lorsque traité avec le moteur XSLT 2.0 ou le moteur Altova XQuery, rendra correctement le graphique.
- *Recharger/Auto* : Si le bouton **Auto** est activé, toute modification du document XML sous-jacent actualisera automatiquement un graphique dans la fenêtre Graphiques sur la base du document XML. Dans le cas contraire, un graphique ne sera mis à jour que sur pression du bouton **Recharger.**

Pour plus d'informations, voir la section Graphiques dans la section XML du manuel de l'utilisateur.

3.1.13 Fenêtre de sortie : XULE

La fenêtre XULE vous permet de requêter interactivement le document d'instance XBRL actif et de consulter les résultats de votre requête. La fenêtre XULE dispose de neuf onglets qui sont divisés en deux volets : (i) un volet d'expression XULE, dans lequel vous pouvez saisir l'expression XULE (ou la règle XULE) que vous souhaitez exécuter ; et (ii) un volet Résultats, qui affiche le résultat de l'exécution.



Pour exécuter de manière interactive une expression XULE dans le document d'instance XBRL actif, suivez les étapes suivantes :

1. Faire du document d'instance XBRL que vous souhaitez requêter le document actif dans la fenêtre principale.
2. Saisir l'expression XULE dans le volet d'expression XULE *(volet de gauche)*.
3. Cliquez sur **Exécuter** dans la barre d'outils de la fenêtre pour exécuter l'expression
4. Les résultats de l'exécution sont affichés dans le volet Résultats *(volet de droite)*.

Pour des informations détaillées, voir la description de la fenêtre XULE dans la section XBRL.

3.1.14 Menu Bar, Toolbars, Status Bar (2)

Barre de menu

La barre de menu (*voir illustration*) contient les menus d'application divers. Les conventions suivantes s'appliquent :

- Si les commandes dans un menu ne **s'**appliquent pas dans d'un mode ou dans un emplacement particulier dans le document, elles ne sont pas disponibles.
- Certaines commandes de menu font apparaître un sous-menu avec une liste d'options disponibles. Les sous-commandes de menus sont indiquées par une flèche orientée vers la droite.
- Certaines commandes de menu font apparaître un dialogue qui vous invite à donner des informations supplémentaires requises pour exécuter la commande sélectionnée. De telles commandes sont indiquées par une ellipse (...) après le nom de la commande.
- Pour accéder une commande de menu, cliquez sur le nom du menu et ensuite sur la commande. Si un sous-menu est valable pour un élément de menu, le sous-menu s'ouvrira lorsque vous ferez passer la souris au-dessus du nom de l'élément de menu. Cliquez sur l'item du sous-menu requis.
- Un menu peut être ouvert depuis le clavier en appuyant sur la combinaison de clé appropriée. La combinaison de clé pour chaque menu est **Alt+*KEY***, où ***KEY*** est la lettre soulignée dans le nom de menu. **Par exemple, la combinaison de touches pour le menu** F**ichier** est **Alt+F**.
- Une commande de menu (c'est-à-dire une commande dans un menu) peut être sélectionnée en choisissant consécutivement (i) le menu avec sa combinaison de touches (voir point précédent), et puis (ii) la combinaison de touches pour la commande spécifique (**Alt+*TOUCHE, où*** TOUCHE est la lettre soulignée dans le nom du menu). Par exemple, pour créer un nouveau fichier (**File | New**), appuyez sur **Alt+F**, puis **Alt+N**.
- Certaines commandes de menu peuvent être sélectionnées **directement** en appuyant sur un **raccourci clavier** spécial ou une combinaison de clé (**Ctrl+*KEY***). Les commandes qui ont des raccourcis y associés sont indiquées par un raccourci clavier ou une combinaison de clé recensée à droite de la commande. Par exemple, vous pouvez utiliser la combinaison de touches de raccourci **Ctrl+N** pour créer un nouveau fichier ; ou bien la touche de raccourci **F8** pour valider un fichier XML. Vous pouvez créer vos propres raccourcis dans l'onglet Clavier du dialogue Personnaliser (**Outils | Personnaliser**).

Barres d'outils

Les barres d'outils (*voir illustration*) contiennent des icônes qui sont des raccourcis à la sélection des commandes de menu. Le nom de la commande apparaît en plaçant la souris sur l'icône. Pour exécuter la commande, cliquer sur l'icône.

Les boutons de barre d'outils sont organisés en groupes. Dans le dialogue **Outils | Personnaliser | Barres d'outils**, vous pouvez spécifier quels groupes de barre d'outils vous souhaitez afficher. Ces paramètres s'appliquent au mode actuel. Pour configurer un paramètre pour un autre mode, veuillez passer à ce mode-là, puis définir les paramètres dans **Outils | Personnaliser | Barres d'outils**. Dans la GUI, vous pouvez glisser les groupes de barre d'outils par leurs poignées (ou barres de titre) vers des emplacements alternatifs à l'écran. Double-cliquer sur la poignée fait que la barre d'outils se détache et flotte ; double-cliquer sur sa barre de titre fait que la barre d'outils s'ancre à son emplacement précédent.

Barre de statut

La barre de statut est située au bas de la fenêtre de l'application (*voir illustration*) et affichera (i) les informations de statut concernant le chargement des fichiers, et (ii) des informations concernant les commandes de menu et les raccourcis de commande dans les barres d'outils si vous placez la souris dessus. Si vous utilisez la

© 2016-2022 Altova GmbH

version 64-bit de XMLSpy, cela est indiqué dans la barre de statut avec le suffixe (x64) placé après le nom de l'application. Il n'y a pas de suffixe pour la version 32-bit.

3.2 L'environnement d'application

Dans cette section, nous décrivons les différents aspects de l'application qui sont importants pour se lancer. La lecture de cette section vous permettra de vous familiariser avec XMLSpy et de maîtriser les fonctions les plus importantes. Elle contient des informations importantes concernant les paramètres et la personnalisation que nous vous recommandons de lire pour obtenir un aperçu général de la palette des paramètres et options de personnalisation disponibles et comment les modifier.

Cette section est organisée comme suit :

- *Paramètres et personnalisation :* Décrit comment et où les paramètres et options de personnalisation importants peuvent être définis.
- *Tutoriels, Projets, Exemples :* Note l'emplacement des fichiers non-programmes divers contenus dans le paquet d'application.
- *Fonctions de produit et documentation, et les produits Altova :* Fournit des liens vers le site Internet Altova, où vous pouvez trouver des informations concernant les fonctions du produit, des formats d'aide supplémentaires et d'autres produits Altova.

3.2.1 Paramètres et personnalisation

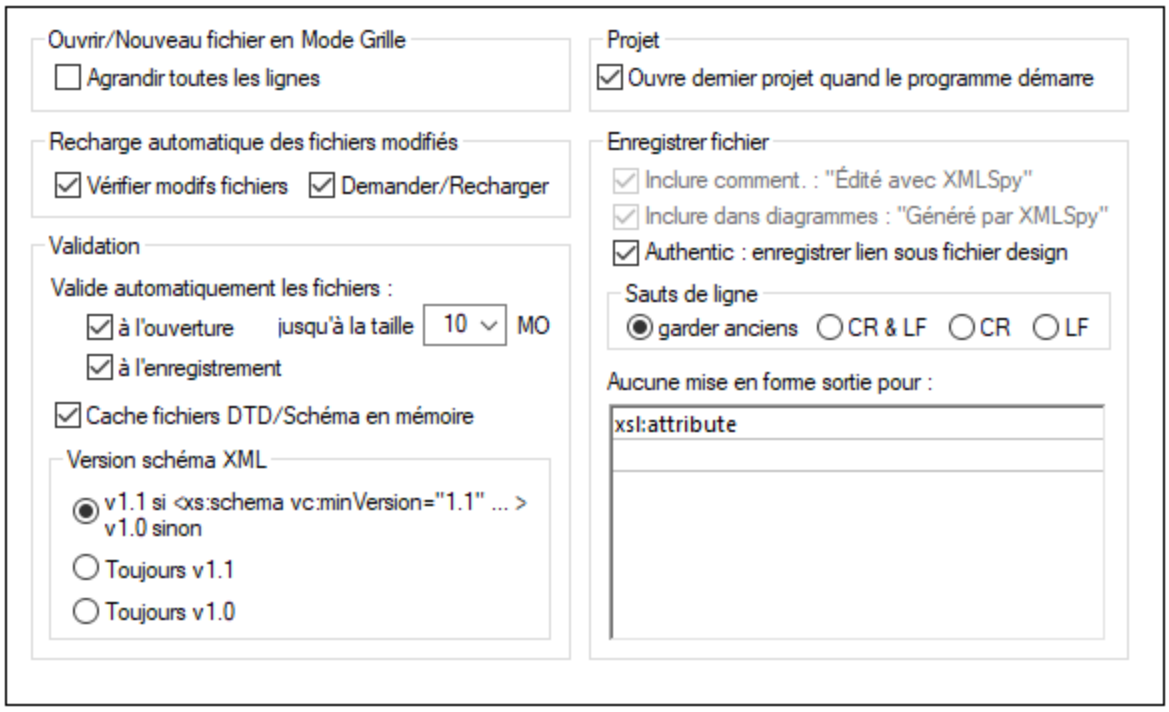
XMLSpy vous permet de choisir parmi plusieurs options de paramètres et de personnalisation. Dans cette section, nous vous guidons vers ces options et expliquons brièvement quelques aspects des menus XMLSpy. Cette section est organisée dans les parties suivantes.

- Paramètres
- Personnalisation
- Menus

Paramètres

Plusieurs paramètres importants de XMLSpy sont définis dans plusieurs onglets dans le dialogue Options (*capture d'écran ci-dessous*, accédés par le biais de la commande de menu **Outils | Options**). Nous vous recommandons de parcourir les options pour vous familiariser avec les options disponibles.

© 2016-2022 Altova GmbH

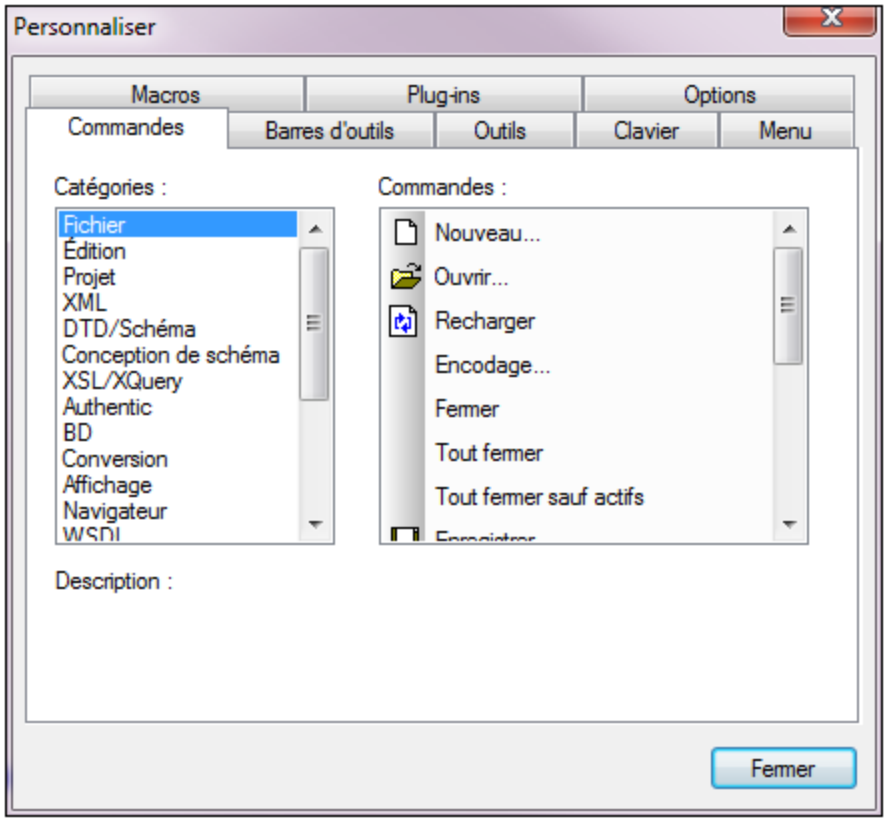


Ci-dessous, vous trouverez une liste des paramètres les plus importants. Pour les détails, voir la description du dialogue des Options dans la section de la Référence de l'utilisateur (User Reference).

- *Types de fichier et modes par défaut :* Dans l'onglet Types de fichiers vous pouvez ajouter des types de fichier que XMLSpy reconnaîtra. Un type de fichier est spécifié par une extension de fichier. Pour chaque type de fichier, vous pouvez ensuite spécifier la conformité à un standard particulier (par exemple, la conformité avec le standard DTD, Xquery ou JSON). Ce paramètre permettra d'activer des aides à l'édition pertinentes pour le standard sélectionné pour un type de fichier particulier. Vous pouvez également spécifier dans quel mode XMLSpy ouvrira les fichiers de chaque type (le mode par défaut pour ce type de fichier).
- *Validation de fichier :* Dans l'onglet Fichier (*capture d'écran* ci-dessus), vous pouvez spécifier si des fichiers doivent être validés automatiquement lors de l'ouverture et/ou l'enregistrement. Dans l'onglet Types de fichier (*voir point précédent*), la validation de fichier peut ensuite être désactivée pour des types de fichiers spécifiques.
- *Fonctions d'édition :* Dans l'onglet Édition, vous pouvez spécifier comment organiser les aides à la saisie, comment générer de nouveaux éléments et si le remplissage automatique est activé. Des options supplémentaires sont disponibles pour des modes supplémentaires dans l'onglet Mode. Dans l'onglet Polices, pour de nombreux modes, vous pouvez configurer les caractéristiques de polices pour les types de nœuds individuels dans chacun de ces modes.
- *Moteurs XSLT et FO :* Dans l'onglet XSL, vous pouvez spécifier l'utilisation d'un moteur XSL externe pour les transformations effectuées depuis la GUI. Vous devez aussi spécifier l'emplacement du processeur FO exécutable à utiliser pour les traitements FO depuis XMLSpy. Plus d'informations, voir la section Traitement XSLT .
- *Encodage :* Les encodages pour les fichiers XML et non XML sont spécifiés dans l'onglet Encodage.

Personnalisation

Vous pouvez aussi personnaliser les différents aspects de XMLSpy, y compris l'apparence de la GUI. Ces options de personnalisation sont disponibles dans le dialogue Personnaliser (*capture d'écran ci-dessous*, accédé via la commande de menu **Outils | Personnaliser**).



Les différentes options de personnalisation sont décrites dans la section **Référence de l'utilisateur**.

Menus

Les commandes de menu sont activées/désactivées selon trois facteurs : (i) le type de fichier, (ii) mode actif et (iii) emplacement actuel du curseur ou le statut actuel du document. Par exemple :

- *Type de fichier:* La commande **DTD/Schéma | Inclure une autre DTD** est uniquement activée si le fichier actif est une DTD. De même, les commandes dans le menu **WSDL** seront uniquement activées lorsqu'un fichier WSDL est actif.
- *Mode actif :* La plupart des commandes dans le menu **Conception de schéma** seront uniquement actives lorsque le mode actif est le Mode Schéma.
- *Emplacement du curseur et statut du document actuels :* Dans le mode Grille, la commande pour ajouter un attribut en tant que nœud enfant (**XML | Ajouter enfant | Attribut**) sera uniquement activée si l'item sélectionné dans le mode Grille est un élément (*Emplacement du curseur actuel*). Lorsqu'un document XSLT est actif, la commande **Stop Debugger** ne sera pas active jusqu'après le lancement d'une séance de débogage (*statut actuel du document*).

© 2016-2022 Altova GmbH

Veuillez également noter que vous pouvez personnaliser des menus (**Outils | Options**) ainsi que les glisser et les réorganiser dans la GUI (voir Barre de menu, barres d'outils, barre de statut).

3.2.2 Tutoriels, Projets, Exemples

Le paquet d'installation de XMLSpy contient des fichiers de tutoriels, de projets et d'exemple.

Emplacement des fichiers de tutoriels, de projets et d'exemple

Les fichiers de tutoriels, de projets et d'exemple de XMLSpy sont installés dans le dossier :

```
C:\Users\<username>\Documents\Altova\XMLSpy2023\Examples\
```

Le dossier `My Documents\Altova\XMLSpy2023` sera installé pour chaque utilisateur enregistré sur un PC dans le dossier `<username>` de cet utilisateur. Ainsi, sous ce système d'utilisation, chaque utilisateur aura son propre dossier `Examples` dans une zone de travail séparée.

Emplacement des fichiers de tutoriel, de projets et d'exemples

Tous les fichiers de tutoriel, de projet et d'exemple sont situés dans le dossier `Examples`. Les emplacements spécifiques sont les suivants :

- Tutoriel XMLSpy : Dossier `Tutoriel`.
- Tutoriel Authentic View : Dossier `Exemples`.
- Tutoriel WSDL : Dossier `Exemples`.
- Fichier de projet : Le projet `Exemples` avec lequel XMLSpy s'ouvre est défini dans `Examples.spp`, qui est situé dans le dossier `Exemples`.
- Fichiers d'exemple : se trouvent dans le dossier `Examples` et dans des sous-dossiers du dossier `Examples`.

3.2.3 Fonctions et Aide Authentic Desktop et les produits Altova

Le site Internet Altova, www.altova.com, contient une foule de renseignements et de ressources concernant XMLSpy. Vous y trouverez entre autres :

Liste des fonctions XMLSpy

Le site Internet d'Altova offre une liste actualisée de fonctions XMLSpy, vous permettant de comparer la prise en charge des différentes fonctions pour les éditions XMLSpy (Enterprise et Professional). Sur le site Internet vous pourrez également consulter une liste de toutes les nouvelles fonctions depuis chaque publication précédente.

Aide XMLSpy

Cette documentation est l'aide fournie par Altova pour XMLSpy. Elle est disponible comme système d'aide built-in XMLSpy, qui est accessible par le menu **Aide** ou en appuyant sur **F1**. De plus, les manuels d'utilisateurs pour tous les produits Altova sont disponibles dans les formats suivants :

- Mauels HTML en ligne, accessibles depuis la page Support du site Internet Altova
- PDF imprimables que vous pouvez télécharger depuis le site Altova et imprimer localement
- Livres imprimés que vous pouvez acheter via un lien sur le site Altova

Options de prise en charge

Si vous nécessitez des informations complémentaires à celles disponibles dans les manuels d'utilisateur (cette documentation) ou si vous avez des questions concernant des produits Altova, veuillez visiter notre Centre d'assistance sur le site Altova. Vous y trouverez :

- Des liens vers nos pages FAQ
- Des forums de discussion concernant les produits Altova et les sujets XML généraux
- Des formulaires d'aide en ligne qui vous permettront de déposer des demandes d'assistance si vous détenez un kit d'assistance. Votre demande d'assistance sera traitée par notre équipe d'assistance.

Produits Altova

Pour consulter une liste de tous les produits Altova, rendez-vous sur le site Altova.

© 2016-2022 Altova GmbH

4 Modes d'édition

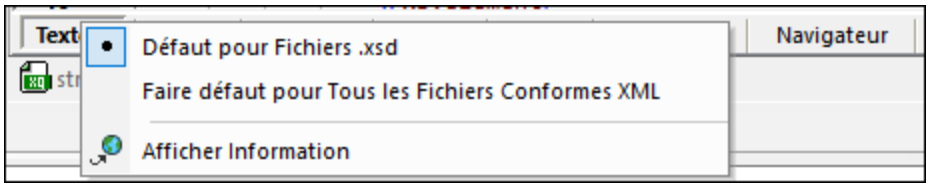
XMLSpy est doté de modes d'édition puissants. Outre un mode Texte comprenant des fonction d'édition intelligentes, l'utilisateur dispose de modes graphiques permettant de simplifier considérablement l'édition des documents. Selon le type de documents actuellement actifs dans XMLSpy, la fenêtre principale permettra d'utiliser un ou plusieurs modes d'édition de XMLSpy. Par exemple, si un fichier Office Open XML ou un fichier ZIP est actif, la fenêtre principale contiendra uniquement un mode d'édition : Mode d'Archive. Si un document HTML est actif, deux modes d'édition seront actifs : Mode Texte et Mode Navigateur. Si un document XML est actif, vous disposerez de sept modes d'édition : Mode Texte, Mode Grille, Mode Schéma, Mode WSDL, Mode XBRL, Mode Authentic et Mode Browser ; parmi ces modes, le Mode Schéma sera activé uniquement pour les documents XML Schéma et le Mode WSDL sera activé uniquement pour les documents WSDL.

Dans cette section, nous décrivons les différents modes d'édition disponibles dans XMLSpy:

- Mode Texte
- Mode Grille
- Mode Schéma
- Mode WSDL
- Mode XBRL
- Mode Authentic
- Mode Navigateur
- Mode Archive

Sélection du mode par défaut

Un document d'un type spécifique (par exemple, un document XML ou un document JSON) peut être consulté de manières variées. Vous pouvez sélectionner le mode d'édition par défaut directement dans l'interface, soit en cliquant soit en cliquant avec la touche de droite dans l'onglet du mode actuel et en sélectionnant l'option appropriée depuis le menu qui apparaît *(voir capture d'écran ci-dessous):*



Les options de menu sont :

- *Défaut pour extension de fichier :* le mode d'édition actuel devient le mode par défaut pour les fichiers ayant la **même extension de fichier** que le fichier actif (par exemple, fichiers `.xml` ou `.xq`).
- *Défaut pour tous les fichiers de conformation semblable :* le mode d'édition actuel devient le mode par défaut pour les fichiers ayant le **même type de conformance** que le fichier actif (par exemple, tous les fichiers conformes à XML ou tous les fichiers conformes à XQuery). Cela peut constituer un ensemble plus grand que celui défini pour une extension de fichier ; par exemple, les deux extensions de fichier différente `.xq` et `.xquery` peuvent toutes les deux être conformes à XQuery.
- *Montrer information :* liens vers une page dans le Site web Altova qui décrit les fonctions du mode d'édition actuel.

Si une option de mode par défaut a été choisie, vous trouverez un bouton radio à côté d'elle *(voir capture d'écran ci-dessous)*.

Dans les cas suivants, aucun changement du mode d'édition par défaut ne sera possible

- *Fichier de Schéma JSON :* Ces fichiers, comme les fichiers d'instance JSON, ont une extension **.json**. Néanmoins, ils seront sélectionnés automatiquement depuis le contenu comme étant des Schémas JSON, et seront toujours ouverts dans Mode Schéma JSON.
- *Fichiers XML activés pour Authentic :* Il s'agit de fichiers XML (avec une extension de fichier **.xml**) qui ont été attribués à une feuille de style Altova StyleVision, qui permet de consulter le document XML en tant que document XML Authentic. Ces fichiers seront toujours ouverts dans Mode Authentic.
- *Fichiers XSD pour XBRL (taxonomies XBRL) :* ces fichiers présentent une extension **.xsd**. Néanmoins, ils seront détectés automatiquement depuis le contenu en tant que taxonomies XBRL (et pas Schémas XML), et seront donc toujours ouverts dans Mode XBRL.

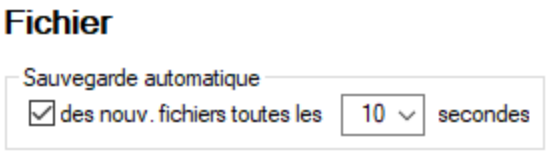
Note : La conformité de fichier de chaque extension de fichier, et le mode d'édition par défaut qui correspond aux types de fichiers individuels, peuvent être définis dans la section Types de fichier du dialogue Options. Néanmoins, la sélection de mode par défaut dans le mode d'édition lui-même (qui a été décrit ici) vous donne l'option de configurer des modes par défaut plus rapidement et de manière plus simple.

© 2016-2022 Altova GmbH

4.1 Sauvegarde automatique des fichiers

Les fichiers qui sont modifiés dans XMLSpy sont automatiquement sauvegardés à des intervalles réguliers. Dans l'onglet *Fichier* du dialogue Options (**Outils | Options | Fichier**) affiché dans la capture d'écran, vous pouvez :

- Allumer/éteindre des sauvegardes automatiques
- Spécifier la fréquence des sauvegardes (5 secondes à 300 secondes)



Tous les types de fichier qui peuvent être édités dans XMLSpy seront sauvegardés ; les archives ZIP ne sont pas sauvegardés.

Indicateurs

Les onglets de fichier en bas de la fenêtre principale contiennent des symboles à droite du nom de fichier qui indiquent l'état enregistré/non-enregistré et l'état de sauvegarde du fichier *(capture d'écran ci-dessous)*.



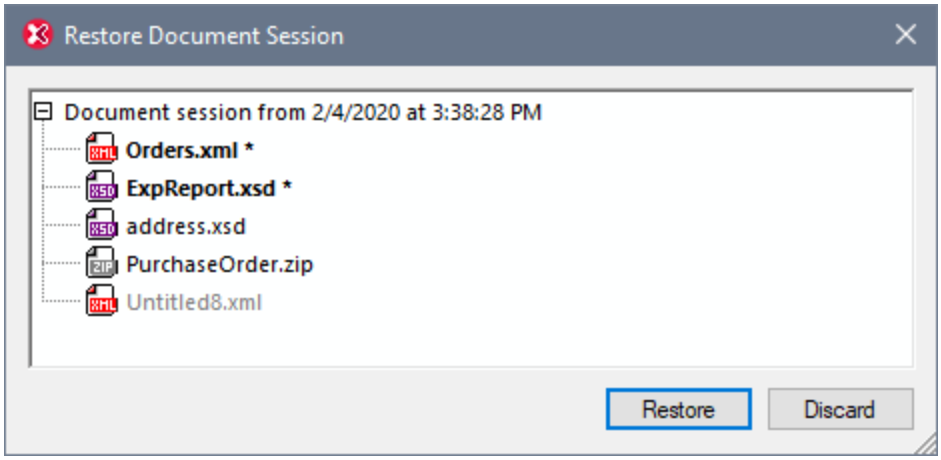
Enregistré / Non-enregistré
Un cercle coloré apparaît si un fichier a été modifié. Si vous ne voyez pas apparaître de symbole, cela signifie que le fichier n'a pas été modifié depuis la dernière ouverture ou depuis le dernier enregistrement. Dans la capture d'écran ci-dessus, par exemple, `address.xsd` n'a pas été modifié depuis son dernier enregistrement et `Untitled8.xml` est un nouveau fichier qui n'a pas été édité ou enregistré depuis qu'il a été créé.

État de sauvegarde
Les couleurs des cercles indiquent l'état de sauvegarde du fichier.

- *Jaune :* Le fichier a été modifié mais la dernière modification n'a pas été sauvegardée (ou enregistrée).
- *Vert :* Le fichier a été sauvegardé et il n'a pas été modifié depuis la dernière sauvegarde. Néanmoins, le fichier n'a pas été enregistré. (S'il avait été enregistré, il n'y aurait pas de cercle.)
- *Rouge :* La sauvegarde n'est pas prise en charge pour ce fichier (par exemple, s'il s'agit du Mode Archive d'un fichier ZIP) ou si une sauvegarde a échoué.
- *Gris :* La fonction de sauvegarde automatique a été désactivée (par le biais du dialogue des Options ; *voir ci-dessus*). La présence du symbole, néanmoins, indique que le fichier n'a pas été enregistré depuis la dernière modification. (S'il avait été enregistré, il n'y aurait pas de cercle.)

Restaurer depuis les sauvegardes

Si XMLSpy cesse de manière imprévue, alors, au prochain démarrage de l'application, un dialogue Restaurer Document est affiché qui contient une liste de tous les documents qui ont été ouverts au moment de l'interruption de l'application *(capture d'écran ci-dessous)*. Vous pouvez planer sur chaque fichier pour voir son chemin. Dans le cas des fichiers temporaires qui n'ont pas encore été enregistrés, le chemin de fichier sera le chemin par défaut actuel où un dialogue Enregistrer sous s'est ouvert pour ce fichier.



Pour chaque fichier dans la liste, le style de police et la présence/absence d'astérisques propose l'information suivante :

- Un style gras et une astérisque indiquent que le fichier contient des modifications non enregistrées. Ce type de fichier sera restauré à leur dernier état de sauvegarde.
- Un style normal indique que le fichier a été enregistré et qu'il n'y a pas de modifications non enregistrée. Les fichiers seront restaurés à leur état enregistré.
- Un style grisé indique que le fichier n'a pas été enregistré ni sauvegardé (par exemple, parce qu'il s'agit d'un nouveau fichier qui n'a pas été édité). Ces fichiers ne seront pas restaurés.

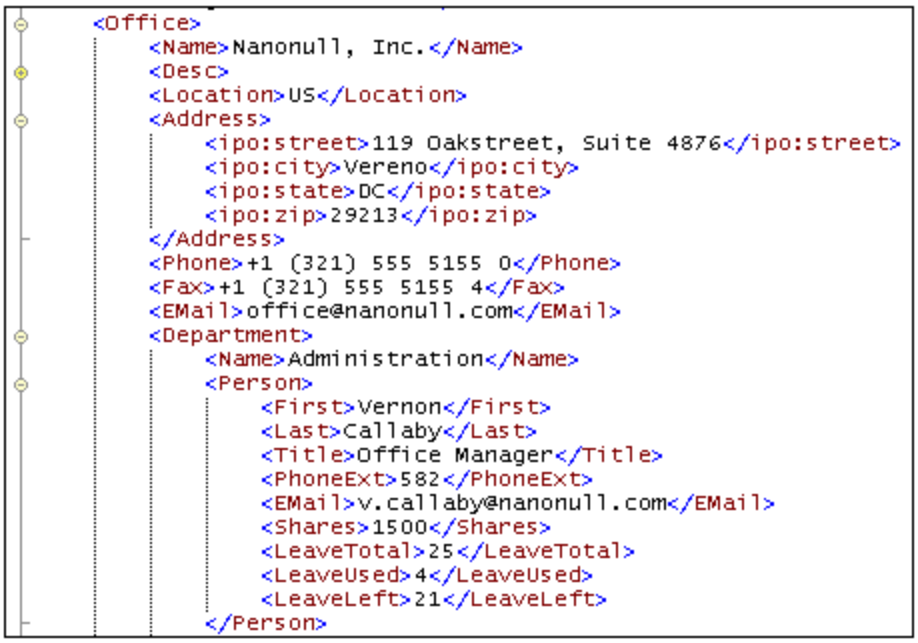
Vous pouvez à présent suivre une des étapes suivantes :

- Cliquer sur **Restaurer** pour restaurer les fichiers dans la GUI depuis leur dernier état sauvegardé.
- Cliquer sur **Éliminer** pour ne pas ouvrir un des fichiers listé et pour éliminer toute sauvegarde disponible.

© 2016-2022 Altova GmbH

4.2 Mode Texte

Dans le Mode Texte (*capture d'écran ci-dessous*), vous pouvez saisir le texte de votre document - aussi bien les balises que le contenu - directement. Tout fichier de texte, y compris les documents non-XML (comme les documents XQuery et HTML) peuvent être édités dans le Mode Texte. Un certain nombre de fonctions vous aident à saisir rapidement et précisément les données dans votre document.



Dans cette section, nous décrivons les fonctions générales du Mode Texte qui sont disponibles pour tous les types de documents. Les types de document spécifiques, comme XML, XQuery et CSS ont certaines fonctions spécifiques aux types décrits dans les sections respectives de ces types de document. Par exemple, les fonctions du Mode Texte spécifiques à XML sont décrites dans la section XML | Éditer un XML dans le Mode Texte.

Les fonctions générales du Mode Texte sont classées comme suit :

- Formatage dans le Mode Texte décrit comment spécifier les propriétés de police, les indentations et les retours à la ligne d'un document.
- Afficher le document contient des informations concernant la numérotation de ligne, les signets, l'agrandissement et la réduction des nœuds et d'autres fonctions liées à l'affichage.
- Édition dans le Mode Texte décrit les fonctions qui sont disponibles pendant l'édition, en particulier les fonctions d'édition intelligente.
- Parcourir le document explique les différentes possibilités de parcourir un document dans le Mode Texte.
- Chercher et remplacer décrit les fonctions de Recherche et de remplacement du Mode Texte.
- Les Assistants à la saisie sont les fenêtres qui fournissent des options de saisie de données sensibles au contexte. Par exemple, les éléments ou les attributs qui peuvent être ajoutés de manière valide à un endroit particulier d'un document sont affichés dans un assistant de saisie et une de ces options peut être insérée sur un double clic.

- Mode Écran divisé partage en deux la fenêtre principale du Mode Texte et affiche le document actif dans les deux affichages. Cela vous permet de voir des parties différentes d'un long document côte à côte.
- Raccourcis Mode Texte recense les raccourcis par défaut des commandes Mode Texte les plus communément utilisées.

Passer en mode Texte

Pour ouvrir le Mode Texte d'un document, cliquer sur le bouton **Texte** situé en bas de la fenêtre Document ou choisir **Mode | Mode Texte**.

4.2.1 Formatage dans le Mode Texte

Le Mode Texte propose un grand nombre d'options de formatage de texte. Ces options sont regroupées ci-dessous.

Polices

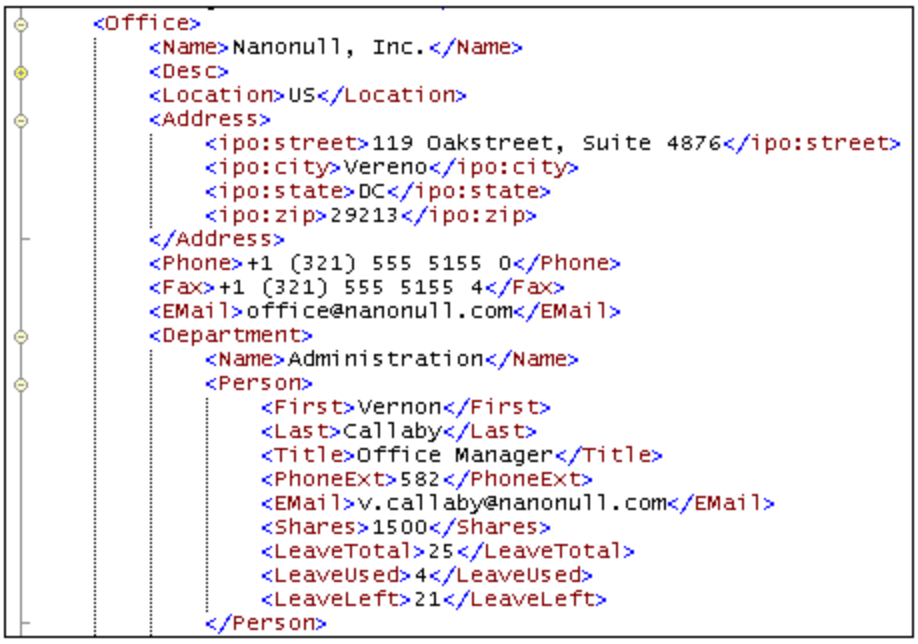
La famille, le style de la police, et la couleur d'arrière-plan peuvent être personnalisés pour les groupes de documents suivants : (i) documents XML génériques (y compris HTML); (ii) les documents XQuery ; et (iii) les documents CSS.

Dans un document, les éléments de texte de sémantique différente peuvent être colorés différemment. Par exemple, vous pouvez colorer des noms d'élément, des noms d'attribut et des contenus d'élément différemment. Lorsque vous définissez des couleurs différentes pour les différents éléments de texte, la fonction de coloration de la syntaxe est activée. Les polices de texte sont personnalisées dans la section Polices et Couleurs du dialogue Options.

Retrait

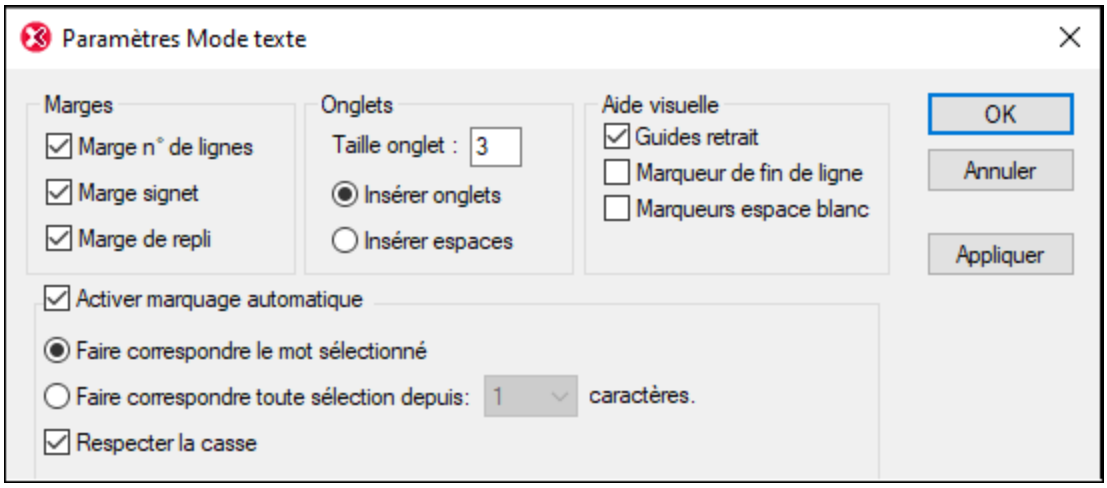
Vous pouvez mettre en retrait a document pour afficher sa structure, tel qu'affiché dans la capture d'écran ci-dessous. Lorsque le document est affiché avec ce type d'indentation hiérarchique, on dit qu'il est en mode pretty-printed. À chaque descente de niveau, le retrait recule d'un cran par rapport à son élément parent. Pour voir le document dans son format pretty-print, vous devez : (i) définir son indentation (conformément à vos préférences) et (ii) appliquer pretty-printing.

© 2016-2022 Altova GmbH



Pour définir et appliquer pretty-printing, exécutez les étapes suivantes. Veuillez noter que la définition de pretty-printing est valable pour toute l'application. Ceci signifie que les paramètres de pretty-printing seront appliqués dans tous les documents XML et XML une fois que les paramètres ont été enregistrés. De manière conséquente, les paramètres doivent être édités uniquement si vous souhaitez modifier l'affichage pretty-printing de vos documents. Une fois que les paramètres valables pour toute l'application ont été définis, vous devez appliquer pretty-printing individuellement à chaque document. Un document est toujours affiché dans le format pretty-printing dans lequel il a été appliqué, et son formatage d'affichage ne changera pas jusqu'à ce que le prochain pretty-printing est appliqué.

1. Dans la section Pretty-printing du dialogue des Options (**Outils | Options**), activez pretty-printing en vérifiant l'option *Utiliser l'indentation déterminée par la configuration de l'onglet du Mode Texte*. Il s'agit de ce paramètre qui active l'indentation que vous verrez dans un document pretty-print. Si l'option *Utiliser retrait* n'est pas cochée, alors chaque ligne dans le document commencera sans retrait.
2. Encore dans la section Pretty-printing (*voir le point précédent*), cliquez sur la touche **Paramètres Mode texte** (en haut à droite) pour aller dans le dialogue des Paramètres du Mode Texte (*voir la capture d'écran ci-dessous*) et définissez la quantité et le type de retrait. (Des manières alternatives d'accéder au dialogue des Paramètres Mode texte sont (i) la commande de menu **Mode | Paramètres Mode Texte** et (ii) l'icône **Paramètres Mode texte** dans la barre d'outils Texte.) Dans le dialogue des Paramètres Mode Texte, le champ *Taille de l'onglet* spécifie le nombre d'espaces qui font un onglet ainsi qu'un retrait. Choisissez quels retraits pretty-printing seront composés d'onglets ou d'espaces en sélectionnant soit *Insérer Onglets*, soit *Insérer Espaces*. Dans les deux cas, la taille de chaque retrait sera équivalente au nombre d'espaces **x** spécifiés dans le champ *Taille de l'onglet* (puisque 1 onglet = X espaces).



3. Après avoir défini l'option pretty-printing valable pour toute l'application, tel que défini dans les deux étapes précédentes, vous pouvez appliquer pretty-printing à des documents individuels. Transformez le document que vous voulez avoir en pretty-print en document actif. Cliquez sur la commande **Édition | Pretty-Print** ou sur l'icône **Pretty Print** dans la barre d'outils Texte. Ceci affichera le texte du document conformément au formatage spécifié dans la section Pretty Print du dialogue des Options. Veuillez noter que la commande **Pretty-- Print** supprime les espaces de début et les espaces blancs de fin inutiles.

Note : les guides de retrait sont des lignes pointillées verticales (*voir la capture d'écran au début de la section*). Elles sont activées ou désactivées par le biais de la case à cocher Guides retrait dans le panneau *Aide visuelle* du dialogue de Paramètres du Mode Texte (*voir la capture d'écran ci-dessus*).

Note : pour information sur la gestion de l'espace blanc, voir la section Espace blanc.

Utiliser les onglets et les espaces pour le formatage

Vous pouvez utiliser les onglets et les espaces pour formater le texte, en particulier pour les documents non-XML, lorsque l'option Pretty-print n'est pas disponible. Si vous appuyez sur **Entrée** ou **Shift+Entrée**, le curseur passera à une position dans la ligne suivante qui correspond au point de démarrage de la ligne précédente.

Word-wrapping

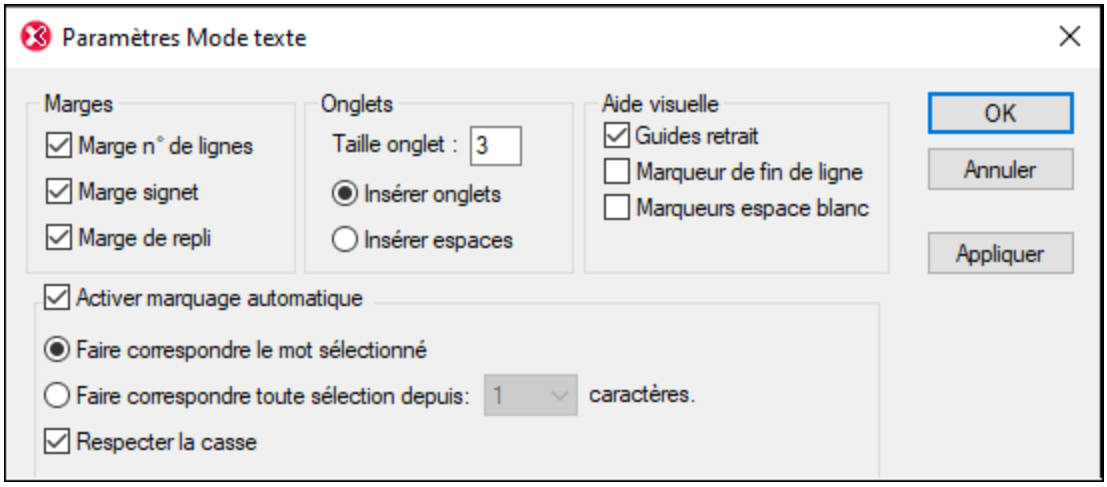
Les lignes de texte qui sont plus longues que la largeur de la Fenêtre principale peuvent être coupées par un retour automatique à la ligne en activant la commande **Mode | Retour à la ligne**; l'icône correspondant se trouve dans la barre d'outils Texte.

4.2.2 Afficher le document

Le Mode Texte dispose de fonctions visuelles pour faciliter l'affichage et l'édition de grandes portions de texte. Les fonctions les plus utiles sont : (i) les Numéros de ligne, (ii) les Signets, (iii) le Pli de source (agrandir et réduire l'affichage des nœuds), (iv) les Guides de retrait, et (v) les Marqueurs de fin de ligne et des espaces blancs. Ces commandes sont disponibles dans le dialogue Paramètres Mode texte (*première capture d'écran ci-dessous*) et la barre outils de texte (*deuxième capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH

Le dialogue Paramètres Mode texte est accessible par la commande **Mode | Paramètres Mode texte**, le bouton **Paramètres Mode texte** dans la barre outils de Texte ou le menu contextuel du Mode Texte. Les paramètres dans le dialogue Paramètres Mode texte s'appliquent à toute l'application et non seulement au document actif.





Les autres fonctions utiles sont le Zoom et la navigation et recherche.

Numérotation des lignes

La numérotation des lignes est affichée dans la marge de gauche (*capture d'écran ci-dessous*), qui peut être activée ou désactivée dans le dialogue Paramètres Mode texte (voir capture d'écran ci-dessus). Lorsqu'une section de texte est réduite, les numéros des lignes de la portion de texte réduite sont également dissimulés. Une commande liée est la commande de Aller à la ligne/Caractère.

Signets

Les lignes dans le document peuvent être marquées séparément pour une référence et un accès rapides. Si la marge des signets est activée, les signets seront affichés dans la marge correspondante ; sinon, les lignes marquées seront soulignées en cyan.

La marge des signets peut être activée ou désactivée dans le dialogue Paramètres Mode texte (*capture d'écran ci-dessus*).

Vous pouvez éditer et parcourir les signets en utilisant les commandes dans le menu **Édition** et dans la barre outils Texte. Vous pouvez insérer les signets peuvent être insérés avec la commande **Édition | Insérer/Supprimer signet**, vous permettant de marquer une ligne dans le document pour la référence. Un signet peut être supprimé en sélectionnant la ligne marquée et puis en sélectionnant la commande **Édition | Insérer/Supprimer signet**. Pour parcourir les signets d'un document, utiliser la commande **Édition | Aller au signet suivant** et **Édition | Aller au signet précédent**. Ces commandes de signet sont aussi disponibles en tant qu'icônes dans la barre d'outils Texte (*capture d'écran ci-dessus*).

Pliage de source

Le pliage de source permet d'agrandir et de réduire des nœuds dans des documents XML, XQuery, JSON et CSS. Les nœuds qui peuvent être agrandis/réduits sont indiqués dans la marge de pli de source par un signe +/- (*voir capture d'écran ci-dessous*). La marge peut être activée ou désactivée dans le dialogue Paramètres Mode texte (*voir capture d'écran ci-dessus*). Dans la capture d'écran ci-dessous, veuillez noter que les trois nœuds ont été réduits : l'élément `shipTo` et deux éléments `item`. Lorsqu'un nœud est réduit, cela est indiqué par une ellipse (*marqué en vert dans la capture d'écran ci-dessous*). Si le curseur de la souris est placé sur une ellipse, le contenu du nœud réduit est affiché dans une fenêtre popup (*marqué en bleu dans la capture d'écran ci-dessous*). Si le contenu est trop volumineux pour une fenêtre popup, cela est indiqué par une ellipse en bas de la fenêtre popup.



L'icône **Basculer tous les plis** dans la barre outils Texte bascule **tous** les nœuds dans leur forme agrandie ou réduit tous les nœuds à l'élément de document top-level.
Les options suivantes sont disponibles lorsque vous cliquez sur l'icône du nœud +/- :

Clic [-]	Réduit le nœud.
Clic [+]	Agrandit le nœud de manière à ce que les nœuds descendants sont affichés sous leur forme agrandie ou réduite selon leur forme avant la réduction du nœud.
Shift+Clic [-]	Réduit tous les nœuds descendants, mais laisse le nœud qui a été cliqué dans sa forme agrandie.
Ctrl+Clic [+]	Agrandit le nœud cliqué ainsi que les nœuds descendants.

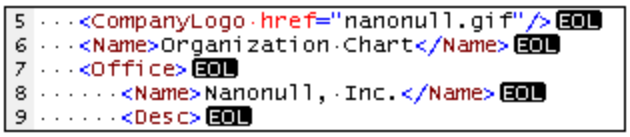
© 2016-2022 Altova GmbH

Guides de retrait

Les guides de retrait sont des pointillés verticaux qui indiquent l'étendue du retrait d'une ligne (*voir capture d'écran ci-dessus*). Ils peuvent être activés ou désactivés dans le dialogue Paramètres Mode texte.

Marqueurs de fin de ligne, marqueurs d'espace blanc

Les marqueurs de fin de ligne (EOL) et les marqueurs d'espace blanc peuvent être activés dans le dialogue Paramètres Mode texte. La capture d'écran ci-dessous montre ces marqueurs dans l'affichage du document ; chaque point représente un espace blanc.



Zoom avant zoom arrière

Vous pouvez zoomer en avant et en arrière dans le Mode Texte en faisant tourner la roue de la souris tout en maintenant la touche **Ctrl** pressée. Cela vous permet d'agrandir et de réduire la taille du texte dans le Mode Texte. Si vous souhaitez augmenter la taille des polices, vous pourrez le faire dans le dialogue Options.

4.2.3 Édition dans le Mode Texte

Les fonctions d'édition du texte suivantes sont généralement disponibles dans le Mode Texte pour tous les types de document. Ces fonctions sont disponibles en plus des fonctions générales d'édition des applications comme **Couper**, **Copier**, **Coller**, **Supprimer**, et **Sélectionner tout** (qui sont disponibles en tant que commandes dans le menu **Édition**).

- Coloration syntaxique
- Édition intelligente
- Saisie automatique
- Correspondance de la balise de démarrage et de la balise de fin
- Déplacer des éléments frères les uns par rapport aux autres
- Sélectionner un élément entier et se rendre sur le parent
- Glisser/déposer et les menus contextuels
- Annulations sans limites
- Vérification orthographique

Pour certains types de document (comme les documents XML et XQuery) des fonctions spécialisées supplémentaires sont disponibles. Celles-ci sont décrites, respectivement dans les sections qui traitent de ces types de document.

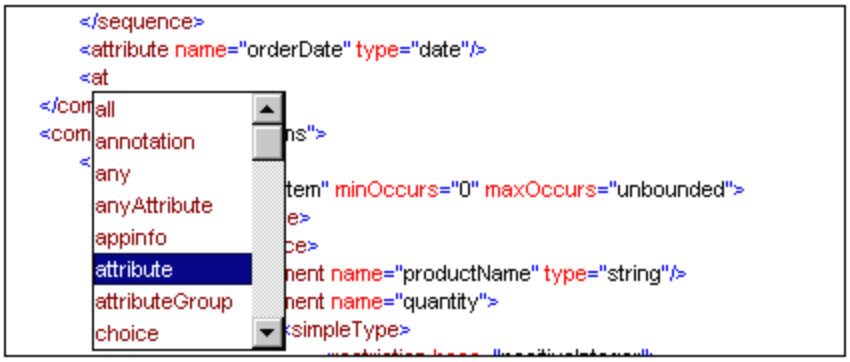
Note : Pour les fichiers volumineux, la saisie automatique et les assistants à la saisie peuvent être désactivés, permettant ainsi un chargement et une édition plus rapide. La taille seuil de fichier est spécifiée par l'utilisateur. Pour plus de détails, voir la section de référence Options | Édition.

Coloration syntaxique

La coloration syntaxique est appliquée conformément à la valeur sémantique du texte. Par exemple, dans des documents XML, selon que le nœud XML soit un élément, un attribut, un contenu, une section CDATA, un commentaire ou une instruction de traitement, le nom de nœud (et dans certains cas le contenu du nœud) est coloré d'une teinte différente. Il existe plusieurs types de documents comme par exemple : (i) XML générique (qui comprend HTML); (ii) XQuery; (iii) CSS ; et (iv) JSON. Les propriétés de texte (y compris color) de chaque groupe peuvent être configurées dans la section Polices de texte dans le dialogue Options (**Outils | Options**).

Édition intelligente

Si vous travaillez avec un document XML sur la base d'un schéma, XMLSpy vous propose plusieurs outils d'édition intelligents dans le Mode Texte. Ceux-ci vous permettent d'insérer rapidement l'élément, l'attribut ou la valeur d'attribut corrects conformément au modèle de contenu défini pour l'élément que vous êtes en train d'éditer. Par exemple, lorsque vous commencez à saisir la balise de début d'un élément, les noms de tous les éléments autorisés par le schéma à cet endroit sont affichés dans une fenêtre pop-up (*capture d'écran ci-dessous*). Il suffit de choisir l'élément souhaité et d'appuyer sur **Entrer** pour insérer le nom de cet élément dans la balise de démarrage. De même, une fois que la balise de démarrage est créée, la balise de fin est ajoutée automatiquement (*voir saisie automatique ci-dessous*).



Les fenêtres popup apparaissent aussi dans les cas suivants :

- Lorsque le curseur se trouve à l'intérieur de la balise de démarrage d'un élément qui a un attribut défini et si la barre espace est appuyée. La fenêtre popup contiendra tous les attributs disponibles.
- Lorsque le curseur se trouve à l'intérieur des guillemets doubles délimitant une valeur d'attribut qui a des valeurs énumérées. La fenêtre popup contiendra les valeurs énumérées.
- Lorsque vous saisissez `</` (qui signifie le début d'une balise de fermeture), le nom de l'élément à fermer apparaît dans la fenêtre popup.
- Lorsque vous souhaitez écrire un élément vide en tant que balise unique ou convertir un élément vide de deux balises en un élément vide d'une seule balise, saisir la barre oblique de fermeture après le nom de l'élément : `<element/`. Un élément vide avec une seule balise sera créé ; si une balise de fermeture existe, elle sera supprimée : `<element/>`.

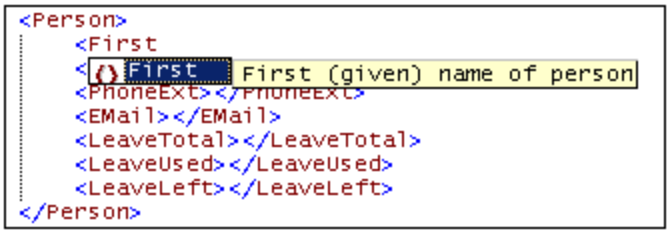
Saisie automatique

L'édition dans le Mode Texte peut facilement donner lieu à des documents XML et à des balises de type différent (comme un document HTML par ex.) mal formés. Des balises de fermeture peuvent manquer, être mal rédigées ou ne pas correspondre structurellement. XMLSpy termine automatiquement les balises de démarrage et de fin des éléments, et insère tous les attributs requis dès que vous avez terminé la saisie du nom d'éléments avec votre clavier. De même, le curseur est positionné automatiquement entre les balises de

© 2016-2022 Altova GmbH

démarrage et de fin de l'élément, pour vous permettre de poursuivre immédiatement l'ajout d'éléments ou de contenus enfants : `<img src="" alt=""> </img>`

XMLSpy se sert des règles XML de bonne formation et de validité pour la saisie automatique. L'information concernant la structure du document est obtenue à partir du schéma sur lequel le document se base. (Dans le cas des schémas bien utilisés, comme des HTML et des XSLT, l'information de schéma est intégrée dans XMLSpy.) La saisie automatique utilise non seulement des informations concernant la structure du document, mais aussi les valeurs stockées dans le document. Par exemple, les énumérations et les annotations de schéma dans un Schéma XML sont utilisées activement par la fonction de la saisie automatique. Si, dans le schéma, les valeurs sont énumérées pour un nœud particulier, alors ces énumérations seront affichées en tant qu'options de saisie automatique lorsque ce nœud est édité. De même, si des annotations existent pour un nœud dans le schéma, ces annotations seront affichées lorsque le nom du nœud est saisi dans le document (*capture d'écran ci-dessous*). (*Prénom de la personne* est l'annotation de schéma de l'élément `First` .)



Le remplissage automatique peut être activé ou désactivé dans la section Édition du dialogue Options (**Outils | Options | Édition**).

Correspondance de la balise de démarrage et de la balise de fin

Lorsque vous placez le curseur à l'intérieur d'une balise de démarrage ou de fin d'un élément de balisage, appuyez sur **Ctrl+E** pour déplacer la sélection vers l'autre membre de la paire. Appuyer plusieurs fois sur **Ctrl+E** pour passer des balises de démarrage aux balises de fin. Cette fonction est une aide excellente pour une localisation rapide des balises de démarrage et de fin d'un élément XML. De plus, les noms des éléments sont marqués dans deux couleurs différentes selon le fait que les noms des balises de démarrage et de fin concordent ou pas. Cela sert d'aide visuelle à l'édition. Les couleurs de marquage peuvent être configurées dans le dialogue des Options. Lorsque vous éditez le nom d'un élément dans une balise de démarrage, la balise de fin sera aussi éditée automatiquement.

Déplacer des éléments frères les uns par rapport aux autres

Lorsque le curseur se trouve à l'intérieur d'un élément, appuyer sur **Alt+FlècheHaut** ou **Alt+FlècheBas** pour déplacer l'élément sélectionné vers le haut ou vers le bas conformément à ses frères.

Sélectionner un élément entier et se rendre sur le parent

Lorsque le curseur se trouve à l'intérieur d'un élément, appuyer sur **Ctrl+Shift+E** pour sélectionner l'élément entier. Si vous cliquez sur **Ctrl+Alt+E**, la balise de démarrage de l'élément parent est marquée. Ces deux raccourcis vous aident à localiser rapidement la position actuelle de votre curseur par rapport à la structure du document.

Glisser/déposer et les menus contextuels

Vous pouvez aussi utiliser la fonction glisser/déposer pour déplacer un bloc de texte dans un nouvel emplacement, ou bien, vous pouvez cliquer avec la touche de droite pour accéder directement aux commandes

d'édition utilisées fréquemment (comme Couper, Copier, Coller, Supprimer, Envoyer par mail, et Aller à la ligne/carac) dans un menu contextuel.

Annulations sans limites

XMLSpy offre des niveaux sans limite d'Annulation et de Rétablissement pour toutes les opérations d'édition.

Vérification orthographique

Dans le Mode Texte, les documents peuvent être soumis à une vérification orthographique à l'aide d'un des dictionnaires intégrés. Un dictionnaire de l'utilisateur peut aussi être créé et édité pour y intégrer des mots non contenus dans le dictionnaire de cette langue. Pour plus de détails, voir les descriptions des commandes d'orthographe et d'options d'orthographe.

4.2.4 Parcourir le document

Vous pouvez utiliser les fonctions suivantes pour parcourir un document dans le Mode Texte:

- Marquage de texte vous permet de trouver toutes les correspondances d'une ligne de texte ou d'un mot que vous sélectionnez. Chaque correspondance est indiquée dans la barre de défilement, vous permettant ainsi de parcourir aisément toutes les correspondances.
- Aperçu du document dans la barre de défilement vous montre l'emplacement relatif du curseur et la sélection du texte dans le document.
- Aller à la ligne/caractère vous mène directement à la ligne et au caractère spécifié.

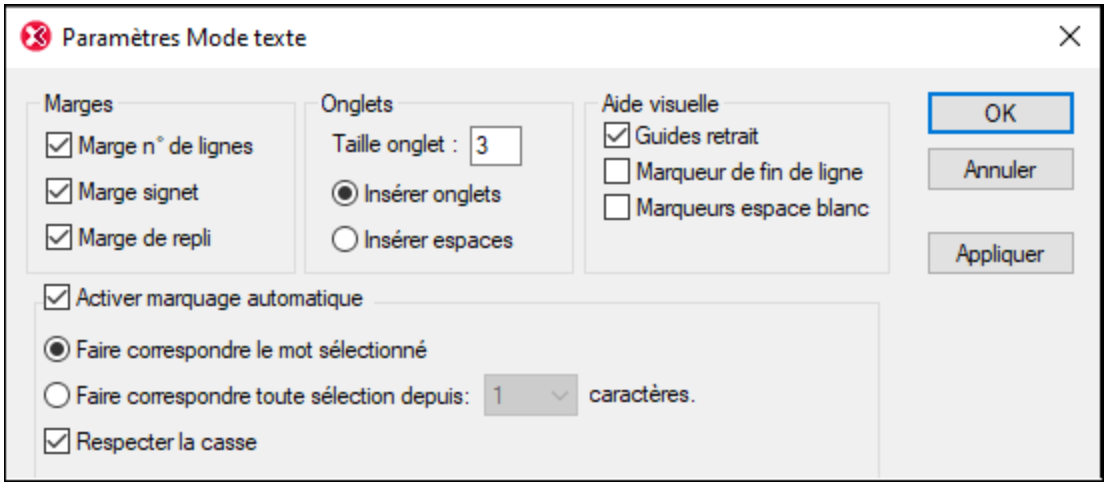
Marquage de texte

Si le marquage de texte est activé dans le dialogue de Paramètres de Mode Texte (**Mode | Paramètres Mode texte**), alors toutes les correspondances de la sélection de texte dans le document effectuées par l'utilisateur seront marquées. La sélection sera marquée en bleu clair et les correspondances seront marquées en orange pâle (*voir capture d'écran ci-dessous*). La sélection et ses correspondances seront indiquées dans la barre de défilement par des blocs de marquage gris. Veuillez aussi noter que la position de curseur actuelle est indiquée par le trait de marquage bleu foncé situé dans la barre de défilement.

© 2016-2022 Altova GmbH

```
<expense-report xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ExpReport.xsd" currency="USD" detailed="false"
total-sum="556.9">
    <Person>
        <First>Fred</First>
        <Last>Landis</Last>
        <Title>Project Manager</Title>
        <Phone>123-456-7890</Phone>
        <Email>f.landis@nanonull.com</Email>
    </Person>
    <expense-item type="Lodging" expto="Sales">
        <Date>2003-01-01</Date>
        <expense>122.11</expense>
    </expense-item>
    <expense-item type="Lodging" expto="Development">
        <Date>2003-01-02</Date>
        <expense>122.12</expense>
        <description>Played penny arcade</description>
    </expense-item>
    <expense-item type="Lodging" expto="Marketing">
        <Date>2003-01-02</Date>
        <expense>299.45</expense>
        <description>Treated Clients</description>
    </expense-item>
    <expense-item type="Entertainment" expto="Development">
        <Date>2003-01-02</Date>
        <expense>13.22</expense>
        <Misc misctype="TeamBuilding"/>
        <description>Bought signed XMLSpy Handbook</description>
    </expense-item>
</expense-report>
```

Pour activer la fonction du marquage de texte, sélectionner *Activer marquage automatique* dans le dialogue de Paramètres de Mode Texte (**Mode | Paramètres Mode texte**, *capture d'écran ci-dessous*). Une sélection peut être un mot entier ou un nombre fixe de caractères. Vous pouvez aussi spécifier si vous souhaitez respecter la casse ou pas.



Veuillez noter les points suivants :

- Pour une sélection de caractère, vous pouvez spécifier le nombre minimum de caractères devant correspondre, en commençant par le premier caractère de la sélection. Par exemple, vous pouvez

choisir de faire correspondre deux caractères ou plus. Dans ce cas, les sélections d'un seul caractère n'auront pas de correspondance, mais une sélection consistant en deux ou plus de caractères pourra avoir des correspondances. Donc, dans ce cas, si vous sélectionnez `t`, aucune correspondance ne sera affichée ; la sélection `ty` montrera toutes les correspondances `ty` ; la sélection `typ` montrera toutes les correspondances `typ` ; etc.

- Pour les recherches de mots entiers, les exemples suivants sont considérés être des mots séparés : des noms d'élément (sans crochets), les crochets de balises d'élément, les noms d'attribut, et les valeurs d'attribut sans guillemets.

Note : Correspondance de la balise de démarrage et de la balise de fin est une fonction séparée qui n'est pas touchée par le paramètre *Activer le marquage automatique*.

Aperçu du document dans la barre de défilement

La barre de défilement fournit les fonctions suivantes :

- Elle met en rapport les tailles des éléments suivants (*voir capture d'écran ci-dessous*) : (i) le document entier (barre de défilement) ; (ii) le segment du document actuellement ouvert dans la fenêtre (barre); (iii) la sélection de texte actuelle (barre bleue clair), le cas échéant ; et (iv) l'emplacement actuel du curseur (trait de marquage du curseur).
- Elle vous permet de parcourir le document soit en : (i) glissant la barre vers le haut ou vers le bas, ou (ii) en cliquant sur les flèches **Page Haut** et **Page Bas** (entourées en vert dans la capture d'écran ci-dessous).



Veuillez noter les points suivants :

- La longueur de la barre de défilement correspond à la longueur du document entier.
- Si uniquement une partie du document entre dans la fenêtre, cette partie s'affichant dans la fenêtre correspond à la barre de défilement gris foncé (*voir capture d'écran ci-dessus*). Vous pouvez faire glisser la barre vers le haut ou le bas pour amener d'autres portions du document dans la fenêtre. La barre représente en quelque sorte la fenêtre et en déplaçant la barre, vous déplacez la fenêtre le long du document pour voir les parties dissimulées.

 © 2016-2022 Altova GmbH

- La sélection de texte actuelle est indiquée dans la barre de défilement par la barre bleu clair. La taille de la barre bleu est proportionnelle à la taille du document entier. Si la sélection de texte ne dépasse pas une ligne, la barre bleue ne sera pas visible.
- La position du curseur est indiquée par un trait de marquage bleu. Sa position relative dans la barre de défilement correspond à la position relative du curseur dans le document.

Aller à la ligne/caractère

Cette commande dans le menu **Affichage** et la barre outils Texte vous permet de vous rendre à une ligne et un caractère spécifique dans le texte du document.

4.2.5 Chercher et remplacer

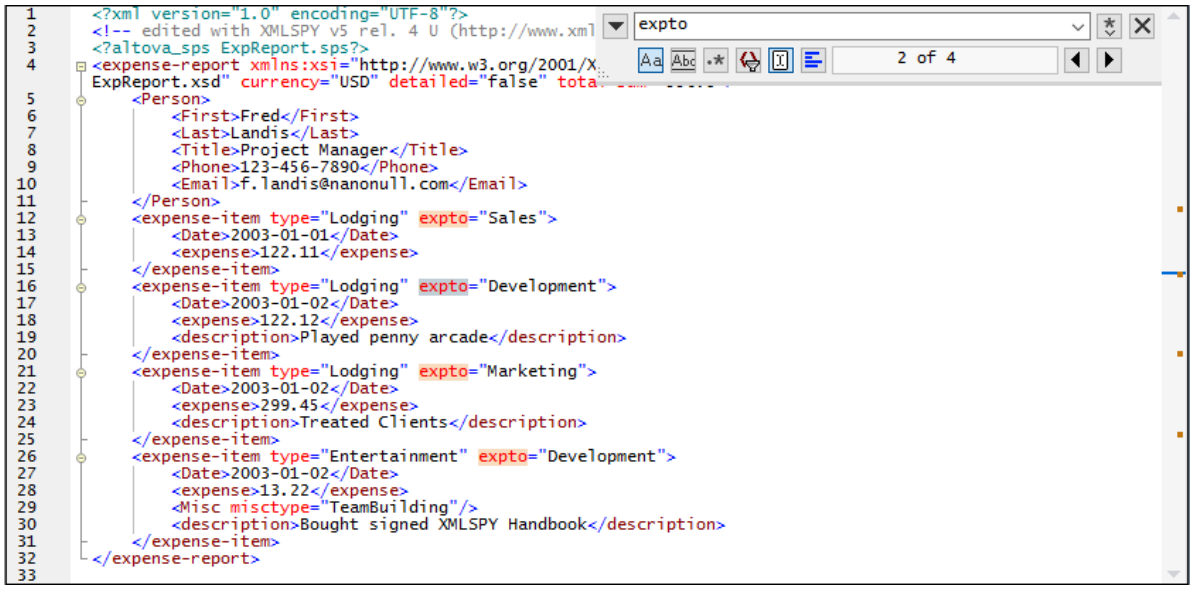
Vous trouverez dans cette section :

- Chercher un texte dans un document
- Trouver des options
- Remplacer du texte
- Utiliser des expressions régulières
- Exemples d'expression régulières

Chercher un texte dans un document

Appuyer sur **Ctrl+F** (ou sélectionner la commande de menu **Édition | Recherche**) pour afficher le dialogue Rechercher (*capture d'écran ci-dessous, coin supérieur droit*). Vous pouvez ensuite rechercher dans la totalité du document ou dans une section de texte pour trouver un terme à saisir dans le dialogue.

- Saisir une ligne de texte pour effectuer une recherche ou utiliser la liste de choix pour sélectionner une ligne de texte parmi les 10 dernières lignes de texte.
- Lorsque vous saisissez ou sélectionnez une ligne de texte à chercher, toutes les correspondances sont marquées et les positions des correspondances sont indiquées par des marqueurs beige dans la barre de défilement (*voir capture d'écran*).
- La correspondance actuellement sélectionnée a une couleur de marquage différente des autres correspondances et sa position est indiquée dans la barre de défilement par le trait de marquage bleu foncé.
- Le nombre total de correspondances est recensé sous le champ de saisie des termes, avec la position d'index de la correspondance actuellement sélectionnée (*voir capture d'écran ci-dessous*). Par exemple, `2 de 4` indique que la deuxième d'un total de quatre correspondances est actuellement sélectionnée.
- Vous pouvez passer d'une correspondance à l'autre, dans les deux sens, en appuyant sur les touches **Précédent (Shift+F3)** et **Suivant (F3)** situés en bas à droite.



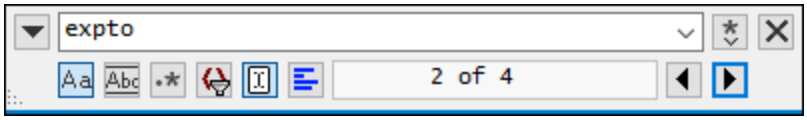
- Pour passer entre les modes Trouver et Remplacer, cliquer sur le bouton Bas situé en haut à gauche.
- Pour fermer le dialogue, cliquer sur le bouton **Fermé** situé en haut à droite.

Veuillez noter les points suivants :

- Le dialogue Chercher n'a *pas de mode*. Cela signifie qu'il peut rester ouvert pendant que vous continuez à utiliser le Mode Texte.
- Si le texte est sélectionné avant l'ouverture de la fenêtre de dialogue, le texte sélectionné sera automatiquement inséré dans le champ de recherche du terme (champ Chercher).
- Pour effectuer une recherche au sein d'une sélection, procéder comme suit : (i) marquer la sélection; (ii) activer l'option *Chercher dans la sélection* pour verrouiller la sélection ; (iii) saisir le terme de recherche. Pour effectuer une recherche dans une autre sélection, déverrouiller la sélection actuelle en désactivant l'option *Chercher dans la sélection*, puis faire la nouvelle sélection et activer l'option *Chercher dans la sélection*.
- Une fois le dialogue Chercher fermé, vous pouvez renouveler la recherche en appuyant sur **F3** pour une recherche vers l'avant, ou **Shift+F3** pour une recherche vers l'arrière. Le dialogue Chercher réapparaîtra dans ce cas.

Trouver des options

Les options pour déterminer les critères Chercher peuvent être spécifiées par le biais de boutons situés en-dessous du champ de saisie du terme (*voir capture d'écran ci-dessous*). Lorsqu'une option est activée, la couleur de sa touche change de couleur et devient bleue (*voir la première option (de casse) dans la capture d'écran ci-dessous*).



Vous pouvez sélectionner une option parmi les suivantes :

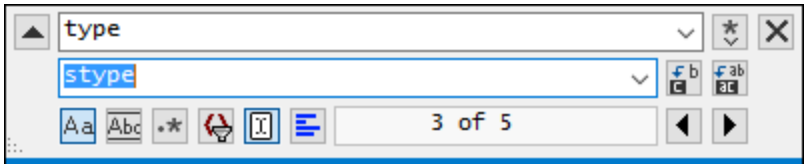
© 2016-2022 Altova GmbH

- *Respecter la casse* : Respecter la casse pendant la recherche si cette option est activée (`Address` n'est pas identique à `address`).
- *Correspondance du mot entier* : Seuls les mots exactement identiques dans le texte seront retenus. Par exemple, pour la saisie `fit`, si l'option *Correspondance du mot entier* est activée, seul le mot `fit` correspondra au terme recherché ; mais `fit` dans `fitness`, par exemple, ne sera pas retenu.
- *Expression régulière* : Si cette option est activée, le terme de recherche sera lu en tant qu'expression régulière. Voir *Expressions régulières* ci-dessous pour une description de l'utilisation des expressions régulières.
- *Filtrer les résultats :* Choisir un ou plusieurs composants de document à l'endroit où vous souhaitez effectuer la recherche.
- *Trouver ancre :* Lorsque vous saisissez un terme de recherche, les correspondances dans le document sont marquées et une de ces correspondances sera marquée en tant que la sélection actuelle. La bascule *Trouver ancre* détermine si cette première sélection actuelle est effectuée relativement à la position du curseur ou pas. Si l'option *Trouver ancre* est activée, la première correspondance actuellement sélectionnée sera la correspondance suivante en partant de l'emplacement actuel du curseur. Si l'option *Trouver ancre* est désactivée, la première correspondance actuellement sélectionnée sera la première correspondance dans le document, en partant du haut.
- *Trouver dans la sélection :* Si cette option est activée, elle verrouille la sélection de texte actuelle et limite la recherche à la sélection. Dans le cas contraire, le document entier sera parcouru. Avant de sélectionner une nouvelle série de texte, déverrouiller la sélection actuelle en désactivant l'option *Chercher dans la sélection*.

Remplacer du texte

Pour remplacer une ligne de texte, procéder comme suit :

1. Appuyer sur **Ctrl+H** (ou sélectionner la commande de menu **Édition | Remplacer**) pour ouvrir le dialogue Remplacer (*capture d'écran ci-dessous*). (En alternative, vous pouvez passer au mode Remplacer du dialogue Chercher/Remplacer en cliquant sur le bouton Flèche bas situé en haut et à gauche du dialogue.)



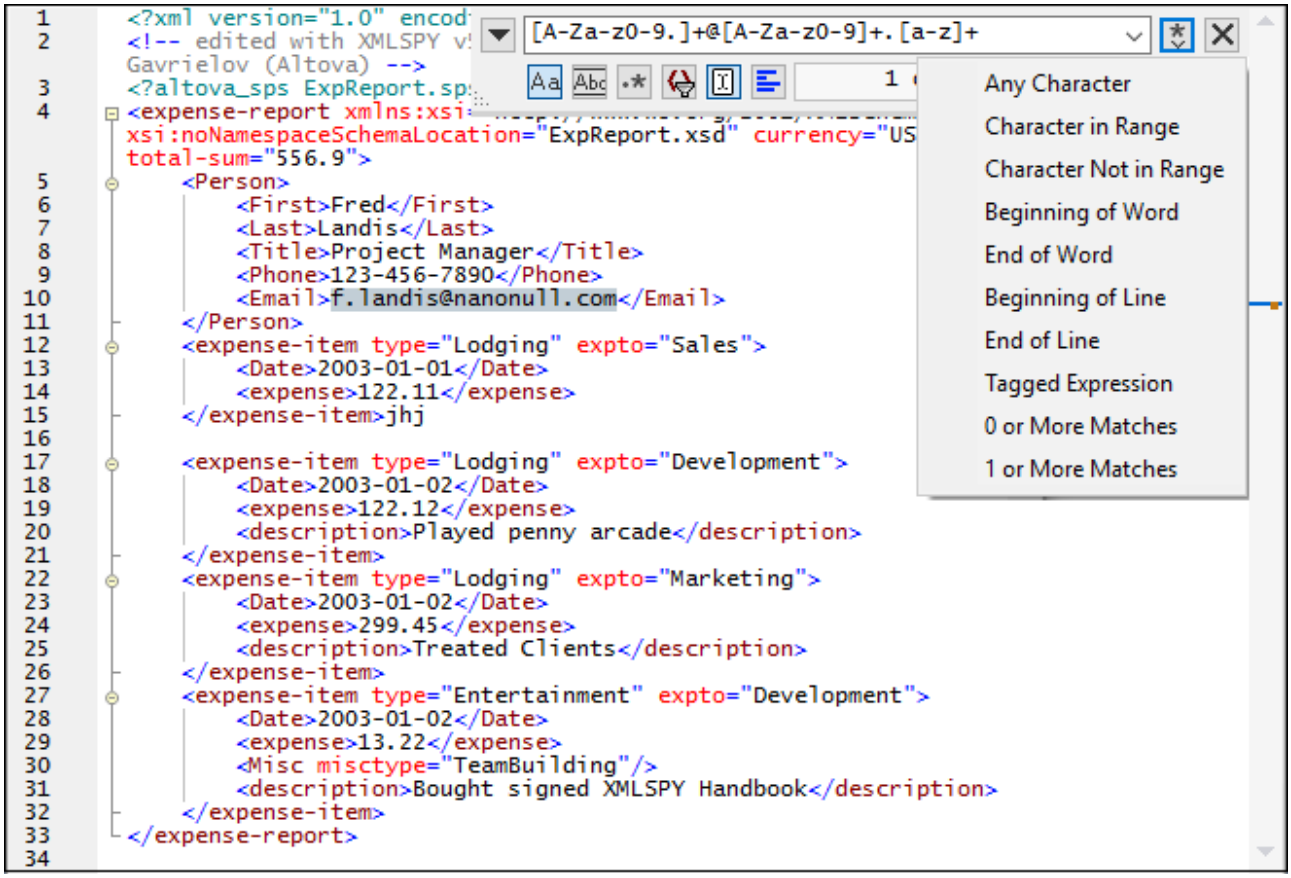
2. Saisir la ligne à remplacer dans le champ de saisie et saisir la nouvelle ligne dans le champ Remplacer. Le nombre de correspondances de texte à remplacer et l'index de la correspondance sélectionnée actuellement sont affichés sous le champ Remplacer. De même, les emplacements des correspondances sont indiqués dans la barre de défilement par des marqueurs beiges (*voir Chercher du texte dans un document ci-dessus pour plus d'informations*). Par exemple, la capture d'écran ci-dessus montre qu'il y a cinq correspondances pour le string `type`, et que la troisième de ces correspondances est sélectionnée actuellement.
3. Les boutons **Remplacer suivant** et **Remplacer tout** se trouvent à droite du champ Remplacer. Si vous cliquez sur **Remplacer suivant**, (i) lorsque le curseur est placé à côté d'une correspondance ou dans une correspondance, la correspondance sera remplacée ; (ii) si le curseur est placé en dehors d'une correspondance, il passera à la correspondance suivante ; cliquer sur **Remplacer suivant** pour remplacer cette correspondance. Cliquer sur **Remplacer tout** pour remplacer toutes les correspondances.

Veuillez noter les points suivants :

- Pour remplacer le texte dans le cadre d'une sélection, au lieu du document entier, procéder comme suit : (i) marquer la sélection ; (ii) activer l'option *Chercher dans la sélection* pour verrouiller la sélection ; (iii) saisir les lignes de texte Chercher et Remplacer ; (iii) Cliquer sur **Remplacer suivant** ou **Remplacer tout** le cas échéant. Pour remplacer le texte avec une autre sélection, déverrouiller la sélection actuelle en désactivant l'option *Chercher dans la sélection*, puis effectuer la nouvelle sélection et activer l'option *Chercher dans la sélection* pour verrouiller la nouvelle sélection.
- Pour annuler une action de remplacement, appuyer sur **Ctrl+Z** ou sélectionner **Édition | Annuler**.

Utiliser des expressions régulières

Vous pouvez utiliser des expressions régulières (regex) pour trouver une ligne de texte. Pour ce faire, veuillez tout d'abord activer l'option *Expression régulière* (voir *Trouver des options* ci-dessus). Cela spécifie que le texte se trouvant dans le champ de saisie doit être évalué en tant qu'expression régulière. Ensuite, saisir l'expression régulière dans le champ de saisie. Pour obtenir de l'aide en ce qui concerne la construction d'une expression régulière, cliquer sur le bouton **Générateur d'expression régulière**, qui est situé à droite du champ de saisie (*voir capture d'écran ci-dessous*). Cliquer sur un élément dans le Générateur pour saisir le/s métacaractère/s regex correspondant dans le champ de saisie. La capture d'écran ci-dessous affiche une expression régulière simple pour trouver des adresses e-mail. Pour consulter une brève description des métacaractères, voir la section *Métacaractères d'expression régulière* ci-dessous.



Métacaractères d'expression régulière

Ci-dessous, vous trouverez une liste des métacaractères d'expression régulière.

© 2016-2022 Altova GmbH

.	Correspond à n'importe quel caractère. Il s'agit d'un espace réservé pour un seul caractère.
(	Marque le début d'une expression balisée.
)	Marque la fin d'une expression balisée.
(abc**)**	Les métacaractères **(** et **)** marquent le début et la fin d'une expression balisée. Les expressions balisées peuvent être utiles lorsque vous devez baliser (« vous souvenir») une région correspondante afin de pouvoir vous y référer ultérieurement (back-reference). Vous pouvez baliser jusqu'à neuf expressions (et puis les back-referencer plus tard, dans le champ Chercher ou Remplacer) Par exemple, **(the) \1** correspond au string the the. Cette expression peut être expliquée littéralement comme suit : correspondre au string "the" (et s'en souvenir en tant que région balisée), suivi d'un espace, suivi d'une back-reference à la région balisée correspondante précédemment
\n	Où n est une **variable** qui peut prendre des valeurs d'entier de 1 à 9. L'expression se réfère à la première jusqu'à la neuvième région balisée lors du remplacement. Par exemple, si le string de recherche est Fred**([1-9])**XXX et que le string de remplacement est Sam**\1**YYY, cela signifie que dans le string de recherche, il y a une expression balisée qui est (implicitement) indexée avec le numéro **1** ; dans le string de remplacement, l'expression balisée est référencée avec **\1**. Si la commande chercher-remplacer est appliquée à Fred2XXX, cela génèrera Sam2YYY.
\<	Correspond au début d'un mot.
\>	Correspond à la fin d'un mot.
\x	Vous permet d'utiliser un caractère x, qui aurait généralement une signification particulière. Par exemple, \[serait interprété comme [et non pas comme le début d'un ensemble de caractères.
[...]	Indique un *ensemble de caractères*. Par exemple, [abc] signifie un des caractères a, b ou c. Vous pouvez aussi utiliser des séries : par exemple [a-z] pour n'importe quel caractère en minuscule.
[^...]	Le complément des caractères dans l'ensemble. Par exemple, [^A-Za-z] signifie importe quel caractère sauf un caractère alphabétique.
^	Correspond au début d'une ligne (sauf si utilisé dans un ensemble, *voir ci-dessus*).
$	Correspond à la fin d'une ligne. Par exemple : A+$ pour trouver un ou plusieurs "A" à la fin de la ligne.
*	Correspond 0 ou plus. Par exemple, Sa*m correspond à Sm, Sam, Saam, Saaam etc.
+	Correspond 1 ou plus. Par exemple, Sa+m correspond à Sam, Saam, Saaam etc.

Représentation de caractères spéciaux
Veuillez noter les expressions suivantes.

\r	Retour de chariot (CR). Vous pouvez utiliser CR (**\r**) ou LF (**\n**) pour trouver ou créer une nouvelle ligne
\n	Saut de ligne (LF). Vous pouvez utiliser CR (**\r**) ou LF (**\n**) pour trouver ou créer une nouvelle ligne

\t	Caractère de tabulation
\\	Utiliser ce signe pour échapper des caractères qui apparaissent dans l'expression regex, par exemple : \\\n

Exemples d'expression régulière

Cet exemple montre comment chercher et remplacer du texte à l'aide d'expressions régulières. Dans la plupart des cas, la recherche et le remplacement de texte est une opération simple ne nécessitant pas l'utilisation d'expressions régulières. Néanmoins, dans certaines instances, il se peut que vous ayez à manipuler des textes qui ne permettent pas des opérations de recherche et de remplacement. Par exemple, vous avez un fichier XML de plusieurs milliers de lignes et que vous devez renommer certains éléments dans une opération, sans toucher au contenu qui se trouve à l'intérieur. Autre exemple : vous souhaitez changer l'ordre de plusieurs attributs d'un élément. Vous pourrez recourir aux expressions régulières, en éliminant une grande partie du travail que vous auriez dû effectuer manuellement normalement.

Exemple 1 : renommer les éléments

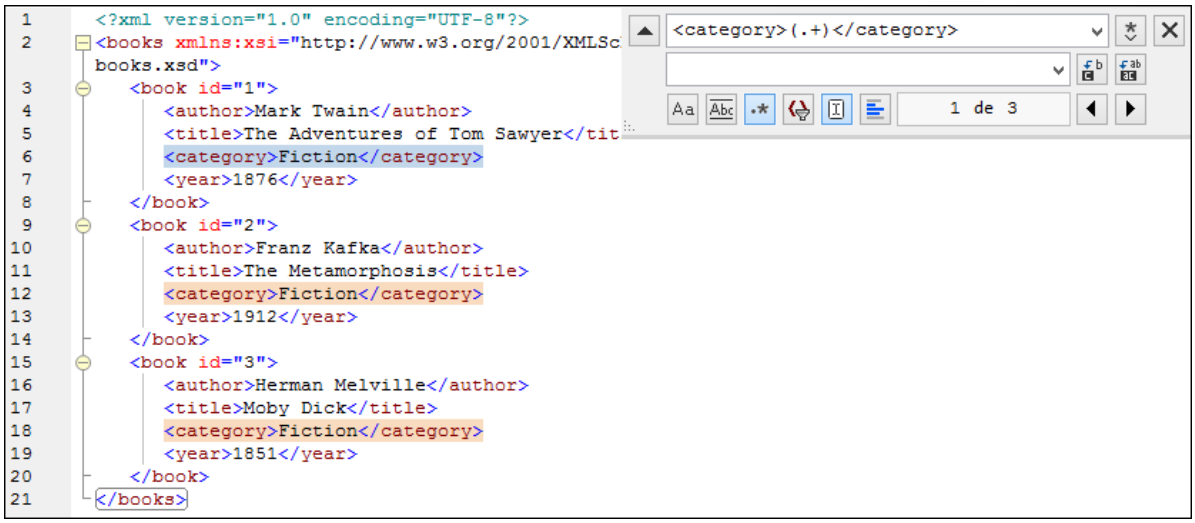
L'échantillon de code ci-dessous contient une liste de livres. Si vous souhaitez remplacer l'élément `<Category>` de chaque livre en `<Genre>`, une des possibilités est d'utiliser des expressions régulières.

```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
   <book id="1">
      <author>Mark Twain</author>
      <title>The Adventures of Tom Sawyer</title>
      <category>Fiction</category>
      <year>1876</year>
   </book>
   <book id="2">
      <author>Franz Kafka</author>
      <title>The Metamorphosis</title>
      <category>Fiction</category>
      <year>1912</year>
   </book>
   <book id="3">
      <author>Herman Melville</author>
      <title>Moby Dick</title>
      <category>Fiction</category>
      <year>1851</year>
   </book>
</books>
```

Pour obtenir le résultat souhaité, suivre les étapes suivantes :

1. Appuyer sur **Ctrl+H** pour ouvrir la fenêtre Chercher et remplacer.
2. Cliquer sur **Utiliser expression régulière** .
3. Dans le champ Chercher, saisir le texte suivant : `<category>\(.+\)</category>` . Cette expression régulière correspond à tous les éléments `category`, et ceux-ci sont marqués.

© 2016-2022 Altova GmbH



Afin de correspondre au texte intérieur de chaque élément (qui n'est pas connu à l'avance), nous utilisons l'expression balisée `\(.+\)` . L'expression `\(.+\)` signifie "correspond avec une ou plusieurs occurrences d'un caractère, dans ce cas `.+` , et se souvenir de cette correspondance". Comme indiqué dans l'étape suivante, nous aurons besoin de la référence à l'expression balisée ultérieurement.

4. Dans le champ Remplacer, saisir le texte suivant : `<genre>\1</genre>` . Cette expression régulière définit le texte de remplacement. Veuillez noter qu'elle utilise une back-reference `\1` à l'expression balisée précédemment du champ Chercher. Dans d'autre mots, `\1` dans ce contexte signifie "le texte intérieur de l'élément actuellement correspondant <category>".
5. Cliquer **Remplacer tout** et observer les résultats. Tous les éléments `category` sont maintenant renommés en `genre`.

Exemple 2 : changer l'ordre des attributs

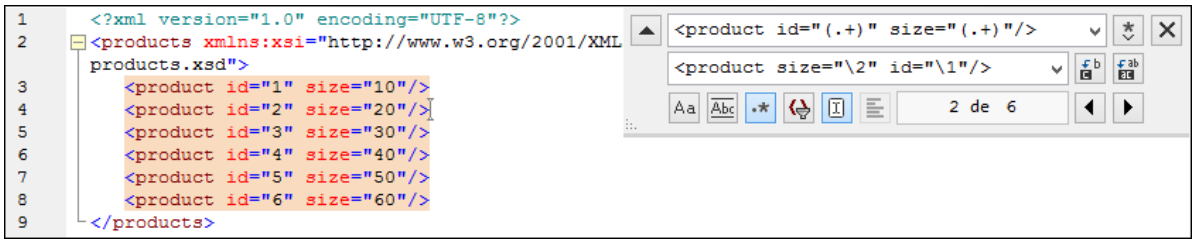
L'échantillon de code XML ci-dessous contient une liste de produits. Chaque élément de produit contient deux attributs : `id` et un `size`. Si votre intention est de changer l'ordre des attributs `id` et `size` dans chaque élément `product` (autrement dit, l'attribut `size` devrait apparaître avant `id`). Un des moyen de régler ce problème est d'utiliser des expressions régulières.

```
<?xml version="1.0" encoding="UTF-8"?>
<products xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="products.xsd">
   <product id="1" size="10"/>
   <product id="2" size="20"/>
   <product id="3" size="30"/>
   <product id="4" size="40"/>
   <product id="5" size="50"/>
   <product id="6" size="60"/>
</products>
```

Pour obtenir le résultat souhaité, suivre les étapes suivantes :

1. Appuyer sur **Ctrl+H** pour ouvrir la fenêtre Chercher et remplacer.

2. Cliquer sur **Utiliser expression régulière** .*.
3. Dans le champ Chercher, saisir le texte suivant : `<product id="\(.+\)" size="\(.+\)"/>` . Cette expression régulière correspond à un élément de produit dans le document XML. Veuillez noter que, pour correspondre à la valeur de chaque attribut (qui n'est pas connu à l'avance), une expression balisée `\(.+\)` est utilisée deux fois. Cette expression `\(.+\)` correspond à la valeur de chaque attribut (présumé avoir une ou plusieurs occurrences de tout caractère, donc `.+`).
4. Dans le champ Remplacer, saisir le texte suivant : `<product size="\2" id="\1"/>` . Cette expression régulière contient le texte de remplacement pour chaque élément de produit correspondant. Veuillez noter qu'elle utilise deux références `\1` et `\2` . Celles-ci correspond aux expressions balisées du champ Chercher. En d'autres termes, `\1` signifie "la valeur de l'attribut `id`" et `\2` signifie "la valeur de l'attribut `size`".

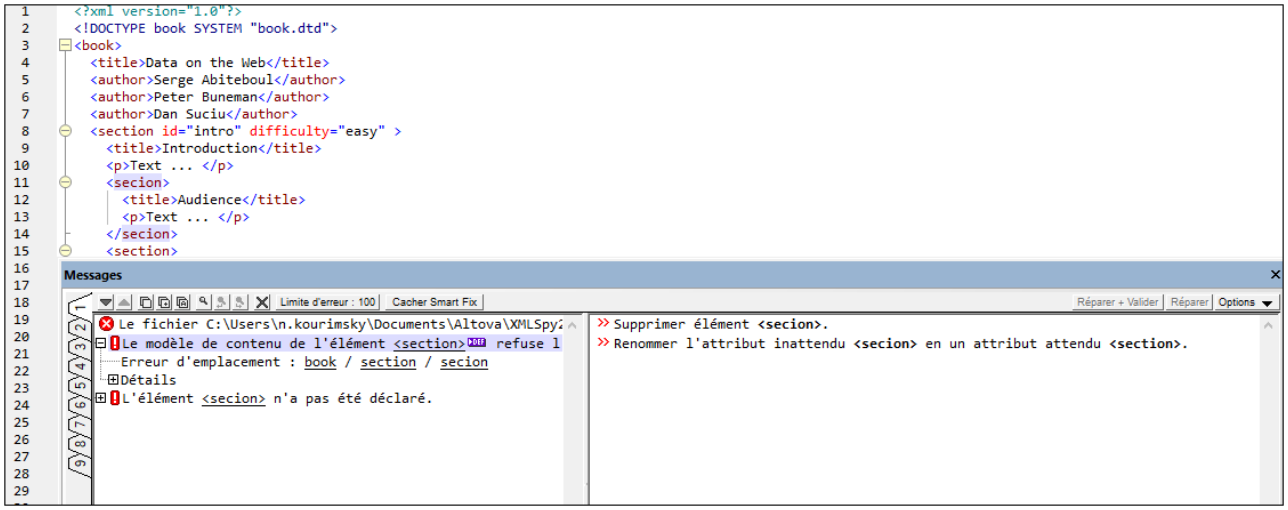


6. Cliquer **Remplacer tout** et observer les résultats. Tous les éléments `product` ont été mis à jour de manière à ce que l'attribut `size` apparaisse avant l'attribut `id`.

4.2.6 Valider un document XML

La commande **XML** | **Valider (F8)** valide un document XML par rapport a un DTD, Schéma XML ou un autre schéma associé. Si un document est valide, un message de validation réussi est affiché dans la fenêtre Messages. Dans le cas contraire, les causes de l'erreur sont affichées dans le panneau de gauche (*voir capture d'écran ci-dessous*). Si une cause est sélectionnée dans le panneau de gauche, des smart fixes, si disponibles, sont affichés dans le panneau de droite. Les suggestions de smart fix sont basées sur des information dans le schéma associé. Pour appliquer un smart fix, soit le (i) double-cliquer, soit (ii) le sélectionner et cliquer soit sur l'option **Fix** ou **Fix + Valider** (*voir capture d'écran ci-dessous*).

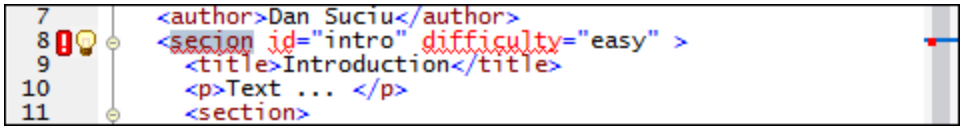
© 2016-2022 Altova GmbH



Indicateurs d'erreur et smart fixes

Indicateurs d'erreur

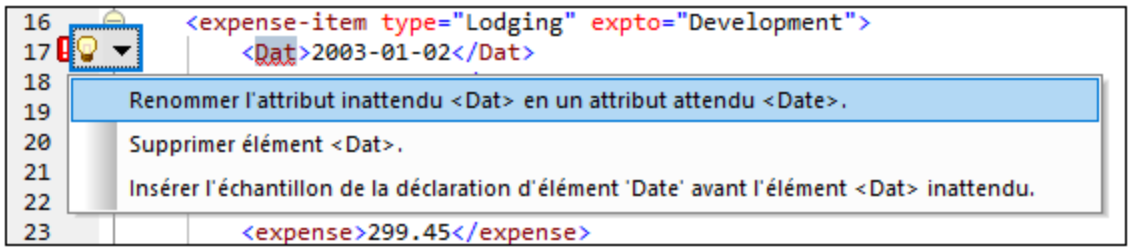
Dans le Mode Texte, vous trouverez deux indicateurs supplémentaires d'une erreur de validation (*voir capture d'écran ci-dessous*): (i) un point d'exclamation rouge dans la marge numérotée et (ii) un carré de marquage rouge dans la barre de défilement (à la droite de la fenêtre).



Veuillez noter que le carré rouge apparaît du côté gauche de la barre de défilement (située du côté droit de la fenêtre ; *voir capture d'écran ci-dessus*). Nous mentionnons ce fait ici parce que la barre de défilement affiche deux autres types de carrés de marquage: (i) pour les occurrences de texte marqué (brun, côté gauche de la barre de défilement ; *voir Parcourir le document*) ; (ii) Trouver des occurrences (brun, côté droit de la barre de défilement ; *voir Chercher et remplacer*).

Smart fixes

Si un smart fix est disponible pour une erreur, une ampoule s'affiche sur la ligne qui génère l'erreur (*voir capture d'écran ci-dessous*). Lorsque vous placez la souris sur l'icône, une fenêtre popup apparaît qui recense les smart fixes disponibles (*voir capture d'écran ci-dessous*). Sélectionner un fix pour l'appliquer immédiatement.



Veuillez noter :

- Les indicateurs d'erreur de validation et les smart fixes sont disponibles pour les types de document qui peuvent être validés dans XMLSpy, par exemple des documents JSON.
- Les indicateurs d'erreur de validation et les smart fixes décrits ci-dessus sont réinitialisés uniquement lorsque la commande **XML | Valider (F8)** est exécutée ; ils ne sont pas mis à jour en arrière-plan. Donc, si vous corrigez une erreur, vous devrez exécuter la commande **Valider (F8)** à nouveau pour vous assurer que l'erreur a bien été réparée.

Pour plus d'informations concernant la validation d'un document XML, voir la description de la commande **Valider**.

Valider sur Édition

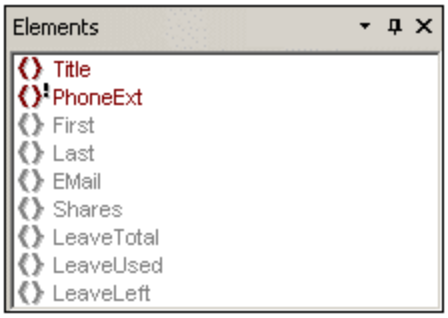
Le mode *Valider sur Édition* est activé par défaut. Lorsqu'il est activé, des vérifications de bonne formation et de validation sont effectués pendant que vous modifiez un document dans le Mode Texte (ainsi que dans le mode JSON Grid). Pour que la validation soit effectuée (en plus des contrôles de bonne formation), un DTD ou un schéma XML doit être attribué au document XML (un schéma JSON doit être attribué à un document JSON). Les erreurs sont indiquées par des indicateurs d'erreur (voir ci-dessus) dans la marge de gauche et sur les lignes contenant les erreurs.

Le mode *Valider sur Édition* peut être activé ou désactivé soit (i) via la commande de menu **XML | Valider sur Édition**, (ii) via la touche de la barre d'outils **Valider sur Édition**, ou (iii) via l'option *Sur Édition* des Paramètres de validation du dialogue Options.

4.2.7 Assistants de saisie dans le Mode Texte

Le genre d'assistant à la saisie disponible dans le Mode Texte dépend du type de document édité. Une liste d'assistants de saisie est indiquée ci-dessous pour les types de documents les plus communs. L'utilisation générale des assistants à la saisie est décrite ci-dessous. Les fonctions supplémentaires pour les types de document spécifiques, le cas échéant, sont décrites dans les sections décrivant les types de documents respectifs.

- *XML:* Éléments (*capture d'écran ci-dessous*), Attributs, Entités



- *HTML :* Éléments, Attributs, Entités
- *CSS :* Outline CSS, Propriétés CSS , Éléments HTML
- *DTD :* Aucun
- *XQuery :* Mots-clés XQuery, Variables XQuery, Fonctions XQuery
- *WSDL :* Aperçu, Détails
- *Texte :* Entités

 © 2016-2022 Altova GmbH

Veuillez noter que plusieurs types de document, comme XSD, XSLT, XHTML, et RDF, sont principalement des documents XML et auront donc des assistants à la saisie Éléments, Attributs et Entités.

Affichage et utilisation des items d'assistants à la saisie

Les différents items dans les assistants à la saisie variés présentent des codes de couleurs différents. Ces codes de couleur sont expliqués dans la documentation des assistants à la saisie pour chaque type de document. Veuillez noter les points suivants concernant les assistants à la saisie :

- Les assistants à la saisie sont sensibles au contexte et affichent des items qui peuvent être insérés à cet endroit.
- Si l'item a déjà été inséré dans l'emplacement sélectionné (ou dans tout autre emplacement équivalent et valide) et ne peut pas être réinséré à cet endroit (par exemple, un attribut XML), il sera affiché en gris.
- Si l'item est obligatoire, une icône de point d'exclamation sera affichée à côté.
- Pour insérer un item d'assistant à la saisie au niveau du point de sélection du curseur dans le texte, double-cliquer l'item d'assistant à la saisie.
- Si un élément est inséré par le biais de l'assistant à la saisie Éléments, ses balises de début et de fin sont insérées dans le texte du document. Les éléments obligatoires sont aussi insérés si cette option a été spécifiée dans le dialogue **Options** (**Outils | Options | Édition**).
- Lorsqu'un attribut est inséré par le biais d'un assistant à la saisie Attributs, l'attribut est inséré au niveau du point du curseur avec un signe "égal à" et des guillemets pour délimiter la valeur de l'attribut. Le curseur est placé entre les guillemets, de manière à ce que vous puissiez commencer à saisir directement la valeur d'attribut.

Note : En ce qui concerne les fichiers volumineux, la saisie automatique et les assistants à la saisie peuvent être désactivés, permettant ainsi un chargement et une édition plus rapide. La taille seuil du fichier est spécifiée par l'utilisateur. Pour plus de détails, voir l'option de référence, Options | Édition.

4.2.8 Mode Écran divisé

Le Mode Écran divisé permet de partager la fenêtre principale du Mode Texte en deux et affiche le document actif dans les deux modes. Vous pouvez parcourir chaque écran séparément, ce qui vous permet de voir des parties différentes du document côte à côte. Néanmoins, les modifications d'édition sont effectuées dans le document sous-jacent et sont donc visibles dans les deux modes.

Passer entre le Mode Écran divisé et le Mode Écran unique

Pour créer des modes divisés du **document actif**, procéder comme suit :

- *Division horizontale :* Faire glisser l'icône de la division horizontale située en haut à droite (encerclée en vert dans la capture d'écran ci-dessous).
- *Division verticale:* Faire glisser l'icône de la division verticale située en bas à gauche (encerclée en bleu dans la capture d'écran ci-dessous) vers la droite.



Pour retourner au Mode Écran unique, procéder comme suit :

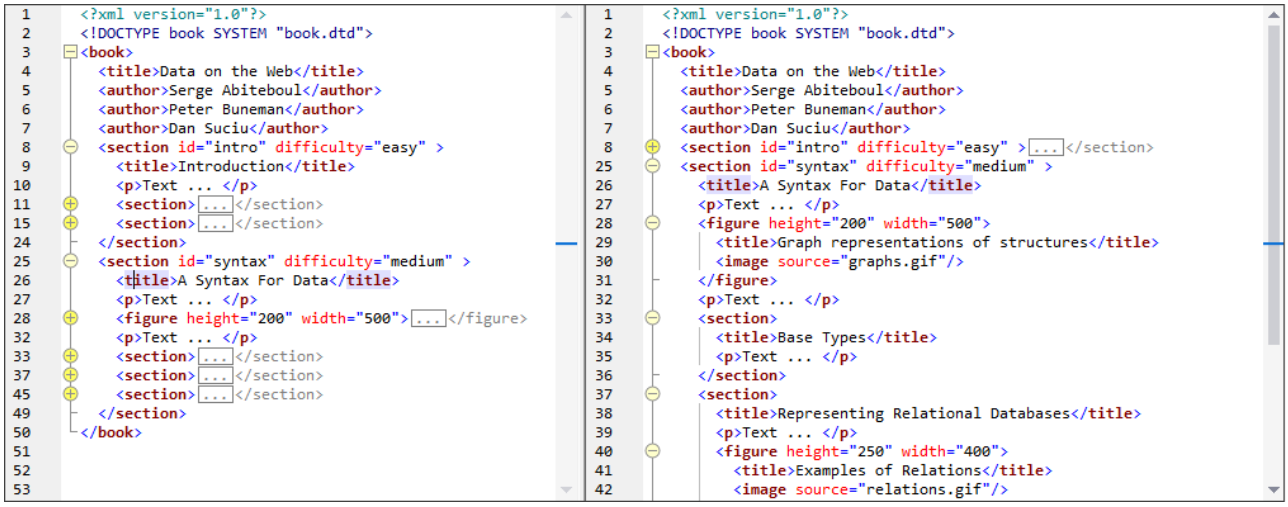
- Double-cliquer sur la barre de division ou
- Déplacer la barre de division vers un des coins parallèles de la fenêtre principale.

Note : Un Mode Écran divisé est créé pour chaque document individuellement.

Parcourir et éditer dans le mode Écran divisé

L'avantage principal du travail avec le Mode Écran divisé est que vous pouvez consulter des parties différentes d'un long document côte à côte, tout en pouvant éditer le document dans les deux affichages. La capture d'écran ci-dessous montre un document dans un Mode Écran divisé vertical.

© 2016-2022 Altova GmbH



Veuillez noter :

- Toutes les fonctions d'affichage (comme la numérotation de ligne et le pliage de source), fonctions d'édition, fonctions de navigation, etc. qui sont disponibles dans l'affichage unique d'un document sont disponibles dans les deux affichages du Mode Écran divisé *(capture d'écran ci-dessus)*.
- Dans chaque affichage du Mode Écran divisé, vous pouvez faire défiler et naviguer séparément.
- Vous pouvez utiliser séparément le pliage de source dans chaque affichage.
- Toutes les actions d'édition du document, y compris les actions de l'assistant à la saisie, sont réfléchies dans les deux affichages.

4.2.9 Raccourcis Mode texte

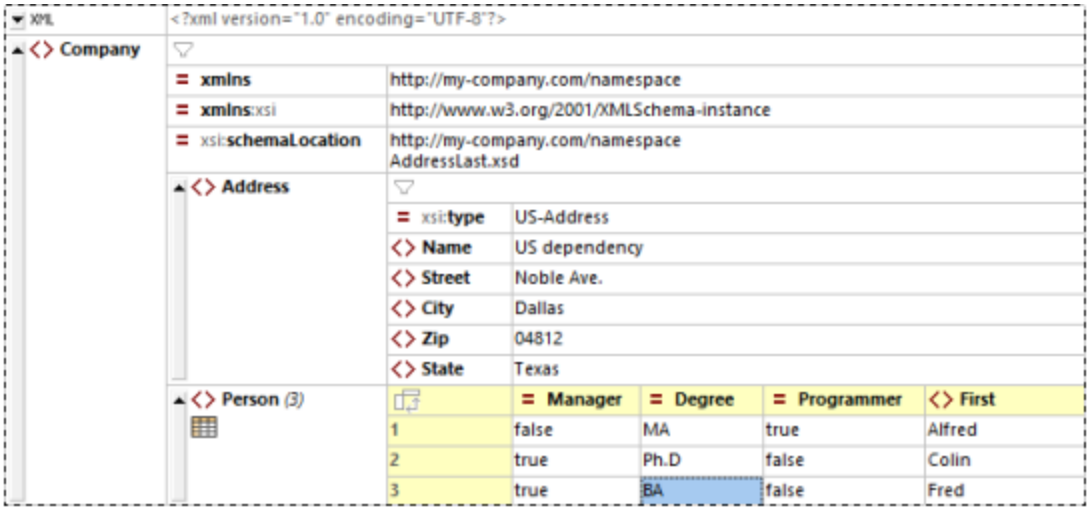
Les raccourcis par défaut des commandes du Mode Texte les plus courantes sont recensées ci-dessous. Vous pouvez changer les raccourcis par défaut dans l'*onglet Clavier du dialogue Personnaliser*.

Commandes de Mode Texte

CTRL + E	Passer entre les balises de début/fin
CTRL + Shift + E	Sélectionner l'élément qui contient le curseur
CTRL + Alt + E	Aller à l'élément parent
CTRL + "+"	Zoom avant
CTRL + "−"	Zoom arrière
CTRL + 0	Réinitialiser Zoom
CTRL + roulette de souris avant	Zoom avant
CTRL + roulette de souris arrière	Zoom arrière

4.3 Mode Grille

Le Mode Grille est disponible pour les documents XML, JSON et les DTD. (*La capture d'écran ci-dessous représente le Mode Grille d'un document XML.*) Le Mode Grille affiche la structure hiérarchique du document par un ensemble de conteneurs imbriqués. Ils peuvent être aisément élargis et réduits pour obtenir une image claire de la structure du document. Dans le Mode Grille, les contenus et la structure peuvent donc être édités très simplement.



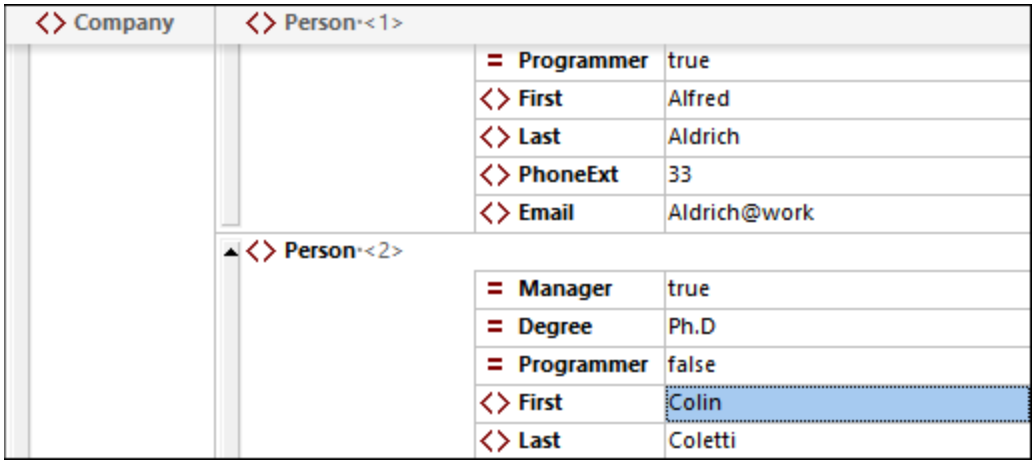
Pour savoir comment travailler dans le mode Grille, voir les sections suivantes :

- XML dans le Mode Grid
- Mode JSON Grille
- DTD

Faire défiler les en-têtes

Quand vous scrollez vers le bas dans un document XML, JSON ou DTD dans le Mode Grille et le nœud du niveau supérieur dépasse la fenêtre Vue, une barre d'en-tête apparaît en haut du Mode Grille et affiche ce nœud de niveau supérieur. Quand vous scrollez plus vers le bas, plus d'éléments ancêtre du nœud actuellement tout en haut de la fenêtre Vue seront affichés dans la barre d'en-tête. Par exemple, dans la capture d'écran ci-dessous, les ancêtres du nœud `/Company/Person[1]/@Programmer` sont affichés dans la barre d'en-tête.

© 2016-2022 Altova GmbH



Si vous cliquez sur un des nœuds dans la barre d'en-tête, alors ce nœud sera sélectionné et affiché dans la fenêtre Vue.

Personnaliser le Mode Grille

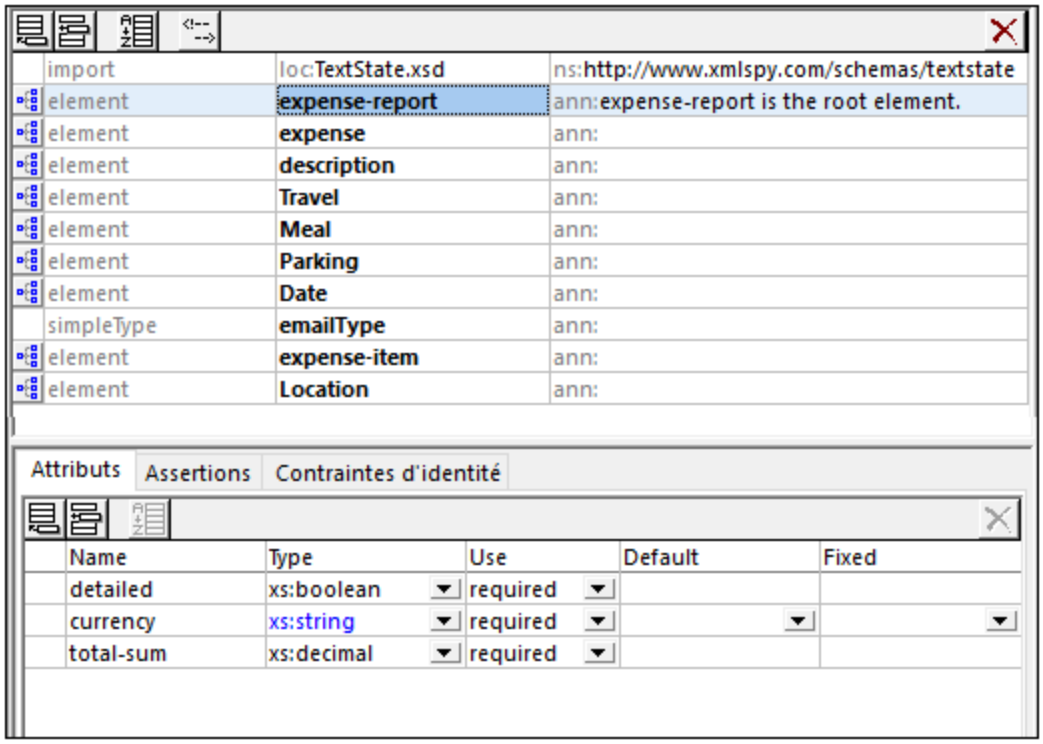
- Pour redimensionner les colonnes, placez le curseur au-dessus du rebord approprié et glissez-le de manière à obtenir la largeur désirée.
- Pour redimensionner une colonne à la largeur de son entrée la plus large, double-cliquez sur la ligne de grille située à droite de cette colonne
- Pour ajuster les largeurs de colonnes afin d'afficher tous les contenus, sélectionnez la commande de l'élément de menu Mode | Largeurs optimales, ou cliquez sur l'icône Largeurs optimales dans la barre d'outils du Mode Grille.
- Les hauteurs des cellules sont déterminées par leurs contenus. Elles peuvent être ajustées avec l'option de menu **Outils | Options | Mode | Mode Grille amélioré,** « Seuil hauteur cellule xx lignes ».

Note : si vous marquez des données dans le Mode Grille et passez au Mode Texte, les données seront aussi marquées dans le Mode Texte.

4.4 Mode Schéma

Site web Altova : Éditeur de schéma XML

Les Schémas XML peuvent être visionnés et édités graphiquement dans le Mode Schéma (*capture d'écran ci-dessous*). L'interface graphique vous permet de générer des schémas rapidement et précisément à l'aide de fonction GUI typiques. Le Mode Schéma présente deux panneaux : (i) un panneau supérieur pour la conception des relations structurelles entre les composants de schéma ; et (ii) un panneau inférieur pour les définitions liées au composant sélectionné dans le panneau supérieur. Il y a donc trois assistants à la saisie qui simplifient considérablement la création de schémas valides : les assistants à la saisie Composant, Détails et Facettes.



Panneau supérieur : conception de schéma

Le panneau supérieur du Mode Schéma peut passer entre deux modes :

- Aperçu de schéma affiche tous les composants globaux du schéma (comme les éléments globaux et les types complexes) dans une liste tabulaire simple (*voir capture d'écran*). En cliquant l'icône du composant global, vous pouvez passer au Mode de Modèle de contenu de ce composant global. Veuillez noter que tous les composants globaux ne peuvent pas forcément avoir un modèle de contenu (par exemple, des types simples).
- Mode de Modèle de contenu affiche le modèle de contenu du composant global sélectionné (*voir capture d'écran*). Pour retourner à l'Aperçu de schéma, cliquer sur l'icône Afficher Globaux dans le coin supérieur gauche du panneau supérieur.

© 2016-2022 Altova GmbH



Passer au Mode de Modèle de contenu : disponible pour les composants globaux qui ont un modèle de contenu. Ouvre le modèle de contenu du composant global dans Mode de Modèle de contenu.



Afficher globaux : disponible dans le Mode de Modèle de contenu. Passe à l'Aperçu de schéma.

Panneau inférieur: attributs, assertions et contraintes d'identité

Le panneau inférieur du Mode Schéma (*voir capture d'écran*) contient des onglets pour les définitions d'Attributs, Assertions et de Contraintes d'identité du composant sélectionné dans le design (panneau supérieur). Nous appelons ce panneau AACID en guise d'abréviation.

- Dans le mode XSD 1.0, le panneau inférieur a deux onglets : (i) Attributs, et (ii) Contraintes d'identité.
- Dans le mode XSD 1.1, le panneau inférieur a trois onglets : (i) Attributs, (ii) Assertions et (iii) Contraintes d'identité.

Le panneau AACID est toujours présent dans l'Aperçu de schéma et peut être présent dans le Mode de Modèle de contenu. Dans ce mode, les trois types de définition (attributs, assertions, CID) peuvent être affichés dans le diagramme au lieu d'être affichés dans le panneau AACID. Pour ce faire, activer les boutons respectifs de la barre d'outils du Design de Schéma : (i) **Afficher les attributs dans le diagramme**, (ii) **Afficher les assertions dans le diagramme**, et (iii) **Afficher les contraintes d'identité dans le diagramme**. En alternative, vous pouvez spécifier ces préférences dans le dialogue de Configuration de l'affichage de schéma (**Conception de schéma | Configurer affichage**). Lorsque tous les types-définitions du panneau AACID sont affichés dans le diagramme, le panneau inférieur ne sera plus visible dans le Mode Modèle de contenu.

Paramètres de schéma

Le dialogue de Paramètres de schéma (**Conception de schéma | Paramètres de Schéma**) est accessible depuis le Mode Schéma et vous permet de définir des paramètres globaux pour le schéma actif. Ces paramètres sont les attributs de l'élément `xs:schema`.

Organisation de cette section

Cette section est classée dans les sous-sections suivantes

- Mode XSD : XSD 1.0 ou 1.1: permet de choisir entre les deux modes d'édition
- Aperçu de Schéma: éditer les propriétés des composants globaux
- Mode de Modèle de contenu: éditer les modèles de contenu des composants globaux individuels
- Attributs, Assertions et Contraintes d'identité : définir ces propriétés particulières des composants
- Assistants à la saisie : utiliser ces assistants pour définir rapidement les différentes propriétés des composants
- Restrictions intelligentes: créer et éditer graphiquement des types dérivés depuis les types de base
- Utiliser xml : attributs préfixés : ajouter les attributs `base`, `id`, `lang`, et `space` graphiquement aux composants de schéma
- Retour et Avant : Se déplacer dans les positions explique une fonction de Mode Schéma qui vous permet de vous déplacer dans des positions consultées précédemment

Connexion à SchemaAgent

À partir de XMLSpy, vous pouvez aussi vous connecter à SchemaAgent pour afficher les composants provenant d'autres schémas dans la GUI et pour utiliser ces composants dans le schéma actuellement édité. Vous trouverez dans la section travailler avec SchemaAgent une description du travail avec SchemaAgent.

Trouver dans les schémas

Les fonctions Trouver dans les Schémas permettant des recherches intelligentes dans les schémas, c. à. d. des recherches qui sont limitées par des critères liés aux schémas variés. Par exemple, des recherches peuvent être limitées à certains types de composant, rendant la recherche plus efficace. La fonction Trouver dans les Schémas est décrite dans la section DTD et Schémas XML.

4.4.1 Mode XSD : XSD 1.0 ou 1.1

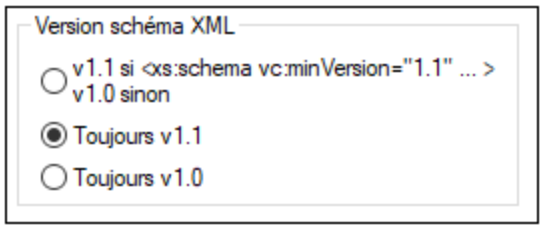
Dans le Mode Schéma, vous pouvez choisir si vous préférez et valider le Schéma XML (XSD) conformément aux spécifications de Schéma XML 1.0 (XSD 1.0) ou aux spécifications de Schéma XML 1.1 (XSD 1.1). Le **mode XSD** utilisé pour éditer un fichier est basé sur deux paramètres : un dans l'application, l'autre dans le document XSD.

Sélectionner le mode XSD

Le mode XSD détermine les fonctions d'édition et de validation du Mode Schéma (XSD 1.0 ou 1.1) disponibles pour le document actif. Vous pouvez soit : (i) procéder à un réglage sur toute l'application, dans lequel cas tous les documents XSD dans le Mode Schéma seront édités dans le mode sélectionné, ou (ii) vous pouvez enregistrer le numéro de version dans le document XSD et laisser l'application sélectionner automatiquement le mode XSD conformément à cette information.

Mode étendu à toute l'application

Le réglage étendu à toute l'application est effectuée dans la section Fichier du dialogue Options (**Outils | Options**, *voir capture d'écran ci-dessous*). Si vous sélectionnez le bouton de radio *Version 1.0* ou *Version 1.1*, le mode sélectionné devient le mode étendu à toute l'application. Tous les documents de Schéma XML ouverts dans le Mode Schéma seront maintenant édités dans ce mode. (Si vous sélectionnez le réglage *v1.1 si <xs:schema vc:minVersion="1.1", v1.0 sinon*, le mode dépendra des informations dans le document et ne sera pas applicable à toute l'application. Voir la section *Mode spécifique au document* et les autres sections ci-dessous pour plus d'informations.)



Vous pouvez passer entre les deux modes applicables à toute l'application (*Version 1.0* et *Version 1.1*) à tout moment en sélectionnant l'option que vous souhaitez dans le réglage Version schéma XML dans le dialogue Options (*capture d'écran ci-dessus*).

Note : Si le réglage actuel est un réglage applicable à toute l'application et que vous changez de mode par le biais du bouton **XSD 1.0** ou **XSD 1.1** dans la barre d'outils de Conception du schéma (*voir section suivante*), le changement de mode sera temporaire et le mode repassera au mode applicable à toute l'application lorsque le document est rechargé. Un rechargement se produira à chaque fois que le mode est modifié ou que le Mode Schéma est réinitialisé (via **Fichier | Recharger**).

© 2016-2022 Altova GmbH

Mode spécifique au document

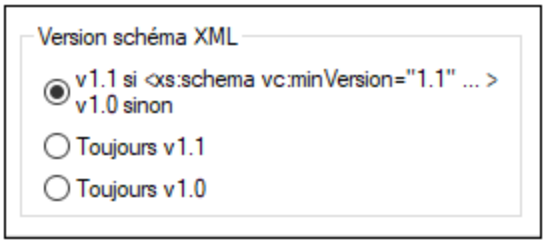
Vous pouvez aussi choisir d'enregistrer les informations du mode XSD dans le document XSD lui-même. Cela devrait permettre au Mode Schéma de passer automatiquement au mode XSD du document lorsque le document est chargé. Vous pouvez ajouter des informations de mode XSD à un document XSD en cliquant sur le bouton **XSD 1.0** ou **XSD 1.1** dans la barre outils de Conception du schéma (*capture d'écran ci-dessous*). Pour ce faire, le mode sélectionné est enregistré dans l'attribut `vc:minVersion` de l'élément `xs:schema` de niveau supérieur. (La valeur de `vc:minVersion` peut aussi être ajoutée manuellement dans le Mode Texte.)



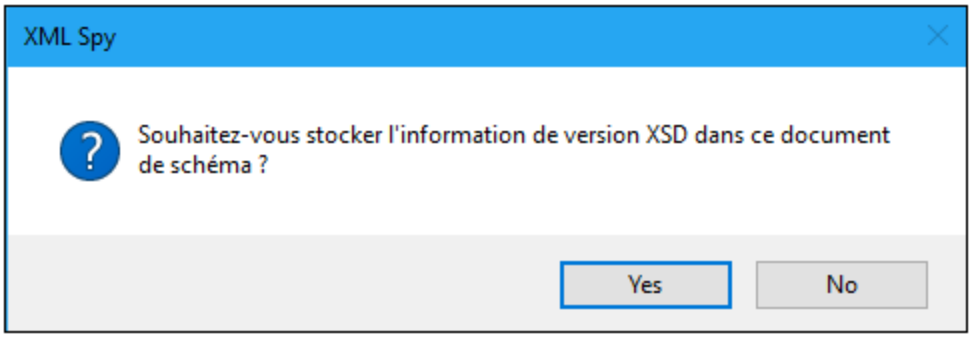
Note : L'attribut `vc:minVersion`, si présent, doit se trouver dans l'espace de noms `http://www.w3.org/2007/XMLSchema-versioning`. Dans ce cas, le document de Schéma XML doit avoir une déclaration d'espace de noms liant le préfixe d'espace de noms `vc:` à cet espace de noms. Si vous utilisez le bouton de barre d'outils **XSD 1.1** (*capture d'écran ci-dessus*), l'espace de noms est ajouté automatiquement. Cliquer sur le bouton de barre d'outils **XSD 1.0** pour supprimer cette déclaration d'espace de noms si aucun autre nom de nœud dans le document ne se trouve dans cet espace de noms.

Pour activer le mode spécifique au document et pour spécifier un mode XSD de document, procéder comme suit:

1. *Activer le mode spécifique au document :* dans la section Fichier du dialogue Options (**Outils | Options**), régler l'option de la Version de Schéma XML sur *v1.1 si <xs:schema vc:minVersion="1.1", v1.0 sinon* (*voir capture d'écran ci-dessous*). Cela indique à XMLSpy que le mode XSD dans le Mode Schéma doit être réglé conformément à l'attribut `vc:minVersion` de l'élément `xs:schema`.



2. *Spécifier la version XSD du document :* dans la barre outils de la Conception de Schéma (*capture d'écran ci-dessus*), cliquer sur le bouton **XSD 1.0** ou **XSD 1.1**. Un dialogue de confirmation (*capture d'écran ci-dessous*) s'affiche.



3. Cliquer sur **Oui** entraîne les processus suivants : (i) saisit la valeur correspondante dans l'attribut `vc:minVersion` de l'élément `xs:schema`, et (ii) si **XSD 1.1** a été sélectionné, déclare l'espace de noms `XMLSchema-versioning` avec une liaison au préfixe `vc:` namespace ; si **XSD 1.0** a été sélectionné, la déclaration d'espace de noms a été supprimée si aucun autre nœud se trouve dans l'espace de noms `XMLSchema-versioning`. Le document de Schéma XML contient à présent le numéro de version XSD. Lors de l'enregistrement du fichier, les informations du mode XSD y sont contenues. Lorsque vous rouvrez ou rechargez le fichier, le Mode Schéma passera automatiquement au mode XSD du document tel que contenu dans l'attribut `vc:minVersion` de l'élément `xs:schema`.

Note: Si l'option de mode spécifique au document est sélectionnée et si le document XSD n'a pas d'attribut `vc:minVersion` ou si la valeur de `vc:minVersion` est différente de `1.0` ou `1.1`, alors le Mode Schéma se trouvera en mode XSD 1.0 par défaut.

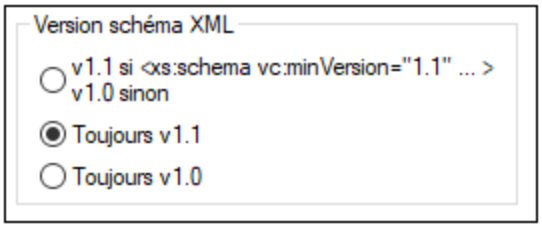
Note: Ne pas confondre l'attribut `vc:minVersion` avec l'attribut `xsd:version`. Le premier contient le numéro de version XSD alors que l'autre contient le numéro de version du document.

Mode XSD des nouveaux documents

Lorsque vous créez un nouveau document XSD, vous serez invité à choisir si vous voulez le créer en tant qu'un document XSD 1.0 ou XSD 1.1. Si XSD 1.1 est choisi, le nouveau document est créé avec l'attribut `/xs:schema/@vc:minVersion="1.1"` et l'espace de noms `XMLSchema-versioning` avec une liaison vers le préfixe d'espace de noms `vc:` est déclaré. Si XSD 1.0 est sélectionné, ni l'attribut `vc:minVersion` ni la déclaration d'espace de noms `XMLSchema-versioning` ne sont ajoutés. Néanmoins, le type de mode XSD qui est réellement activé dans le Mode Schéma dépend aussi de la Version de Schéma XML Schema sélectionnée dans la section Fichier du dialogue Options (**Outils | Options**). Voir la section suivante pour plus de détails concernant l'interaction de ces deux réglages.

Le mode XSD activé

Le mode XSD qui est activé dans le Mode Schéma dépend des deux (i) la présence/absence—et, le cas échéant, de la valeur—de l'attribut `/xs:schema/@vc:minVersion` du document XSD, et (ii) de l'option de la Version de Schéma XML sélectionnée dans la section Fichier du dialogue Options (**Outils | Options**, *capture d'écran ci-dessous*).



Les situations suivantes sont possibles. *Version de Schéma XML* dans la table ci-dessous se réfère à la sélection dans le panneau Version de Schéma XML affiché ci-dessous. Les valeurs `vc:minVersion` dans la table ci-dessous se réfèrent à la valeur de l'attribut `xs:schema/@vc:minVersion` dans le document de Schéma XML.

Version Schéma XML	**Attribut `vc:minVersion`**	**Mode XSD**
Toujours v1.0	Est absent, ou est présent avec toute valeur	1.0

 © 2016-2022 Altova GmbH

Toujours v1.1	Est absent, ou est présent avec toute valeur	1.1
Valeur de @vc:minVersion	Attribut a la valeur de `1.1`	1.1
Valeur de @vc:minVersion	Attribut est absent, ou attribut est présent avec une valeur autre que `1.1`	1.0

Note: Dans les situations décrites dans les premières deux lignes, il est possible qu'un schéma XSD 1.1 soit ouvert dans le mode XSD 1.0 et vice-versa. La description de la gestion des incohérences se trouve dans les sections ci-dessous.

Fonctions du mode XSD

Les fonctions d'interface et d'édition du Mode Schéma changeront conformément au mode XSD (XSD 1.0 ou XSD 1.1) activé actuellement.

Si le **mode XSD 1.0** est activé :

- La prise en charge de l'édition pour les nouveaux composants et propriétés de Schéma XML 1.1 n'est pas disponible. Néanmoins, si les composants ou les propriétés XSD 1.1 sont déjà présents dans le document XSD, ceux-ci seront affichés et disponibles pour une suppression.
- La validation est effectuée par rapport aux spécifications XSD 1.0. Donc si un composant ou une propriété exclusivement XSD 1.1 existe (déjà) dans le schéma, une erreur de validation est rapportée.

Si le **mode XSD 1.1** est activé, la prise en charge de l'édition est fournie pour toutes les fonctions du Schéma XML 1.1. La validation est effectuée conformément aux spécifications du Schéma XML 1.1.

Gestion des fonctions XSD 1.1 dans le mode XSD 1.0

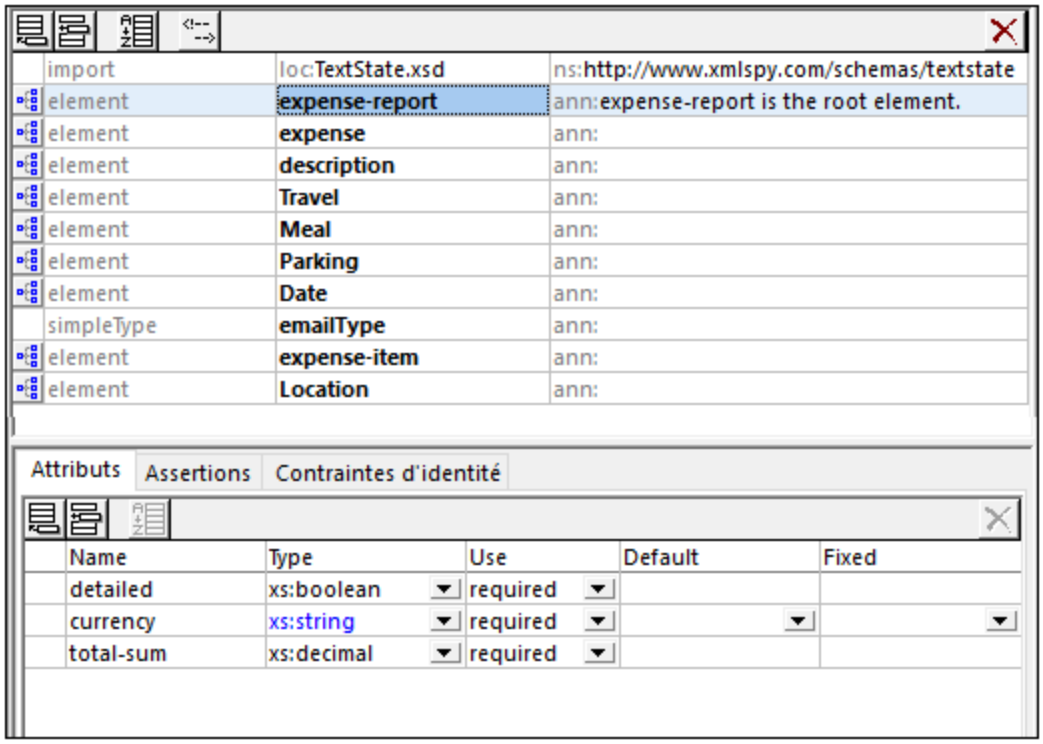
Si une fonction XSD 1.1 qui n'est pas prise en charge dans XSD 1.0 est présente dans le document (par exemple, une assertion), l'affichage et la gestion de ce genre de fonction dans le mode XSD 1.0 est décrit ci-dessous.

- ***Assertions :*** Si au moins une assertion est présente sur le type simple sélectionné, l'onglet *Assertions* est présent dans l'assistant à la saisie Facettes. Aucune édition n'est possible sauf une suppression de l'assertion.
- ***Asserts :*** L'assertion est affichée dans le diagramme du type complexe si présente. Aucun onglet d'Assertions n'est disponible dans le panneau AAIDC. L'assertion ne peut pas être ajoutée par le menu contextuel. Aucune édition des propriétés n'est possible sauf pour la suppression.
- ***Attributs :*** La nouvelle propriété `inheritable` est affichée si présente. Aucune édition n'est possible sauf pour la sélection de la valeur vide (il s'agit en fait d'une suppression des propriétés).
- ***Types complexes :*** La nouvelle propriété `defaultAttributesApply`, si présente, est affichée dans l'assistant de saisie des Détails. Aucune édition n'est possible sauf pour sélectionner la valeur vide (il s'agit en fait d'une suppression des propriétés).
- ***Documentation :*** Les nouveaux composants et propriétés spécifiques à XSD 1.1 ne sont pas inclus dans la documentation du Mode Schéma.
- ***Facettes :*** Des facettes inconnues entraînent les erreurs de validation et sont affichées en rouge.
- ***Trouver dans les schémas :*** Les nouveaux composants et propriétés spécifiques à XSD 1.1 sont ignorés.
- ***Contraintes d'identité (IDC) :*** La propriété `isRef` est affichée en cas de référence et peut être éteinte. Elle sera éteinte dès que le nom de l'IDC sera modifiée.

- ***Groupes de substitution multiples :*** Liste de choix pour sélectionner un groupe de substitution unique (seuls des groupes de substitution uniques sont autorisés dans XSD 1.0).
- ***Contenu ouvert :*** Affiché dans le diagramme si présent. Impossible d'être ajouté par menu contextuel. Aucune édition n'est possible sauf pour la suppression. Le Contenu Ouvert par Défaut n'est pas affiché dans le cadre des types complexes.
- ***Contournement :*** Affiché dans la grille des globaux si présent. Impossible d'être ajouté par le menu. Aucune édition (de l'emplacement) n'est possible sauf pour la suppression. Le contournement des composants (c'est à dire, des enfants de `xs:override`) sont ignorés et ne seront pas inclus dans l'assistant de saisie des Composants.
- ***Paramètres de schéma :*** Les nouvelles propriétés `defaultAttributes` et `xpathDefaultNamespace` sont affichées dans le dialogue si elles y sont présentes. Aucune édition n'est possible sauf pour sélectionner la valeur vide (il s'agit en fait d'une suppression des propriétés).
- ***Types simples :*** Des types inconnus entraînent des erreurs de validation et sont affichés en rouge.
- ***Alternatives de type :*** Affiché dans le diagramme si présent. Impossible d'être ajouté par menu contextuel. Aucune édition n'est possible sauf pour sélectionner la valeur vide (il s'agit en fait d'une suppression des propriétés).
- ***Caractères génériques :*** Les nouvelles propriétés sont affichées si présentes. Aucune édition n'est possible sauf pour sélectionner la valeur vide (il s'agit en fait d'une suppression des propriétés).

4.4.2 Aperçu de schéma

L'aperçu de schéma (*capture d'écran ci-dessous*) affiche une liste de tous les composants globaux du schéma (éléments `import`, éléments globaux, types complexes, etc).



Vous pouvez insérer, ajouter ou supprimer des composants globaux et vous pouvez modifier leurs propriétés. Pour ce faire, sélectionner les composants globaux dans la liste d'Aperçu de schéma. Selon le type de

© 2016-2022 Altova GmbH

composant global dont il s'agit, ses propriétés peuvent être éditées dans l'Assistant de saisie Détails, l'Assistant de saisie Facettes, et/ou le panneau Attributs/Assertions/Contrainte d'identité (AAIDC).

Un composant global qui peut avoir un modèle de contenu présente une icône **Passer au Mode de Modèle de contenu** situé à sa gauche dans la liste des composants globaux. Cliquer sur les boutons d'icône pour passer au Mode de Modèle de contenu de ce composant où le modèle de contenu de ce composant peut être édité.



Passer au Mode de Modèle de contenu: disponible pour les composants globaux qui ont un modèle de contenu. Ouvre le modèle de contenu du composant global dans Mode de Modèle de contenu.



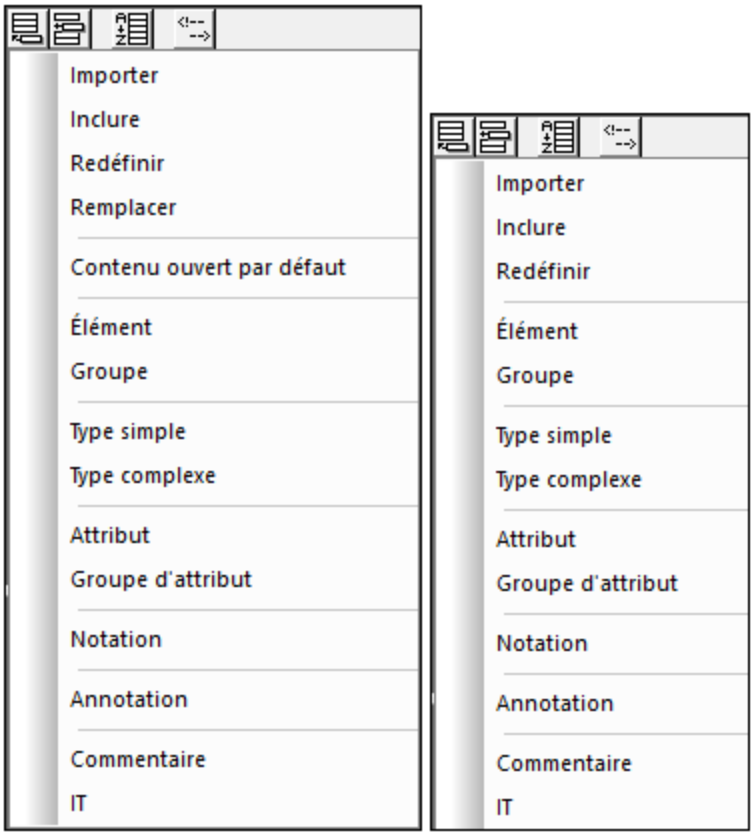
Afficher globaux : disponible dans le Mode de Modèle de contenu. Passe à l'Aperçu de schéma.

Dans cette section, nous allons tout d'abord décrire les mécanismes de GUI de l'Aperçu de Schéma puis nous allons décrire les particularités des différents composants globaux.

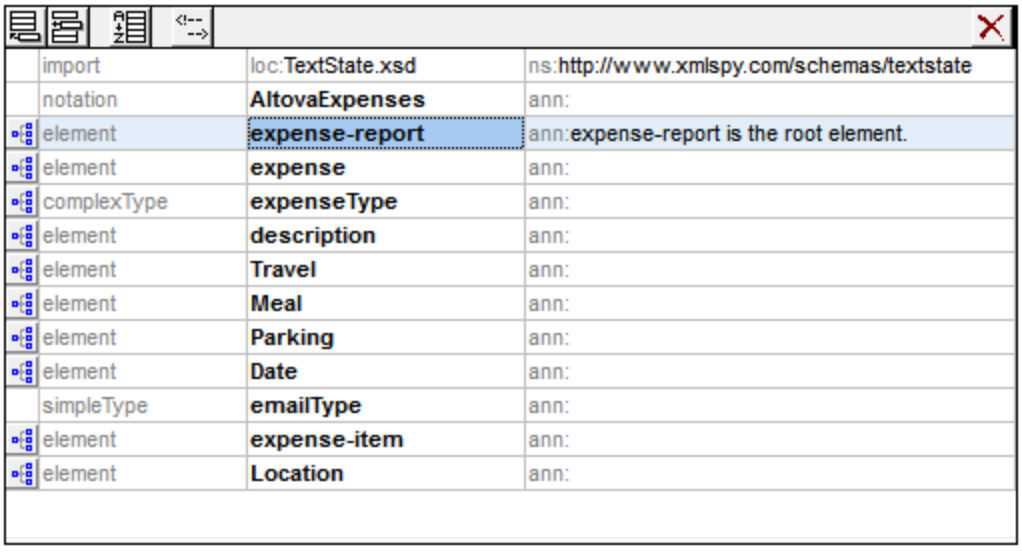
4.4.2.1 Mécanismes de la GUI

Les composants globaux sont ajoutés en tant qu'enfants du niveau supérieur de l'élément `xs:schema`. Ajouter un composant global en cliquant sur l'icône **Ajouter** ou **Insérer** situées en haut à gauche du panneau supérieur (*voir la liste des icônes ci-dessous*), puis sélectionner, à partir du menu des composants globaux (*captures d'écran ci-dessous*), le composant global que vous souhaitez ajouter.

Les captures d'écran ci-dessous montrent les composants globaux qui peuvent être ajoutés : mode XSD 1.1 à gauche, mode XSD 1.0 à droite. (*Contourner* et *contenu ouvert par défaut* sont des fonctions XSD 1.1.)



Vous pouvez ajouter autant de composants globaux que vous voulez. Tous les composants globaux dans le schéma sont affichés dans une liste tabulaire dans l'Aperçu de schéma (*capture d'écran ci-dessous*).



import	loc:TextState.xsd	ns:http://www.xmlspy.com/schemas/textstate
notation	**AltovaExpenses**	ann:
element	**expense-report**	ann:expense-report is the root element.
element	**expense**	ann:
complexType	**expenseType**	ann:
element	**description**	ann:
element	**Travel**	ann:
element	**Meal**	ann:
element	**Parking**	ann:
element	**Date**	ann:
simpleType	**emailType**	ann:
element	**expense-item**	ann:
element	**Location**	ann:

© 2016-2022 Altova GmbH

Édition dans l'Aperçu de Schéma

Veuillez noter les fonctions d'édition suivantes de l'Aperçu de Schéma:

- Vous pouvez repositionner les composants dans la liste d'Aperçu de Schéma par glisser/déposer.
- Vous pouvez naviguer en utilisant les touches fléchées et la touche **Onglet** de votre clavier.
- Vous pouvez utiliser la fonction couper/copier et coller pour copier ou déplacer les composants, attributs, assertions et contraintes d'identité globaux depuis un des diagrammes vers une position différente dans le diagramme, vers d'autres diagrammes et depuis un schéma vers un autre.
- Cliquer avec la touche de droite sur un composant pour ouvrir un menu contextuel vous permettant de couper, de copier, de coller, de supprimer ou d'éditer les données d'annotation de ce composant.
- Pour saisir une nouvelle ligne dans les commentaires globaux et les annotations globales, appuyer sur **Ctrl+Entrée**. Pour saisir un onglet, appuyer sur **Ctrl+Onglet**.

Aperçu de Schéma et les icônes d'y rapportant

	*Ajouter les composants globaux :* ajoute les composants globaux au bas de la liste des composants globaux. Si le composant doit, par définition, se produire au début du document, il est ajouté en haut de la liste.
	*Insérer le composant global :* ajoute les composants globaux au-dessus du composant sélectionné. si le composant doit, par définition, se produire au début du document, il est ajouté en haut de la liste..
	*Trier :* affiche le dialogue Trier les composants, dans lequel la précédence des critères de tri peut être définie (nom avant type, ou vice-versa), avant de poursuivre le tri. *Voir la description ci-dessous*.
	*Commentaires :* affiche un menu permettant de choisir entre un affichage multi-ligne et ligne unique des commentaires globaux. *Voir la description ci-dessous*.
	*Passer au Mode Modèle de contenu :* disponible pour les composants globaux qui ont un modèle de contenu. Ouvre le modèle de contenu du composant global dans Mode de Modèle de contenu.
	*Afficher les globaux :* disponible dans le Mode de Modèle de contenu. Passe à l'Aperçu de Schéma.

Passer entre le mode Aperçu de Schéma et Modèle de contenu

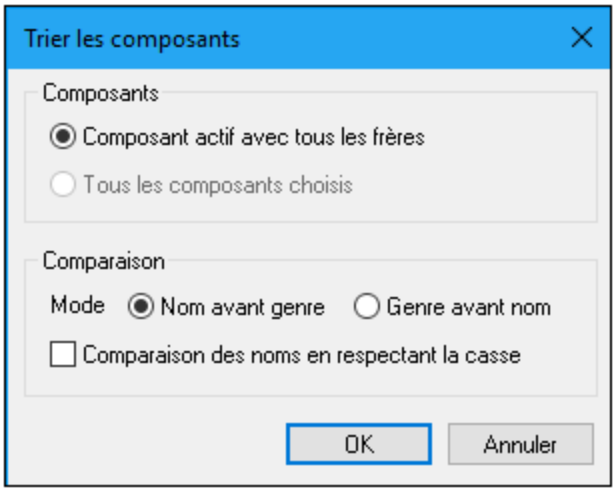
Certains composants globaux (comme des types complexes, des déclarations d'élément et des groupes de modèles) présentent une icône **Passer au Mode de Modèle de contenu** situé à gauche du nom de composant (*voir la liste des icônes ci-dessus*). Cela indique que ces composants globaux ont une modèle de contenu qui décrit la structure et les contenus du composant.

Cliquer cette icône pour faire passer le mode d'Aperçu de Schéma au Mode de Modèle de contenu de ce composant global. Les autres composants globaux (comme les annotations, les types simples et les groupes d'attributs) n'ont pas de modèle de contenu et n'ont donc pas d'icône **Passer au Mode de Modèle de contenu**. Vous pouvez retourner à l'Aperçu de Schéma depuis le Mode Modèle de contenu en cliquant sur l'icône **Afficher globaux** (*voir la liste des icônes ci-dessus*).

Trier les composants globaux

Vous pouvez trier les composants globaux en cliquant sur l'icône **Trier** dans la barre d'outils Aperçu de Schéma (*voir la liste des icônes ci-dessus*). Dans le dialogue Trier les composants qui apparaît (*capture d'écran ci-dessous*), vous pouvez choisir soit tous les composants globaux triables, soit l'ensemble de des composants sélectionnés. Vous pouvez utiliser **Shift**+clic pour sélectionner une plage et **Ctrl**+clic pour ajouter des composants supplémentaires à la sélection.

Note : Les composants globaux qui doivent se produire au démarrage du document (comme `include` et `import`) ne sont pas concernés par la fonction de tri. Ils ne font pas partie de la plage de composants globaux qui peuvent être triés.

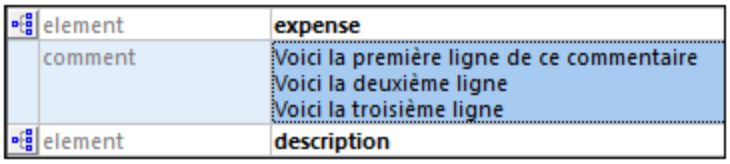


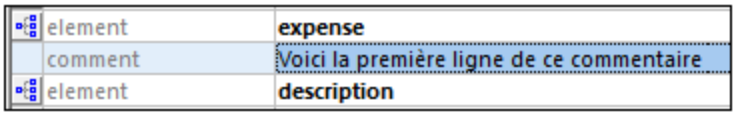
Une fois avoir défini la plage, vous pouvez choisir de trier la plage triable par ordre alphabétique (*Nom avant le type*), ou de l'organiser tout d'abord par type et puis par leur nom.

L'ordre de tri est mis en place dans le texte du schéma.

Commentaires globaux : Mode d'affichage en ligne

Les commentaires globaux peuvent être affichés dans un champ de texte multi-ligne (par défaut) ou à ligne unique (*captures d'écran ci-dessous*).





Afin de passer entre ces deux modes d'affichage de commentaires, cliquer sur l'icône **Commentaires** située en haut du panneau Aperçu de Schéma et choisir l'option que vous souhaitez. Dans le texte d'un commentaire, si vous souhaitez créer une nouvelle ligne (et donc transformer le commentaire en un commentaire multi-ligne), appuyer sur **Ctrl+Entrée**. Lorsque les commentaires sont dans le mode d'affichage de ligne simple, placer le

© 2016-2022 Altova GmbH

curseur au-dessus d'un commentaire multi-ligne pour faire apparaître une fenêtre multi-ligne qui affiche toutes les lignes.

4.4.2.2 Composants globaux

Les composants globaux sont ceux qui sont ajoutés en tant qu'enfants de l'élément de niveau supérieur `xs:schema` (contrairement aux composants locaux, qui sont créés dans le cadre d'autres composants). Certains composants globaux, comme des types complexes, des éléments et des attributs peuvent être référencés par d'autres composants dans le schéma.

Créer des composants globaux dans l'Aperçu de schéma

Les composants globaux sont généralement créés et édités dans Aperçu de schéma. Dans Aperçu de schéma, ils sont ajoutés par le biais des icônes **Ajouter** ou **Insérer**. Le modèle de contenu d'un composant global (si le composant global peut en avoir un, *voir la table ci-dessous*) est créé et édité dans le Mode de Modèle de contenu de ce composant global. (Cliquer sur l'icône **Passer au Mode de Modèle de contenu** situé à gauche du nom d'un composant pour vous rendre au Mode de Modèle de contenu.)

Certains composants globaux, une fois créés dans l'Aperçu de schéma, sont aussi ajoutés dans l'Assistant de saisie du Composants. Si un composant a un modèle de contenu, double-cliquer sur son nom dans l'Assistant de saisie de Composants ouvrira le modèle de contenu pour l'édition dans le Mode de Modèle de contenu.

Si le composant global a une définition de type (type simple ou type complexe), alors cliquer sur la commande du menu contextuel du composant **Aller à la définition de type** pour aller dans la définition de type. Dans le cas des types simples intégrés, une fenêtre de message apparaît qui contient des informations concernant le type simple.

Note : Vous pouvez aussi créer des composants globaux (éléments, attributs, types simples, types complexes et groupes de modèle) tout en réalisant l'édition dans le Mode de Modèle de contenu. Cliquer avec la touche de droite à n'importe quel endroit dans la fenêtre et sélectionner **Nouveau global | < Type de Composant global >**.

Note : Tout en éditant dans le Mode de Modèle de contenu, vous pouvez faire d'un élément local un élément global — ou un type complexe global si l'élément a un enfant élément ou attribut. Sélectionner l'élément local, cliquer avec la touche de droite à n'importe quel endroit dans la fenêtre et sélectionner **Rendre Global | Élément** ou **Rendre Global | Type complexe**.

Composant global	Emplacement dans le Schéma	Modèle de contenu
`include`	Au début	Non
`import`	Au début	Non
`redefine`	Au début	Non
`override` [1.1]	Au début	Non
`defaultOpenContent` [1.1]	Après Includes, Imports, Redefines Overrides; avant tout autre	Oui

element	N'importe où après defOpenCont	Oui
group	N'importe où après defOpenCont	Oui
simpleType	N'importe où après defOpenCont	Non
complexType	N'importe où après defOpenCont	Oui
attribute	N'importe où après defOpenCont	Non
attributeGroup	N'importe où après defOpenCont	Non
notation	N'importe où après defOpenCont	Non
annotation	N'importe où après defOpenCont	Non
Commentaire	N'importe où	Non
Instruction de traitement	N'importe où	Non

Ci-dessous, vous trouverez les points principaux concernant l'édition de ces composants dans l'Aperçu de schéma.

Includes, Imports, Redefines et Overrides

Ces quatre composants globaux permettant à d'autres documents de schéma d'être réutilisés dans le cadre du document de schéma actuel.

- Includes réutilisent des documents qui ont le même espace de nom cible que le document actuel.
- Imports réutilisent des documents qui ont d'autres espaces de nom cible que le document actuel.
- Redefines et Overrides sont des types d'Includes dans le sens qu'ils ont le même espace de nom cible que le document actuel. En revanche, ils modifient des parties des schémas inclus. Redefines sont des fonctions 1.0 et sont obsolètes dans 1.1. Overrides, qui sont une fonction de la version 1.1, sont plus flexibles et ont été conçus pour remplacer les Redefines dans 1.1.

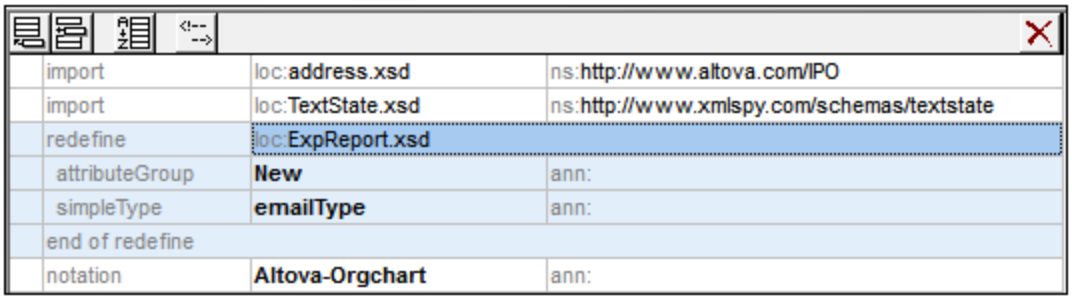
Tous les quatre ont un attribut de schemaLocation qui pointe vers le schéma à réutiliser. Dans le Mode Schéma, lorsque vous double-cliquez dans le champ loc de ces composants, vous pouvez parcourir le fichier à réutiliser et définir son chemin par rapport au document actuel. Le document import a en plus un attribut namespace qui détient l'espace de nom cible du schéma importé.

Lorsqu'un schéma est réutilisé dans le document de schéma actuel (via includes, imports, redefines ou overrides), ses composants globaux, espaces de noms et contraintes d'identité sont affichés dans l'Assistant de saisie des Composants du document actuel.

Redefines

Dans un composant redefine, vous pouvez modifier des types complexes, des types simples, des groupes de modèle et des groupes d'attribut. Le composant à redéfinir se trouvera dans le schéma spécifié dans le champ loc du composant redefine (dans la capture d'écran ci-dessous les composants à redéfinir se trouvent dans le schéma ExpReport.xsd). Une fois qu'un composant redefine a été ajouté, vous devez ajouter le composant à redéfinir dans une position entre les lignes redefine et end of redefine de la liste des composants globaux (*voir capture d'écran ci-dessous, où les composants* New *et* emailType *sont redéfinis*). Ces deux composants existent dans le schéma ExpReport.xsd et sont redéfinis pour le schéma actuel.

© 2016-2022 Altova GmbH



Pour redéfinir un composant, procéder comme suit :

1. Sélectionner la ligne `end of redefine`.
2. Cliquer l'icône **Insérer** située en haut à gauche de l'Aperçu de schéma.
3. Sélectionner le type de composant que vous souhaitez définir (types complexes, types simples, groupes de modèle ou groupe d'attribut). Le composant est ajouté dans le cadre du composant redefine.
4. Lui donner le même nom que le composant que vous souhaitez redéfinir. Le composant aura maintenant toutes les propriétés du composant depuis le schéma qui est réutilisé.
5. Redéfinir le composant en le sélectionnant et en modifiant ses propriétés dans les assistants à la saisie Détail et Facettes, ou en modifiant son modèle de contenu dans le Mode de Modèle de contenu (le cas échéant).

Note : Vous devriez pouvoir insérer les composants à redéfinir de la manière suivante : soit depuis un autre endroit dans la liste des composants globaux, soit depuis l'assistant de saisie Composants, par le biais d'un glisser/déposer ou copier/coller.

Les composants redéfinis peuvent être référencés par d'autres composants dans le schéma.

Overrides

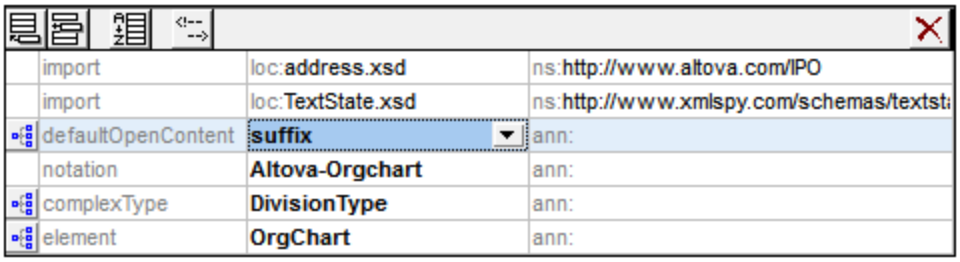
Dans un élément `override`, vous pouvez définir les composants suivants : types complexes, types simples, éléments globaux, attributs globaux, groupes de modèle, groupes d'attributs, et notations. Si, dans le cadre d'un élément `override`, un des ces composants est défini, alors ce composant remplacera, dans le schéma contourné, tous les composants du même type qui portent le même nom que le composant de contournement. Le schéma contourné est spécifié dans le champ `loc` de Override.

Les Overrides diffèrent des Redefines (*voir ci-dessus*) dans le sens qu'ils sont des composants définis à partir de zéro et non basés sur des composants réutilisés. Dans le Mode Schéma, vous ajoutez des composants pour surmonter comme si vous ajoutez des composants pour la redéfinition. Insérer le composant de contournement au-dessus de la ligne `end of override` et puis définissez ses propriétés. *Voir la section 'Redefine' ci-dessus*. La différence principale entre un Override et un Redefine est que lorsqu'un composant est ajouté à un Override, il n'est pas basé sur un composant provenant du schéma réutilisé.

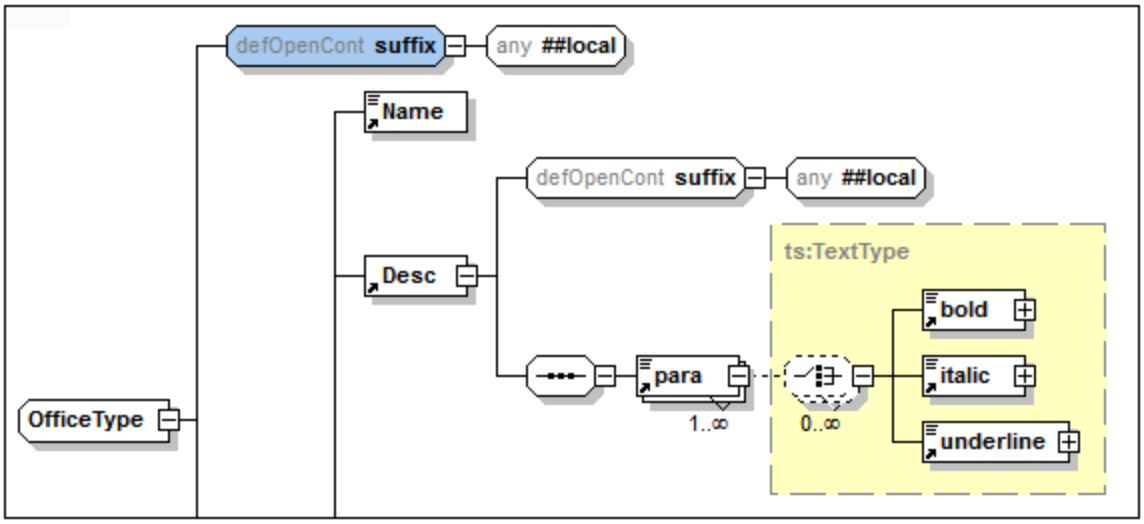
Contenu ouvert par défaut

L'élément `defaultOpenContent` est nouveau dans XSD 1.1 et spécifie qu'un ou plusieurs éléments peuvent être ajoutés à tout type complexe de contenu mélangé ou d'élément-uniquement. Il est semblable à l'élément `openContent` (donc nouveau pour XSD 1.1), la différence principale étant qu'alors que l'élément `openContent` s'applique à un seul type complexe, l'élément `defaultOpenContent` s'applique à tous les types complexes dans le schéma.

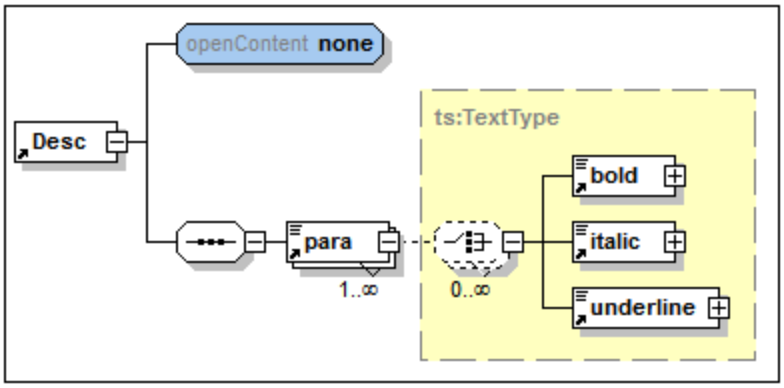
L'élément `defaultOpenContent` se produit une fois dans le document (*voir capture d'écran ci-dessous*), après Includes, Imports, Redefines et Overrides, et avant les définitions des composants. Il a un attribut `mode` qui peut prendre une valeur soit de `interleave` soit de `suffix`. Le défaut est `interleave`.



L'élément `defaultOpenContent` a un modèle de contenu que vous pouvez éditer dans le Mode de Modèle de contenu. Une fois déclaré, l'élément `defaultOpenContent` s'appliquera à tous les types complexes dans le schéma. Dans la capture d'écran ci-dessous, vous pouvez voir que `defaultOpenContent` a été appliqué automatiquement à `OfficeType` et aux types complexes `Desc`.



Afin de contourner l'élément `defaultOpenContent` lorsqu'il est appliqué à un type complexe en particulier, ajouter un élément enfant `openContent` à ce type de complexe. Dans la capture d'écran ci-dessous, l'élément `Desc` avec l'élément `defaultOpenContent` (*voir capture d'écran ci-dessus*) s'est vu rajouter un élément `openContent` qui contourne l'élément `defaultOpenContent`.



© 2016-2022 Altova GmbH

Éléments Globaux (élément)

Dans l'Aperçu de schéma, vous pouvez créer un élément global. Si celui-ci doit avoir un modèle de contenu, cela est défini dans le Mode de Modèle de contenu de l'élément global. Une fois l'élément sélectionné dans un des modes, vous pouvez définir les attributs, les assertions et les contraintes d'identité dans les onglets respectifs du panneau AACID. Les facettes et les autres propriétés peuvent être définies dans les assistants à la saisie Facettes et Détails de l'élément. Les éléments globaux peuvent ensuite être référencés par des types complexes.

Les groupes de modèle (groupe)

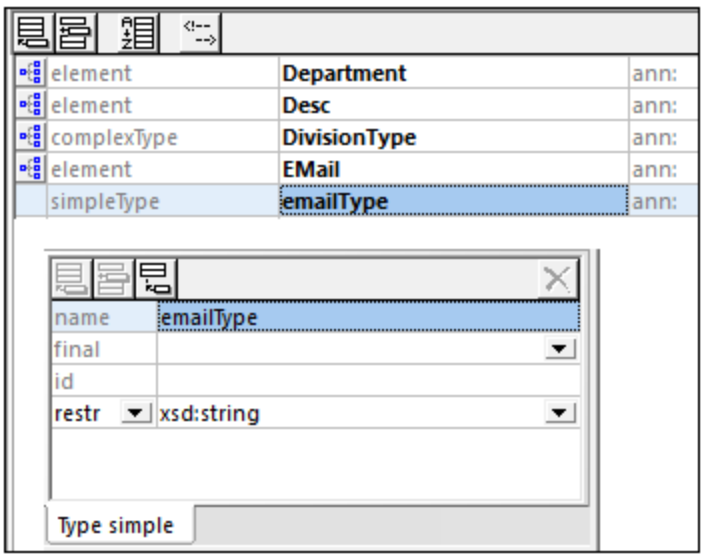
Dans l'Aperçu de schéma, vous pouvez créer des groupes de modèle nommés qui peuvent ensuite être référencés dans des types complexes. Un groupe de modèle nommé (l'élément `xs:group`) vous permet de prédéfinir un modèle de contenu qui peut être réutilisé. Il peut contenir un de trois types de groupe de modèle enfant : un groupe `sequence`, un groupe `choice`, ou un groupe `all`.

Vous créez un groupe de modèle nommé dans l'Aperçu de schéma en ajoutant un composant de groupe, en lui donnant un nom, et puis en définissant son modèle de contenu dans le Mode de Modèle de contenu. Le groupe de modèle nommé peut ensuite être ajouté au modèle de contenu d'un type complexe.

Types simples nommés (simpleType)

Dans l'Aperçu de schéma, vous pouvez créer des types simples nommés (*voir capture d'écran ci-dessous*), qui peuvent ensuite être référencés dans les déclarations élément et attribut.

Dans l'Assistant à la saisie de Détails, vous spécifiez le contenu du type simple (`restriction`, `list`, `union`) et le type correspondant : respectivement, le type de base, le type d'item et un type membre. Dans la capture d'écran ci-dessous, par exemple, le type base de la restriction du type simple est `xs:integer`. Voir la section Assistant à la saisie de Détails pour plus d'informations. Pour limiter un type simple avec des facettes, utiliser les options dans l'Assistant à la saisie de facettes.



Note : Les types anonymes peuvent être déclarés sur un élément ou un attribut de contenu simple soit dans Aperçu de schéma ou dans Mode de Modèle de contenu. Lorsque vous définissez la propriété `derivedBy` (dans l'Assistant à la saisie de Détails) sur `restriction`, `list` ou `union`, vous créez un

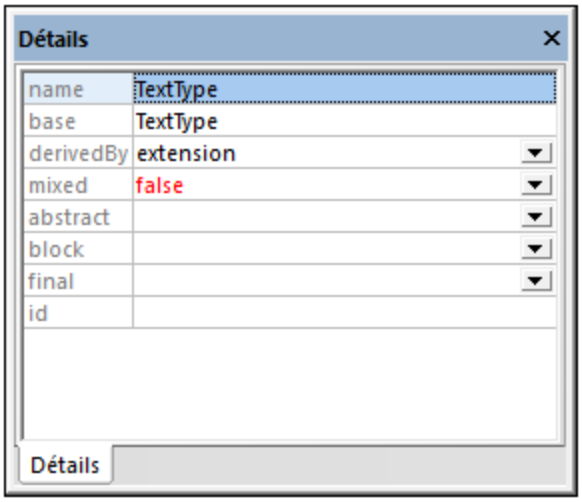
type simple anonyme dans le cadre de cette déclaration d'élément ou d'attribut. Vous pouvez définir des facettes de restriction (dans l'Assistant à la saisie de Facettes) et d'autres propriétés dans l'Assistant à la saisie de Détails.

Types complexes nommés (complexType)

Dans l'Aperçu de schéma, vous pouvez créer des types complexes nommés qui peuvent ensuite être référencés dans des déclarations d'élément. Une fois le type complexe nommé sélectionné dans un des modes, vous pouvez définir ses attributs et assertions dans les onglets respectifs du panneau AACID.

Un type complexe peut avoir quatre types de contenu (*voir liste ci-dessous*). Vous spécifiez les différents types de contenu dans l'Assistant à la saisie de Détails tel que décrit ci-dessous et, si vous le souhaitez, (et si vous y êtes autorisé), un modèle de contenu dans le Mode de Modèle de contenu.

- *Contenu simple :* définir le type de base du contenu simple (*voir capture d'écran ci-dessous*). L'attribut `mixed` (pour le contenu mixte) doit avoir une valeur de `true` (la valeur par défaut) ; C'est pourquoi la valeur `false` dans la capture d'écran ci-dessous est affichée en rouge. Aucun modèle de contenu n'est autorisé.



- *Contenu uniquement à élément :* créer des éléments enfant dans le diagramme de modèle de contenu. Il n'y aura pas de type de base.
- *Contenu mixte :* l'attribut `mixed` ne doit pas être défini sur `true`. Les données de caractère peuvent être présentes à tout endroit de l'élément parmi les nœuds élément enfants. Les données de caractère ne contiennent pas de type de données, il ne doit donc y avoir aucun type de base (*voir capture d'écran ci-dessus*). Les éléments enfants peuvent être créés dans le diagramme de modèle de contenu.
- *Contenu vide :* l'élément n'aura ni des données de caractère, ni des éléments enfants. Il ne doit pas y avoir de type base et `mixed` doit être faux. Les données dans les éléments vides de contenu sont généralement stockées dans les attributs.

Note : Les attributs et les assertions peuvent être définies (dans le panneau AACID) pour les quatre types de contenu.

Note : Les types complexes anonymes sont créés dans le cadre d'un élément en créant un modèle de contenu pour cet élément dans Mode de Modèle de contenu.

© 2016-2022 Altova GmbH

Attributs globaux et groupes d'attribut (attribut, attributeGroup)

Les attributs globaux et les groupes d'attribut sont ajoutés dans l'Aperçu de schéma.

- Les propriétés d'un **attribut global** sont définies dans l'assistant à la saisie des Détails de l'attribut.
- Après avoir créé un **groupe d'attribut global**, vous pouvez ajouter des attributs au groupe comme suit : (i) Sélectionner le groupe d'attribut global dans la liste des composants globaux ; (ii) Ajouter des attributs dans l'onglet Attributs du panneau AACID; et (iii) Définir les propriétés de chaque attribut dans l'assistant à la saisie des Détails de l'attribut sélectionné.

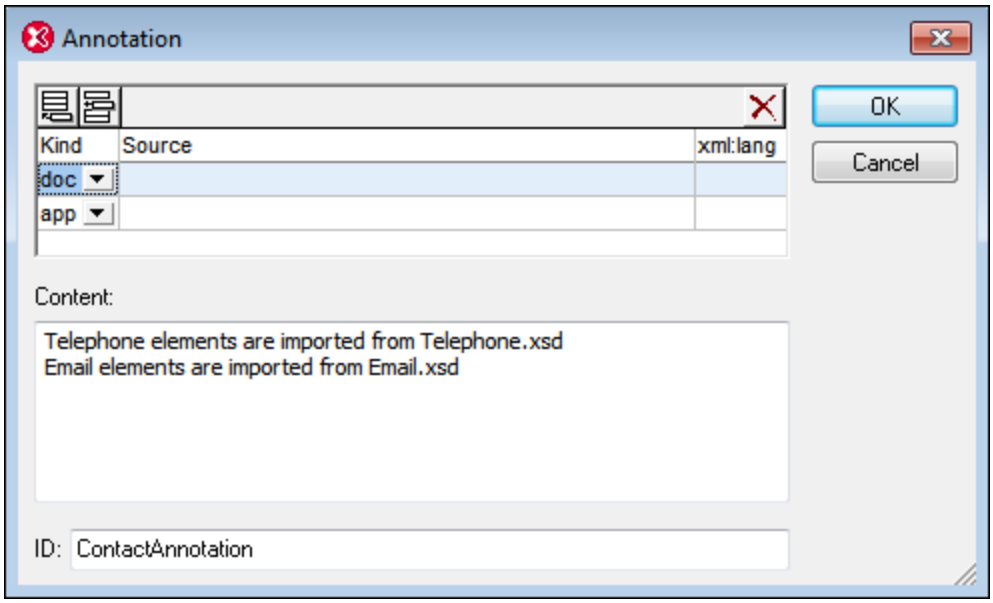
Une fois que les attributs globaux et les groupes d'attribut ont été créés, ils peuvent être référencés dans les déclarations des éléments et des types complexes.

Notations (notation)

Les notations sont toujours globales ; il n'y a pas de notations locales. Les propriétés d'une notation sont spécifiées dans l'assistant à la saisie des Détails de la notation. Le nom de la notation peut être spécifié directement dans la liste des composants globaux. Toutes les notations dans le schéma sont affichées dans l'assistant à la saisie des Composants pour faciliter la référence.

Annotations globales

Les annotations globales sont des composants globaux et ne sont pas identiques aux annotations optionnelles et qui sont disponibles pour certains composants globaux. Vous pouvez éditer une annotation globale dans le dialogue Annotation (*capture d'écran ci-dessous*), accessibles en cliquant avec la touche de droite sur le composant global de l'annotation et en choisissant **Toutes les données d'annotation**.



Chaque annotation peut avoir un attribut `id` et une `documentation` enfant multiple et/ou des éléments `appinfo`. Vous pouvez ajouter des éléments `documentation` ou `appinfo` en cliquant sur les boutons **Ajouter** ou **Insérer** situés en haut à gauche du dialogue et puis en choisissant l'item `doc` ou `app` depuis les listes de choix respectives. Choisir un item `doc` ou `app` dans la partie supérieure du panneau du dialogue et saisir son contenu dans le panneau *Contenu*. Si vous souhaitez créer une nouvelle ligne dans le contenu (et donc de rendre le contenu multi-ligne), appuyez sur **Entrée**. Dans la capture d'écran ci-dessus, l'élément `documentation` est

sélectionné et présente un contenu sur deux lignes. Pour chaque élément `documentation` ou `appinfo`, vous pouvez aussi saisir des attributs optionnels `source` et `xml:lang`.

Dans l'Aperçu de schéma, seul le premier élément `documentation` ou `appinfo` de l'annotation globale est affiché et peut être édité directement dans la liste des composants globaux. Si ce contenu est multi-ligne, placer le curseur par-dessus pour révéler toutes les lignes dans une fenêtre popup multiligne. Pour afficher ou éditer les contenus des autres éléments `documentation` et/ou `appinfo`, rendez-vous sur le dialogue d'annotation de cette annotation globale.



Note : Les annotations optionnelles qui sont disponibles pour certains composants globaux peuvent aussi être édités par le biais du dialogue d'annotation de la même manière que décrit ci-dessus.

Instructions de commentaires et de traitement

Les instructions de commentaires et de traitement peuvent être insérées à tout endroit de la liste des composants globaux dans l'Aperçu de schéma. Elle ne peuvent pas être ajoutées dans le Mode de Modèle de contenu. Si une ou plusieurs instructions de commentaires ou de traitement se trouvent dans les types simples ou les types complexes, elles sont collectées et déplacées vers l'extrémité de l'objet contenant. Dans ces cas, il est donc recommandé d'utiliser les annotations au lieu des commentaires.

4.4.3 Mode de Modèle de contenu

Un modèle de contenu est une description de la structure et du contenu d'un composant. Les composants suivants peuvent avoir des modèles de contenu :

- Types complexes
- Éléments
- Groupes de modèle
- Contenu ouvert par défaut

Ils sont indiqués dans la liste des composants globaux dans le Mode Schéma avec une icône **Passer au Mode de Modèle de contenu** située à gauche du nom du composant.

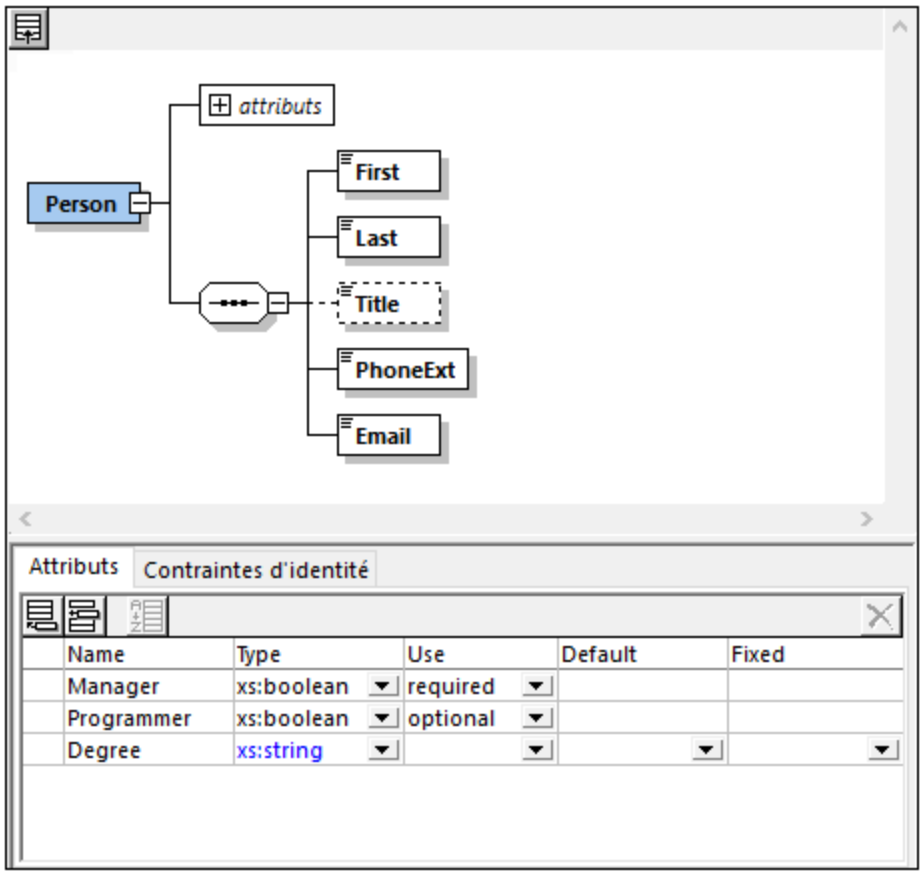
	*Passer au Mode de Modèle de contenu :* disponible pour les composants globaux qui ont un modèle de contenu. Ouvre le modèle de contenu du composant global dans le Mode de Modèle de contenu.
	*Afficher Globaux :* disponible dans le Mode de Modèle de contenu. Passe au Mode Schéma.

Cliquer sur l'icône **Passer au Mode de Modèle de contenu** pour ouvrir le Mode de Modèle de contenu pour ce composant global (*voir capture d'écran ci-dessous*). En alternative, dans le Mode Schéma : (i) choisir un composant puis sélectionner l'option de menu **Conception de schéma | Afficher diagramme,** ou (ii) double-cliquer sur le nom d'un composant dans l'Assistant à la saisie des Composants. Veuillez noter que seul un Modèle de contenu dans le schéma peut être ouvert à la fois. Lorsqu'un modèle de contenu est ouvert, vous pouvez passer au modèle de contenu d'un composant dans le cadre du modèle de contenu actuel en appuyant sur **Ctrl** et en double-cliquant sur le composant requis.

 © 2016-2022 Altova GmbH

Description générale du Mode de Modèle de contenu

Le modèle de contenu est affiché dans le Mode de Modèle de contenu en tant qu'arborescence (*voir capture d'écran ci-dessous*). Vous pouvez configurer l'apparence de l'arborescence dans le dialogue de Configuration de l'Affichage de Schéma (item de menu **Conception de Schéma | Configurer affichage**).



Note :

- Il existe deux types d'objets contenus dans l'arborescence de modèle de contenu : les compositeurs et les composants. De plus, les attributs, assertions, contraintes d'identité et le contenu ouvert peuvent être affichés dans des boîtes attachées au composant.
- Chaque niveau dans l'arborescence est joint aux niveaux adjacents par un compositeur. Le modèle de contenu peut s'étendre sur un nombre illimité de couches.
- Un objet peut être ajouté par rapport à un autre objet par le biais du menu contextuel de ce dernier (accessible en cliquant avec la touche de droite sur le dernier objet).
- Les composants dans le modèle de contenu peuvent être des composants locaux ou peuvent référencer des composants globaux.
- Une fonction glisser/déposer permet de déplacer des objets.
- Les raccourcis de clavier peuvent être utilisés pour copier (**Ctrl+C**) et coller (**Ctrl+V**) des objets.
- Les propriétés d'un objet peuvent être éditées dans l'Assistant de saisie des Détails et dans le panneau AACID.
- Les attributs, assertions et contraintes d'identité d'un composant sont affichés dans un panneau en dessous du Mode de Modèle de contenu, le panneau AACID. Les attributs et les contraintes d'identité

peuvent aussi être affichées dans le diagramme du modèle de contenu au lieu du panneau AACID. Cette option d'affichage peut être définie dans le dialogue de Configuration de l'affichage de schéma. En alternative, vous pouvez utiliser les trois boutons **Affichage dans le Diagramme** situés dans la barre outils de Conception de Schéma (*capture d'écran ci-dessous*).



- Les composants frères peuvent être triés en les sélectionnant, en cliquant avec la touche de droite et en choisissant la commande **Trier Composants** à partir du menu contextuel. Vous pouvez prioriser avec un des deux critères suivants : (i) nom local, et (ii) type de composant.

Ces fonctions sont expliquées en détail dans les sous-sections de cette section et dans le tutoriel.

Pour retourner au Mode Schéma, cliquer sur l'icône **Afficher globaux** ou sélectionner l'option de menu **Conception de schéma | Afficher tous les globaux**.

4.4.3.1 Objets de Modèle de contenu

Dans le Mode Modèle de contenu, les objets affichés dans le diagramme sont classés en trois groupes :

- Compositeurs : (i) séquence, (ii) choix, (iii) tous
- Composants : (i) élément, (ii) type complexe, (iii) groupe de modèle, (iv) caractère générique
- Divers : (i) attribut, (ii) groupe d'attribut, (iii) assertion, (iv) contrainte, (v) contenu ouvert

Les représentations graphiques de ces objets sont décrits individuellement ci-dessous.

Compositeurs

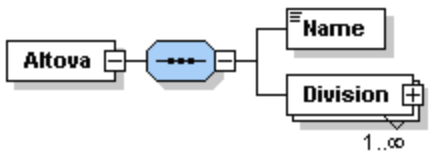
Un **compositeur** définit l'ordre dans lequel les éléments enfants se produisent. Il y a trois compositeurs : `sequence`, `choice`, et `all`.

Pour insérer un compositeur :

1. Cliquer avec la touche de droite sur l'élément auquel vous souhaitez ajouter des éléments enfants.
2. Sélectionner **Ajouter enfant | Séquence** (ou **Choix** ou **Tous**).

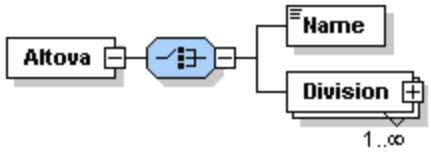
Le compositeur est ajouté, et ressemblera à la capture d'écran ci-dessous :

- **Séquence**

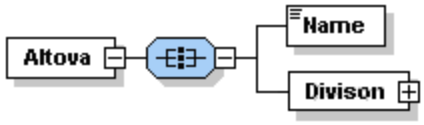


- **Choix**

 © 2016-2022 Altova GmbH



- **Tout**



Pour changer le compositeur, cliquer avec la touche de droite sur le compositeur et choisir **Changer Modèle | Séquence** (ou **Choix** ou **Tout**). Une fois avoir ajouté le compositeur, vous pourrez ajouter le/s élément/s enfant ou un groupe de modèle.

Composants

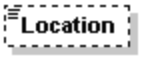
Ci-dessous, vous trouverez une liste des composants utilisés dans les modèles de contenu. La représentation graphique de chacun permet d'obtenir des informations détaillées concernant les propriétés de type et structurelles du composant.

- Élément unique obligatoire

  Détails : Le rectangle indique un élément et la bordure solide indique que l'élément est nécessaire. L'absence d'une fourchette de chiffres indique un élément unique (par ex. `minOcc=1` et `maxOcc=1`). Le nom de l'élément est `Country`. La couleur bleu indique que l'élément est sélectionné actuellement ; (un composant est sélectionné en cliquant dessus). Si un composant n'est pas sélectionné, il est blanc.

- Élément unique optionnel

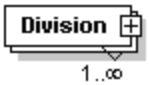
  Détails : Le rectangle indique un élément et la bordure en pointillés indique que l'élément est optionnel. L'absence d'une fourchette de chiffres indique un élément unique (par ex. `minOcc=0` et `maxOcc=1`). Le nom de l'élément est `Location`.
  Note: L'option de menu contextuel **Optionnel** convertit un élément obligatoire en un élément optionnel.

- Élément multiple obligatoire

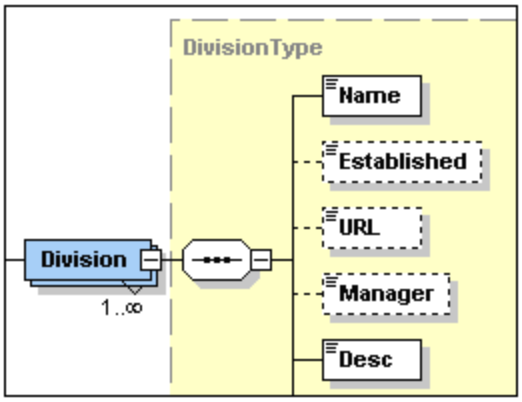
  

Détails : Le rectangle indique un élément et la bordure solide indique que l'élément est nécessaire. La fourchette de chiffres 1..5 signifie que `minOcc=1` et `maxOcc=5`. Le nom de l'élément est `Alias`.

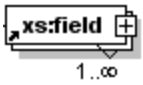
- Élément multiple obligatoire contenant des éléments enfant



Details: Le rectangle indique un élément et la bordure solide indique que l'élément est nécessaire. La fourchette de chiffres 1..infini signifie que `minOcc=1` et `maxOcc=unbounded`. Le signe plus signifie un contenu complexe (par ex. au moins un enfant élément ou attribut). Le nom de l'élément est `Division`.
Note : L'option de menu contextuel **Non lié** change `maxOcc` en `unbounded`.
Cliquer sur le signe + de l'élément pour agrandir l'affichage en arborescence et afficher les éléments enfant.



- Élément référençant un élément global



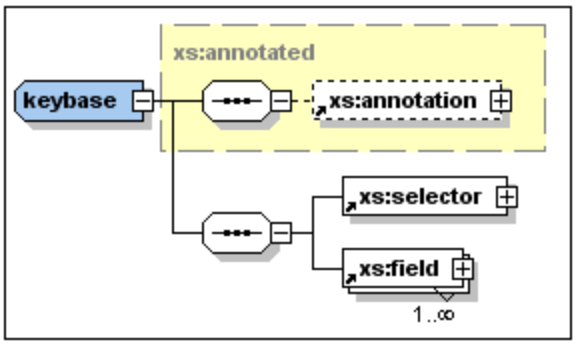
Détails : la flèche située en bas à gauche signifie que l'élément référence un élément global. Le rectangle indique qu'un élément et la bordure solide indique que l'élément est nécessaire. La fourchette de chiffres 1..infini signifie que `minOcc=1` et `maxOcc=unbounded`. Le signe plus indique un contenu complexe (par ex. au moins un élément ou un enfant d'attribut). Le nom de l'élément est `xs:field`.
Note : un élément global peut être référencé depuis des définitions de type simple et complexe, vous permettant ainsi de réutiliser une déclaration globale dans plusieurs endroits dans votre schéma. Vous pouvez créer une référence à un élément global de deux manières : (i) en saisissant un nom pour l'élément local qui est le même que celui de l'élément global ; et (ii) en cliquant avec le bouton de droite sur l'élément local et en sélectionnant l'option **Référence** à partir du menu contextuel. Vous pouvez consulter la définition d'un élément global en appuyant sur **Ctrl** et en double cliquant sur l'élément. En alternative, cliquer avec la touche de droite et sélectionner sur **Aller à la définition**. Si vous créez une référence à un élément qui n'existe pas, le nom de l'élément apparaît en rouge pour vous avertir qu'il n'y aura pas de définition à laquelle se référer.

- Type complexe

 © 2016-2022 Altova GmbH



Détails : l'hexagone irrégulier avec un signe plus indique un type complexe. Le type complexe montré ici porte le nom `keybase`. Ce symbole (l'hexagone irrégulier avec un signe plus) indique un type complexe global. Un type complexe global est déclaré dans l'Aperçu de Schéma et son modèle de contenu est généralement défini dans le Mode de Modèle de contenu. Un type complexe global peut être utilisé soit en tant que (i) le type de données d'un élément, soit (ii) le type de base d'un autre type complexe en l'attribuant à l'élément ou au type complexe, respectivement, dans l'Assistant à la saisie des Détails (soit dans le Mode de Modèle de contenu soit dans l'Aperçu de Schéma).



Le type complexe `keybase` affiché ci-dessus a été déclaré dans l'Aperçu de Schéma avec un type de base de `xs:annotated`. Le type de base est affiché en tant qu'un rectangle avec une bordure à tirets et un arrière-plan jaune. Ensuite, dans le Mode de Modèle de contenu, les éléments enfant `xs:selector` et `xs:field` ont été créés. (Veuillez noter les petites flèches en bas à gauche du coin des rectangles `xs:selector` et `xs:field`. Celles-ci indiquent que les deux éléments référencent des éléments globaux de ces noms.)

Un type complexe local est défini directement dans le Mode de Modèle de contenu en créant un élément enfant ou un attribut pour un élément. Il n'y a pas de symbole séparé pour les types complexes locaux.

Note : le type de base d'un modèle de contenu est affiché en tant que rectangle dans une bordure grise à tirets et avec un arrière-plan jaune. Vous pouvez vous rendre au modèle de contenu du type de base en double-cliquant son nom.

- Groupe de modèle



Détails : L'octogone irrégulier avec un signe plus indique un groupe de modèle. Un groupe de modèle vous permet de définir et de réutiliser des déclarations d'élément.
Note : Lorsque le groupe de modèle est déclaré (dans l'Aperçu de schéma) un nom lui est attribué. Par conséquent, vous définissez son modèle de contenu (dans le Mode de Modèle de contenu) en lui attribuant un compositeur enfant qui contient les déclaration d'élément. Lorsque le groupe de modèle est utilisé, il est inséré en tant qu'un enfant, ou inséré ou ajouté dans le cadre du modèle de contenu d'un autre composant (dans le Mode de Modèle de contenu).

- Caractères génériques

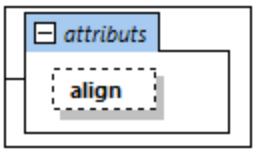


Détails : L'octogone irrégulier avec `any` à gauche indique un caractère générique.
Note : Les caractères génériques sont utilisés en tant qu'espaces réservés pour permettre aux éléments non spécifiés dans le schéma ou à partir d'autres espaces de noms. `##other` = éléments peuvent appartenir à tout espace de noms différent de l'espace de noms cible défini dans le schéma ; `##any` = éléments peuvent appartenir à tout espace de noms; `##targetNamespace` = éléments doivent appartenir à l'espace de noms cible défini dans le schéma ; `##local` = éléments ne peuvent pas appartenir à un espace de noms; *`anyURI`* = éléments appartiennent à l'espace de noms que vous spécifiez.

Objets divers

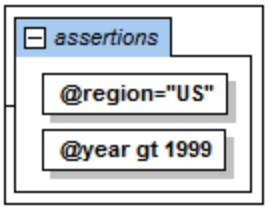
Les objets divers sont des attributs, des groupes d'attribut, des assertions, des contraintes d'identité et du contenu ouvert.

- Attributs, Groupes d'attribut



Détails : indiqué par le mot *'attributs'* en italique dans un rectangle qui peut être agrandi. Chaque attribut est affiché dans un rectangle avec une (i) bordure à tirets si l'attribut est optionnel, ou une (ii) bordure solide si l'attribut est nécessaire (obligatoire). Les groupes d'attribut et les caractères génériques d'Attribut sont aussi inclus dans le rectangle *'attributs'*.
Note : les attributs peuvent être édités dans le diagramme et dans l'Assistant à la saisie des Détails. Les attributs peuvent être affichés dans le diagramme de Mode de modèle de contenu ou dans le panneau AACID situé en-dessous du Mode de modèle de contenu. Vous pouvez basculer entre ces deux modes en cliquant sur l'icône Afficher les attributs. Pour changer l'ordre des attributs d'un élément, glissez l'attribut puis déposez-le lorsque la flèche apparaît à l'emplacement requis.

- Assertions

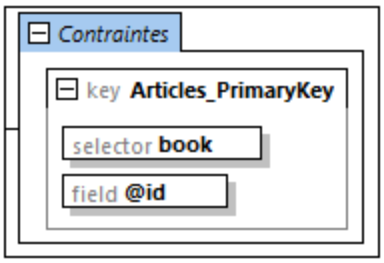


Détails : indiqué par le mot *'assertions'* en italique dans un rectangle qui peut être agrandi. Chaque assertion est affichée dans un rectangle dans la fenêtre Assertions.
Note : les assertions peuvent être éditées dans le diagramme et dans l'Assistant à la saisie des Détails. Elles peuvent être affichées dans le diagramme de Mode de modèle de contenu ou dans le panneau AACID situé en-dessous du Mode de modèle de contenu. Vous pouvez basculer entre ces

 © 2016-2022 Altova GmbH

deux modes en cliquant sur l'icône Afficher les Assertions . Pour changer l'ordre des assertions d'un élément, glissez l'assertion déposez-la lorsque la flèche apparaît à l'emplacement requis.

- Contraintes d'identité

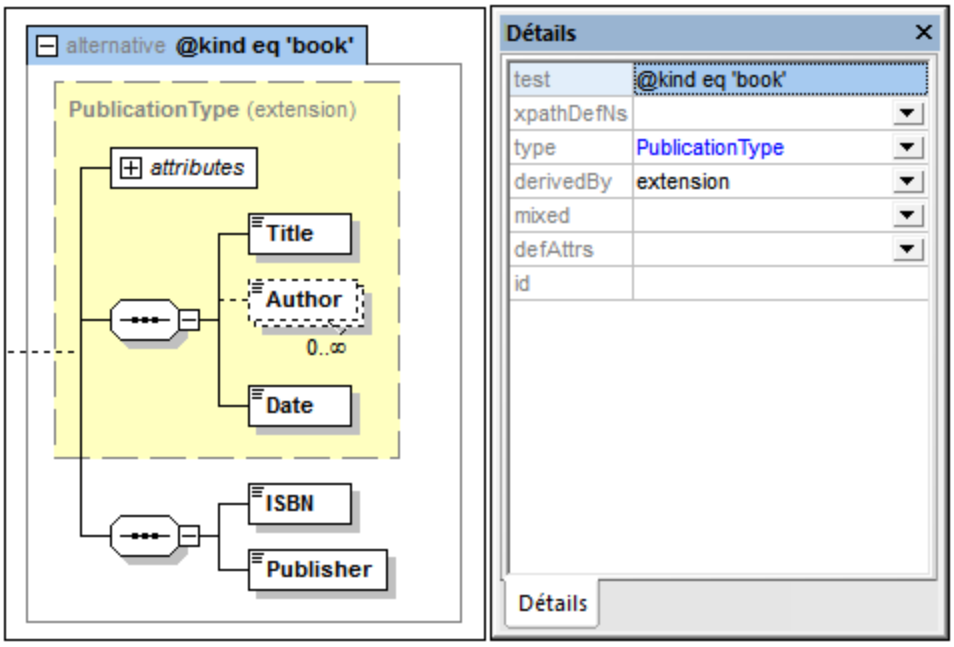


Détails : indiqué par le mot '*contraintes*' en italique dans un rectangle qui peut être agrandi.
Note : la contrainte d'identité recensée dans le modèle de contenu d'un composant montre les contraintes telles que définies avec les éléments `key` et `keyref`, et avec l'élément `unique`. Les contraintes d'identité définies à l'aide du type de données `ID` ne sont pas affichées dans le diagramme de modèle de contenu mais dans l'Assistant à la saisie des Détails. Les contraintes d'identité peuvent être affichées et éditées dans le Mode de modèle de contenu ou dans l'onglet Contraintes d'identité de l'Aperçu de schéma. Dans le Mode de modèle de contenu, vous pouvez allumer et éteindre la fenêtre Contraintes avec l'icône Afficher les Contraintes .

- Attribution de Type conditionnelle



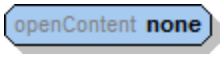
Détails : l'élément `alternative` est un rectangle contenant l'expression XPath qui sera testée (*voir capture d'écran ci-dessus*). Le type de l'élément `alternative` est spécifié dans Assistant à la saisie des Détails. Si le type est un type complexe, il est affiché dans le rectangle agrandi de l'élément `alternative` et peut encore être édité ici (*voir capture d'écran ci-dessous*). Les types simples ne sont pas affichés dans le diagramme, mais peuvent être définis dans l'onglet Type Simple de l'Assistant à la saisie des Détails.



Note : l'élément `alternative` est nouveau dans XSD 1.1. Si l'expression XPath évalue à vrai, le type spécifié par l'élément `alternative` sera le type sélectionné. Le premier élément `alternative` parmi les frères `alternative` à évaluer vrai sera sélectionné. L'ordre des éléments alternatifs est donc important. Celui-ci peut être modifié en glissant les fenêtres d'élément `alternative` dans l'ordre souhaité. Voir la section Attribution de type conditionnelle pour une description plus détaillée.

- Contenu ouvert par défaut, Contenu ouvert





Détails : les éléments `defaultOpenContent` et `openContent` sont indiqués dans le Mode de modèle de contenu avec les libellés `openContent` et `defOpenContent`. Le contenu d'élément de caractères génériques est indiqué par une fenêtre `any` (*voir capture d'écran ci-dessus*).
Note : les éléments `defaultOpenContent` et `openContent` sont nouveaux dans XSD 1.1. Le Contenu ouvert par défaut est un composant global et est créé dans l'Aperçu de schéma. Dans le Mode de modèle de contenu d'un modèle de contenu particulier, vous pouvez remplacer le Contenu ouvert par défaut avec le Contenu ouvert spécifique à ce composant qui surmontera le Contenu ouvert par défaut du Schéma. Il suffit d'ajouter le Contenu ouvert en tant qu'enfant du composant. La fenêtre Contenu ouvert par défaut sera remplacée par une fenêtre de Contenu ouvert. Dans le Mode de modèle de contenu, vous pouvez éditer l'attribut `mode` du Contenu ouvert et l'espace de noms de son caractère générique est un élément de carte générique, aussi bien dans le diagramme que dans l'Assistant à la saisie des Détails. Vous pouvez aussi modifier le Contenu ouvert par défaut (pour tout le schéma) depuis sa représentation dans le Mode de modèle de contenu de tout type complexe.

Note :

© 2016-2022 Altova GmbH

- Les détails prédéfinis que vous avez spécifiés dans le dialogue de Configuration d'affichage du Schéma peuvent être allumés et éteints en cliquant sur l'icône de barre d'outils Ajouter les détails prédéfinis.
- Vous pouvez basculer les Attributs, les Assertions et les Contraintes d'Identité pour qu'ils apparaissent soit dans le diagramme du modèle de contenu lui-même soit dans le panneau AACID (en-dessous du Mode de modèle de contenu) en cliquant sur les icônes Afficher dans le Diagramme pour les attributs, assertions et contrainte d'identité respectivement.
- Dans le Mode de modèle de contenu, vous pouvez passer au mode de modèle de contenu de tout composant global dans le cadre du modèle de contenu actuel en maintenant appuyée la touche **Ctrl** et en double-cliquant sur le composant requis. Vous pouvez vous rendre sur le modèle de contenu d'un type de base en double-cliquant sur le nom du type de base.
- Le menu contextuel des composants contient des commandes pour (i) se rendre à la définition d'un composant et (ii) se rendre dans la définition de type d'un composant, le cas échéant.

4.4.3.2 Édition dans le Mode de Modèle de contenu

La description de l'édition dans le Mode de Modèle de contenu est organisée dans les sections suivantes :

- Configurer le Mode de Modèle de contenu
- Attributs, Assertions et Contraintes d'identité
- Icônes de Mode de Modèle de contenu
- Exploitation du menu contextuel
- Raccourcis du clavier et glisser/déposer
- Propriétés du composant
- Annotations
- Instructions de commentaires et de traitement
- Documenter le modèle de contenu

Configurer le Mode de Modèle de contenu

Vous pouvez configurer le mode de modèle de contenu pour tout le schéma dans le dialogue de configuration d'affichage du Schéma (**Conception de Schéma | Configurer affichage**). Pour consulter des détails concernant les options de configuration, voir la section Configurer affichage située plus loin dans la Référence de l'Utilisateur. Veuillez noter que les paramètres que vous définissez ici s'appliquent au schéma dans son ensemble et à la sortie de documentation du schéma ainsi qu'à la sortie de l'imprimante.

Attributs, Assertions et Contraintes d'identité

Les attributs, les assertions et les contraintes d'identité d'un composant peuvent apparaître dans un panneau situé en-dessous du Mode de Modèle de contenu, le panneau AACID, ou en tant que fenêtres dans le Mode de Modèle de contenu lui-même, c'est à dire, dans le diagramme. La deuxième option de consultation peut être définie dans le dialogue de configuration d'Affichage du Schéma. En alternative, vous pouvez utiliser les trois boutons de barre d'outils **Afficher dans le diagramme** situés dans la barre d'outils de la barre d'outils de Conception de Schéma (*capture d'écran ci-dessous, voir aussi la liste d'icônes ci-dessous*).



Pour consulter une description de la manière d'insérer et d'éditer des attributs, des assertions et des contraintes d'identité, voir la section, Attributs, Assertions et Contraintes d'identité.

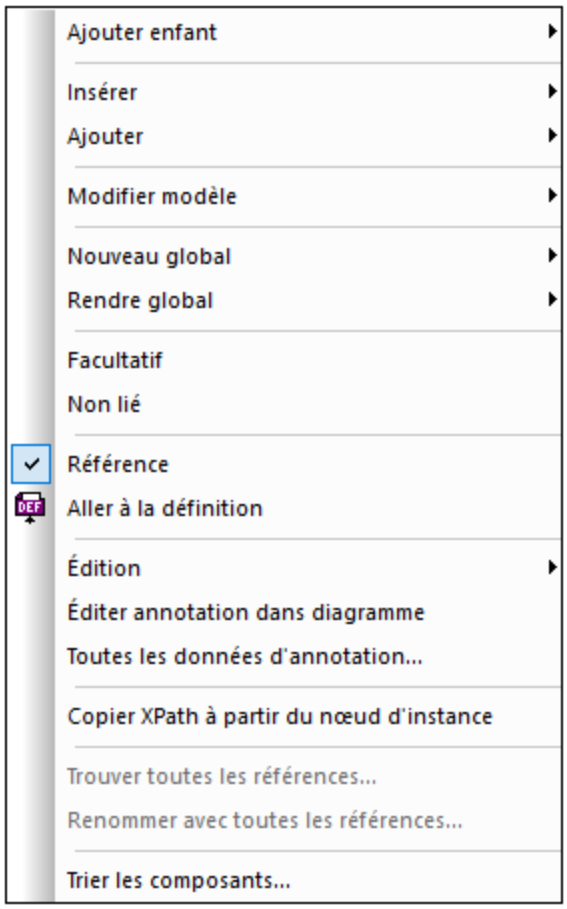
Icônes de Mode de Modèle de contenu

	*Affichage global :* disponible dans le Mode de Modèle de contenu. Passe à l'Aperçu de schéma.
	*Ajouter des détails prédéfinis :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des détails prédéfinis dans les composants
	*Affiche les Attributs dans le Diagramme :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des attributs entre le diagramme (activé) et l'onglet Attributs.
	*Affiche les Assertions dans le Diagramme :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des assertions entre le diagramme (activé) et l'onglet Assertions.
	*Affiche les Contraintes dans le Diagramme :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des IDC entre le diagramme (activé) et l'onglet Contraintes d'identité.
	*Visualiser les Contraintes d'identité :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des informations IDC.

Exploitation du menu contextuel

Plusieurs opérations d'édition dans le Mode de Modèle de contenu sont effectuées par le biais du menu contextuel (*capture d'écran ci-dessous*) qui apparaît lorsque vous cliquez avec la touche de droite dans le Mode de Modèle de contenu. Seules les commandes pour les opérations autorisées à cet endroit dans le diagramme de modèle de contenu sont activées. Les opérations sont effectuées par rapport à l'objet cliqué (avec le bouton de droite). Par exemple, lorsqu'un enfant est ajouté, il est ajouté par rapport à l'objet cliqué avec le bouton de droite.

© 2016-2022 Altova GmbH



Ci-dessous, vous trouverez une liste des opérations disponibles par le biais du menu contextuel.

- *Ajouter des compositeurs et composants enfants :* la commande **Ajouter enfant** ouvre un sous-menu dans lequel vous pouvez sélectionner le compositeur ou le composant à ajouter.

- *Insérer/Ajouter des compositeurs et composants :* insère le compositeur ou le composant au même niveau hiérarchique que l'objet sélectionné, avant l'objet sélectionné (**Insérer**) ou après son dernier frère (**Ajouter**).

- *Changer un compositeur :* cliquer avec la touche de droite sur un compositeur, sélectionner **Changer Modèle | <nouveau compositeur>**.

- *Créer des composants globaux :* (i) la commande **Nouveau Global** est accessible en cliquant n'importe où dans le Mode de Modèle de contenu. Un sous-menu s'affiche dans lequel vous pouvez sélectionner le nouveau composant global que vous souhaitez créer. (ii) Si un objet peut être créé en tant que composant global, la commande **Rendre global** est activée dans son menu contextuel. En sélectionnant cette commande, l'objet sera créé en tant que composant global. Dans le Mode de Modèle de contenu, il contiendra une référence au nouveau composant global créé.

- *Changer la définition d'occurrence :* utiliser ensemble les commandes **Optionnel** et **Non-lié** pour obtenir les paramètres d'occurrence désirés : (i) *optionnel* = 0 ou 1; (ii) *optionnel* + *non-lié* = 0 à infini ;

(iii) *non-lié* = 1 à infini ; (iv) *non optionnel* + *non non-lié* = 1. (Note : *optionnel* définit l'attribut `minOccurs` du composant, *non-lié* définit l'attribut `maxOccurs`.)

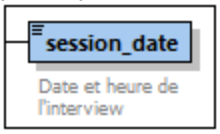
- *Basculer entre les définitions locales et globales :* si un élément global existe qui a le même nom qu'un élément local, utiliser la commande de bascule **Référence** pour passer entre le référencement de la définition globale (actif) et en utilisant la définition locale (désactiver).

- *Passer à un autre modèle de contenu :* si un composant a son propre modèle de contenu (par exemple, s'il référence un composant global), alors la commande **Aller à la définition** sera active et vous pourrez la sélectionner pour aller dans le modèle de contenu. En alternative, vous pouvez appuyer sur **Ctrl** et double-cliquer sur le composant.

- *Aller à la définition de type du composant :* Si un composant a une définition de type (type simple ou type complexe), alors cliquer sur la commande **Aller à la définition de type** pour aller dans la définition de type. Dans le cas des types simples intégrés, une fenêtre de message apparaît qui contient des informations concernant le type simple.

- *Éditer des détails prédéfinis :* si les détails prédéfinis ont été réglés pour être affichés dans le diagramme (avec l'icône **Ajouter les détails prédéfinis** dans la barre outils Conception de Schéma), la commande **Édition** affichera un sous-menu contenant les détails prédéfinis pouvant être édités. Sélectionner le détail prédéfini nécessaire et éditer sa valeur dans le diagramme.

- *Créer et éditer une annotation compositeur/composant :* la commande **Éditer l'annotation** crée un espace d'annotation en-dessous du compositeur/composant (*voir capture d'écran ci-dessous*). Vous pouvez saisir et éditer l'annotation ici. Si l'annotation existe déjà, cliquer sur la commande pour souligner l'annotation pour l'édition de texte. Double-cliquer sur l'annotation existante est un moyen plus rapide de lancer une édition.

  Dans le document Schéma XML, l'annotation est créée à l'intérieur de l'élément `annotation/documentation` du compositeur ou du composant. *Voir aussi la section ci-dessous à propos de la documentation.*

- *Copie XPath du nœud d'instance :* la commande **Copier XPath à partir du nœud d'instance** est activée pour les éléments et les attributs définis dans un élément global ou un type de complex global. Il copie sur le presse-papier une expression XPath qui localise le nœud sélectionné. L'expression XPath d'emplacement commence au niveau du composant global dont le modèle de contenu est actuellement affiché dans le Mode de Modèle de contenu.

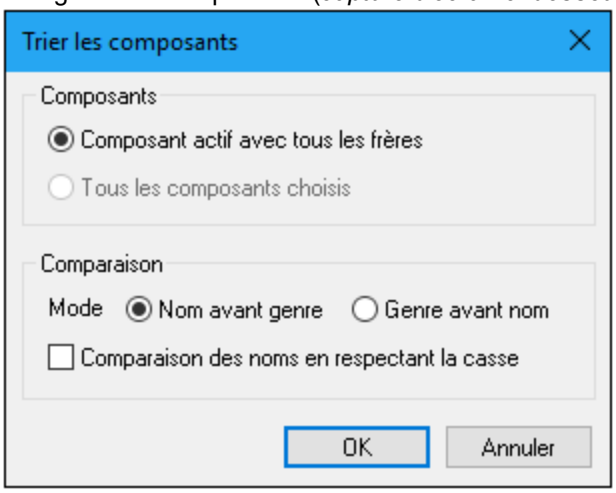
- *Trouver et renommer le composant :* les commandes **Trouver toutes les références** et **Renommer avec toutes les références** sont activées pour les éléments globaux. Ceux-ci, respectivement, trouvent toutes les occurrences du composant sélectionné et renomment toutes les occurrences du composant sélectionné dans le document actif et, en option, dans tous les fichiers de schéma liés au document actif.

 © 2016-2022 Altova GmbH



Dans la capture d'écran ci-dessus, le nom `Email` remplacera le nom du composant cliqué avec la touche de droite et de toutes ses références dans le cadre du champ de recherche. Voir Trouver et renommer les Globaux pour plus de détails.

- *Trier les déclarations et les références :* utiliser la commande **Trier**, tous les composants sélectionnés ou les frères du composant sélectionnés peuvent être triés. Effectuez vos paramètres de tri dans le dialogue Trier composants (*capture d'écran ci-dessous*) et cliquer sur **OK**.



Afin de sélectionner des composants multiples, appuyer sur la touche **Shift** ou **Ctrl** tout en cliquant. Vous pouvez choisir soit les noms de composant en tant que la première clé de tri et le type de composant en tant que la deuxième clé de tri, soit vice-versa.

Note : Vous pouvez sélectionner un composant et le copier, le couper, le supprimer ou le glisser. Dans quelques cas, comme la restriction d'attributs d'une limite `complexType`, cela peut être refusé.

Raccourcis du clavier et glisser/déposer

Vous pouvez copier et coller des éléments dans le Mode de Modèle de contenu en utilisant les raccourcis **Ctrl+c** et **Ctrl+v**. Les objets copiés sont collés en tant qu'objets enfants de l'objet sélectionné. Si cela n'est pas possible pour des raisons structurelles, un message s'affiche.

Vous pouvez aussi effectuer un glisser/déposer : (i) des objets à d'autres emplacements dans le diagramme, (ii) certains composants, comme les attributs, depuis l'Assistant à la saisie des Composants dans le diagramme.

Propriétés du composant

Si le Mode de Modèle de contenu est configuré de manière à ce que les composants sont affichés avec des détails prédéfinis dans la fenêtre de composant, alors vous pouvez éditer ces informations directement dans le diagramme. L'affichage des détails prédéfinis peut être allumé et éteint en cliquant sur l'icône de barre d'outils **Ajouter les détails prédéfinis** (*voir la liste d'icône ci-dessus*).

En alternative, vous pouvez éditer les propriétés de composant dans l'Assistant à la saisie des Détails, et les changements seront visibles dans les champs des espaces réservés , si ceux-ci sont configurés pour un affichage.

Annotations

Les annotations de Schéma XML sont contenus dans l'élément `annotation`. Il existe deux types d'annotations, chacun d'entre eux est contenu dans un élément enfant différent d'`annotation`:

- enfant `documentation` : contient les informations qui peuvent être utiles pour les éditeurs du schéma
- enfant `appinfo` : vous permet d'insérer un script ou une information pouvant être utilisé par une application de traitement

Ci-dessous, se trouve le texte d'un élément d'annotation qui contient les deux types d'éléments enfants.

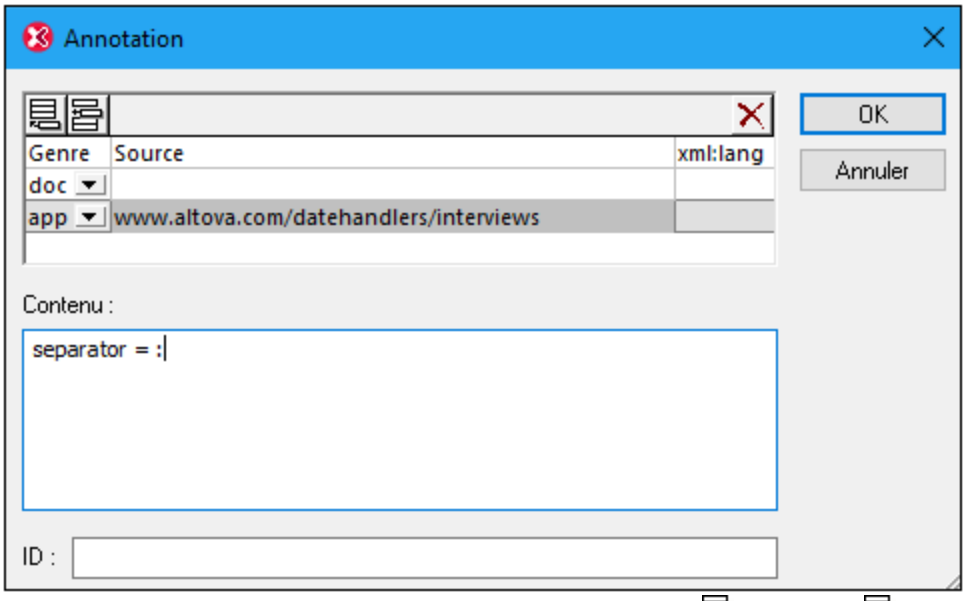
```
<xs:element name="session_date" type="xs:dateTime" nillable="true">
   <xs:annotation>
      <xs:documentation>Date and time when interview was held</xs:documentation>
      <xs:appinfo source="http://www.altova.com/datehandlers/interviews">separator =
:</xs:appinfo>
   </xs:annotation>
</xs:element>
```

Dans le Mode de Modèle de contenu, vous pouvez créer l'annotation des compositeurs et composants individuels comme suit.

1. Cliquer avec la touche de droite sur le compositeur ou le composant.
   

2. Sélectionner l'option de menu contextuel **Toutes les données d'annotation**. Le dialogue d'annotation s'ouvre (*voir capture d'écran ci-dessous*). Si une annotation (soit `documentation` ou `appinfo`) existe pour cet élément, alors cela sera indiqué par une ligne correspondante dans le dialogue.

© 2016-2022 Altova GmbH



3. Afin de créer un élément `appinfo`, cliquer sur l'icône Ajouter ou Insérer située en haut à gauche pour ajouter ou insérer une nouvelle ligne, respectivement.
4. Dans le champ *Genre* de la nouvelle ligne, sélectionner l'option `app` depuis le menu déroulant.
5. Dans le panneau Contenu du dialogue, saisir le script ou l'info que vous souhaitez voir traité par une application de traitement.
6. En option, dans le champ *Source*, vous pouvez saisir une URI de source où plus d'informations seront accessibles pour l'application de traitement.

Instructions de commentaires et de traitement

Lorsque les documents de Schéma XML sont chargés dans XMLSpy, ou lorsque des modes sont modifiés, les instructions de commentaires et de traitement dans les types simples et les types complexes sont collectées et déplacées vers l'extrémité de l'objet contenant. Il est donc recommandé d'utiliser les annotations à la place des commentaires dans ces cas.

Documenter le modèle de contenu

Vous pouvez générer une documentation détaillée de votre schéma dans des formats HTML et MS Word. Une documentation détaillée est générée pour chaque composant global et la liste des composants globaux est affichée dans une page table-des-matières qui vous permet d'établir des liens vers les modèles de contenu des composants individuels. De plus, les éléments liés (comme les éléments enfants ou les types complexes) sont référencés par des hyperliens, vous permettant ainsi de naviguer d'un élément à un autre. Afin de générer une documentation de schéma, sélectionnez la commande de menu **Conception de schéma | Générer documentation**.

4.4.3.3 Assignation de type conditionnel

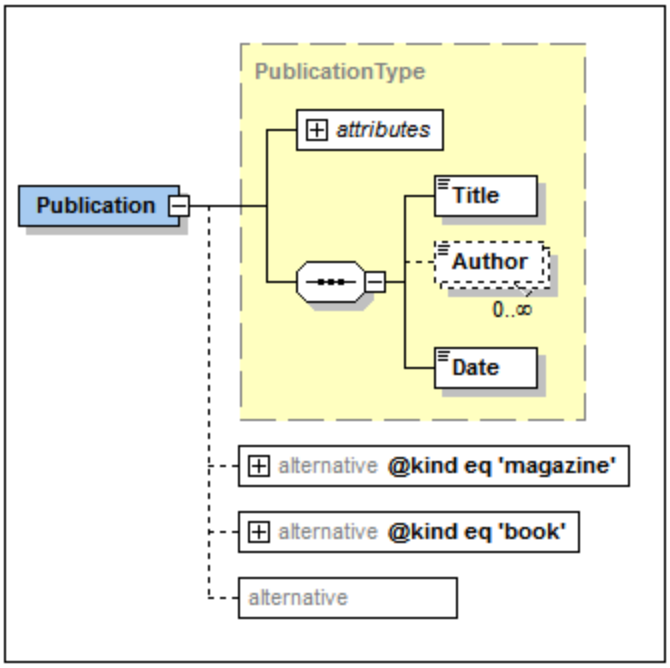
L'assignation de type conditionnel est une fonction XSD 1.1 qui permet de déterminer le type d'un élément par le biais du contenu dans le document XML, en particulier par la valeur des attributs de l'élément ou par la présence ou l'absence d'attributs. Si, par exemple, le document XML a l'élément suivant :

```
<publication kind="magazine">
```

```
    ...
</publication>
```

Dans le schéma, il est possible de faire varier le type de l'élément `publication` conformément à la valeur de la valeur de l'attribut `@kind` de l'élément de l'instance. Dans le schéma, cela est fait en utilisant l'élément `alternative`, qui est nouveau dans XSD 1.1. Les types multiples sont spécifiés, chacun dans un élément `alternative`.

Dans la capture d'écran ci-dessous, l'élément `Publication` est déclaré avec trois éléments enfants `alternative`, dont deux ont des attributs `test` (`@kind eq 'magazine'` et `@kind eq 'book'`). Le troisième élément `alternative` n'a aucun attribut `test` et une assignation de type simple de `xs:error` (assigné dans l'Assistant à la saisie Détails, non affiché dans le diagramme), qui, si déclenchée, retourne une erreur de validation XML.



La liste pour les déclarations ci-dessus est indiquée ci-dessous :

```
<xs:complexType name="PublicationType">
   <xs:sequence>
      <xs:element name="Title" type="xs:string"/>
      <xs:element name="Author" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Date" type="xs:gYear"/>
   </xs:sequence>
   <xs:attribute name="kind" type="xs:string"/>
</xs:complexType>

<xs:complexType name="MagazineType">
   <xs:complexContent>
      <xs:restriction base="PublicationType">
         <xs:sequence>
            <xs:element name="Title" type="xs:string"/>
            <xs:element name="Date" type="xs:gYear"/>
         </xs:sequence>
      </xs:restriction>
```

 © 2016-2022 Altova GmbH

```
   </xs:complexContent>
</xs:complexType>

<xs:element name="Publication" type="PublicationType">
   <xs:alternative test="@kind eq 'magazine'" type="MagazineType"/>
   <xs:alternative test="@kind eq 'book'">
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="PublicationType">
               <xs:sequence>
                  <xs:element name="ISBN" type="xs:string"/>
                  <xs:element name="Publisher" type="xs:string"/>
               </xs:sequence>
            </xs:extension>
         </xs:complexContent>
      </xs:complexType>
   </xs:alternative>
   <xs:alternative type="xs:error"/>
</xs:element>
```

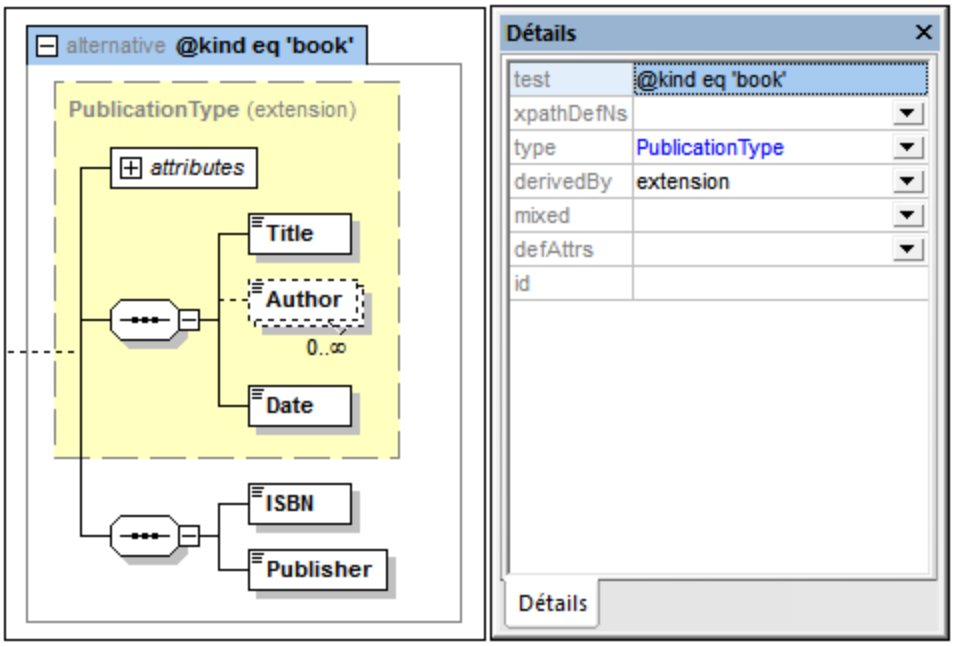
Veuillez noter les points suivants :

- Le premier élément `alternative` parmi les frères `alternative` à évaluer sur "vrai" est sélectionné. L'ordre des éléments alternatifs est donc important. Dans le Mode de Modèle de contenu, l'ordre peut être changé en glissant les fenêtres d'élément `alternative` dans l'ordre désiré.
- Veuillez noter que l'élément `Publication` a un type (`PublicationType`). Ce type sert en tant que le type par défaut si aucun autre des éléments `alternative` ne sont utilisés. Dans notre exemple ci-dessus, néanmoins, l'élément `alternative` du type `xs:error` sera utilisé si les deux conditions d'élément `alternative` précédentes retournent `faux`.
- Si aucune condition d'élément `alternative` n'est remplie et que l'élément n'a pas de type de défaut, un type `anyType` de l'élément est assigné. Dans ce cas, l'élément peut avoir un contenu XML bien formé de tout genre.
- L'élément `alternative` et le type simple `xs:error` sont nouveaux dans XSD 1.1.

Édition de Mode de Modèle de contenu

Vous pouvez ajouter un type alternatif à une déclaration d'élément en tant qu'enfant par le biais du menu contextuel de l'élément (*voir le modèle de contenu dans la capture d'écran ci-dessus*).

Le type de l'élément `alternative` est spécifié dans l'Assistant à la saisie des détails. Si le type est un type complexe, il est affiché dans le rectangle agrandi de l'élément `alternative` et peut y être édité ultérieurement (*voir capture d'écran ci-dessous*). Les types simples ne sont pas affichés dans le diagramme, mais peuvent être définis dans l'onglet Type simple de l'Assistant de saisie Détails.



Note : Vous pouvez spécifier un espace de noms pour l'expression XPath par le biais de l'attribut `xpathDefaultNamespace` dans l'Assistant de saisie Détails. Pour plus d'informations concernant les espaces de noms par défaut XPath, voir la section ci-dessous.

Utiliser `xpathDefaultNamespace`

Un espace de noms par défaut déclaré dans le document de Schéma XML est l'espace de noms par défaut du document de Schéma XML. Il s'applique aux noms d'élément sans préfixe dans le document du schéma mais pas aux noms d'éléments sans préfixes dans les expressions XPath dans le document schéma.

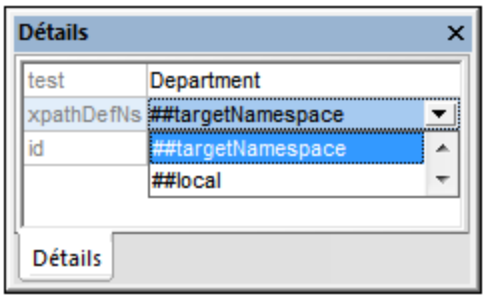
L'attribut `xpathDefaultNamespace` (une nouvelle fonction dans XSD 1.1) est le mécanisme utilisé pour spécifier l'espace de noms auxquels appartiennent les noms d'élément sans préfixe dans les expressions XPath.

Les espaces de noms XPath par défaut portent sur les éléments de Schéma XML sur lesquels ils sont déclarés. L'attribut `xpathDefaultNamespace` peut se produire sur les éléments de Schéma XML 1.1 suivants :

- `xs:schema`
- `xs:assert` et `xs:assertion`
- `xs:alternative`
- `xs:selector` et `xs:field` (dans les contraintes d'identité)

`xpathDefaultNamespace` sur `xs:schema` est déterminé, dans le mode XSD 1.1, dans le dialogue Paramètres de Schéma (**Conception de Schéma | Paramètres de schéma**). En ce qui concerne les autres éléments contenus dans la liste ci-dessus, l'attribut `xpathDefaultNamespace` est déterminé dans leurs Assistants à la saisie Détails respectifs (*voir capture d'écran ci-dessous par exemple*).

© 2016-2022 Altova GmbH



La déclaration d'espace de noms XPath par défaut dans `xs:schema`, déclare l'espace de noms XPath par défaut pour l'étendue de tout le schéma. Vous pouvez contourner cette déclaration sur les éléments dont l'attribut `xpathDefaultNamespace` est autorisé (*voir liste ci-dessus*).

Au lieu de contenir un espace de noms réel, l'attribut `xpathDefaultNamespace` peut contenir un des trois mots-clé :

- `##targetNamespace`: l'espace de noms XPath par défaut sera le même que l'espace de noms cible du schéma
- `##defaultNamespace`: l'espace de noms XPath par défaut sera le même que l'espace de noms par défaut du schéma
- `##local`: il n'y a pas d'espace de noms XPath par défaut

Si aucun espace de noms XPath par défaut n'est déclaré dans le document, des éléments sans préfixes dans les expressions XPath ne seront pas dans des espaces de noms.

Note : La déclaration d'espace de noms XPath par défaut ne s'applique pas aux attributs.

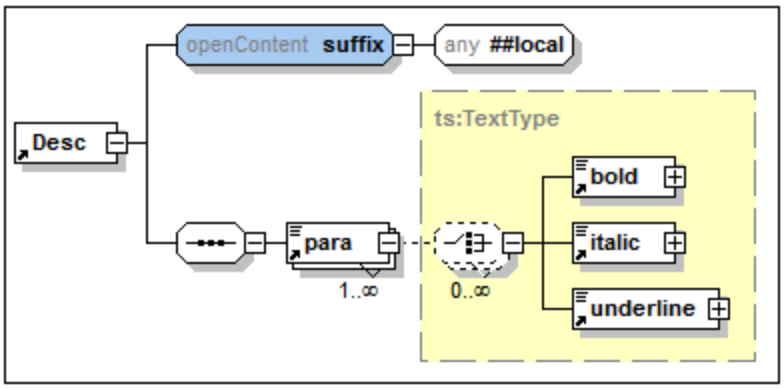
4.4.3.4 Modèles de contenu ouverts

Les modèles de contenu ouverts sont nouveaux dans XSD 1.1. Ils sont déclarés sur des types complexes et permettent à tout élément (c'est à dire, un élément non défini dans le modèle de contenu du type complexe) de se produire un nombre illimité de fois soit (i) entre les éléments définis dans le modèle de contenu, soit (ii) après les éléments définis dans le modèle de contenu.

L'élément `openContent` est un enfant du type complexe et se produit une fois avant le modèle de contenu du type complexe (*voir capture d'écran ci-dessous*).

Mode

L'élément `openContent` a un attribut `mode` obligatoire qui peut prendre les valeurs `interleave`, `suffix`, ou `none` (*voir capture d'écran ci-dessous*). La valeur par défaut est `interleave`.



La signification de ces valeurs est la suivante :

- Si `mode="interleave"` ou `mode="suffix"`, alors le contenu de l'élément de caractère générique (`xs:any`) doit être présent sans un nombre minimum ou maximum d'occurrences. Cela implique que tout nombre d'éléments non défini (caractères génériques) est autorisé.
- Si le mode est `interleave`, tout nombre d'éléments non définis peut se produire avant ou après les éléments définis individuels dans le modèle de contenu. Ils sont intercalés entre des éléments définis.
- Si le mode est `suffix`, tout nombre d'éléments non définis peut se produire après le dernier élément défini du modèle de contenu.
- Si le mode est `none`, aucun élément non défini (enfant `xs:any`) peut se produire ; le modèle de contenu n'est pas ouvert. La valeur `none` est utilisée pour contourner l'élément `defaultOpenContent` dont la portée est répartie sur tout le schéma.

Dans le Mode de Modèle de contenu, vous pouvez ajouter un contenu ouvert en tant qu'un enfant du type complexe (par le biais d'**Ajouter enfant** dans le menu contextuel). Spécifier le mode soit en double-cliquant la fenêtre `openContent` dans le diagramme (*voir capture d'écran ci-dessus*) et sélectionner une valeur (`interleave`, `suffix`, ou `none`), ou en sélectionnant une valeur dans l'Assistant à la saisie des Détails.

Propriétés de caractère générique (xs:any)

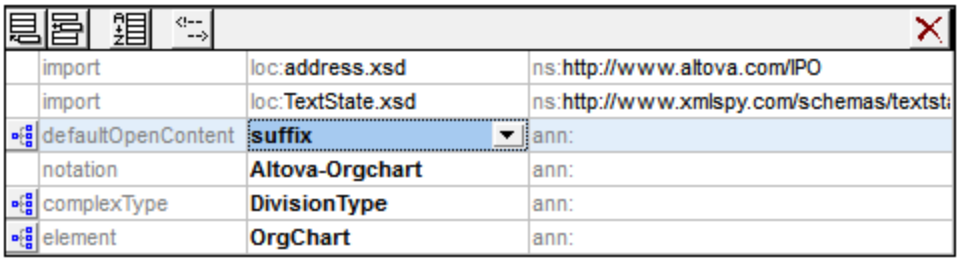
Les propriétés de caractère générique sont spécifiées dans l'Assistant à la saisie des Détails. Sélectionner le caractère générique dans le diagramme et saisir les valeurs de propriété dans l'Assistant à la saisie Détails.

Contenu ouvert par défaut

Cet élément `defaultOpenContent` est nouveau dans XSD 1.1 et spécifie que l'un ou plusieurs éléments non définis peut être ajouté à tout type complexe de contenu mixte ou element-only. Il est semblable à l'élément `openContent` (également nouveau dans XSD 1.1), la différence principale étant qu'alors que l'élément `openContent` s'applique à un seul type complexe, l'élément `defaultOpenContent` s'applique à tous les types complexes dans le schéma.

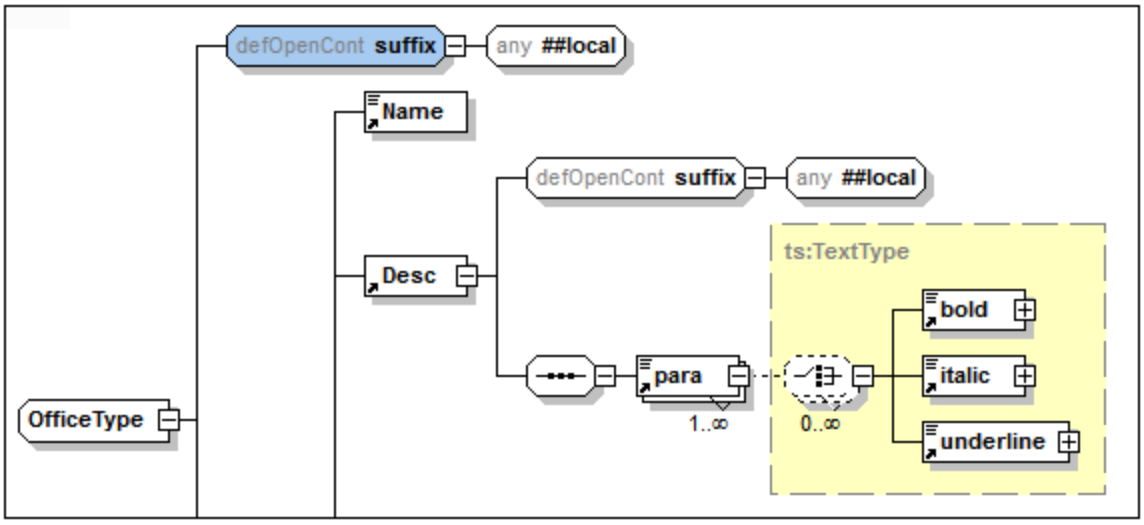
L'élément `defaultOpenContent` est un composant global et se produit une fois dans le document (*voir capture d'écran ci-dessous*), après Inclut, Importe, Redéfinit et Substitue, et avant les définitions des composants. Il a un attribut `mode` qui peut prendre une valeur de soit `interleave` ou `suffix`. Le défaut est `interleave`.

© 2016-2022 Altova GmbH

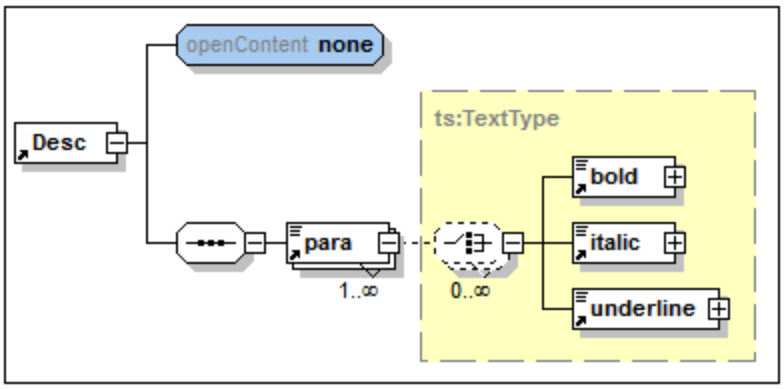


	import	loc:address.xsd	ns:http://www.altova.com/IPO
	import	loc:TextState.xsd	ns:http://www.xmlspy.com/schemas/textst
	defaultOpenContent	suffix	ann:
	notation	Altova-Orgchart	ann:
	complexType	DivisionType	ann:
	element	OrgChart	ann:

L'élément `defaultOpenContent` a un contenu de modèle que vous pouvez éditer dans le Mode de Modèle de contenu, exactement de la même manière que dans la définition de l'élément openContent (*voir ci-dessus*). Une fois déclaré, l'élément `defaultOpenContent` s'appliquera automatiquement à tous les types complexes dans le schéma et sera affiché dans leurs modèles de contenu. Dans la capture d'écran ci-dessous, vous pouvez voir que l'élément `defaultOpenContent` a été appliqué automatiquement aux types complexes `OfficeType` et `Desc`.

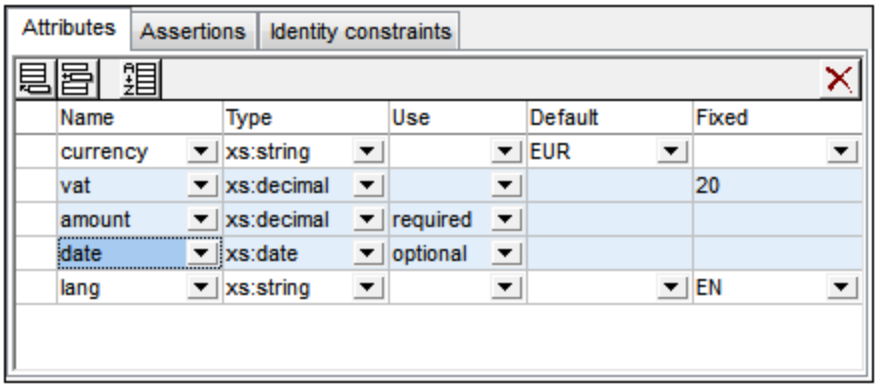


Afin de surmonter l'élément `defaultOpenContent` lorsqu'il est appliqué à un type complexe particulier, ajouter un élément enfant `openContent` à ce type complexe. Dans la capture d'écran ci-dessous, l'élément `Desc` avec l'élément `defaultOpenContent` (*voir capture d'écran ci-dessus*) s'est vu rajouter un élément `openContent` qui surmonte l'élément `defaultOpenContent`.



4.4.4 Attributs, Assertions et Contraintes d'identité

Le panneau Attributs/Assertions/Contraintes d'identité (AACID) (*capture d'écran ci-dessous*) est situé sous le panneau principal dans l'aperçu de Schéma et le Mode de Modèle de contenu. Le panneau et ses onglets sont fixes. En revanche, dans le Mode de Modèle de contenu, l'aperçu de chaque onglet peut être activé individuellement afin de consulter et éditer les composants des onglets dans le diagramme dans le Mode de Modèle de contenu au lieu du panneau AACID. Lorsque les aperçus des trois onglets sont activés dans le diagramme, le panneau AACID disparaît.



Les aperçus peuvent être basculés entre le panneau AACID et le diagramme par le biais du dialogue Configuration d'affichage du schéma (**Conception de schéma | Configurer Mode | onglet Élément**) ou en cliquant sur l'icône respective dans la barre outils Conception de schéma (*affiché ci-dessous*).

	Affiche les Attributes dans le Diagramme : dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des attributs entre le diagramme (activé) et l'onglet Attributs.
	Affiche les Assertions dans le Diagramme : dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des assertions entre le diagramme (activé) et l'onglet Assertions.

© 2016-2022 Altova GmbH

	Affiche les Contraintes dans le Diagramme : dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des IDC entre le diagramme (activé) et l'onglet Contraintes d'identité.

Utiliser les onglets

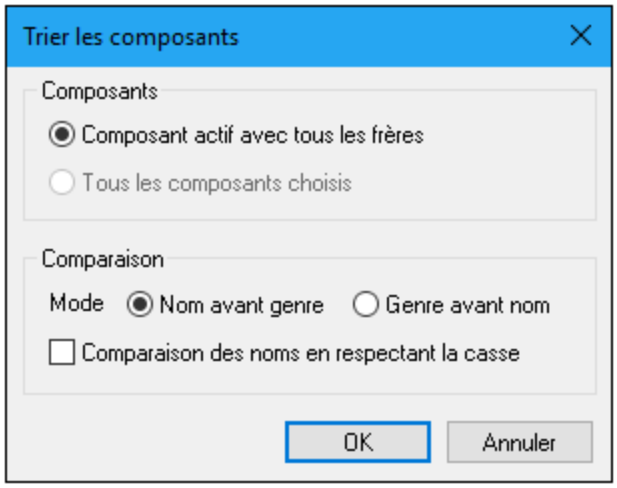
Les onglets dans le panneau AACID peuvent être activés individuellement selon le type de composant sélectionné dans le panneau principal supérieur de l'Aperçu de schéma ou du Mode de Modèle de contenu. Par example, puisqu'il est possible d'ajouter un attribut à un type complexe, l'onglet Attributs sera activé lorsqu'un type complexe est sélectionné dans le panneau principal. (Un onglet est considéré activé lorsque ses commandes sont activées.)

Vous trouverez une description de l'utilisation des onglets dans les sous-sections de cette section :

- Attributs, Groupes d'attributes, Caractères génériques d'attribut
- Assertions
- Contraintes d'identité

Trier les attributs et les contraintes d'identité

Vous pouvez trier les attributs et les contraintes d'identité dans leurs onglets respectifs en cliquant sur l'icône **Trier** dans la barre outils de l'onglet. Dans le dialogue Trier les composants qui s'affiche (*capture d'écran ci-dessous*), vous pouvez décider soit de trier le composant unique et ses frères soit l'ensemble des composants sélectionnés. Dans la capture d'écran ci-dessus par exemple, trois attributs ont été sélectionnés (*marqués en bleu*). Vous pouvez utiliser cliquer+**Shift** pour sélectionner une plage pour sélectionner et cliquer+**Ctrl** pour ajouter des composants supplémentaires à la sélection.



Une fois avoir sélectionné l'ensemble des composants que vous souhaitez trier, vous pouvez choisir un tri par ordre alphabétique du nom et puis du genre (*Nom avant le genre*), ou vice-versa (*Genre avant le nom*). L'ordre de tri est immédiatement mis en place dans le texte du document de schéma, il ne s'agit pas d'un masque d'interface.

4.4.4.1 Attributs, Groupes d'attribut, Caractères génériques d'attribut

Dans l'onglet Attributs du panneau Attributs/Assertions/Contraintes d'Identité (AAIDC) (*capture d'écran ci-dessous*), vous pouvez :

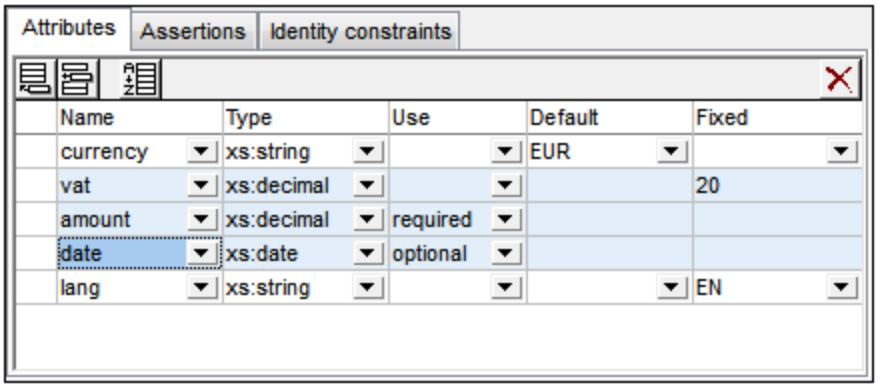
- Déclarer des attributs localement sur un type complexe sélectionné
- Référencer des groupes d'attribut pour une utilisation sur un type complexe sélectionné
- Définir des caractères génériques d'attribut sur le type complexe sélectionné

Note : Si vous avez choisi l'option d'afficher les attributs dans le diagramme (**Conception de schéma | Configurer affichage**) au lieu du panneau AACID, vous pouvez éditer les propriétés des attributs, des référence de groupe d'attribut et des caractères génériques d'attribut dans le diagramme et l'assistant à la saisie des Détails.

	*Affiche les Attributes dans le Diagramme :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des attributs entre le diagramme (activé) et l'onglet Attributs.

Attributs

Dans l'onglet Attributs du panneau Attributs/Assertions/Contraintes d'Identité (AAIDC) (*capture d'écran ci-dessous*), vous pouvez déclarer les attributs locaux des éléments et des types complexes et les attributs qui constituent les groupes d'attribut.



Pour créer les attributs, procéder comme suit :

1. Dans l'Aperçu de schéma, sélectionner le type complexe ou le groupe d'attribut pour lequel vous souhaitez créer l'attribut.
2. Dans l'onglet Attributs, cliquer sur l'icône **Ajouter** ou **Insérer** située en haut à gauche et sélectionner **Attribut**.
3. Dans la ligne qui est créée pour l'attribut, saisir les détails de l'attribut (nom, type, utilisation et valeur par défaut ou fixée). La propriété `name` est obligatoire et la valeur par défaut de `use` est `optional`. Le type de données et les propriétés de valeur par défaut/fixe sont optionnelles.

Note : Les attributs peuvent être ajoutés aux groupes d'attributs uniquement dans l'Aperçu de schéma, mais ils peuvent être ajoutés aux types complexes aussi bien dans l'Aperçu de schéma que dans le Mode de Modèle de contenu.

© 2016-2022 Altova GmbH

Note: Si un attribut a une définition de type, cliquer sur sa commande de menu contextuel **Aller à la définition de type** vous emmènera dans la définition de type. Dans le cas des types simples intégrés, une fenêtre de message apparaît qui contient des informations concernant le type simple.

Valeurs par défaut et valeurs fixes

Une valeur par défaut ou fixe, si spécifiée dans une déclaration d'attribut, est appliquée dans le document d'instance lorsque cet attribut est absent dans le document d'instance. Il est possible de spécifier soit une valeur par défaut soit une valeur fixe, mais pas les deux (*voir capture d'écran ci-dessus*). La valeur par défaut ou fixe doit être valide conformément au type de données de l'attribut. Si `use` est défini sur `required`, alors ni la valeur par défaut ni la valeur fixe n'est autorisée.

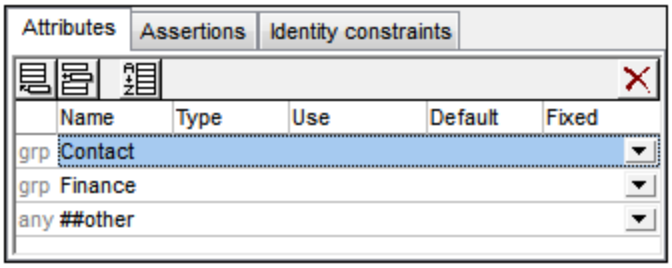
Veuillez noter :

- *Valeurs par défaut :* une valeur par défaut est insérée uniquement si l'attribut est manquant. Si l'attribut est présent et a une valeur valide, la valeur par défaut n'est pas insérée. Si la valeur dans les documents d'instance est invalide, une erreur est générée.
- *Valeurs fixes :* une valeur fixe est appliquée non seulement lorsque l'attribut est manquant, mais aussi si la valeur dans le document d'instance n'est pas égale à la valeur fixe spécifiée dans la déclaration de l'attribut.

Note : Les valeurs par défaut et fixes peuvent être spécifiées aussi bien sur les attributs locaux que globaux. Sur les attributs locaux, ils peuvent être définis aussi bien dans l'onglet Attributs du panneau AACID (*capture d'écran ci-dessus*) que dans l'Assistant à la saisie des Détails. Sue les attributs globaux, ils peuvent être spécifiés dans l'Assistant à la saisie des Détails.

Références du groupe d'attribut

Si un groupe d'attribut global a été déclaré, vous pouvez ajouter une référence à ce groupe d'attributs dans la définition d'un type complexe. Pour ce faire, sélectionner le composant de type complexe dans l'Aperçu de schéma ou le Mode de Modèle de contenu, puis cliquer sur l'icône **Ajouter** ou **Insérer** située en haut à gauche de l'onglet Attributs du panneau AACID et sélectionner **Groupe d'attribut**. Dans la ligne du groupe d'Attribut qui a été créée, saisir le nom du groupe d'attribut à référencer (*voir capture d'écran ci-dessous, qui a deux références de groupe d'attribut*). Vous pouvez ajouter plusieurs groupes d'attribut.

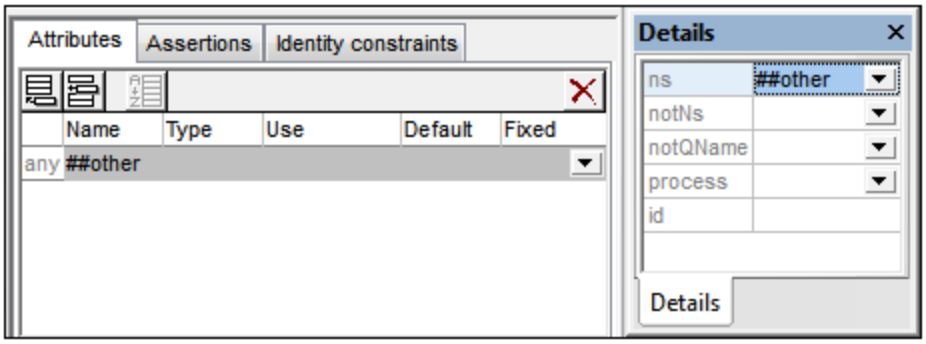


Lorsque le groupe d'attribut a été sélectionné dans l'onglet Attributs, ses propriétés peuvent aussi être éditées dans l'Assistant à la saisie des Détails.

Caractères génériques d'attribut : anyAttribute

Un caractère générique d'attribut peut être ajouté à un type complexe pour permettre l'utilisation de tout attribut sur un élément. Un caractère générique d'attribut est défini par un seul élément `anyAttribute`. Il permet à tout nombre d'attributs provenant de l'espace de noms spécifié de se produire sur l'élément dans le document d'instance.

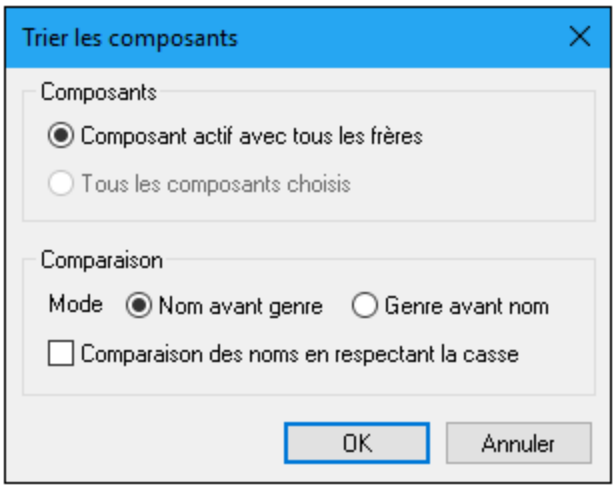
Ajouter un caractère générique d'attribut en sélectionnant le composant de type complexe dans l'Aperçu de schéma ou le Mode de Modèle de contenu, puis cliquer sur l'icône **Ajouter** ou **Insérer** située en haut à gauche de l'onglet Attributs du panneau AACID et sélectionner **Tout attribut**. Une ligne pour le caractère d'attribut `anyAttribute` est créé (*voir capture d'écran ci-dessous*).



Dans l'onglet Attributs, vous pouvez déterminer la propriété `namespace` de `anyAttribute`. Une fois le caractère générique d'attribut sélectionné dans l'onglet Attributs, vous pouvez déterminer des propriétés supplémentaires dans l'Assistant à la saisie des Détails (*voir capture d'écran ci-dessus*). Veuillez noter que les propriétés `notNamespace` et `notQName` sont des fonctions XSD 1.1 et ne seront pas disponibles dans le mode XSD 1.0.

Les attributs de tri et les groupes d'attribut

Vous pouvez trier les attributs et les groupes d'attribut dans l'onglet Attributs en cliquant l'icône **Trier** dans la barre outils de l'onglet. Dans le dialogue Trier les composants qui s'affiche (*capture d'écran ci-dessous*), vous pouvez choisir de trier soit le composant unique sélectionné et ses frères, soit l'ensemble des composants sélectionnés. Vous pouvez utiliser **Shift**+clic pour sélectionner une plage et **Ctrl**+clic pour ajouter les composants supplémentaires à la sélection.



Après avoir déterminé la plage, vous pouvez choisir de trier toute la plage des attributs et des groupes des attributs par ordre alphabétique (*Nom avant genre*), ou des attributs triés alphabétiquement avant les groupes d'attribut triés alphabétiquement.

L'ordre de tri est immédiatement mis en place dans le texte du document de schéma, il ne s'agit pas d'un masque d'interface.

© 2016-2022 Altova GmbH

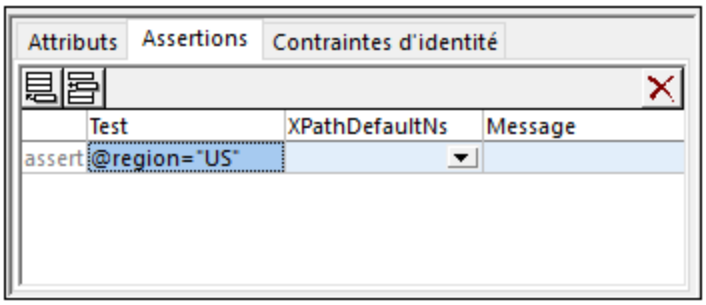
Note : Les caractères génériques d'attribut ne seront pas inclus dans la plage à trier puisqu'ils doivent toujours se produire à la fin d'Une déclaration de type complexe et seul un caractère générique d'attribut est autorisé dans une déclaration de type complexe unique.

4.4.4.2 Assertions

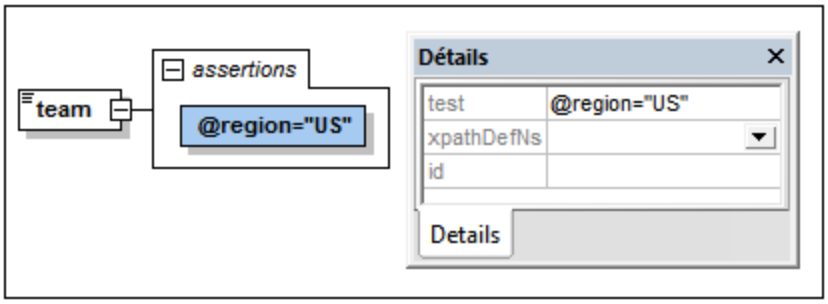
Les assertions décrites dans cette section sont des **assertions de types complexes**. Une telle assertion est définie dans un élément `xs:assert` et sert de contrainte de validité du type complexe. (L'autre type d'assertion est une assertion de type simple qui est définie dans un élément `xs:assertion` et est créée et éditée dans l'Assistant à la saisie Facettes d'un type simple.)

Dans le Mode Schéma, les assertions de type complexe peuvent être créées et éditées par le biais des points d'accès de la GUI suivants :

- *Dans l'Aperçu de schéma :* dans l'onglet Assertions du panneau Attributs/Assertions/Contrainte d'identité (AAIDC) (*capture d'écran ci-dessous*).



- *Dans Mode de Modèle de contenu :* les assertions peuvent être éditées dans l'onglet Assertions (*capture d'écran ci-dessus*) ou dans le diagramme (*capture d'écran ci-dessous*). Afin de sélectionner l'option d'affichage du diagramme, cliquer sur l'icône **Afficher les assertions dans le diagramme** dans la barre outils de la Conception de schéma. Dans le diagramme, sélectionner la fenêtre Assertion du type complexe ou de l'élément contenu-complexe. Ensuite, saisir ou éditer la définition de l'Assertion dans l'Assistant à la saisie Détails.



	*Affiche les Assertions dans le Diagramme :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des assertions entre le diagramme (activé) et l'onglet Assertions.

Note: Les assertions sont une fonction XSD 1.1. Les fonctions d'édition Assertions seront donc uniquement disponibles dans le mode XSD 1.1.

Étendue de l'assertion

L'expression XPath utilisée pour définir la contrainte de l'assertion doit se trouver dans le cadre de l'étendue du type complexe sur lequel il est défini. Donc si l'expression XPath est nécessaire pour accéder à un nœud particulier, l'assertion doit être définie sur un ancêtre de ce nœud.

Ajouter et supprimer des assertions

Un type complexe peut avoir des assertions multiples. L'expression XPath de chaque assertion doit évaluer à une valeur booléenne `true` pour l'élément dans le document d'instance pour être valide. Pour ajouter une assertion à un type complexe, procéder comme suit :

- *Dans l'Aperçu de schéma :* choisir le type complexe. Ensuite, dans l'onglet Assertions du panneau AACID (*voir capture d'écran ci-dessus*), cliquer sur l'icône **Ajouter** ou **Insérer** située en haut à gauche de l'onglet. Vous pouvez ajouter plusieurs assertions. Pour supprimer une assertion, la sélectionner et cliquer sur l'icône **Supprimer** située en haut à droite de l'onglet.
- *Dans Mode de Modèle de contenu (voir capture d'écran ci-dessus):* cliquer avec la touche de droite sur le type complexe et choisir **Ajouter enfant | Assertion**. En alternative, cliquer avec la touche de droite sur une assertion existante dans le diagramme de type complexe et sélectionner **Ajouter | Assertion** ou **Insérer | Assertion**. Vous pouvez ajouter plusieurs assertions à un type complexe. Pour supprimer une assertion, la sélectionner et appuyer sur la touche **Supprimer**.

Définir l'expression XPath de l'assertion

L'expression XPath d'une assertion de type complexe définit la contrainte de validation à appliquer sur le type complexe dans le document d'instance. Par exemple, dans la capture d'écran ci-dessus, l'assertion est dans l'élément de type complexe `team` et l'expression XPath de l'assertion est : `@region="US"`. Dans le document de Schéma XML document, l'assertion apparaît en tant que :

```
<xs:assert test='@region="US"'/>
```

L'assertion spécifie que, dans le document d'instance, l'élément `team` doit avoir un attribut `region` avec une valeur de `US`. Dans le cas contraire, le document sera non valide.

Veuillez noter les points suivants :

- Les expressions XPath doivent être écrites dans le langage XPath 2.0
- Les nœuds testés dans l'expression XPath doivent se trouver dans le cadre de l'assertion (*voir ci-dessus*)
- Si une expression n'évalue pas à une valeur booléenne `true/false`, la valeur retournée est convertie en une valeur booléenne. Une séquence non-vide est convertie en `true`, alors qu'une séquence vide est convertie en `false`.
- Les erreurs de syntaxe dans l'expression sont marquées en affichant l'expression en rouge. Les erreurs de contexte ne sont pas marquées. Par exemple, si l'expression XPath teste un attribut et que cet attribut n'est pas défini dans le schéma, aucune erreur ne sera marquée.

Le message de l'assertion

Il est toujours utile de proposer une explication de l'assertion avec sa définition de manière à ce qu'un message approprié puisse être affiché dans le cas ou une assertion n'est pas remplie lors de la validation d'un document d'instance XML. Puisque la spécification de Schéma XML n'impose aucun message, XMLSpy permet de fournir un message dans l'espace de noms `http://www.altova.com/xml-schema-extensions` Altova xml-schema-

© 2016-2022 Altova GmbH

extensions (ou tout autre espace de noms) avec la définition de l'assertion et de l'utiliser dans la validation du document d'instance XML. Par exemple :

```
<xs:assert test="count(//MyNode) ge 1" altova:message="There must be at least one MyNode element"/> or
<xs:assertion test="count(//MyNode) ge 1" altova:message="There must be at least one MyNode element"/>
```

Si la restriction spécifiée dans l'assertion n'est pas remplie, le moteur de validation de XMLSpy s'affichera, ainsi qu'avec le message de validation-erreur, le message associé avec l'assertion en tant qu'indice. Le validateur rapportera la valeur d'un attribut `assert/@message` ou d'un attribut `assertion/@message` quels que soient les espaces de noms dans lesquels se trouvent l'attribut `message`. Néanmoins, dans le Mode Schéma, vous pouvez éditer uniquement les attributs `message` qui se trouvent dans l'espace de noms Altova xml-schema-extension. Pour éditer les attributs `message` dans d'autres espaces de noms, utiliser le Mode Texte.

Voir les Messages d'assertion pour plus de détails.

Utiliser `xpathDefaultNamespace`

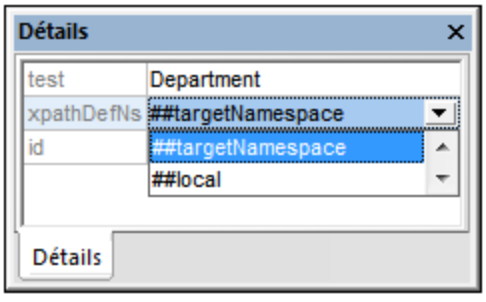
Un espace de noms par défaut déclaré dans le document de Schéma XML est l'espace de noms par défaut du document de Schéma XML. Il s'applique aux noms d'élément sans préfixe dans le document du schéma mais pas aux noms d'éléments sans préfixes dans les expressions XPath dans le document schéma.

L'attribut `xpathDefaultNamespace` (une nouvelle fonction dans XSD 1.1) est le mécanisme utilisé pour spécifier l'espace de noms auxquels appartiennent les noms d'élément sans préfixe dans les expressions XPath.

Les espaces de noms XPath par défaut portent sur les éléments de Schéma XML sur lesquels ils sont déclarés. L'attribut `xpathDefaultNamespace` peut se produire sur les éléments de Schéma XML 1.1 suivants :

- `xs:schema`
- `xs:assert` et `xs:assertion`
- `xs:alternative`
- `xs:selector` et `xs:field` (dans les contraintes d'identité)

`xpathDefaultNamespace` sur `xs:schema` est déterminé, dans le mode XSD 1.1, dans le dialogue Paramètres de Schéma (**Conception de Schéma | Paramètres de schéma**). En ce qui concerne les autres éléments contenus dans la liste ci-dessus, l'attribut `xpathDefaultNamespace` est déterminé dans leurs Assistants à la saisie Détails respectifs (*voir capture d'écran ci-dessous par exemple*).



La déclaration d'espace de noms XPath par défaut dans `xs:schema`, déclare l'espace de noms XPath par défaut pour l'étendue de tout le schéma. Vous pouvez contourner cette déclaration sur les éléments dont l'attribut

`xpathDefaultNamespace` est autorisé (*voir liste ci-dessus*).

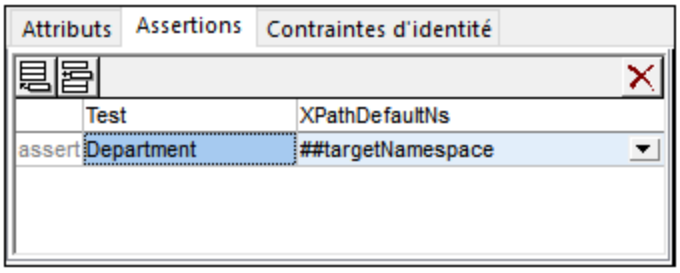
Au lieu de contenir un espace de noms réel, l'attribut `xpathDefaultNamespace` peut contenir un des trois mots-clé :

- `##targetNamespace`: l'espace de noms XPath par défaut sera le même que l'espace de noms cible du schéma
- `##defaultNamespace`: l'espace de noms XPath par défaut sera le même que l'espace de noms par défaut du schéma
- `##local`: il n'y a pas d'espace de noms XPath par défaut

Si aucun espace de noms XPath par défaut n'est déclaré dans le document, des éléments sans préfixes dans les expressions XPath ne seront pas dans des espaces de noms.

Note : La déclaration d'espace de noms XPath par défaut ne s'applique pas aux attributs.

Pour les expressions XPath dans les assertions, vous pouvez aussi spécifier l'espace de noms XPath par défaut sur la définition de l'assertion. Dans l'onglet Assertions du panneau Attributs/Assertions/Contrainte d'identité (AAIDC) (*capture d'écran ci-dessous*), sélectionner le mot-clé nécessaire depuis la liste déroulante du champ *XPathDefaultNS*.



L'espace de noms sélectionné se trouvera dans l'étendue de l'assertion.

4.4.4.3 Contraintes d'identité

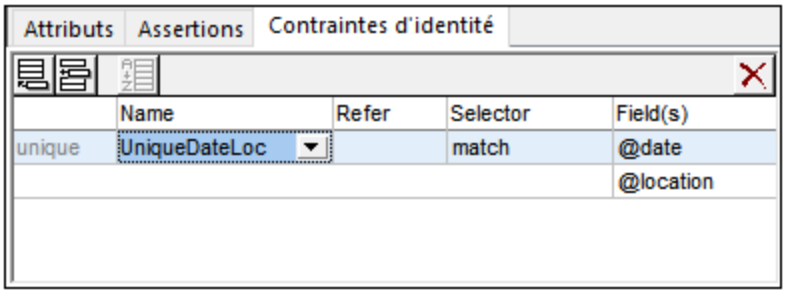
Les contraintes d'identité (CID) peuvent être définies dans des déclarations globales ou locales. Elles spécifient l'unicité des nœuds et permettent le référencement correct entre des nœuds uniques.

Mécanismes de déclaration

Les mécanismes suivants sont disponibles pour la définition d'une CID (`unique`, `key`, `keyref`):

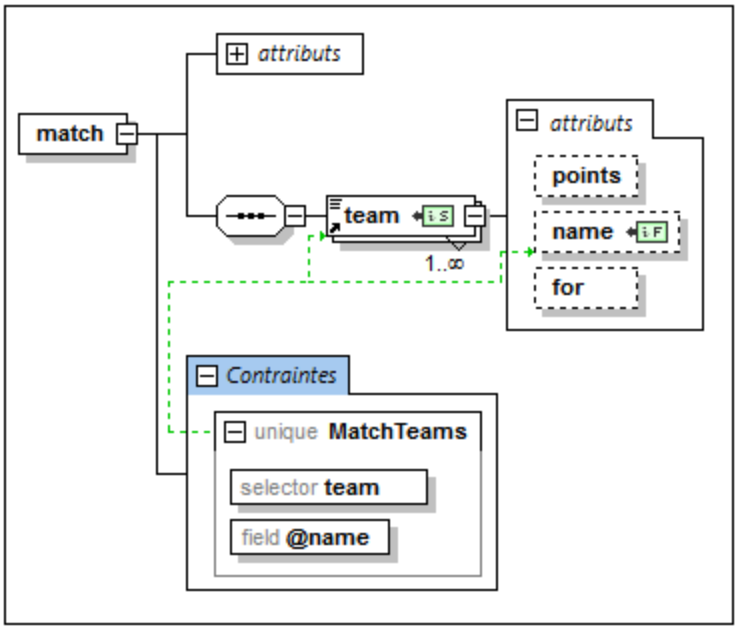
- Dans l'Aperçu de schéma, les CID peuvent être déclarés sur des éléments globaux. Sélectionner un élément global et définir les CID dans l'onglet Contraintes d'identité du panneau Attributs/Assertions/Contraintes d'identité (AACID) (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Ajouter une CID (`unique`, `key`, `keyref`) en utilisant l'icône **Insérer** ou **Ajouter** de l'onglet Contraintes d'identité. Ces icônes peuvent aussi être utilisées pour ajouter un `field` à la CID sélectionnée. Utiliser l'icône **Supprimer** pour supprimer le `field` ou CID sélectionné.

- Dans le Mode Modèle de contenu d'un élément global, les CID peuvent être définis sur l'élément global ou sur un élément descendant local. Dans ce mode, les CID peuvent être éditées soit dans l'onglet Contraintes d'identité (*capture d'écran ci-dessus*) soit dans la fenêtre *Contraintes* d'un élément dans le diagramme (*capture d'écran ci-dessous*, dans laquelle l'élément `match` a une contrainte unique qui comprenant un sélecteur `team`). Cette alternative peut être sélectionnée dans le dialogue de Configuration d'affichage du schéma (**Conception du schéma | Configurer affichage**). En alternative, vous pouvez cliquer sur l'icône **Afficher les contraintes dans le diagramme** située dans la barre d'outils de la Conception de schéma. Le diagramme fournit une représentation graphique des CID et une fonction d'édition par glisser/déposer.



Pour ajouter une CID (`unique`, `key`, `keyref`) dans le diagramme lorsque le mode de diagramme pour les CID est activé, cliquer avec la touche de droite sur l'élément à mettre sous contrainte et choisir **Ajouter enfant | [CID]** depuis le menu contextuel. L'item `field` sera activé dans le menu contextuel uniquement lorsqu'une CID est sélectionnée dans le diagramme. Appuyer sur la touche **Supprimer** pour supprimer le `field` ou la CID sélectionnés.

Vous disposez de trois méthodes possibles pour saisir l'expression XPath dans les fenêtres `selector` et `field` : (i) en la saisissant, (ii) en sélectionnant le nœud requis depuis une liste déroulante qui

apparaît automatiquement lorsque vous cliquez sur la fenêtre `selector` ou `field`, ou (iii) en glissant le nœud cible dans la fenêtre `selector` ou `field` et en la déposant à l'intérieur lorsque la fenêtre change de couleur ; l'expression XPath sera créée automatiquement.

Note : En outre, un Aperçu de toutes les contraintes d'identité dans le schéma est disponible dans l'onglet Contraintes d'identité de l'Assistant de saisie des Composants.

Icônes de contrainte d'identité

	*Affiche les Contraintes dans le Diagramme :* dans la barre outils de Conception de Schéma et activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des CID entre le diagramme (activé) et l'onglet Contraintes d'identité.
	*Visualiser les Contraintes d'identité :* activé dans le Mode de Modèle de contenu. Active et désactive l'affichage des informations de CID.
	*Nœud de sélecteur, nœud de champ :* indiqué dans les fenêtres de nœud dans le diagramme, ces deux icônes identifient, respectivement, le nœud sélectionné (dans les CID) par l'expression XPath pour `selector` et pour `field`.

Visualiser les CID

Lorsque l'icône Visualiser les Contraintes d'identité  est activée, les informations de la CID sont affichées dans le diagramme et peuvent être visualisées plus confortablement. Lorsque la visualisation est activée, les nœuds sélectionnés par les expressions XPath `selector` et `field` sont indiqués au moyen d'icônes dans leurs fenêtres (*voir la section sur les icônes ci-dessus*), et la fenêtre CID est connectée à son sélecteur et ses fenêtres de champ par des lignes vertes (*voir capture d'écran ci-dessus*).

L'icône Visualiser les Contraintes d'identité permet également d'activer la fonction de validation de CID dans le Mode Schéma. Si une expression XPath est incorrecte ou si une CID est incorrecte, les erreurs sont indiquées par du texte rouge, les avertissements sont affichés en orange. Lors de la validation du document de Schéma XML, les messages d'erreur ou d'avertissement sont affichés dans la fenêtre Messages.

Vous pouvez aussi désactiver la validation en désactivant l'icône Visualiser les Contraintes d'identité .

Liste XML

Les exemples CID ci-dessous indiqués dans cette section sont basés sur le document d'instance valide suivant.

```
<results xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="Scores.xsd">
   <!----- Groups ----->
   <group id="A">
          <team name="Brazil"/>
          <team name="Germany"/>
          <team name="Italy"/>
          <team name="Holland"/>
   </group>
```

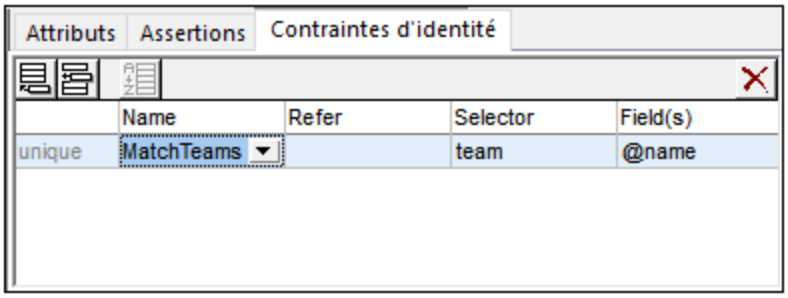
 © 2016-2022 Altova GmbH

```
    <group id="B">
           <team name="Argentina"/>
           <team name="France"/>
           <team name="England"/>
           <team name="Spain"/>
    </group>
    <!----- Matches ----->
    <match group="A" date="2012-06-12" location="Munich">
           <team name="Brazil" for="2" points="3"/>
           <team name="Germany" for="1" points="0"/>
    </match>
    <match group="A" date="2012-06-12" location="Frankfurt">
           <team name="Italy" for="2" points="1"/>
           <team name="Holland" for="2" points="1"/>
    </match>
    <match group="B" date="2012-06-13" location="Munich">
           <team name="Argentina" for="2" points="3"/>
           <team name="France" for="0" points="0"/>
    </match>
    <match group="B" date="2012-06-13" location="Berlin">
           <team name="England" for="0" points="1"/>
           <team name="Spain" for="0" points="1"/>
    </match>
</results>
```

Contraintes d'unicité (unique)

Une contrainte d'unicité spécifie que la valeur d'un élément ou d'un attribut (ou d'un ensemble d'éléments et/ou d'attributs) doit être unique dans le cadre d'une étendue définie. Dans la liste XML indiquée ci-dessus, nous souhaitons nous assurer que les deux équipes se disputant un match ne sont pas deux fois la même équipe. Donc, dans le cadre de l'élément `match`, nous contraignons les valeurs du nœud `team/@name` à être uniques. Pour ce faire, nous procédons comme suit :

1. Dans l'Aperçu Schéma, choisir l'élément `match`. Celui-ci représentera donc l'étendue de la définition de la contrainte d'identité.
2. Dans l'onglet Contrainte d'identité, cliquer sur l'icône **Ajouter** ou **Insérer** située en haut à gauche de l'onglet, et, dans le menu qui s'affiche, cliquer **Unique**. Cela rajoute une ligne pour la contrainte d'unicité (*voir capture d'écran ci-dessous*).



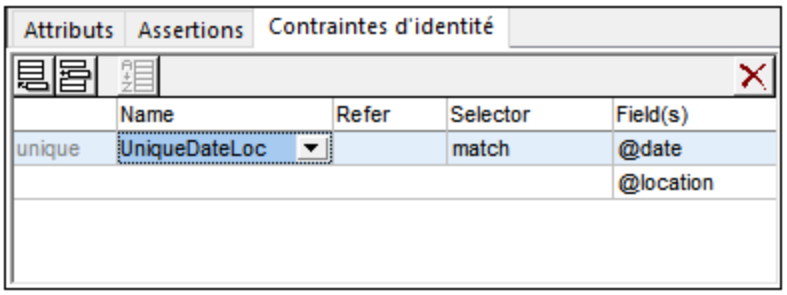
3. Nommer la contrainte d'identité. (Dans la capture d'écran ci-dessous, `MatchTeams` est le nom.)
4. Saisir une expression XPath dans la colonne *Selector* pour sélectionner l'élément `team`. Veuillez noter que l'élément `match` est le nœud contextuel. L'élément `team` devient maintenant le sélecteur de la CID, c'est à dire, le nœud auquel la contrainte d'unicité s'applique.

5. Dans la colonne *Field*, saisir le nœud `@name` qui doit être unique. La valeur de ce nœud est la valeur qui doit être unique.

La contrainte d'unicité décrite ci-dessus spécifie que dans le cadre de chaque élément `match`, chaque élément `team` doit avoir un attribut-valeur unique `@name`.

Vous pouvez ajouter des champs supplémentaires pour vérifier l'unicité. Par exemple, une contrainte d'unicité peut être définie sur l'élément `results` pour contrôler que toutes les correspondances présentent une combinaison unique de date et d'emplacement : pas plus d'une correspondance ne peut se produire à un endroit à la même date. La contrainte d'unicité doit avoir, pour chaque élément `match` (le sélecteur), sa combinaison de valeurs `@date` et `@location` uniques dans le cadre de l'élément `results`.

Définir la contrainte d'unicité dans l'élément `results` de la même manière que ce qui a été décrit plus haut. Le sélecteur sera `match`, et les champs seront `@date` et `@location` (*voir capture d'écran ci-dessous*). Ajouter le second champ en cliquant sur l'icône **Ajouter** et puis **Champ**.

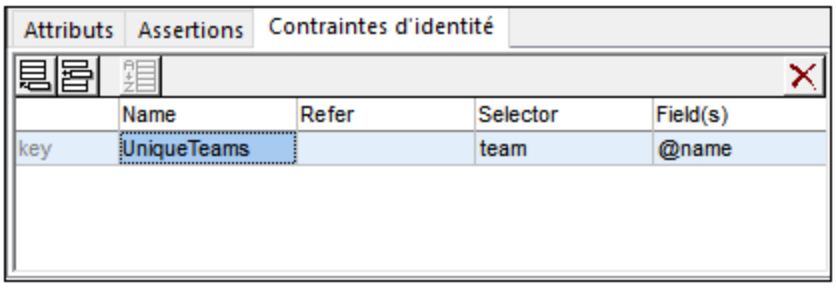


Note : La colonne *Refer* dans l'onglet Contrainte d'identité est uniquement activé pour les contraintes `keyref`, pas pour les contraintes `unique` ou `key`.

Contraintes de clé (key)

Une contrainte de clé spécifie: (i) que la valeur d'un élément ou d'un attribut (ou d'un ensemble d'éléments et/ou d'attributs) doit être unique dans le cadre d'une étendue définie, et (ii) que ces éléments de champ et/ou attributs doivent être présents dans le document XML d'instance ; c'est pourquoi, les éléments ou attributs optionnels ne devraient pas être sélectionnés en tant que champs d'une contrainte de clé. Une contrainte de clé est donc (dans le point (i) ci-dessus) exactement identique à une contrainte d'unicité. Elle stipule une contrainte supplémentaire : que ses éléments/attributs de champ doivent être présents dans le document XML.

La capture d'écran ci-dessous montre une contrainte de clé définie sur un élément `match` qui est semblable à la première contrainte d'unicité décrite ci-dessus.



© 2016-2022 Altova GmbH

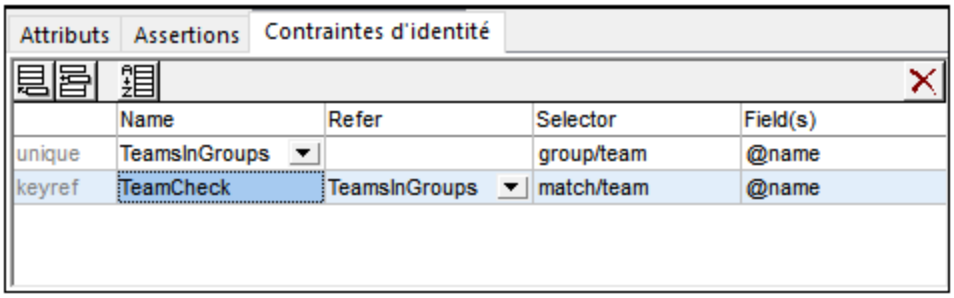
Cette contrainte de clé spécifie que dans le cadre de chaque élément `match`, chaque élément `team` doit avoir un attribut-valeur unique `@name`. De plus, elle spécifie que l'attribut `@name` doit être présent dans chaque élément `match/team`.

Note: La colonne *Refer* dans l'onglet Contrainte d'identité est uniquement activée pour les contraintes `keyref`, pas pour les contraintes `unique` ou `key`.

Références de clé (keyref)

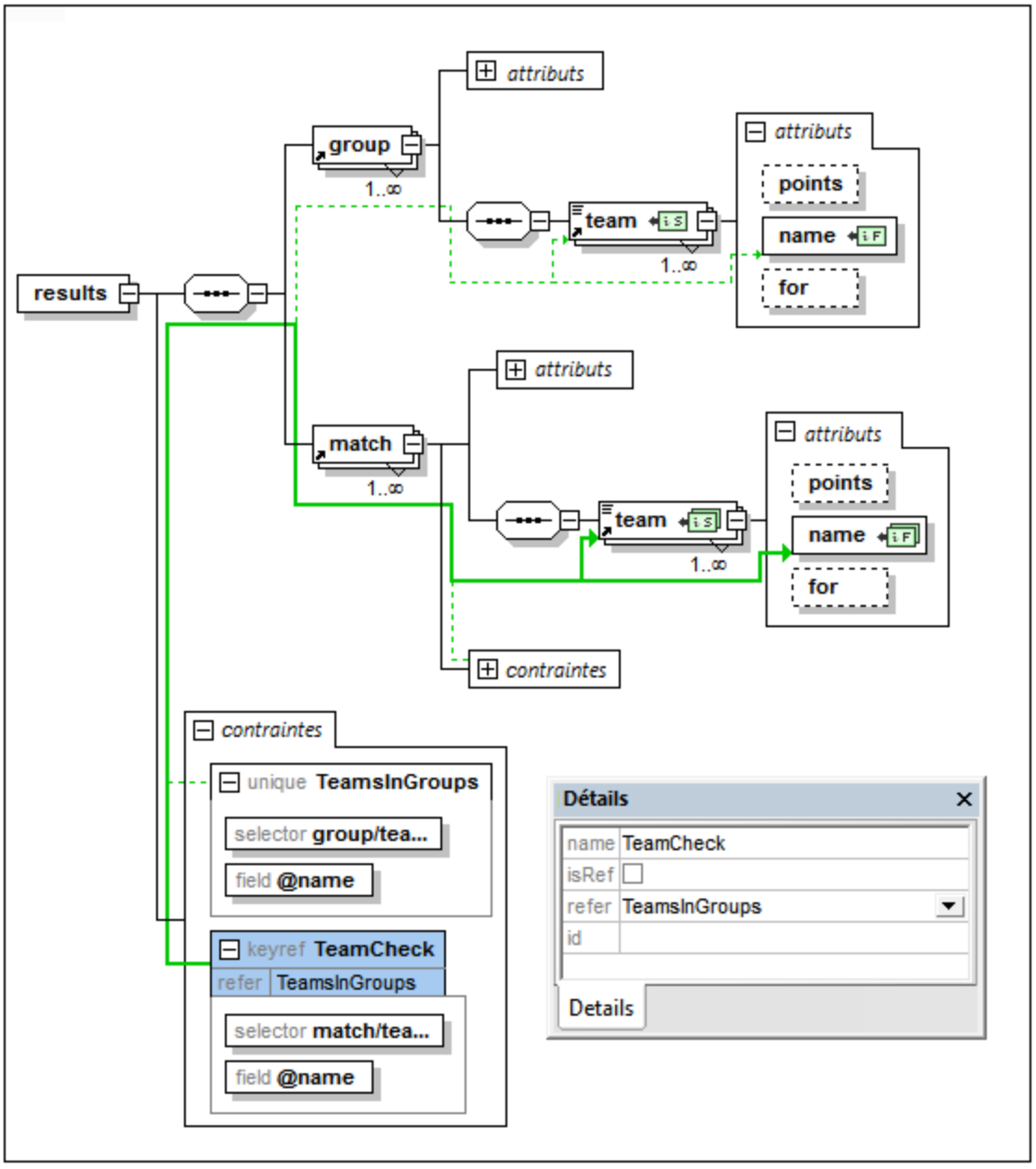
Les références de clé vérifient un ensemble de valeurs dans un document d'instance les uns par rapport aux autres. Dans notre liste XML, par exemple (*voir liste ci-dessus*), nous pouvons utiliser une référence de clé pour vérifier si les équipes qui s'affrontent dans les matches font partie des équipes recensées dans les groupes respectifs. Si ce n'est pas le cas, le document XML sera invalide.

Tout d'abord, nous créons une constante d'unicité ou une contrainte de clé. La capture d'écran ci-dessous montre une contrainte d'unicité (`unique`), `TeamsInGroups`, créée dans l'élément `results`. Cette contrainte stipule que chaque `team` dans le `group` a un attribut `@name` unique.



Ensuite, nous créons la référence de clé (`keyref`), `TeamCheck`, qui sélectionne l'enfant `team` de `match` et vérifie si son attribut-valeur `@name` est présent parmi les valeurs retournées par `TeamsInGroups`, qu'il référence (dans la colonne *Refer*).

La capture d'écran ci-dessous montre l'affichage graphique de cette référence clé (marquée en bleu) ainsi que l'Assistant à la saisie Détails (dans lequel vous pouvez aussi sélectionner la CID référencée). Les relations entre la CID sélectionnée est indiquée par une ligne épaisse verte, alors que les CID non sélectionnés sont affichés avec une ligne verte pointillée. De même, pour chaque contrainte d'identité, les nœuds sélectionnés par l'expression XPath pour `selector` et `field` sont affichés avec les icônes iS et iF respectivement. Si un nœud est réduit, la ligne de relation qui y mène se termine par une ellipse.



Utiliser `xpathDefaultNamespace`

Un espace de noms par défaut déclaré dans le document de Schéma XML est l'espace de noms par défaut du document de Schéma XML. Il s'applique aux noms d'élément sans préfixe dans le document du schéma mais pas aux noms d'éléments sans préfixes dans les expressions XPath dans le document schéma.

L'attribut `xpathDefaultNamespace` (une nouvelle fonction dans XSD 1.1) est le mécanisme utilisé pour spécifier l'espace de noms auxquels appartiennent les noms d'élément sans préfixe dans les expressions XPath.

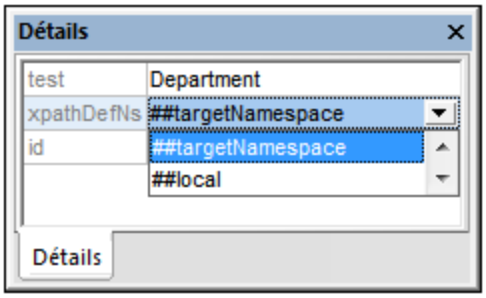
Les espaces de noms XPath par défaut portent sur les éléments de Schéma XML sur lesquels ils sont déclarés. L'attribut `xpathDefaultNamespace` peut se produire sur les éléments de Schéma XML 1.1 suivants :

- `xs:schema`

© 2016-2022 Altova GmbH

- `xs:assert` et `xs:assertion`
- `xs:alternative`
- `xs:selector` et `xs:field` (dans les contraintes d'identité)

`xpathDefaultNamespace` sur `xs:schema` est déterminé, dans le mode XSD 1.1, dans le dialogue Paramètres de Schéma (**Conception de Schéma | Paramètres de schéma**). En ce qui concerne les autres éléments contenus dans la liste ci-dessus, l'attribut `xpathDefaultNamespace` est déterminé dans leurs Assistants à la saisie Détails respectifs (*voir capture d'écran ci-dessous par exemple*).



La déclaration d'espace de noms XPath par défaut dans `xs:schema`, déclare l'espace de noms XPath par défaut pour l'étendue de tout le schéma. Vous pouvez contourner cette déclaration sur les éléments dont l'attribut `xpathDefaultNamespace` est autorisé (*voir liste ci-dessus*).

Au lieu de contenir un espace de noms réel, l'attribut `xpathDefaultNamespace` peut contenir un des trois mots-clé :

- `##targetNamespace`: l'espace de noms XPath par défaut sera le même que l'espace de noms cible du schéma
- `##defaultNamespace`: l'espace de noms XPath par défaut sera le même que l'espace de noms par défaut du schéma
- `##local`: il n'y a pas d'espace de noms XPath par défaut

Si aucun espace de noms XPath par défaut n'est déclaré dans le document, des éléments sans préfixes dans les expressions XPath ne seront pas dans des espaces de noms.

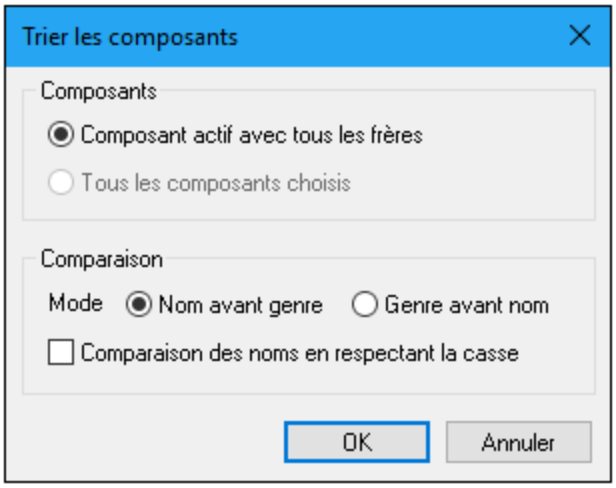
Note : La déclaration d'espace de noms XPath par défaut ne s'applique pas aux attributs.

ID des contraintes d'identité

Une ID peut être attribuée à une contrainte d'identité, son sélecteur, et/ou le/s champ/s. Pour attribuer une ID, sélectionner le composant requis et, par le biais de l'Assistant à la saisie, ajouter l'ID dans la ligne `id`.

Trier les contraintes d'identité

Vous pouvez trier les CID dans l'onglet Contrainte d'identité en cliquant sur l'icône **Trier** dans la barre outils de l'onglet. Dans le dialogue Trier les composants qui apparaît (*capture d'écran*), vous pouvez choisir de trier soit le composant unique sélectionné et ses frères soit le jeu des composants sélectionnés. Vous pouvez utiliser la combinaison de touches clic+**Shift** pour sélectionner une plage et clic+**Ctrl** pour ajouter des composants supplémentaires à la sélection.



Une fois avoir déterminé la plage, vous pouvez choisir de trier toute la plage par ordre alphabétique (*Nom avant type*), ou de l'organiser par type dans l'ordre alphabétique (c'est à dire : unicité de la contrainte en premier, puis les contraintes, puis les références clé).

L'ordre de tri est mis en place dans le texte du schéma.

4.4.5 Assistants à la saisie dans le Mode Schéma

Vous disposez de trois Assistants à la saisie dans le Mode Schéma. Ceux-ci sont décrits en détail dans la sous-section de cette section:

- Assistants à la saisie Composants
- Assistants à la saisie Détails
- Assistants à la saisie Facettes

Les Assistants à la saisie sont les mêmes dans le Mode Aperçu de Schéma et Modèle de contenu. Ils vous permettent d'ajouter et d'éditer graphiquement des définitions des composants de schémas. Généralement, vous pouvez glisser des composants depuis un Assistant à la saisie ou sélectionner un composant dans le design puis y définir les propriétés dans un Assistant à la saisie.

4.4.5.1 Composants

L'Assistant à la saisie Composants dans le Mode Schéma (*voir captures d'écran ci-dessous*) dessert trois objectifs :

- Organiser des composants globaux dans un mode arborescent par le type et l'espace de noms du composant *(voir captures d'écran ci-dessous)*. Cela permet des aperçus organisés de tous les composants globaux et des composants globaux conformément à l'espace de noms.
- Vous permettre de parcourir et d'afficher le Mode de Modèle de contenu d'un composant global—si le composant a un modèle de contenu. Si un composant n'en a pas, le composant sera marqué dans

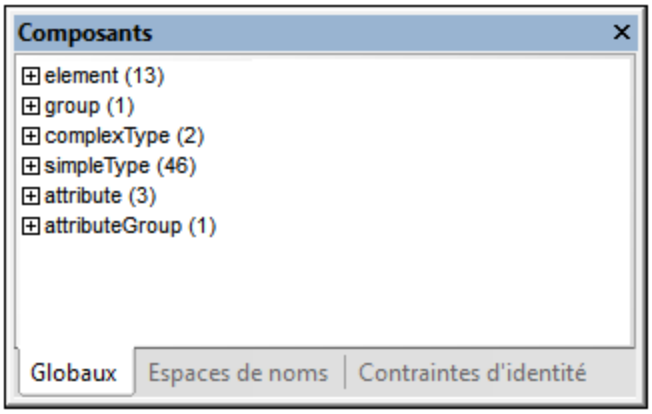
 © 2016-2022 Altova GmbH

l'Aperçu de schéma. Les composants globaux qui sont inclus ou importés depuis d'autres schémas sont aussi affichés dans Assistant à la saisie des Composants.
- Afin de fournir un aperçu des contraintes d'identité définies dans le document de schéma. Pour une description de l'onglet Contraintes d'identité, voir Contraintes d'identité.

Note : Le fait que les types de données intégrés de XSD 1.0 ou 1.1 soient affichés dépend de quel mode XSD (XSD 1.0 ou 1.1) a été sélectionné.

Onglet Globaux

Dans l'onglet Globaux *(voir capture d'écran ci-dessous),* les composants globaux sont regroupés dans une arborescence conformément à leur type de composant. Le nombre de chaque composant global présent dans le schéma est indiqué à côté de chaque type de composant.

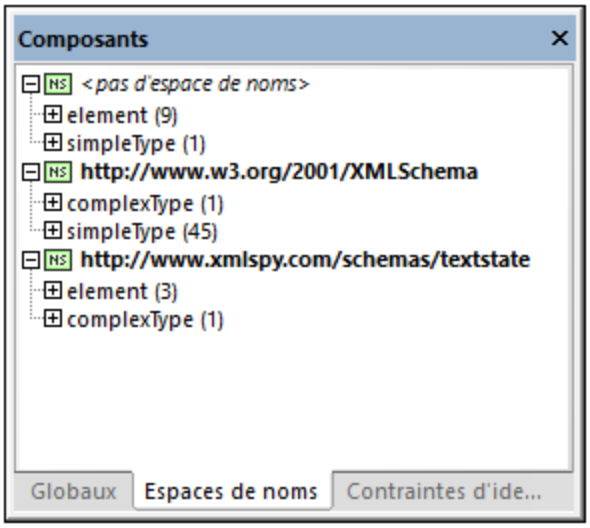


Dans l'arborescence, les composants globaux sont organisés dans les sept groupes suivants. Veuillez noter qu'un type de composant est recensé dans une arborescence uniquement si au moins un composant de ce type existe dans le schéma.

- Déclarations d'élément (Éléments)
- Groupes de modère (Groupes)
- Types Complexes
- Types Simples
- Déclarations d'attribut (Attributs)
- Groupes d'attribut
- Notations

Onglet Espace de noms

Dans l'onglet d'espace de noms *(voir capture d'écran ci-dessous)*, les composant sont organisés tout d'abord conformément à l'espace de noms puis conformément au type de composant.



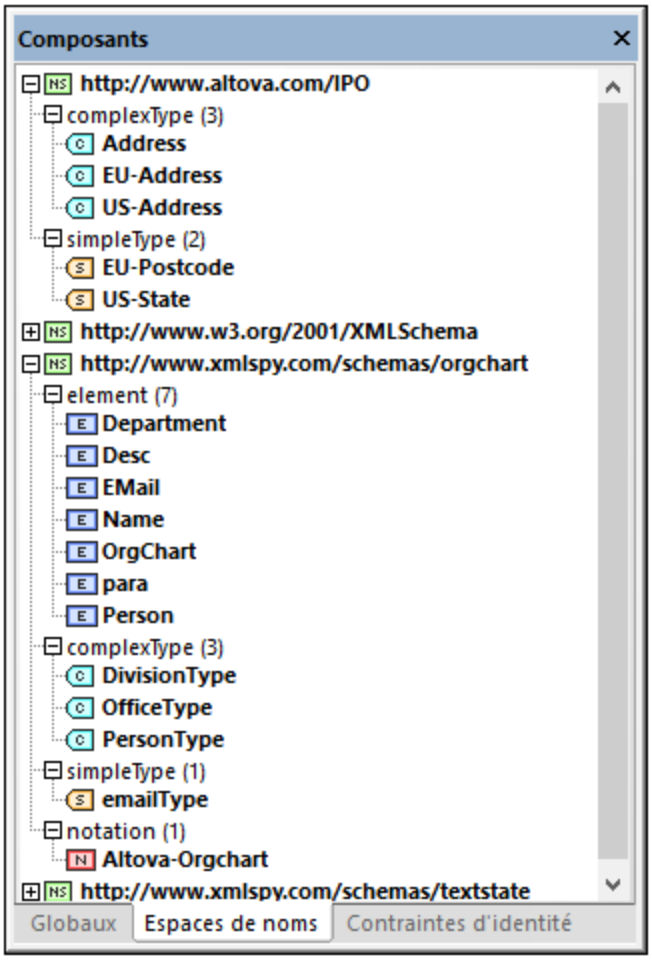
Dans l'arborescence, les composants globaux sont organisés dans les sept groupes suivants. Veuillez noter qu'un type de composant est recensé dans une arborescence uniquement si au moins un composant de ce type existe dans le schéma.

- Déclarations d'élément (Éléments)
- Groupes de modère (Groupes)
- Types Complexes
- Types Simples
- Déclarations d'attribut (Attributs)
- Groupes d'attribut
- Notations

Groupes de type de composant dans les onglets Globaux et Espaces de noms

Agrandir un groupe de type de composant dans l'onglet Globaux ou Espaces de noms pour afficher tous les composants dans ce groupe *(voir capture d'écran ci-dessous)*. Cela vous permet de vous rendre facilement à un composant défini par l'utilisateur. Double-cliquer sur le composant dans l'onglet Composant pour afficher sa définition dans la fenêtre principale.

© 2016-2022 Altova GmbH

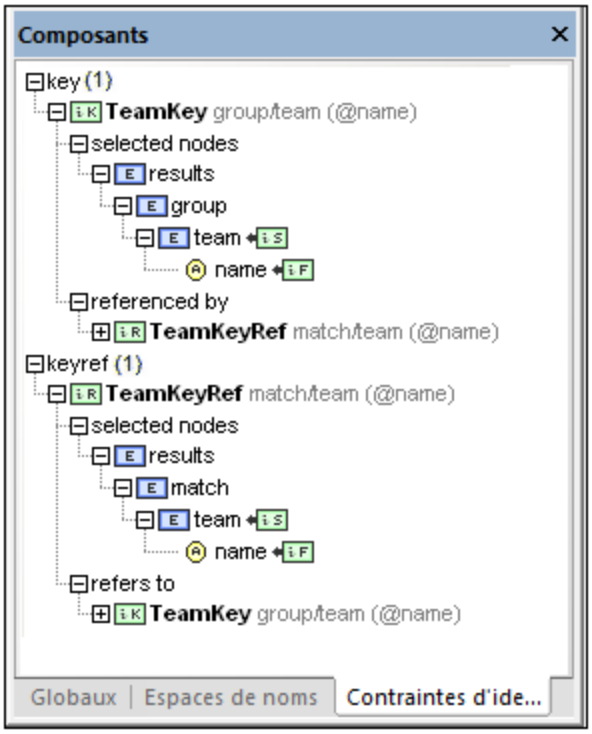


Si un composant a un modèle de contenu (c'est à dire, s'il s'agit d'un Élément, d'un Groupe ou d'un Type Complexe), le double-cliquer pour afficher le modèle de contenu du composant dans le Mode Modèle de contenu (dans la fenêtre principale). Si le composant n'a pas de modèle de contenu (donc s'il s'agit d'un Type Simple, d'un Attribut, d'un Groupe d'attribut ou d'une Notation), le composant est marqué dans l'Aperçu de Schéma (dans la fenêtre principale).

Note : Si le composant se trouve dans un schéma inclus ou importé, le schéma inclus/importé s'ouvrira (s'il n'est pas déjà ouvert) et soit le modèle de contenu du composant est affiché dans le Mode Modèle de contenu soit le composant sera marqué dans l'Aperçu de Schéma.

Contraintes d'identité

L'onglet Contraintes d'identité de l'Assistant à la saisie (*capture d'écran ci-dessous*) confère un aperçu de des contraintes d'identité d'un document. Dans cet onglet, les contraintes d'identité sont recensées par le type de contrainte d'identité (`unique`, `key`, `keyref`) et sont affichées en tant qu'arborescence agrandissable/réduisible.



Les entrées en gras sont présentes dans le schéma actuel, alors que celles en écriture normale sont présentes dans les sous-schémas. Double-cliquer sur une entrée pour l'onglet Contraintes d'identité pour sélectionner ce composant de schéma dans le Mode de Modèle de contenu.

Les commandes de menu contextuel suivantes sont disponibles lorsqu'un item dans l'onglet Contraintes d'identité est sélectionné :

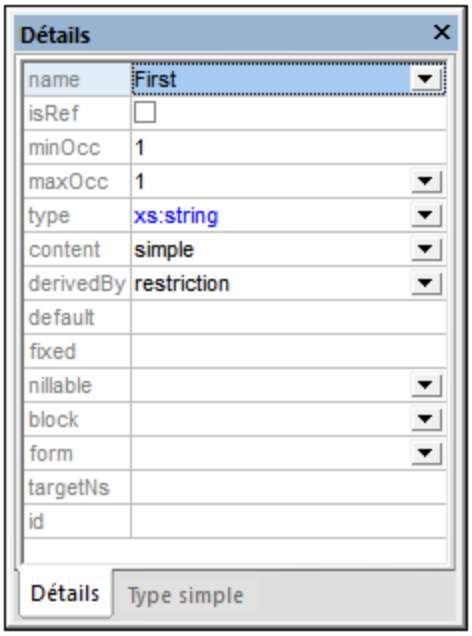
- *Afficher dans le diagramme* : sélectionne le composant de schéma dans le Mode de Modèle de contenu.
- *Afficher Cible Sélecteur/Champ dans le diagramme* : sélectionne, dans le Mode de Modèle de contenu, le composant de schéma ciblé par le sélecteur ou le champ de la contrainte d'identité: dans le cas de champs multiples, un dialogue invite l'utilisateur à utiliser le champ requis.
- *Aller à la Contrainte d'identité* : sélectionne la contrainte d'identité dans l'Aperçu de schéma.
- *Agrandir/réduire tout* : agrandit ou réduit l'aborescence.

Pour une description de l'onglet Contraintes d'identité, voir la section, Contraintes d'identité.

4.4.5.2 Détails

L'Assistant de saisie Détails du Mode Schéma affiche des informations éditables concernant le composant ou le compositeur actuellement sélectionné dans la Fenêtre principale. Si vous éditez un fichier de schéma qui contient des extensions de base de données, un onglet supplémentaire contenant des informations concernant les extensions de BD peut être visible.

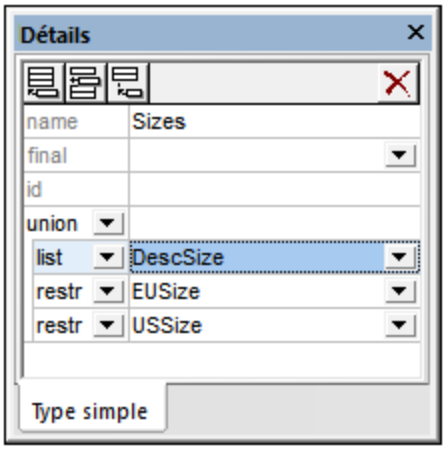
© 2016-2022 Altova GmbH



Afin de changer les propriétés du composant ou du compositeur actuellement sélectionné, double-cliquer le champ à éditer pour pouvoir le modifier ou saisir du texte directement. Si une liste de choix est disponible dans le champ à éditer, sélectionner la valeur désirée depuis la liste déroulante. Les changements que vous effectuez via l'Assistant de saisie Détails apparaîtront immédiatement dans le design.

Dérivations Type Simple

Vous pouvez utiliser l'Assistant de saisie Détails pour créer rapidement et précisément des types simples dérivés : `restriction`, `list` et `union`. Lorsqu'un type simple est sélectionné dans le design, l'Assistant de saisie Détails contiendra un onglet Type Simple (*voir capture d'écran ci-dessous*).



Dans la liste de choix de type-dérivation de l'onglet TypeSimple, sélectionner le type de dérivation (`restriction`, `list`, ou `union`) et, dans la liste de choix de type de membre correspondant situé à droite, sélectionner un type simple depuis les types simples disponibles. Utiliser les icônes dans la barre d'outils pour ajouter ou insérer un type du même niveau, pour ajouter un autre sous-niveau de dérivation ou pour supprimer

un type de dérivation Pour se rendre dans une définition de type, cliquer dessus avec la touche de droite et choisir **Aller sur la définition de Type**. Dans le cas des types simples intégrés, une fenêtre de message apparaît qui contient des informations concernant le type simple.

4.4.5.3 Facettes

Un nouveau type simple (nommé ou anonyme) est créé en limitant le type de base du type simple (qui est un type simple existant). Une telle limitation est effectuée en ajoutant des facettes pour limiter les valeurs du type de base. Dans le Mode Schéma, l'Assistant de saisie Facettes (*voir captures d'écran ci-dessous*) vous permet d'éditer graphiquement en toute simplicité les facettes d'un type simple. Les facettes disponibles sont organisées dans les onglets de l'Assistant de saisie Facettes comme indiqué dans le tableau ci-dessous.

Onglet	**Facettes disponibles**
Facettes	`minInclusive`, `maxInclusive`, `minExclusive`, `maxExclusive`, `length`, `minLength`, `maxLength`, `totalDigits`, `fractionDigits`, `whiteSpace`, `explicitTimezone`
Modèles	`pattern`
Énumérations	`enumeration`
Assertions	`assertion`
Échantillons	`altova:exampleValues` est une annotation, **ce n'est pas une facette**. Cette annotation est utilisée pour générer des valeurs échantillon dans le document XML d'instance généré par XMLSpy à partir du Schéma XML.

Chacun de ces onglets est décrit dans les sections ci-dessous.

Sélectionner le type simple dans le design

Un type simple (nommé ou anonyme) peut être sélectionné dans les environnements de design suivants :

- Dans l'Aperçu de Schéma (soit dans la liste des composants globaux soit dans l'onglet Attributs en-dessous de la liste des composants globaux), ou
- Dans le Mode de Modèle de contenu (soit dans le diagramme soit dans l'onglet Attributs en-dessous du diagramme).

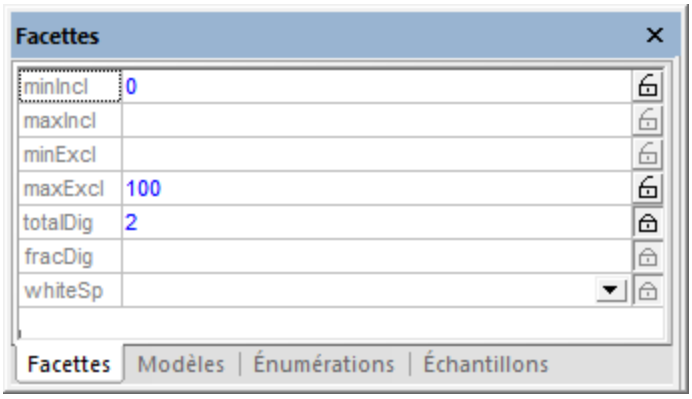
Lorsqu'un type simple est sélectionné dans le design, dans un des environnements de design recensés ci-dessus, les Facettes applicables dans l'Assistant de saisie Facettes s'activent et peuvent être éditées dans l'Assistant de saisie Facettes.

Onglet Facettes

Dans l'onglet Facettes, seules les facettes applicables au type sélectionné dans le design seront affichées. Par exemple, si c'est le type `xs:string` qui est limité, les facettes non-applicables comme `totalDigits` ne seront pas affichées.

© 2016-2022 Altova GmbH

- Les quatre facettes liées (`minInclusive`, `maxInclusive`, `minExclusive`, `maxExclusive`) sont uniquement applicables aux types numérique et de date/heure et aux types dérivés de ces types.
- Les trois facettes de longueur (`length`, `minLength`, `maxLength`) sont uniquement applicables à des types string-based, aux types binaires, et à `anyURI`.
- La facette `totalDigits` s'applique à `xs:decimal` et aux types entiers, ainsi qu'aux types dérivés de ces types. La facette `fractionDigits` peut être uniquement appliquée à `xs:decimal`.

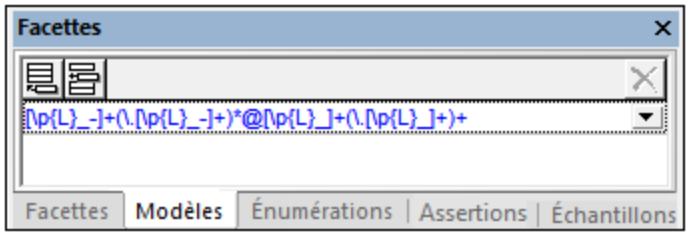


Pour saisir une valeur, soit sélectionner une valeur depuis la liste déroulante d'une liste de choix (si présente) ou double-cliquer le champ de valeur et saisir une valeur. Si une valeur invalide est saisie, les conflits résultants s'affichent en rouge. Les valeurs valides sont affichées en bleu. Par exemple, une facette `minInclusive` et une facette `maxInclusive` ne peuvent pas exister ensemble ; donc si une valeur est saisie pour la seconde de ces facettes, les valeurs des deux facettes sont affichées en rouge.

Afin de spécifier une **facette fixe** (donner à la facette une attribute-value de `fixed="true"`), cliquer sur le symbole du cadenas ouvert à la droite de la facette ; le symbole est maintenant fermé. Dans la capture d'écran ci-dessus, la facette `totalDigits` a été configurée en tant que facette fixe. Il est possible de fixer plus d'une seule facette. Pour annuler la fixation d'une facette, cliquer sur le symbole du cadenas fermé pour l'ouvrir.

Onglet Modèle

Dans l'onglet Modèles (*capture d'écran ci-dessous*), vous pouvez ajouter une ou plusieurs facettes `pattern` à une limitation. Le modèle (d'une facette `pattern`) est spécifié avec la syntaxe d'expression régulière. Le modèle dans la capture d'écran ci-dessous spécifie le modèle des adresses e-mail.

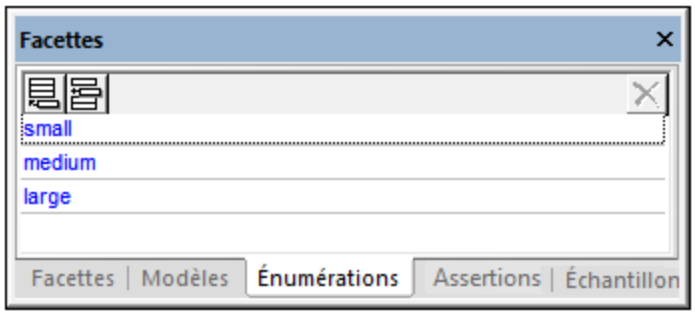


Si plusieurs facettes `pattern` sont spécifiées, la valeur d'instance XML doit correspondre à un des modèles spécifiés. Par exemple, un modèle destiné à limiter des codes postaux pourrait avoir deux facettes `pattern`, une pour un code postal US et un code postal UE. Une valeur d'instance XML doit alors correspondre à un des modèles pour qu'elle soit valide.

Ajouter une facette `pattern` en cliquant sur l'icône **Ajouter** ou **Insérer** située en haut à gauche puis saisir une expression régulière pour définir le modèle requis. Afin de supprimer un `pattern`, le sélectionner et cliquer sur l'icône **Supprimer** en haut à droite.

Onglet Énumérations

Dans l'onglet Énumérations (*capture d'écran ci-dessous*), vous pouvez ajouter une ou plusieurs facettes `enumeration` à une limitation. Chaque facette `enumeration` spécifie une valeur valide pour le type. Pris en groupe, un ensemble de facettes `enumeration` spécifie une gamme de valeurs autorisées. Dans la capture d'écran ci-dessous, les facettes `enumeration` spécifient la plage autorisée de valeurs de taille pour la limitation.



Ajouter une facette `enumeration` en cliquant sur l'icône **Ajouter** ou **Insérer** située en haut à gauche puis saisir la valeur `enumeration`. Pour supprimer une `enumeration`, la sélectionner et cliquer sur l'icône **Supprimer** en haut à droite.

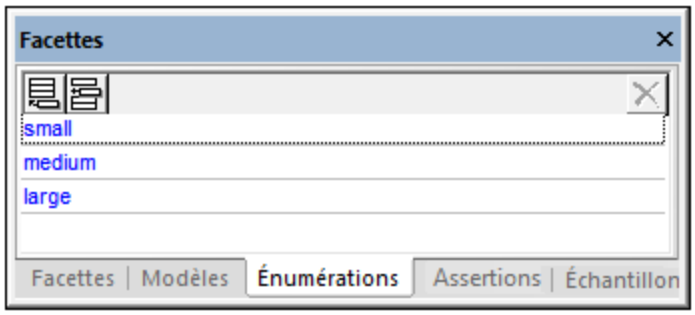
Onglet Assertions

Les Assertions sont une fonction XSD 1.1. C'est pourquoi l'onglet Assertions ne sera activé que dans le mode XSD 1.1 . Les facettes d'assertion définies dans l'onglet d'Assertions des assistants de saisie Facettes sont des **assertions pour des types simples** —par opposition aux assertions pour des types complexes (qui peuvent être définis et édités dans l'Aperçu de schéma ou le Modèle de contenu, **et non** dans l'assistant de saisie Facettes).

Lorsqu'un type simple (élément ou attribut de contenu simple) est sélectionné dans le design, une assertion peut lui être spécifié en passant à l'onglet Assertions (*voir capture d'écran ci-dessous*), cliquer sur l'icône **Ajouter** ou **Insérer** située en haut à gauche puis saisir l'expression XPath 2.0 qui sera utilisée pour définir l'`assertion`. Une variable spéciale nommée **`$value`** doit être utilisée dans l'expression XPath pour contenir la valeur du type simple. (Veuillez noter que, puisqu'il n'y a pas de descendants à tester, mais uniquement une valeur, l'étape de chemin normale `self::node()` (ou l'abréviation de période de cette étape de chemin '.') ne peut pas être utilisée dans l'expression XPath.)

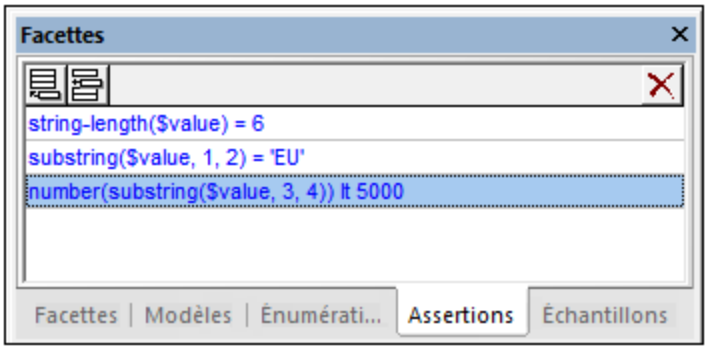
Par exemple, l'expression XPath `string-length($value) = 6` (*voir capture d'écran ci-dessous*) teste si la valeur du type simple a six caractères. Si l'élément ou l'attribut dans le document d'instance a bien six caractères, il est valide conformément à l'assertion.

© 2016-2022 Altova GmbH



Note : L'expression passe au rouge pour signaliser des erreurs de syntaxe dans l'expression XPath. Mais puisque le type de données est déterminé lors de la marche, les erreurs de type ne seront pas signalisées lorsque vous saisissez l'expression XPath. Vous devez prendre soin de construire les types selon vos besoins. Pour un exemple d'une construction de type, voir la troisième expression XPath dans la capture d'écran ci-dessus, qui convertit une valeur de string (en partant du principe que l'assertion est définie sur un type simple `xs:string`) en un nombre avant d'effectuer une comparaison numérique.

Les assertions multiples peuvent être spécifiées sur un type simple unique, comme indiqué dans la capture d'écran ci-dessus. Dans ce cas, toutes les assertions doivent être satisfaites pour que l'élément ou l'attribut dans le document d'instance soit valide. Les assertions dans la capture d'écran ci-dessus spécifient que la valeur du document d'instance doit être un string de six caractères commençant avec les caractères `EU` et ayant des caractères numériques d'une valeur numérique de `0000` à `4999` pour ses quatre derniers chiffres. Pour éditer les détails d'une assertion, cliquer avec la touche de droite sur l'assertion dans l'Assistant de saisie Facettes puis cliquer sur **Détails** dans le menu qui s'affiche. La fenêtre Détails d'Assertion s'affiche (*voir capture d'écran ci-dessous*).



Il est toujours utile de proposer une explication de l'assertion avec sa définition de manière à ce qu'un message approprié puisse être affiché dans le cas ou une assertion n'est pas remplie lors de la validation d'un document d'instance XML. Puisque la spécification de Schéma XML n'impose aucun message, XMLSpy permet de fournir un message dans l'espace de noms `http://www.altova.com/xml-schema-extensions` Altova xml-schema-extensions (ou tout autre espace de noms) avec la définition de l'assertion et de l'utiliser dans la validation du document d'instance XML. Par exemple :

```
<xs:assert test="count(//MyNode) ge 1" altova:message="There must be at least one
MyNode element"/> or
<xs:assertion test="count(//MyNode) ge 1" altova:message="There must be at least one
MyNode element"/>
```

Si la restriction spécifiée dans l'assertion n'est pas remplie, le moteur de validation de XMLSpy s'affichera, ainsi

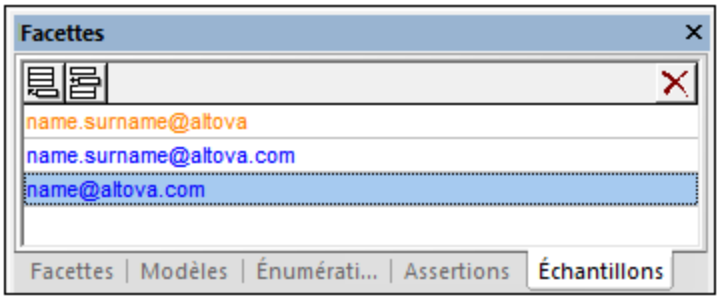
qu'avec le message de validation-erreur, le message associé avec l'assertion en tant qu'indice. Le validateur rapportera la valeur d'un attribut `assert/@message` ou d'un attribut `assertion/@message` quels que soient les espaces de noms dans lesquels se trouvent l'attribut `message`. Néanmoins, dans le Mode Schéma, vous pouvez éditer uniquement les attributs `message` qui se trouvent dans l'espace de noms Altova xml-schema-extension. Pour éditer les attributs `message` dans d'autres espaces de noms, utiliser le Mode Texte.

Voir Assertion de Message pour plus de détails.

Note : Il est recommandé de préférer l'utilisation d'autres facettes au lieu des assertions, lorsque possible. Par exemple, la restriction spécifiée par la première assertion dans la capture d'écran ci-dessus serait mieux spécifiée par la facette `length` (dans l'onglet Facettes).

Onglet Échantillons

Dans l'onglet Échantillons (*capture d'écran ci-dessous*), vous pouvez spécifier les valeurs d'échantillon qui peuvent être utilisées lors de la génération d'un fichier XML depuis le Schéma XML(avec la commande de menu **DTD/Schéma | Générer un fichier XML Échantillon**). Si une valeur d'échantillon est invalide, un avertissement est indiqué en affichant la valeur d'échantillon en orange. Dans la capture d'écran ci-dessous, la première valeur est invalide parce qu'elle ne correspond pas à la facette `pattern` spécifiée pour les e-mails (voir onglet *Échantillons* ci-dessus).



Note : Cliquer sur l'icône **Afficher avertissements de validation** dans la barre d'outils pour passer à l'affichage des avertissements de valeur d'échantillon invalide. Une valeur d'échantillon invalide n'invalide pas le fichier XSD si le fichier est valide autrement.

Les valeurs échantillon sont placées dans un élément d'annotation `altova:example` qui se trouve dans l'espace de noms `http://www.altova.com/xml-schema-extensions`. Ajouter une annotation `altova:example` en cliquant sur l'icône **Ajouter** ou **Insérer** située en haut à gauche puis saisir la valeur `altova:example`. Pour supprimer une annotation `altova:example`, la sélectionner et cliquer sur l'icône **Supprimer** en haut à droite.

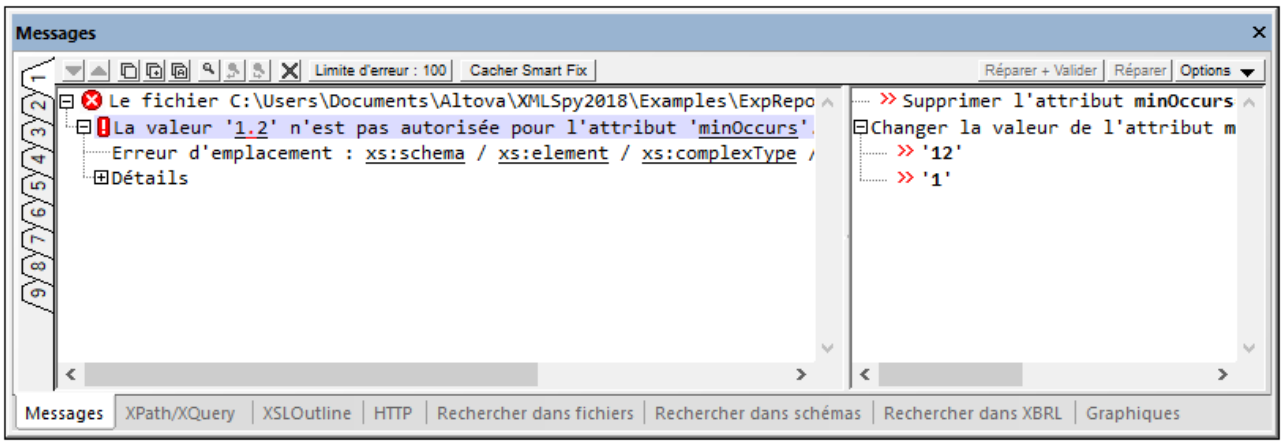
4.4.6 Validation et Smart Fixes

Un document de Schéma XML peut être validé pour son exactitude. Pour ce faire, cliquer sur la commande de menu **XML | Valider XML (F8)**.

Si le document est valide, un message apparaîtra dans la fenêtre Messages.

© 2016-2022 Altova GmbH

Si le document est invalide, la fenêtre Messages affichera deux panneaux (*voir capture d'écran ci-dessous*). Le volet de gauche (le volet Erreurs) contient la liste des premières `x` erreurs, ou toutes les erreurs. Le volet de droite est le volet Smart Fix ; il contient une liste des réparations possibles pour l'erreur sélectionnée dans le volet de gauche. Par exemple, dans la capture d'écran ci-dessous, la sélection de la deuxième erreur dans le volet Erreurs a entraîné l'affichage de réparations possibles dans le volet droit Smart Fix pour cette erreur. Si vous sélectionnez une des réparations, puis cliquez sur **Réparer+Valider** ou **Réparer**, l'erreur dans le document sera corrigée par le biais de cette réparation-là.



Volet d'erreurs

La barre d'outils de la fenêtre permet les fonctions suivantes :

- Défiler parmi les erreurs à l'aide des flèches **Haut** et **Bas**.
- Copier un message, ou un message et ses descendants, ou tous les messages sur le presse-papiers.
- Chercher les mots que vous souhaitez utiliser avec les fonctions Trouver, Trouver suivant, et Trouver précédent. Cela peut être utile si plusieurs erreurs ont été trouvées.
- Effacer toutes les erreurs du volet Erreurs.
- Définir une limite du nombre d'erreurs trouvées et affichées (`1` à `999`). Le réglage par défaut est `100`. Cliquer sur le bouton pour éditer la limite.
- Afficher/Dissimuler le volet Smart Fix. Une fois qu'il est dissimulé, le bouton **Afficher Smart Fix** apparaît dans la barre d'outils ; cliquer sur le bouton pour faire réapparaître le volet Smart Fix et le bouton affichera **Dissimuler Smart Fix**. Si le bouton **Afficher/Dissimuler Smart Fix** est désactivé, aucun smart fix ne sera disponible.

Volet Smart Fix

La barre d'outils de la fenêtre permet les fonctions suivantes :

- Le bouton **Réparer+Valider** corrige l'erreur sélectionnée avec le Smart Fix choisi et revalide le document. Toute autre erreur sera rapportée dans le volet Erreurs.
- Cliquer sur le bouton **Réparer** pour réparer l'erreur mais sans revalider.
- Le bouton **Options** ouvre une liste contenant un choix de comportements lorsque vous double-cliquerez une Smart Fix : soit le double-clic permettra de **Réparer+Valider** soit de **Réparer**.

4.4.7 Messages d'assertion

Dans le Schéma XML 1.1, les assertions peuvent être définies pour des types complexes (à l'aide d'éléments `xs:assert`) et de types simples (à l'aide d'éléments `xs:assertion`).

Il est toujours utile de proposer une explication de l'assertion avec sa définition de manière à ce qu'un message approprié puisse être affiché dans le cas ou une assertion n'est pas remplie lors de la validation d'un document d'instance XML. Puisque la spécification de Schéma XML n'impose aucun message, XMLSpy permet de fournir un message dans l'espace de noms `http://www.altova.com/xml-schema-extensions` Altova xml-schema-extensions (ou tout autre espace de noms) avec la définition de l'assertion et de l'utiliser dans la validation du document d'instance XML. Par exemple :

```
<xs:assert test="count(//MyNode) ge 1" altova:message="There must be at least one
MyNode element"/> or
<xs:assertion test="count(//MyNode) ge 1" altova:message="There must be at least one
MyNode element"/>
```

Si la restriction spécifiée dans l'assertion n'est pas remplie, le moteur de validation de XMLSpy s'affichera, ainsi qu'avec le message de validation-erreur, le message associé avec l'assertion en tant qu'indice. Le validateur rapportera la valeur d'un attribut `assert/@message` ou d'un attribut `assertion/@message` quels que soient les espaces de noms dans lesquels se trouvent l'attribut `message`. Néanmoins, dans le Mode Schéma, vous pouvez éditer uniquement les attributs `message` qui se trouvent dans l'espace de noms Altova xml-schema-extension. Pour éditer les attributs `message` dans d'autres espaces de noms, utiliser le Mode Texte.

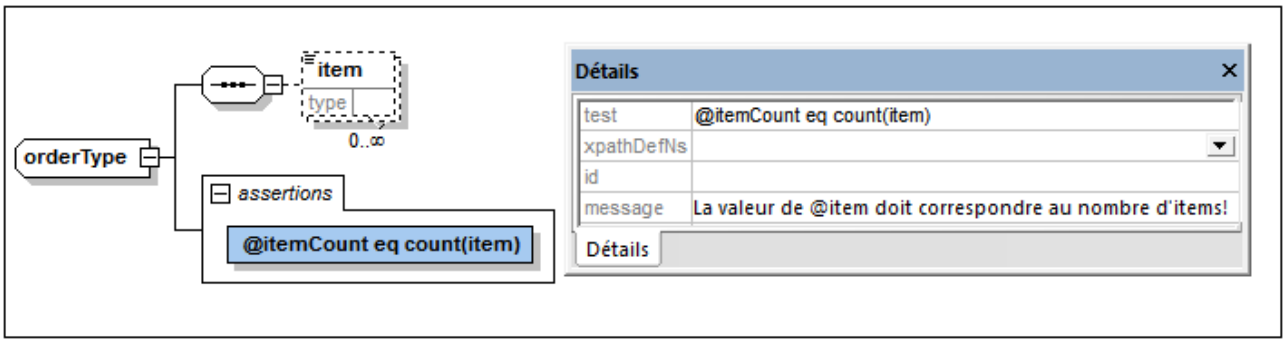
Éditer des messages xs:assert

Dans le Mode Schéma, les éléments `xs:assert` (pour les types complexes) peuvent être créés et édités dans le panneau Attributs/Assertions/Contraintes d'Identité (AAIDC) ou dans l'Assistant de saisie Détails du type complexe pertinent. La capture d'écran ci-dessous montre une assertion pour le type complexe `orderType`. L'assertion (un `xs:assert` dans ce cas) est définie dans l'onglet Assertions (de l'Aperçu de schéma) avec un message d'assertion.



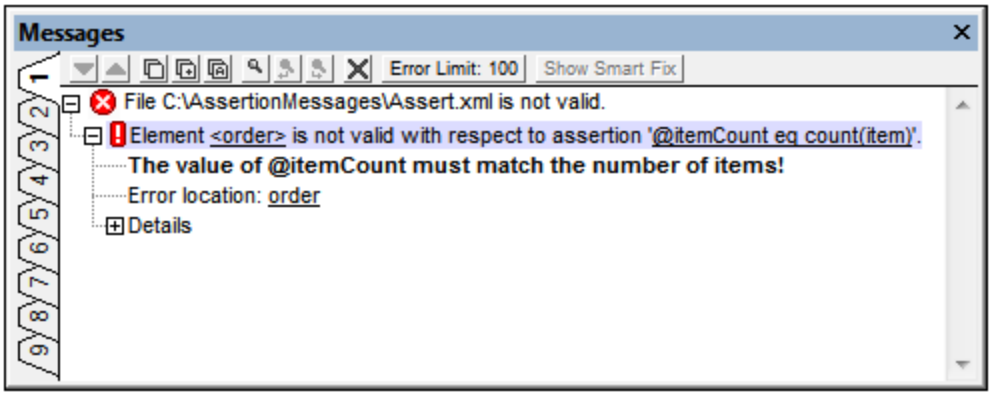
Si l'option **Afficher les assertions dans le diagramme** a été choisie, les assertions sur des types complexes peuvent aussi être créées et éditées dans le Mode Modèle de contenu. Pour ajouter ou éditer un message d'assertion, sélectionner l'assertion et saisir le message d'assertion dans l'assistant à la saisie Détails (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



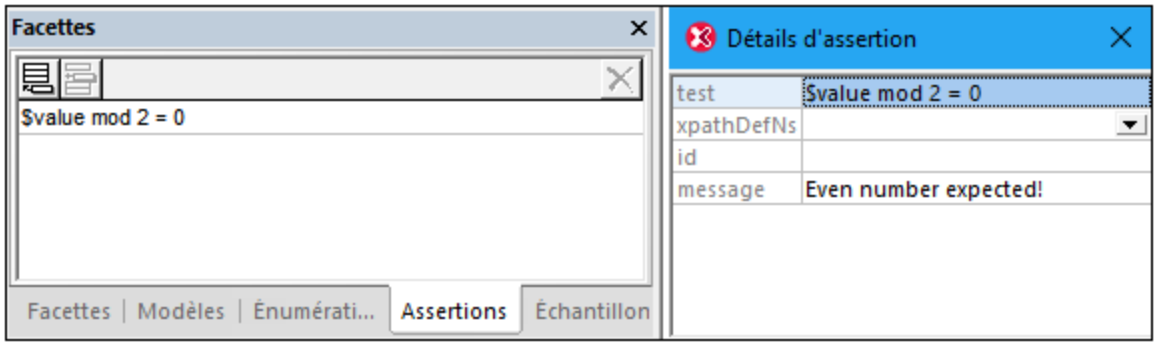
Veuillez noter que les messages d'assertion créés de cette manière se trouvent dans l'espace de noms Altova xml-schema-extensions `http://www.altova.com/xml-schema-extensions`. Lorsque vous ajoutez le premier message d'assertion dans le document de schéma XML via le panneau AACID ou l'Assistant de saisie Détails, l'espace de noms Altova xml-schema-extensions est déclaré automatiquement sur l'élément `xs:schema`.

Si un fichier XML est validé et que le test d'assertion n'est pas satisfaisant, le message défini pour l'assertion est affiché avec un message d'erreur (*voir capture d'écran ci-dessous*).

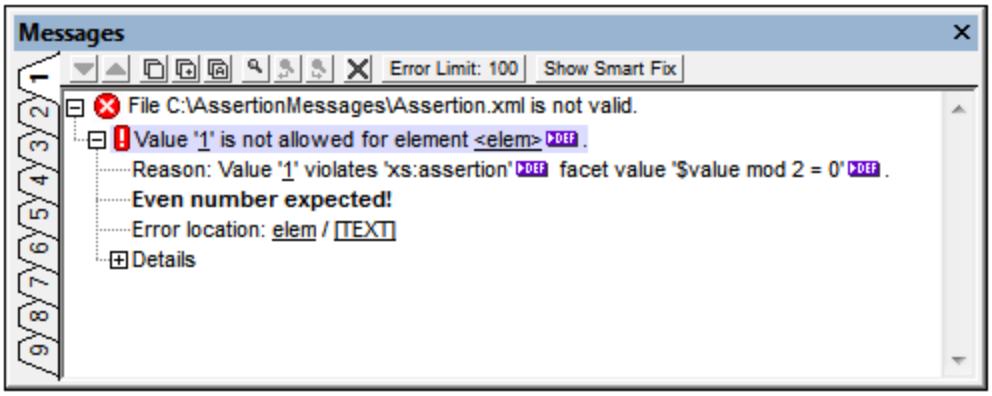


Éditer les messages xs:assertion

Dans le Mode Schéma, les éléments `xs:assertion` (pour les types simples) peuvent être créés et édités dans l'Assistant à la saisie Facettes du type simple pertinent. Pour éditer le message d'assertion, cliquer avec la touche de droite sur l'assertion dans l'Assistant à la saisie Facettes (*voir capture d'écran ci-dessous*), cliquer sur **Détails** dans le menu qui s'affiche et éditer le message dans la fenêtre modale des Détails d'assertion (*voir capture d'écran ci-dessous*). Veuillez noter que les messages d'assertion créés de cette manière se trouvent dans l'espace de noms Altova xml-schema-extensions `http://www.altova.com/xml-schema-extensions`. Lorsque vous ajoutez le premier message d'assertion dans le document de schéma XML par le biais de la fenêtre modale des Détails d'assertion, l'espace de noms est automatiquement déclaré dans l'élément `xs:schema`.



Si un fichier XML est validé et que le test d'assertion n'est pas satisfaisant, le message défini pour l'assertion s'affichera avec le message d'erreur (*voir capture d'écran ci-dessous*).



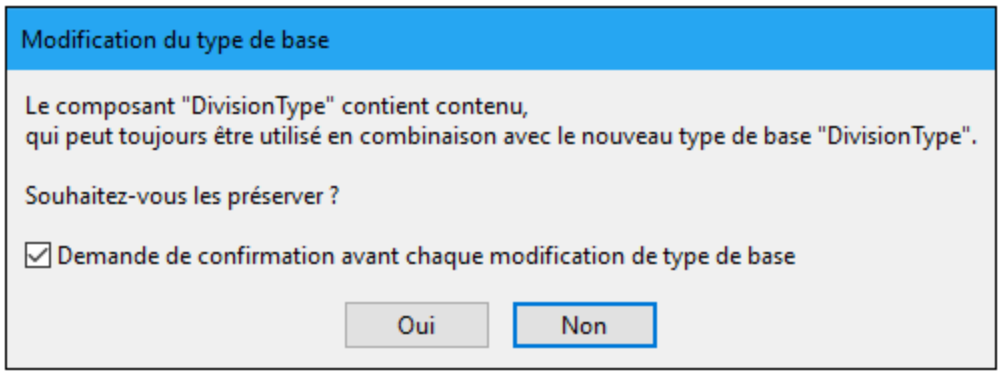
4.4.8 Modification de Type de Base

Si le type de base d'un type dérivé est modifié dans le Mode Schéma, le contenu, les attributs, les facettes et les valeurs d'échantillon définis dans le type dérivé peuvent être gérés de deux manières :

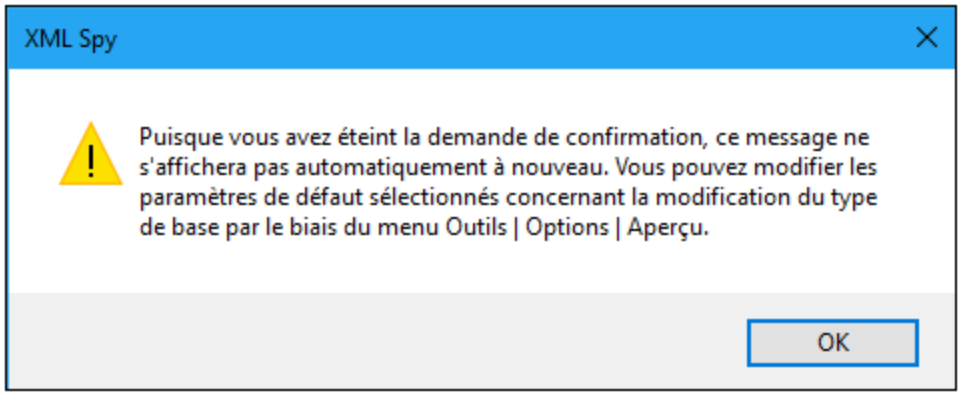
- Ils peuvent être préservés s'ils sont toujours applicables en combinaison avec le nouveau type de base.
- Ils peuvent être supprimés automatiquement qu'ils soient toujours applicables en combinaison avec le nouveau type de base ou pas.

En cas de modifications du type de base d'un type dérivé qui contient le contenu, les attributs, les facettes et les valeurs d'échantillon, le dialogue de Modification de type de base (*capture d'écran ci-dessous*) s'affiche.

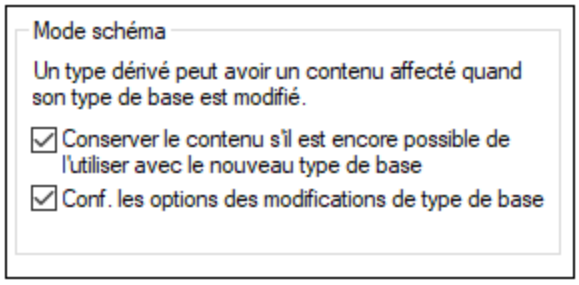
© 2016-2022 Altova GmbH



Si la case à cocher *Demander confirmation* est décochée, une fenêtre pop-up (*capture d'écran ci-dessous*) indique que la confirmation peut être rallumée dans la section Affichage du dialogue Options (**Outils | Options | Affichage**).



Dans le panneau du Mode Schéma (*capture d'écran ci-dessous*) de la section Affichage du dialogue Options (**Outils | Options | Mode Texte**), vous pouvez spécifier si le contenu doit être préservé et si la une confirmation de l'utilisateur est nécessaires pour chaque modification de type de base.



Cochez les cases correspondantes pour préserver le contenu et demander une confirmation si vous souhaitez en faire vos options par défaut.

4.4.9 Smart Restrictions

Lors de la restriction d'un type complexe, certaines parties du modèle de contenu du type de base sont réécrites dans le type dérivé. Cela peut prêter à confusion si le contenu du modèle est complexe parce que tout en éditant le type dérivé, il peut être difficile de se souvenir exactement à quoi ressemble le modèle de contenu du type de base.

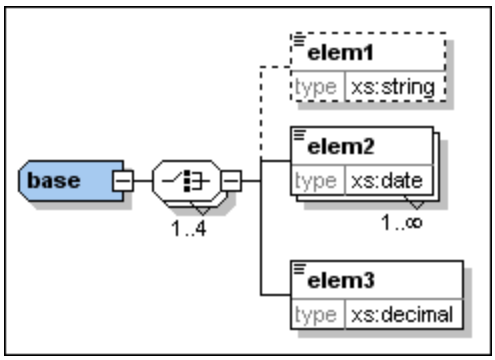
Les Smart Restrictions combinent et lient entre elles deux modèles de contenu dans le mode graphique du modèle de contenu dérivé. Dans le type complexe dérivé, toutes les particules du type de base complexe et la manière dont ils sont liés au type dérivés, peuvent être vus. De plus, les Smart Restrictions fournissent des indices visuels pour vous montrer les possibilités de limiter le type de base. Cela permet de limiter correctement le type dérivé.

Pour activer les Smart Restrictions:

- Cliquer sur l'icône Smart Restrictions . dans la barre d'outils Conception de Schéma.

L'exemple qui suit sert à illustrer les fonctions des Smart Restrictions.

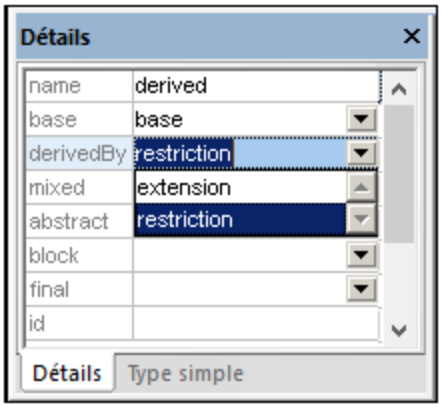
Le type complexe suivant est le type de base utilisé dans cet exemple :



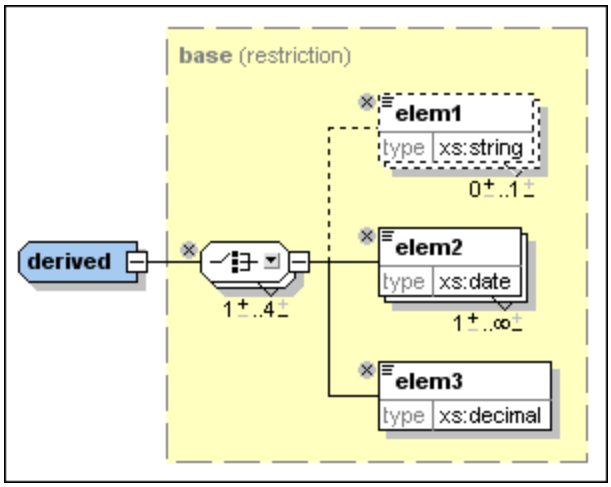
Le type complexe "dérivé" est dérivé du type de "base" comme suit :

1. Créer un nouveau type complexe dans le schéma et le nommer "dérivé".
2. Dans l'assistant à la saisie Détails sélectionner "base" depuis la liste déroulante **base** et "restriction" depuis la liste déroulante **derivedBy**.

© 2016-2022 Altova GmbH

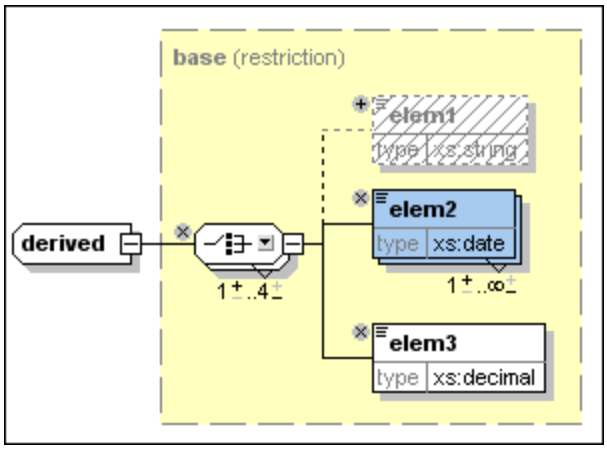


Une fois Smart Restrictions activé, le nouveau type dérivé ressemble à :

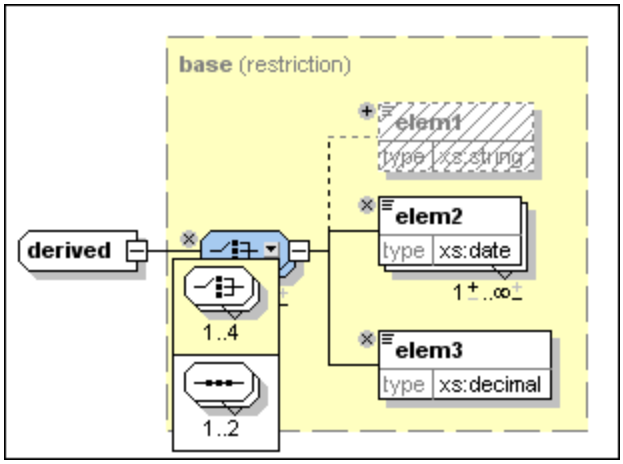


Veuillez noter les différentes commandes qui peuvent être utilisées pour limiter le type dérivé dans cet exemple :

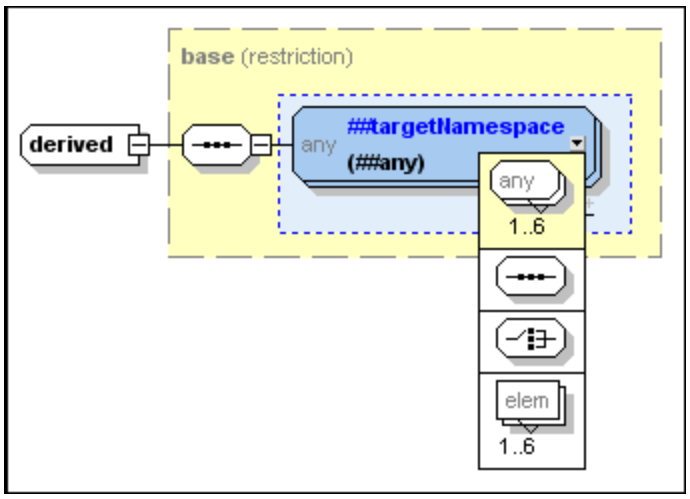
- Utiliser cette icône pour retirer les éléments qui se trouvent dans le type de base depuis le type dérivé. Ici, elem1 a été supprimé. Pour le rajouter, cliquer sur cette icône .



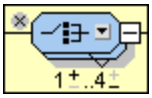
- Cliquer sur la flèche vers le bas sur le compositeur de choix pour obtenir la liste suivante, qui vous permet de changer le groupe de modèle Choix en un groupe de modèle Séquence :



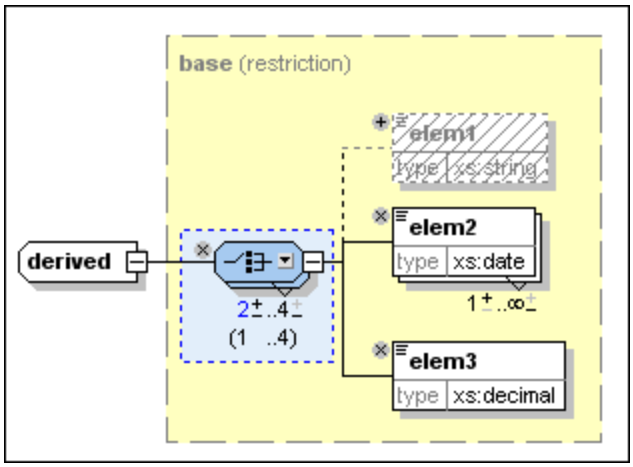
Il est également possible de changer les caractères génériques de la même manière, comme dans l'exemple suivant :



Pour obtenir une liste complète des particules pouvant être remplacées par d'autres particules, voir la spécification de schéma XML.

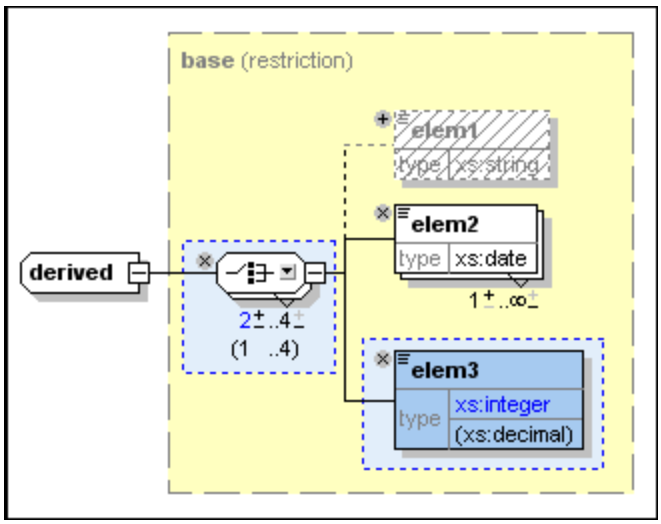
- Changer le nombre d'occurrences du groupe de modèle à l'aide de la commande suivante  afin d'augmenter le nombre minimum d'occurrences en cliquant sur le signe plus à côté du "1", o pour réduire le nombre minimum d'occurrences en cliquant sur le signe moins sous le "4". Ces commandes sont affichées si la gamme de l'occurrence dans la base décrit une vraie plage (par ex., 2-5) et non une certaine quantité (par ex. 4-4). Elles sont aussi affichées si la plage d'occurrence est erronée.

© 2016-2022 Altova GmbH

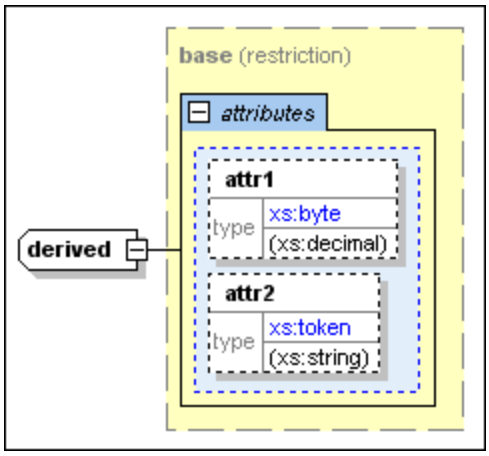


Ici, vous pouvez constater que le minimum d'occurrences pour cet élément a été modifié à 2. Veuillez noter que le groupe de modèle présente maintenant un fond bleu, ce qui signifie qu'il n'est plus le même que le groupe de modèle dans le type complexe de base. Ainsi, la plage d'occurrence permise du groupe de modèle dans la particule de base est maintenant affichée entre parenthèses.

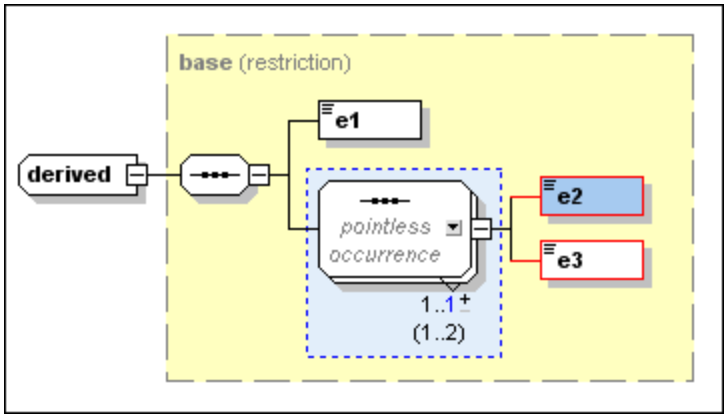
- Il est possible de modifier les types de données des attributs ou des éléments si le nouveau type de données est une restriction valide du type de données de base tel que défini dans la spécification de schéma XML . Par exemple, vous pouvez changer le type de données d'elem3 dans le type de données "dérivé" de décimal en entier. Ensuite, l'élément aura un fond bleu pour vous avertir qu'il est différent de l'élément contenu dans le type de base, et le type que l'élément a dans le type de base est affiché entre parenthèses :



Cet exemple montre des attributs dont les types de données ont été limités dans le type complexe dérivé :



- Smart Restrictions vous préviennent en cas d'*occurrences inutiles* dans le modèle de contenu. Une occurrence inutile se produit par exemple, lorsqu'une séquence qui est présente dans le modèle de contenu n'est pas nécessaire. Cet exemple montre une occurrence inutile :



Note : les occurrences inutiles sont uniquement affichées si le modèle de contenu contient une erreur. Il est possible qu'un modèle de contenu contienne une occurrence inutile et soit valide, dans ce cas, l'occurrence inutile n'est pas affichée explicitement afin d'éviter toute confusion.

Voir la spécification de schéma XML pour plus d'informations concernant les occurrences inutiles.

4.4.10 xml:base, xml:id, xml:lang, xml:space

L'espace de noms `http://www.w3.org/XML/1998/namespace` est lié, conformément aux spécifications de XML Namespaces, par définition au préfixe `xml:` . Cela signifie que c'est l'espace de noms qui doit être utilisé avec le préfixe `xml:` et qu'il est réservé à cet effet. Cet espace de noms contient quatre attributs qui peuvent être les enfants de tout élément XML dans tout document XML (schéma ou instance) :

- `xml:base` (pour configurer l'URI de base d'un élément)
- `xml:id` (pour spécifier l'ID unique d'un élément)
- `xml:lang` (pour identifier la langue utilisée dans cet élément)
- `xml:space` (pour spécifier comment un espace blanc dans l'élément doit être géré)

© 2016-2022 Altova GmbH

Dans le Mode Schéma, une fois que l'espace de noms XML Namespaces a été importé dans le document du Schéma XML, ces quatre attributs `xml:` peuvent être référencés pour une utilisation sur n'importe quel élément dans le schéma.

Afin de déclarer un de ces attributs dans un élément, procéder comme suit :

1. Déclarer l'espace de noms XML Namespaces pour ce document de schéma et lier l'espace de noms au préfixe `xml:` prefix. Si un des quatre attributs `xml:` est utilisé dans le document, son nom sera étendu pour inclure la partie de l'espace de noms correct.
2. Importer l'espace de noms XML Namespaces. Le validateur de XMLSpy reconnaîtra l'espace de noms et rendra les quatre attributs `xml:` disponibles en tant qu'attributs globaux, qui peuvent être référencés dans ce schéma.
3. Insérer l'attribut `xml:` requis en tant que l'enfant d'un élément. L'attribut est déclaré en tant que référence à l'attribut global "importé".

Déclarer l'espace de noms XML Namespaces

Vous pouvez déclarer l'espace de noms XML Namespaces (`http://www.w3.org/XML/1998/namespace`) en le saisissant par le biais du dialogue Configuration de Schéma, où tous les espaces de noms déclarés pour ce schéma sont stockés et peuvent être édités. L'espace de noms doit être lié au préfixe `xml:`. (En alternative, vous pouvez déclarer l'espace de noms (avec le préfixe `xml:`) sur l'élément `:schema` dans le Mode Texte.)

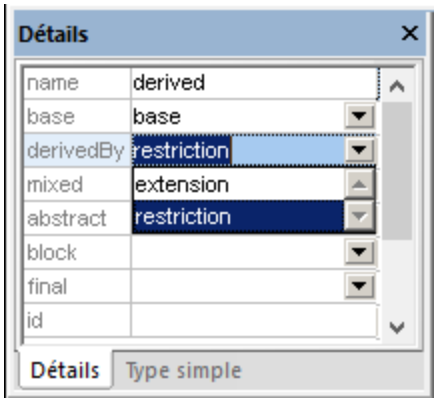
Importer l'espace de noms XML Namespaces

Dans l'Aperçu de Schéma, créer une déclaration d'importation globale pour l'espace de noms XML Namespaces. Pour ce faire, cliquer sur l'icône Insérer  ou Ajouter  située en haut de la fenêtre Aperçu de Schéma et sélectionner **Importer** depuis le menu qui apparaît. Saisir l'espace de noms XML Namespaces en tant que l'espace de noms à importer. Dans le Mode Texte, la déclaration d'importation devrait ressembler à l'exemple suivant :

```
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/XML/1998/namespace"/>.
```

Ajouter l'attribut xml:

Dans l'Aperçu de Schéma, sélectionner l'élément pour lequel vous souhaitez ajouter l'attribut `xml:`, puis ajouter un attribut. Dans l'Assistant à la saisie Détails (*capture d'écran ci-dessous*), cliquer sur la flèche orientée vers le bas de la liste de choix et choisir l'attribut requis `xml:`, par exemple `xml:base`. Lorsque vous serez invité à référencer l'attribut global, cliquer sur **Oui**. L'attribut est ajouté en tant qu'une référence.



XInclude et xml:base

Lorsque l'élément XInclude's `include` est remplacé par le fichier XML spécifié dans l'attribut `href` de l'élément include, l'élément de niveau supérieur du document XML parsé est inclus avec un attribut `xml:base`. Si ce document doit être validé, alors ce schéma doit définir un attribut xml:base sur le/s élément/s pertinent/s.

4.4.11 Avant et Arrière : se déplacer d'une position à une autre

Les commandes **Arrière** et **Avant** dans le Mode Schéma vous permettent de vous déplacer dans des positions visualisées précédemment dans le Mode Schéma. Cela peut s'avérer utile lorsque vous cliquez d'un composant de schéma à l'autre dans le Mode Schéma, et que vous souhaitez consulter un composant que vous aviez ouvert précédemment. Cliquer sur la touche **Arrière** une fois dans la barre outils pour vous ramener à la position consultée précédemment. En cliquant plusieurs fois sur la touche **Arrière**, vous pouvez consulter les 500 dernières positions consultées. Après vous être déplacé dans les positions précédentes, vous pouvez vous déplacer vers l'avant à travers ces positions en utilisant la touche **Avant** dans la barre outils.

Les clés de raccourcis pour les deux commandes sont :

- ***Arrière:*** **Alt + Flèche de gauche**
- ***Avant:*** **Alt + Flèche de droite**

Arrière/Avant par rapport à Annuler / Rétablir

Veuillez noter que les commandes **Arrière** et **Avant** ne sont pas les mêmes que les commandes **Annuler (Ctrl+Z)** et **Rétablir (Ctrl+Y)**. Ces deux ensembles de commandes représentent deux séries d'étapes différentes. Cliquer une fois sur la commande **Arrière** pour vous ramener au composant visionné juste auparavant dans l'affichage précédent. Cliquer sur la commande **Annuler** une fois pour annuler la dernière modification d'édition quel que soit le moment de la modification d'édition.

Informations complémentaires

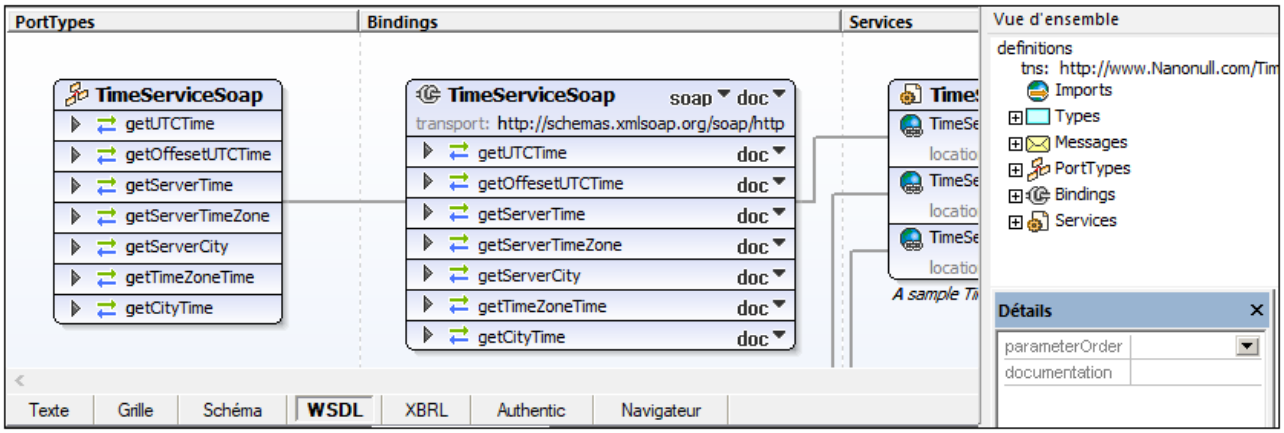
Veuillez noter les points suivants :

© 2016-2022 Altova GmbH

- Le bouton **Arrière** vous permet de visionner les 500 dernières positions.
- La fonction Arrière/Avant est activée sur tous les schémas. Si un schéma a été fermé depuis ou s'il est actuellement ouvert dans un autre mode, il sera ouvert dans le Mode Schéma ou il passera au Mode Schéma, respectivement.
- Si un composant qui a été visualisé dans un position précédente est supprimé, celui-ci ne pourra pas être consulté. Si un tel composant faisait partie d'une position précédente, cette position sera affichée sans le composant supprimé. Si le composant contenait toute la position, la totalité de la position sera indisponible et si vous cliquez sur la touche **Arrière** à ce moment, dans la série Arrière, vous reviendrez à la position précédente à la position non disponible.

4.5 Mode WSDL

Le Mode WSDL (*capture d'écran ci-dessous*) fournit une interface pour une édition graphique de documents WSDL 1.1 et WSDL 2.0. Le Mode WSDL est disponible lorsqu'un document WSDL est actif et que l'onglet Mode WSDL est cliqué. La structure et les composants d'un document WSDL sont créés dans la Fenêtre principale à l'aide de mécanismes de design graphiques, et une édition supplémentaire est activée depuis les Assistants à la saisie.



La Fenêtre principale (consistant en les PortTypes (WSDL 1.1) ou les sections Interfaces (WSDL 2.0), Bindings et Services) et les assistants à la saisie (Aperçu et Détails) sont décrits dans les sous-sections de cette section. (Pour une description de l'utilisation des projets, voir Menu Projet dans la section Référence de l'utilisateur.)

Fonctions disponibles dans le Mode WSDL

Les fonctions suivantes sont disponibles dans le Mode WSDL :

- Un affichage graphique dans la Fenêtre principale de tous les éléments WSDL, regroupés par PortTypes (WSDL 1.1) ou Interfaces (WSDL 2.0), Bindings et Services.
- Une manipulation directe des éléments WSDL à l'aide de glisser/déposer.
- La capacité d'ajouter, d'apposer et de supprimer tout élément WSDL visible dans le mode graphique (menu sensible au contexte).
- La capacité de saisir et d'éditer des valeurs dans l'assistant à la saisie Détails.
- Validation WSDL par rapport au W3C Working Draft.
- Importer ou intégrer des schémas XML dans le document WSDL.
- Passer au Mode Schéma pour l'édition de schémas.
- L'édition de types de schéma depuis le Mode WSDL.
- Génération d'une documentation WSDL dans MS Word ou HTML.
- Génération d'un diagramme (image PNG) du document WSDL dans la Fenêtre principale.
- Imprimer le Mode dans la fenêtre WSDL.

Consulter le fichier

Veuillez noter les points suivants concernant la consultation du fichier :

- Lorsque vous ouvrez un fichier WSDL, le fichier s'ouvre automatiquement dans le Mode WSDL.

© 2016-2022 Altova GmbH

- Vous pouvez aussi consulter un document WSDL dans le Mode Texte et le Mode Grille avancé. Pour ce faire, cliquer sur l'onglet approprié.
- Si le fichier WSDL contient une référence à un Schéma XML, alors le schéma peut être consulté en sélectionnant la commande de menu **WSDL | Types | Éditer le Schéma dans le Mode Schéma**. Cela permet d'ouvrir le fichier de schéma dans le Mode Schéma.
- Si un fichier de schéma associé est ouvert, vous ne serez pas autorisé à changer le mode du fichier WSDL (par exemple, du Mode WSDL au Mode Texte). Avant d'essayer de changer les modes du fichier WSDL, veuillez vous assurer que vous avez enregistré les changements dans le fichier de schéma et fermé le fichier.

Il y a deux assistants à la saisie pour vous aider à éditer les documents WSDL : Aperçu et Détails. Ces deux assistants à la saisie peuvent être ancrés ou détachés en double-cliquant sur la barre de titre. En position ancrée, la fonction de masquage automatique peut être activée en cliquant sur la punaise dans la barre de titre. Une fois dissimulé, l'assistant à la saisie est minimisé et apparaît en tant qu'onglet au bord de la fenêtre d'application. Un assistant à la saisie dissimulé peut être ancré en le faisant réapparaître depuis le rebord (en passant sur l'onglet avec la souris) et en cliquant sur la punaise dans la barre de titre.

Voir aussi : Plus d'informations concernant le travail avec les documents WSDL est disponible dans les sections, Tutoriel WSDL et Référence à l'utilisateur | Menu WSDL.

4.5.1 Fenêtre principale

La Fenêtre principale est l'endroit où vous éditez votre document WSDL. Elle se compose de trois sections verticales : (i) Port Types (WSDL 1.1) ou Interfaces (WSDL 2.0); (ii) Liaisons et (iii) Services. La relation entre un type de port et une liaison et entre une liaison et un service est indiquée avec une ligne de connexion. Chacune de ces trois sections est décrite en détail ci-dessous.

Les symboles dans la Fenêtre principale

Les symboles suivants sont utilisés dans la Fenêtre principale :

 Type de port dans WSDL 1.1, Interface dans WSDL 2.0

 Liaison

 Service

 Faute

 Opération. La flèche verte représente les entrées et les flèches bleues représentent les sorties. Selon le type d'opération, le symbole approprié sera utilisé.

 Message

 Partie du message (paramètre)

 Élément XSD

 XSD Type simple ou type complexe

 Port

Ajouter des nouveaux types de port, des interfaces, des liaisons et des services

Pour ajouter un nouveau type de port (dans les documents WSDL 1.1), une interface (dans les documents WSDL 2.0), une liaison ou un service, cliquer avec la touche de droite à un endroit quelconque de la Fenêtre

principale mais à l'extérieur d'une fenêtre de composant, et sélectionner la commande pertinente depuis le menu contextuel qui apparaît.

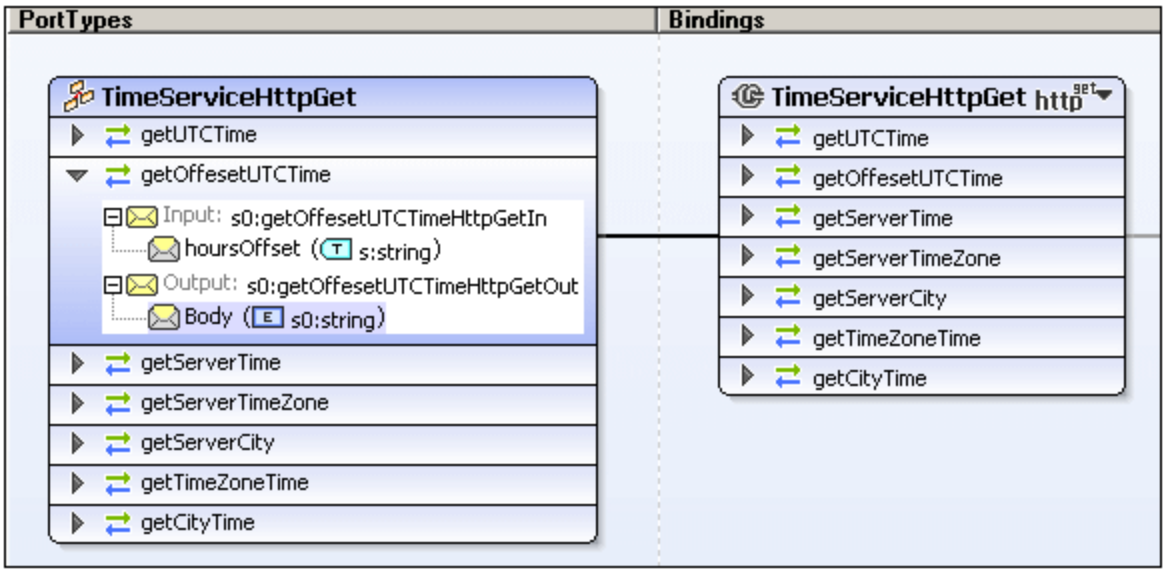
Fonction glisser/déposer

La fonction de glisser/déposer suivante est disponible :

- Dans la Fenêtre principale, les associations entre les PortTypes (WSDL 1.1) ou les Interfaces (WSDL 2.0) et les liaisons et entre Liaisons et Services peuvent être établies par glisser/déposer.
- Dans les documents WSDL 2.0, les éléments dans l'assistant à la saisie Aperçu peut être glisser sur les fautes d'interface aussi bien dans la Fenêtre principale que dans assistant à la saisie Aperçu.

PortTypes (WSDL 1.1), Interfaces (WSDL 2.0)

La section PortTypes (documents WSDL 1.1) contient tous les portTypes définis dans le document WSDL (*la capture d'écran ci-dessous montre uniquement un portType dans la section PortTypes*). La section Interfaces (dans les documents WSDL 2.0) contient toutes les interfaces définis dans le document WSDL.



Chaque portType ou interface est représenté en tant que champ contenant les opérations définies pour ce portType ou interface. Les composants peuvent être édités directement dans le champ. Les fonctions principales du type de port et des champs d'interface sont recensés ci-dessous :

- Les opérations peuvent être étendues pour afficher leurs messages en cliquant sur l'icône situé à gauche d'un nom d'opération.
- Dans WSDL 1.1, un message peut contenir une partie de message. Ces types de messages peuvent être agrandis pour montrer la partie de message.
- Cliquer avec la touche de droite sur un composant d'un champ portType (soit un portType, une opération, un message ou une partie de message), pour qu'un menu contextuel apparaisse à partir duquel des actions pertinentes peuvent être sélectionnées. Par exemple, cliquer avec la touche de droite sur un nom de portType vous permet, entre autre d'apposer une nouveau portType, d'apposer une opération au portType sélectionner ou de créer une liaison pour le portType sélectionné.
- Les propriétés d'interface optionnelles WSDL 2.0, *extends*, *styleDefault* et *documentation* sont dissimulées tant qu'elles sont vides. Elles peuvent être éditées par le biais de la commande **Édition** dans le menu contextuel de l'interface.

© 2016-2022 Altova GmbH

- Dans les documents WSDL 2.0, les propriétés des opérations peuvent être éditées par le biais de la commande **Édition** dans le menu contextuel de l'opération. La valeur de la propriété de style est sélectionnée par le biais d'une liste de choix recensant les options.
- Veuillez noter que lorsqu'un composant a été sélectionné, ses détails peuvent être édités dans l'assistant à la saisie Détail.
- La documentation pour les types de port et les interfaces apparaissent en bas des champs individuels.

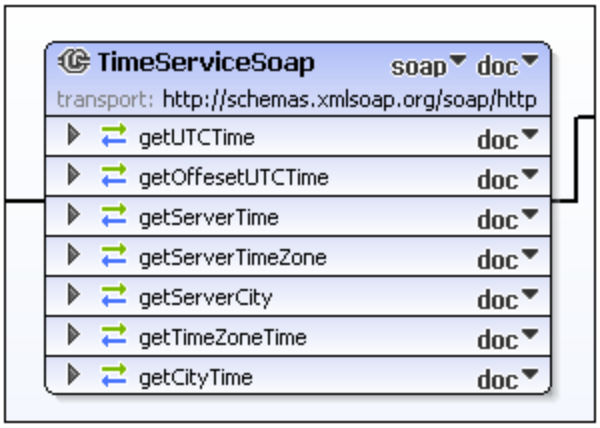
L'association d'un portType ou d'une interface avec une liaison est indiquée dans la Fenêtre principale avec une ligne de connexion noire reliant le champ portType box ou le champ d'interface au champ de liaison, celui-ci se trouvera dans la section Liaisons de la Fenêtre principale.

Liaisons

Une liaison définit des formats de message et des détails de protocole pour :

- Les opérations définies pour un portType (WSDL 1.1) particulier ou
- Les opérations et fautes définies par une interface particulière (WSDL 2.0).

Dans WSDL 1.1, des liaisons peuvent être créées pour les points terminaux SOAP 1.1 ou SOAP 1.2, ou pour les verbes GET et POST de HTTP 1.1. Dans WSDL 2.0, des liaisons peuvent être créées pour les points terminaux SOAP 1.1 ou SOAP 1.2, ou pour HTTP. Chaque liaison est représentée par un champ de liaison (*capture d'écran ci-dessous*) dans la section Liaison de la Fenêtre principale. Le champ de liaison contient toutes les opérations et/ou les fautes du portType associé ou de l'interface (*voir capture d'écran ci-dessous*).



Une liaison peut être associée avec un type de port ou une interface des manières suivantes :

- Cliquer avec la touche de droite sur un type de port ou une interface et choisir la commande **Créer une liaison pour portType** ou **Créer une liaison pour l'interface**, respectivement.
- Cliquer avec la touche de droite sur une liaison WSDL 1.1 et éditer la propriété *PortType*.
- Cliquer avec la touche de droite sur une liaison WSDL 2.0 et choisir la commande **Édition | Interface**.

Afin de définir la liaison, dans le premier champ à liste de choix situé à la droite du nom de liaison (*capture d'écran ci-dessous*), sélectionner le protocole requis. Dans WSDL 1.1, il s'agit soit de `soap 1.1`, `soap 1.2`, `http-get` ou `http-post` pour définir le type de liaison. Si vous sélectionnez un protocole SOAP, vous pouvez également définir (à l'aide de la deuxième liste de choix) si le style doit être `doc` ou `rpc`. Dans les documents WSDL 2.0, la propriété `wsoap:protocol` peut être ajoutée ou éditée par le biais de la commande **Édition** du menu contextuel de la liaison.



Dans WSDL 1.1, les encodages MIME (également référés en tant que liaisons MIME) sont définies au niveau du message. Pour définir un encodage MIME, cliquer avec la touche de droite sur le message (*capture d'écran ci-dessous*) et apposer la définition MIME appropriée. Dans la capture d'écran ci-dessous, les définitions MIME ont été créées pour le message `Output`.



Cliquer avec la touche de droite sur un item spécifique dans un champ de liaison pour ouvrir un menu sensible au contexte. À l'aide des menus contextuels, par exemple, des liaisons peuvent être apposées ou supprimées ; les items d'extensibilité peuvent être édités ; et les messages définis. Veuillez également noter que lorsqu'un champ de liaison ou un item contenu dans un champ de liaison est sélectionné, les définitions sont affichées dans l'assistant à la saisie Détails et peuvent y être éditées.

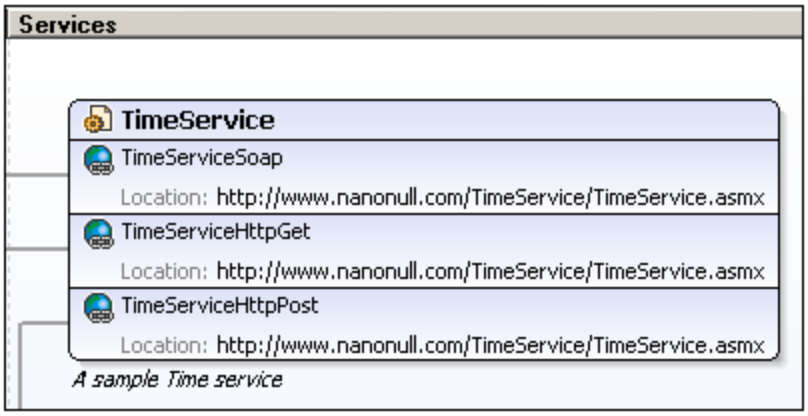
Un port peut être créé pour une liaison en cliquant avec la touche de droite sur la barre de titre d'un champ de liaison et en sélectionnant la commande **Créer port pour liaison** (documents WSDL 1.1) ou **Créer un point de terminaison pour liaison** (documents WSDL 2.0). Le port ou le point de terminaison associé est créé dans un champ de service (dans la section Services de la Fenêtre principale). L'association entre une liaison et un port est indiqué par une ligne de connexion noire.

La documentation pour les liaisons apparaît en bas des champs de liaison individuels.

Services

Un service regroupe un ensemble de ports liés (WSDL 1.1) ou de points de terminaison (WSDL 2.0). Il est représenté par un champ de service dans la section Services de la Fenêtre principale (*capture d'écran ci-dessous*). Chaque champ de service consiste en une ou plusieurs déclarations de port ou de point de terminaison (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



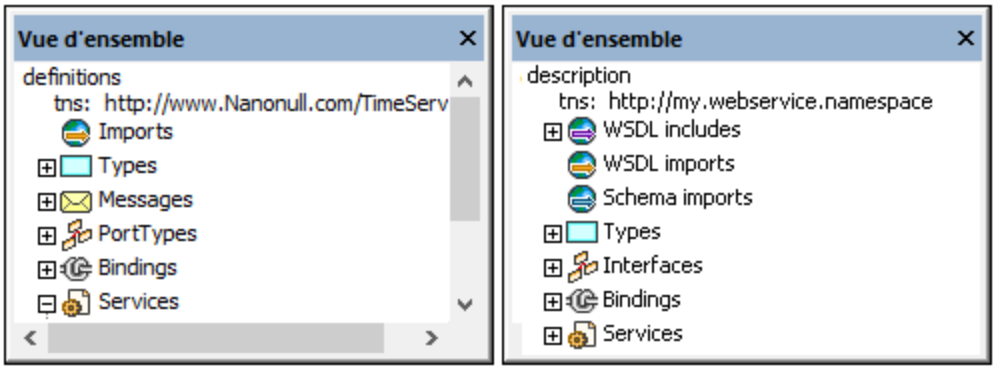
Le nom de service, le nom de port ou de point de terminaison, la liaison associée avec un port ou un point de terminaison, et les informations de l'adresse d'un port ou d'un point de terminaison peuvent être édités directement dans le champ de service ou dans les assistants à la saisie Détails. Cliquer avec la touche de droite sur un champ de service ou sur un item spécifique dans le champ de service pour ouvrir un menu contextuel dans lequel les commandes pertinentes au service ou à cet item sont disponibles.

La documentation pour les services apparaît en bas des champs de services individuels.

4.5.2 Assistant à la saisie Vue d'ensemble

L'**Assistant à la saisie Vue d'ensemble** (*capture d'écran ci-dessous*) fournit un aperçu du document WSDL en regroupant les différents composants du document dans des catégories structurelles et en recensant l'espace de noms cible, les schémas importés et les documents WSDL inclus/importés. Outre les types de port (ou les interfaces dans WSDL 2.0), les messages (WSDL 1.1), liaisons et services, les différents types définis dans le document sont également recensés.

Vous pouvez aussi gérer les imports et includes des fichiers de Schéma XML et WSDL dans l'Assistant à la saisie Vue d'ensemble.



Assistant à la saisie Vue d'ensemble dans WSDL 1.1 (gauche) et WSDL 2.0 (droite).

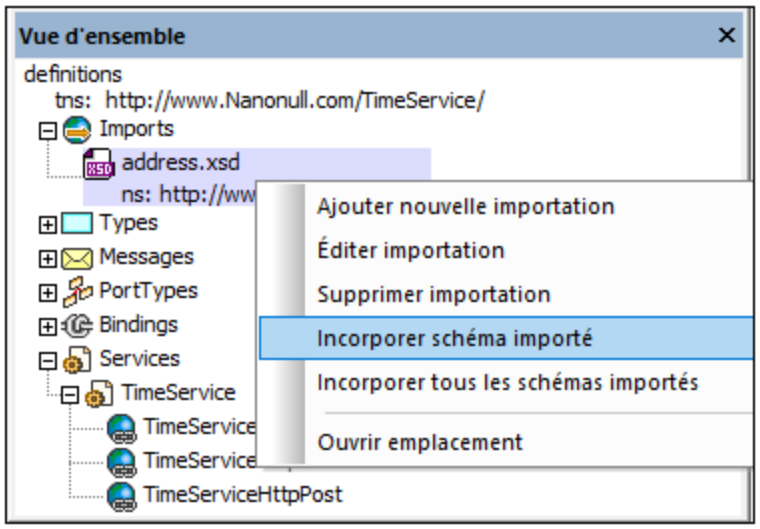
Dans chaque catégorie, les composants sont affichés dans un mode d'arborescence. Un item d'arborescence peut être agrandi et réduit, respectivement, pour révéler et dissimuler son contenu. La sélection d'un composant dans l'Assistant à la saisie Vue d'ensemble permet de l'afficher ainsi que ses priorités dans l'Assistant à la saisie Détails, où les propriétés peuvent être éditées. Les noms de WSDL et des composants de schéma qui sont affichés dans l'arborescence peuvent être édités directement dans les arborescences. Les composants définis en externe (ceux dans les documents WSDL inclus ou importés ou dans les schémas et affichés en gris), néanmoins, ne peuvent pas être édités. Les catégories individuelles dans l'Assistant à la saisie Aperçu sont expliquées ci-dessous.

Espace de noms cible (WSDL 1.1 et 2.0)

Indiqué dans l'arborescence par tns. L'espace de noms cible peut être édité dans l'Assistant à la saisie Vue d'ensemble. Tous les autres espaces de noms doivent être édités dans le Mode Texte.

Imports (WSDL 1.1)

Les fichiers de Schéma XML (XSD) et les fichiers WSDL peuvent être importés dans le document actif WSDL. Pour importer un Schéma XML ou un fichier WSDL, cliquer avec la touche de droite sur l'item Imports ou sur un fichier déjà importé dans la liste *Imports*, et choisir **Ajouter nouvelle importation**. Cliquer avec la touche de droite sur un fichier importer dans la liste Imports pour faire apparaître un menu contextuel dans lequel vous pouvez choisir d'ajouter une nouvelle importation, sélectionner un autre fichier pour remplacer le fichier sélectionné en tant qu'importation (**Éditer importation**), ou supprimer le fichier importé (**Supprimer importation**). Vous pouvez aussi ouvrir le fichier depuis son emplacement. Le fichier s'ouvre dans le Mode WSDL (fichier `.wsdl`) ou dans le Mode Schéma (fichier `.xsd`), et peut y être édité.

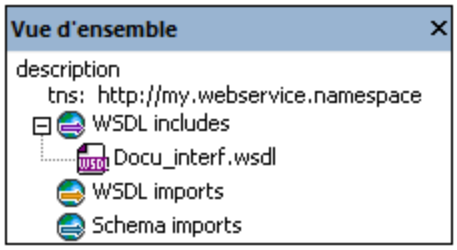


Un Schéma XML importé peut ensuite être incorporé dans le fichier WSDL. L'incorporation d'un schéma intégré crée le schéma en tant qu'un schéma inline dans le cadre de l'élément `types`, et l'élément `import` est supprimé. Afin d'intégrer un schéma importé, cliquer avec la touche de droite sur l'entrée du schéma dans la liste *Imports* et sélectionner la commande **Incorporer Schéma importé** ou **Incorporer tous les Schémas importés**. La dernière commande, qui s'applique à tous les schémas importés est aussi activée dans le menu contextuel de l'item *Imports*.

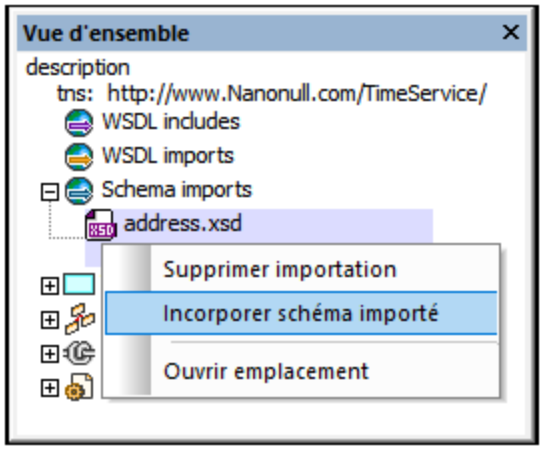
© 2016-2022 Altova GmbH

WSDL includes, WSDL imports, Schema imports (WSDL 2.0)

Les Schémas XML (XSD) peuvent être importés et les fichiers WSDL peuvent être inclus ou importés dans le document WSDL actif. Pour inclure ou importer un fichier, cliquer avec la touche de droite sur l'item respectif (*WSDL Includes, WSDL Imports, Schema Imports*), chercher le fichier que vous souhaitez inclure ou importer et l'ajouter. L'espace de noms d'un fichier importé est généré automatiquement depuis l'espace de noms cible du fichier importé.



Cliquer avec la touche de droite sur un fichier inclus ou importé pour faire apparaître un menu contextuel dans lequel vous pouvez choisir de supprimer un fichier ou de l'ouvrir dans XMLSpy. Le fichier s'ouvre dans le Mode WSDL (fichier `.wsdl`) ou Schéma (fichier `.xsd`), et peut y être édité. Un Schéma XML peut ensuite être incorporé dans le fichier WSDL (*voir capture d'écran ci-dessous*).



L'incorporation d'un schéma importé crée le schéma en tant qu'un schéma inline dans le cadre de l'élément `types`, et l'élément `import` est supprimé. Pour incorporer un schéma importé, cliquer avec la touche de droite sur l'entrée du schéma dans la liste *Imports* et sélectionner la commande **Incorporer le Schéma importé** ou **Incorporer tous les Schémas importés**. Cette deuxième commande, qui s'applique à tous les schémas importés est également activée dans le menu contextuel de l'item *Imports*.

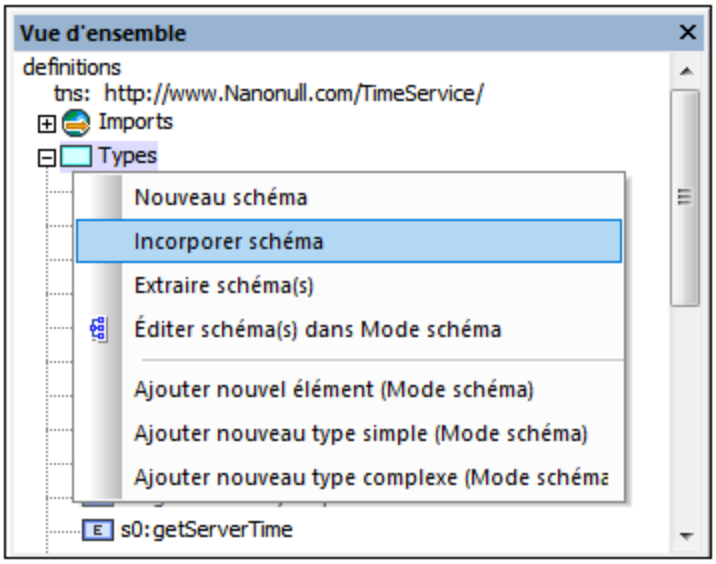
Types (WSDL 1.1 et 2.0)

Regroupe tous les types définis dans le document WSDL (en noir) et dans tout schéma importé ou document WSDL (en gris).



Les fonctions suivantes sont disponibles :

- *Créer un nouveau schéma :* cliquer avec la touche de droite sur l'item *Types* et choisir **Nouveau Schéma** (*voir capture d'écran ci-dessous*). Un nouveau schéma incorporé vide est créé dans le fichier WSDL et tous les schémas incorporés, y compris le nouveau schéma vide, sont ouverts dans le Mode Schéma. Une fois avoir édité le document de schéma, l'enregistrement de vos changements dans le fichier de schéma entraînera l'écriture des changements sur le document WSDL. Vous pouvez ensuite fermer le document de schéma. Cette fonction est utile si vous souhaitez créer un nouveau schéma intégré dans le document WSDL.



- *Incorporer des schémas:* cliquer avec la touche de droite sur l'item *Types* et choisir **Incorporer le Schéma**. Un dialogue Ouvrir-fichier s'ouvre dans lequel vous pouvez chercher le ficher de schéma que vous souhaitez intégrer. Cliquer sur **OK** pour créer le schéma en tant qu'un schéma inline dans l'élément `types`. Si le schéma sélectionné a déjà été importé, vous serez invité à choisir si vous voulez incorporer le schéma déjà importé.

- *Extraire des schémas:* cliquer avec la touche de droite sur l'item *Types* et choisir **Extraire (des) schéma(s)**. Chacun des schémas incorporés est ouvert en tant qu'un fichier temporaire dans le Mode Schéma et un dialogue Enregistrer en tant que s'ouvre pour chaque fichier. Si vous choisissez d'enregistrer le fichier de schéma, ce schéma sera extrait, enregistré sur l'emplacement que vous spécifiez puis importé dans le fichier WSDL. Le fichier de schéma n'existera plus en tant que schéma inline mais en tant qu'un schéma externe, importé.

 © 2016-2022 Altova GmbH

- *Éditer des schémas:* vous pouvez éditer des schémas incorporés dans le Mode Schéma. Cliquer avec la touche de droite soit dans l'item *Types* soit sur le nom d'un component de schéma dans la liste *Types*, puis choisir **Éditer les Schéma(s)** ou **Éditer le Schéma**, respectivement. Cela entraîne la génération immédiate d'un fichier XSD temporaire depuis les définitions `types` dans le document WSDL. Ce document XSD est affiché dans le Mode Schéma et peut être édité. Une fois avoir terminé l'édition du document XSD, l'enregistrement des changements entraînera l'enregistrement des changements sur les définitions `types` dans le document WSDL. Si vous fermez le document XSD sans enregistrer les changements, les définitions `types` définitions du document WSDL ne seront pas modifiés.

- *Ajouter des composants de schéma :* vous pouvez ajouter un élément de Schéma XML (WSDL 1.1 et 2.0), simpleType (WSDL 1.1) ou complexType (WSDL 1.1). Pour ce faire, cliquer avec la touche de droite soit sur l'item *Types* soit sur le nom d'un composant de schéma dans la liste *Types*, puis sélectionner la commande **Ajouter** pertinente. Un fichier XSD temporaire sera généré immédiatement depuis les définitions `types` dans le document WSDL qui sera affiché dans le Mode Schéma. Ce fichier contient le nouveau composant, sans nom. Vous pouvez ensuite éditer ce composant. Lors de l'enregistrement du fichier, le nouveau composant sera écris dans les définitions `types` dans le document WSDL.

- *Supprimer des composants de schéma:* un composant de schéma peut être supprimé en cliquant avec la touche de droite dans la liste de *Types* et en sélectionnant **Supprimer** dans le menu contextuel.

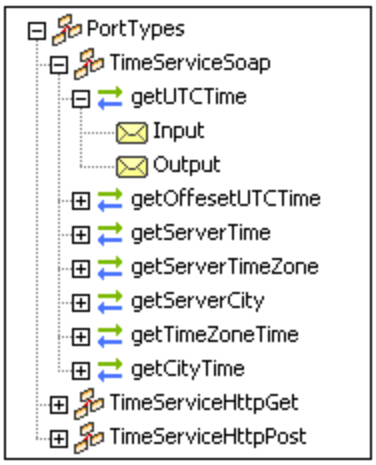
Messages (WSDL 1.1)

Lorsqu'un message ou son sous-composant est sélectionné, les propriétés de ce message ou de son sous-composant sont affichées dans l'Assistant à la saisie Détails, où elles peuvent être éditées. De plus, le menu contextuel vous permet de procéder aux étapes suivantes :

- Lorsqu'un message est sélectionné dans l'assistant à la saisie Vue d'ensemble, vous pouvez ajouter une partie de message à ce message ou bien vous pouvez supprimer le message ou ajouter un nouveau message.
- Lorsqu'une partie de message est sélectionné dans l'assistant à la saisie Vue d'ensemble, vous pouvez ajouter un autre partie de message à ce message ou bien vous pouvez supprimer la part de message sélectionnée.
- La commande **Synchroniser** marque le message sélectionné ou la partie de message dans le champ portType pertinent.

PortTypes (WSDL 1.1)

Les fonctions suivantes sont disponibles pour portTypes via les menus contextuels.



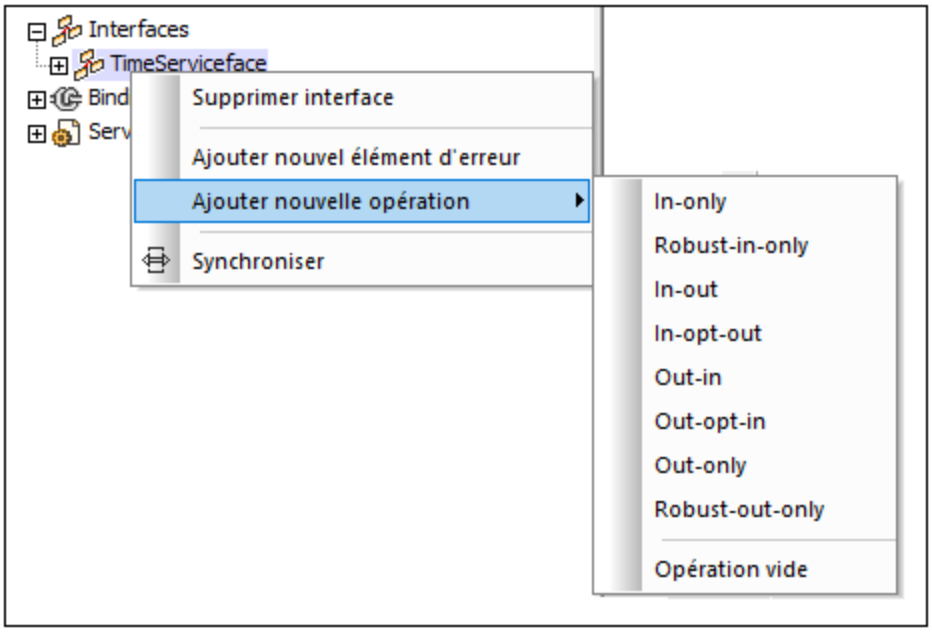
- Si l'item *PortTypes* est sélectionné, il est possible d'ajouter un portType.
- Si un portType est sélectionné, les portTypes peuvent être ajoutés, le portType sélectionné peut être supprimé et les opérations peuvent être ajoutées au portType sélectionné.
- Si une opération est sélectionnée, des opérations supplémentaires peuvent être ajoutées, l'opération sélectionnée peut être supprimée, et les éléments (entrée, sortie ou faute) peuvent être ajoutés à l'opération sélectionnée.
- Si un élément de message (entrée, sortie ou faute) est sélectionné, des messages supplémentaires peuvent être ajoutés et le message sélectionné peut être supprimé.
- La commande **Synchroniser** marque le portType, l'opération ou le message sélectionné.

Interfaces (WSDL 2.0)

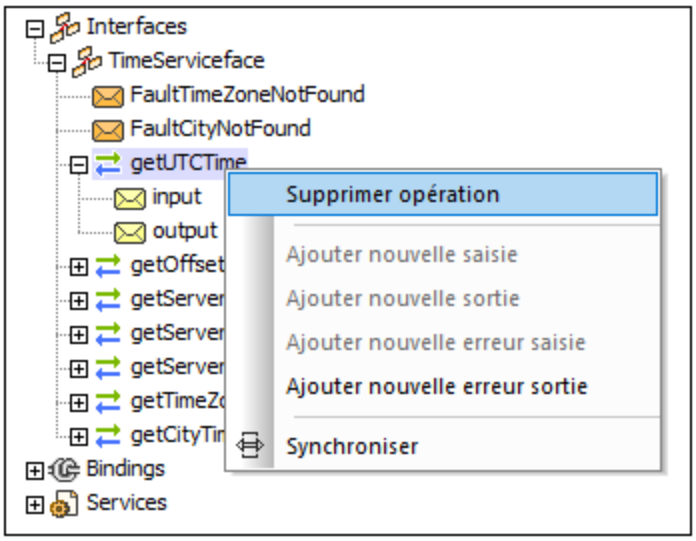
Les interfaces peuvent être gérées à l'aide des menus contextuels.

- Pour ajouter une interface, cliquer avec la touche de droite l'item *Interfaces* et choisie la commande de menu, **Ajouter une nouvelle interface**.
- Cliquer avec la touche de droite sur une interface pour ouvrir un menu (*voir capture d'écran ci-dessous*) avec des commandes permettant de supprimer l'interface sélectionnée, et d'ajouter des fautes et des opérations à la définition de l'interface sélectionnée. Le type de l'opération à ajouter peut être spécifié dans le sous-menu de la commande **Ajouter une nouvelle opération**. Une opération de liaison correspondante est ajoutée à toutes les liaisons référençant à l'interface. De la même manière, lorsqu'une opération est supprimée, le référencement aux opérations de liaison est également supprimé.

© 2016-2022 Altova GmbH



- Cliquer avec la touche de droite sur une opération pour faire apparaître des commandes (*voir capture d'écran ci-dessous*) et permettre de supprimer l'opération sélectionnée et d'ajouter des éléments (comme `infault` et `outfault`) dans l'opération.



- Cliquer avec la touche de droite sur un élément de message pour faire apparaître un menu par lequel vous pouvez supprimer le message sélectionné.
- Cliquer sur la commande **Synchroniser** pour souligner l'interface, l'opération ou le message sélectionné dans le design.

Liaisons (WSDL 1.1 et 2.0)

Une fois qu'une liaison a été sélectionnée, des liaisons supplémentaires peuvent être ajoutées aux liaisons existantes, la liaison sélectionnée peut être supprimée, et il est possible d'insérer des opérations pour la liaison sélectionnée. En sélectionnant une opération ou un message, les mêmes options sont disponibles

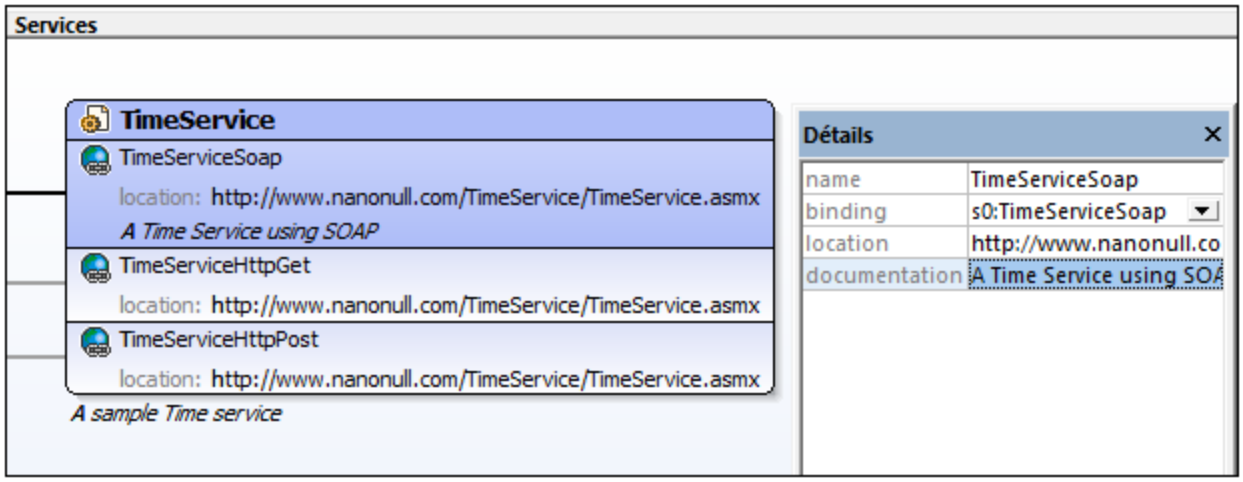
telles que décrites pour les opérations et les messages dans la catégorie PortTypes (WSDL 1.1) ou Interfaces (WSDL 2.0). Cliquer sur la commande **Synchroniser** pour souligner la liaison, l'opération ou le message sélectionné.

Services (WSDL 1.1 et 2.0)

Une fois qu'un service a été choisi, des services supplémentaires peuvent être ajoutés, le service sélectionné peut être supprimé et les ports peuvent être ajoutés pour le service sélectionné. Cliquer sur la commande **Synchroniser** pour souligner le service ou le port sélectionné.

4.5.3 Assistant à la saisie Détails

L'assistant à la saisie Détails affiche les propriétés de l'item sélectionné dans la Fenêtre principale ou dans l'assistant à la saisie Aperçu (*capture d'écran ci-dessous*). Ces propriétés peuvent être éditées dans l''assistant à la saisie Détails.



Par exemple, tel que montré dans la capture d'écran ci-dessus, si, dans la Fenêtre principale, le port `TimeServiceSoap` (du service `TimeService`) est sélectionné, alors les propriétés de `TimeServiceSoap` sont affichées dans l'assistant à la saisie Détails, où ils peuvent être édités. Pour éditer un champ de texte comme `name` ou `description`, double-cliquer dans le champ et éditer le texte. En ce qui concerne certaines propriétés, comme `binding` dans la capture d'écran exemple ci-dessus, où une sélection peut être effectuées depuis les options, une liste de choix vous permet de faire votre choix depuis les options disponibles.

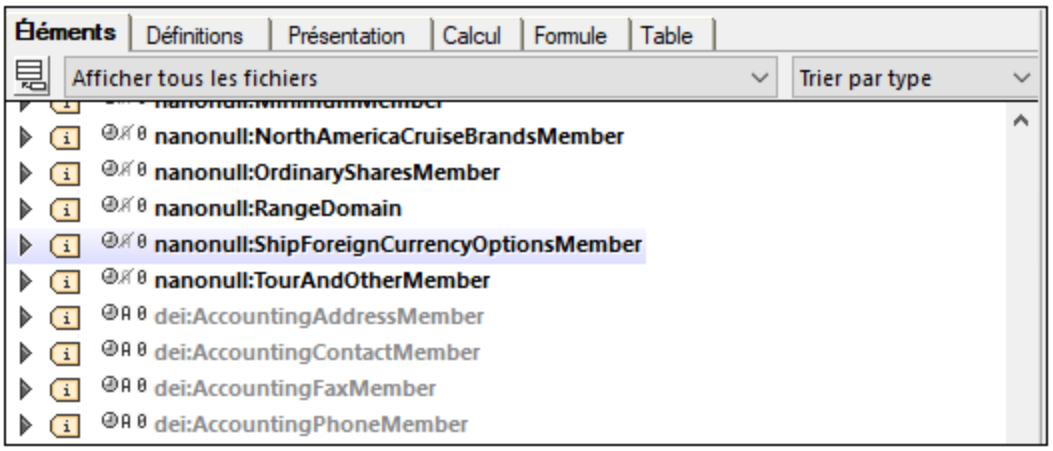
© 2016-2022 Altova GmbH

4.6 Mode XBRL

Le Mode XBRL est une interface graphique qui vous permet d'éditer des taxonomies XBRL. Il permet une validation rapide et des messages d'erreur, ce qui aide les utilisateurs à développer rapidement et précisément des taxonomies.

Le Mode XBRL est constitué des parties suivantes:

- Une Fenêtre principale à six onglets : Éléments, Définitions, Présentation, Calcul, Formule, Table (*capture d'écran ci-dessous*). Les fonctions principales de ces onglets sont décrites dans les sous-sections de cette section.



- Trois puissants assistants à la saisie : Aperçu, Éléments globaux, Détails. Ceux-ci vous permettent de gérer les fichiers de taxonomie et d'éditer la taxonomie dans la Fenêtre principale.
- Une Fenêtre Messages qui affiche des messages concernant la validité de la taxonomie.

Cette section apporte une description de la Fenêtre principale et des assistants à la saisie du Mode XBRL et contient des informations concernant leur utilisation. Pour plus d'informations, voir les sections XBRL et la description des commandes dans le menu XBRL.

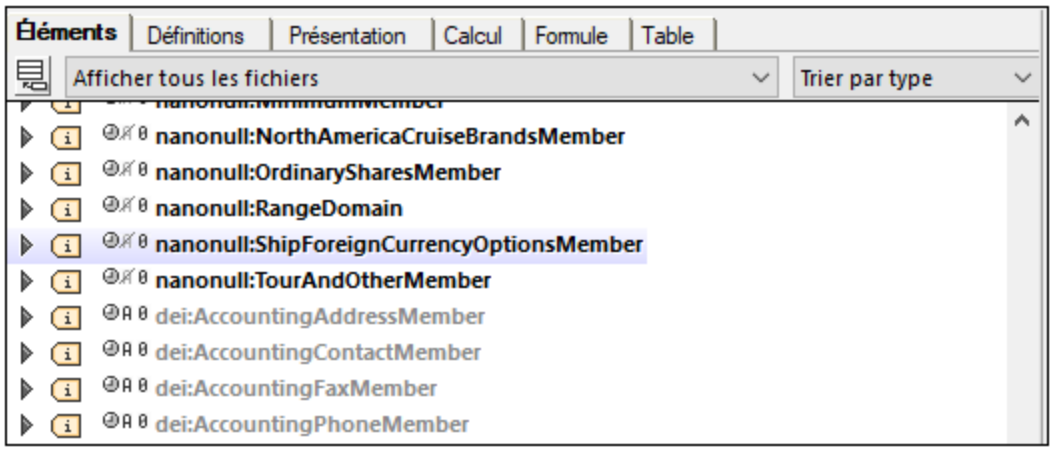
Fonctions supplémentaires du Mode XBRL

Outre les fonctions d'édition décrites dans cette section, les fonctions suivantes sont disponibles :

- Générer une documentation : crée des fichiers d'informations détaillées en formats HTML, Word et RTF.
- Imprimer le mode actuel permet d'imprimer le mode actuel à prendre à l'aide de la commande **Fichier | Imprimer** de XMLSpy.

4.6.1 Fenêtre principale : onglet Éléments

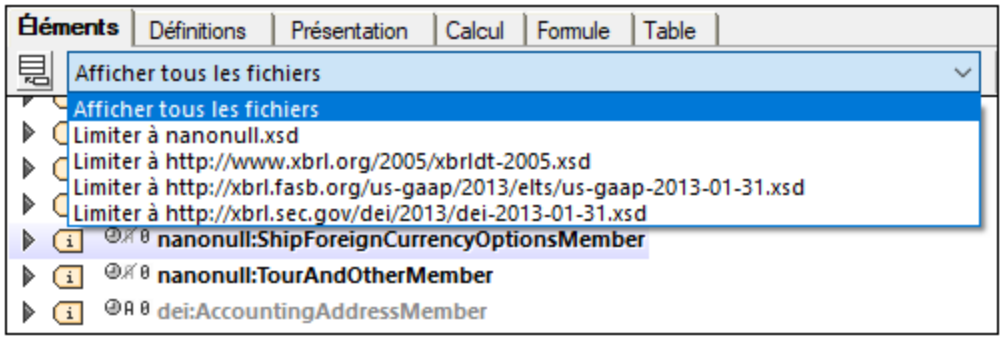
L'onglet **Éléments** de la Fenêtre principale (*capture d'écran ci-dessous*) affiche les concepts de la taxonomie, y compris, par défaut, les concepts contenus dans les taxonomies importées. Les concepts de la taxonomie actuelle sont affichés en noir ; les concepts dans les taxonomies importées sont affichés en gris.



Pour plus d'informations, voir les sections XBRL et la description des commandes dans le menu XBRL.

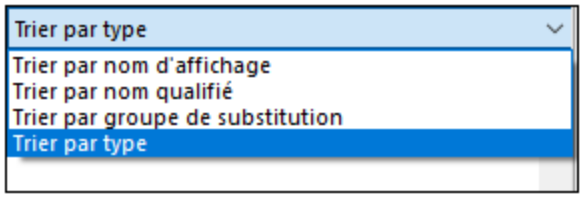
Sélectionner la taxonomie à afficher

Dans la liste de choix *Fichier* située en-dessous des onglets de la Fenêtre principale (*capture d'écran ci-dessous*), vous pouvez sélectionner si vous souhaitez afficher uniquement les concepts provenant de la taxonomie actuelle, ou les concepts provenant de la taxonomie actuelle plus ses taxonomies importées. Choisir *Afficher tous les fichiers* pour montrer les taxonomies importées. La possibilité de filtrer des taxonomies volumineuses importées à partir de l'écran permet de booster la vitesse d'édition.



Éléments de tri

Dans la liste de choix *Trier* située en-dessous des onglets de la Fenêtre principale et à droite (*capture d'écran ci-dessous*), vous pouvez trier les éléments dans la Fenêtre principale.



Le critère de tri peut être un des suivants :

© 2016-2022 Altova GmbH

- *Trier par nom :* les éléments sont listés dans l'ordre alphabétique de leur nom. Les noms comprennent le préfixe. Donc **abc:yname** se produira avant **bcd:xname**.
- *Trier par nom qualifié :* les noms d'élément sont entièrement résolus (ce qui signifie que leurs espaces de noms sont agrandis), et les noms agrandis sont listés dans l'ordre alphabétique. Donc si le préfixe d'espace de noms **abc** est lié à l'espace de noms **http:people.altova.com**, le nom d'élément **abc:yname** sera résolu à **http:people.altova.com:yname**
- *Trier par groupe de substitution :* il existe quatre groupes de substitution : *items*; *tuples*; *hypercubeItems*; *dimensionItems*.
- *Trier par type :* réfère à l'attribut type de l'élément XBRL.

Trouver des éléments dans la Fenêtre principale

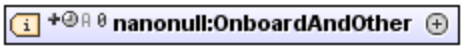
Afin de trouver un élément dans la Fenêtre principale, appuyer sur la combinaison de touches **Ctrl+F**. Le dialogue Recherche s'ouvre dans lequel vous pourrez saisir le string avec lequel vous souhaitez effectuer une recherche. Les préfixes d'espace de noms utilisés dans la taxonomie peuvent aussi être cherchés.

À propos des concepts (les éléments <element>)

Chaque concept de taxonomie est défini dans un élément de Schéma XML `<element>` (*voir liste ci-dessous*). Cette définition de texte clair peut être visionnée en passant au Mode texte du document de taxonomie. (Afin de voir les définitions de texte d'un document de taxonomie importé, celui-ci devra être ouvert dans XMLSpy.)

```
<xs:element id="icui_UnrealizedHoldingLoss"
       name="UnrealizedHoldingLoss"
       substitutionGroup="xbrli:item"
       type="xbrli:monetaryItemType"
       xbrli:balance="credit"
       xbrli:periodType="instant"
       abstract="false"
       nillable="true"/>
```

Chaque élément (ou concept) est affiché dans l'onglet Éléments avec une icône indiquant son groupe de substitution (item, tuple, hypercube ou dimension). De plus, les icônes indiquant les valeur des attributs `balance`, `periodType`, `abstract` et `nillable` du concept sont affichés à la gauche du nom de concept (*capture d'écran ci-dessous*).



Pour éditer le nom du concept, double-cliquer sur le nom du concept et éditer le nom.

Substitution group

La valeur de groupe de substitution d'un concept est indiquée par l'icône du concept :

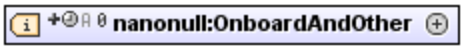
 `xbrli:item`

 `xbrli:tuple`

 `xbrldt:hypercubeItem`

 `xbrldt:dimensionItem`

The capture d'écran ci-dessous montre un élément avec une valeur `substitutionGroup` de `xbrli:item`.



Les attributs balance, periodType, abstract, nillable

Les icônes supplémentaires situées à la gauche d'un nom d'élément (*voir capture d'écran ci-dessus*) indiquent les valeurs des attributs principaux du concept, de la gauche à la droite :

- `xbrli:balance`: un signe Plus pour une valeur de `credit`, un signe Moins pour `debit`
- `xbrli:periodType`: un symbole de montre avec un segment gris entre les aiguilles de la montre pour une valeur de `duration`, un segment blanc pour `instant`
- `xs:abstract`: icône noire `A` si `true`, grise si `false`
- `xs:nillable`: icône noire `0` si `true`, grise si `false`

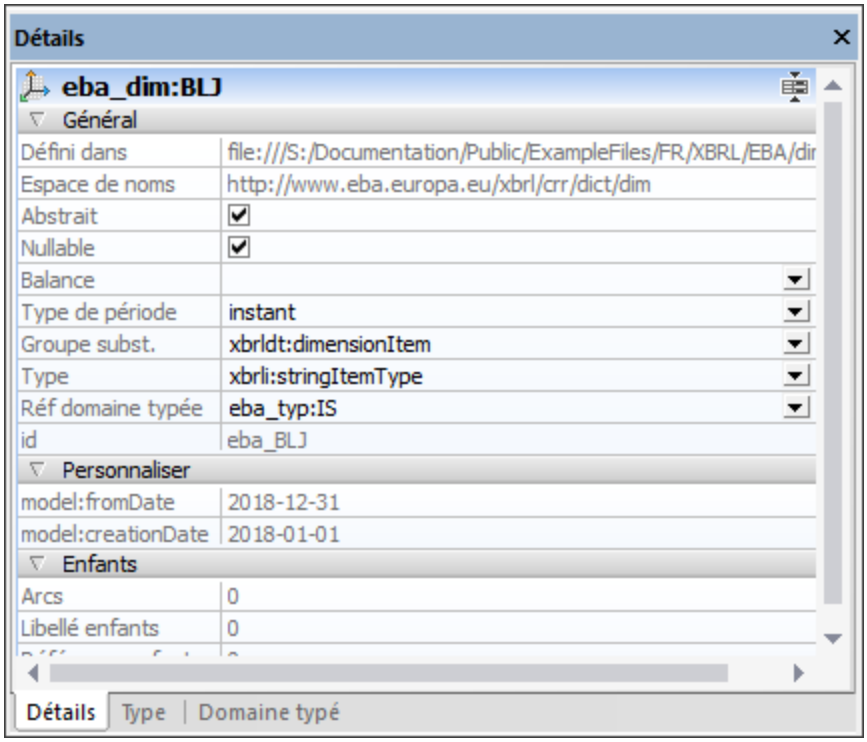
Veuillez noter que les attributs `xbrli:` recensés ci-dessus proviennent du schéma XBRL et que les attributs `xs:` proviennent du schéma Schéma XML.

Dans la capture d'écran ci-dessus, le signe Plus indique que la valeur de l'attribut `xbrli:balance` est `credit`. Les valeurs des autres attributs dans la capture d'écran (`xbrli:periodType`, `xs:abstract`, `xs:nillable`), sont respectivement : `duration`, `false`, (par ex. pas abstrait), et `true` (par ex. nullable).

Les valeurs de ces quatre attributs sont affichés dans une fenêtre popup lorsque le curseur est placé sur une des quatre icônes. Cliquer sur une de ces icônes pour faire apparaître une liste de choix contenant les valeurs autorisées pour cet attribut. Vous pouvez sélectionner un des choix autorisés, et éditer rapidement la valeur des attributs de concept.

Il est également possible d'éditer ces valeurs d'attribut dans l'assistant à la saisie Détails (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Les attributs substitutionGroup, type

Cliquer sur le symbole fléché situé à gauche du nom de l'élément pour agrandir l'affichage de l'élément de manière à ce que les valeurs des deux attributs, `xs:substitutionGroup` et `xs:type`, sont affichés graphiquement (*capture d'écran ci-dessous*).

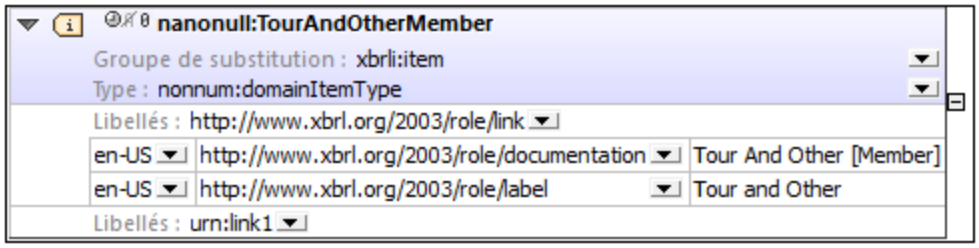


Dans la capture d'écran ci-dessus, la valeur des attributs `xs:substitutionGroup` et `xs:type` attributes sont respectivement : `xbrli:item` et `xbrli:monetaryItemType`. Ces valeurs peuvent être éditées en sélectionnant des valeurs alternatives depuis la liste déroulante des listes de choix correspondantes. Les deux valeurs d'attribut peuvent être éditées dans l'assistant à la saisie Détails (*voir capture d'écran plus haut ci-dessus*) en double-cliquant le champ de valeur respectif dans assistant à la saisie Détails et en éditant la valeur par le biais du clavier.

Liens Label

Pour ajouter un enfant de libellé, cliquer avec la touche de droite sur un élément et sélectionner **Ajouter rôle de lien de libellé** depuis le menu contextuel. Cela permet d'ajouter une ligne de libellé (*capture d'écran ci-*

dessous), dans laquelle vous pouvez spécifier les propriétés du libellé (langue, type de libellé et valeur). Vous pouvez agrandir et réduire les libellés en cliquant sur les signes + et - du côté droit de la fenêtre.



Les propriétés de type de langue et de libellé de chaque libellé peuvent être éditées en sélectionnant la valeur de propriété requise depuis les listes de choix respectives (*capture d'écran ci-dessus*).

Liens Reference

Pour ajouter un enfant de référence, cliquer avec la touche de droite sur un élément et sélectionner **Ajouter rôle de la liaison de référence** depuis le menu contextuel. Cela permet d'ajouter une fenêtre de référence dans laquelle vous pourrez éditer les propriétés de la référence en cliquant dans le champ Référence et en éditant l'URN de référence. Vous pouvez agrandir et réduire les références en cliquant sur les signes + et - du côté droit de la fenêtre.

4.6.2 Fenêtre principale : onglets Définitions, Présentation, Calcul, Formule, Table

La Fenêtre principale contient des onglets pour chacune des trois relations entre les concepts :

- relations Définition relationships, affichées dans l'onglet Définitions
- relations Présentation relationships, affichées dans l'onglet Présentation
- relations Calcul relationships, affichées dans l'onglet Calcul
- définitions et relations Formule, affichées dans l'onglet Formule
- définitions Table, affiché dans l'onglet Table

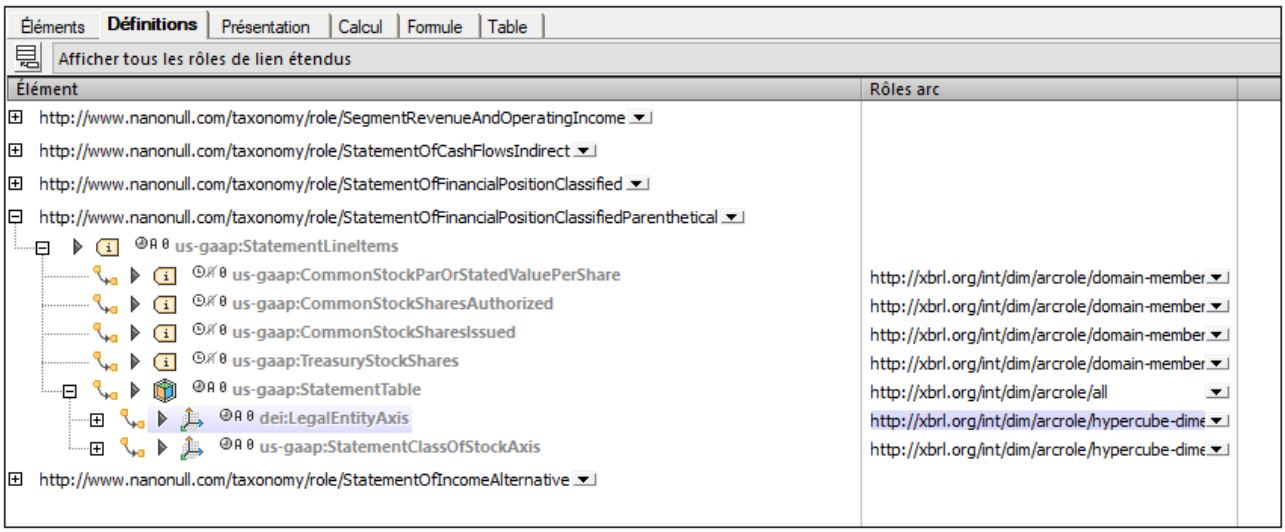
Chaque onglet (Définition, Présentation, Calcul, Formule et Table) affiche les relations de taxonomie de ce type (*capture d'écran ci-dessous*) et vous permet d'éditer les relations graphiquement. Une relation entre deux concepts (qu'il s'agisse d'une relation de définition, de présentation ou de calcul) est créée en construisant un arc s'étendant d'un concept à un autre. Cet arc *De–vers* est indiqué dans l'affichage graphique en tant qu'une flèche courbe. La relation entre les deux concepts (dans la direction *de–vers*) est spécifiée et est connue en tant que son arcrole. L'arcrole d'un arc est affiché dans l'assistant à la saisie Détails lorsque l'élément à l'extrémité *vers* d'une relation est sélectionné, et, dans le cas de relations de définition, dans une colonne Arcrole dans l'onglet Définitions (*voir capture d'écran ci-dessous*).

La manière dont sont affichés les arcs de relation est identique pour tous les types de relation (définition, présentation et calcul). Dans cette section, chaque onglet est considéré séparément, avec la description de base de la manière dont les relations sont affichées dans la section de l'onglet Définitions. Des informations supplémentaires ou spécifiques concernant les arcs de présentation et de calcul se trouvent dans leurs sections respectives. Pour plus d'informations voir les sections, Créer des relations : Partie 1 et Créer des relations : Partie 2.

© 2016-2022 Altova GmbH

Onglet Définitions

L'**onglet Définitions** montre toutes les définitions de la taxonomie. Ces définitions sont spécifiées dans des arcs de définition contenus dans les fichiers de relation de définition (`.xml`). Dans l'onglet Définition du Mode XBRL, la structure résultant de l'ensemble des arcs de définition est affichée dans une forme d'arborescence agrandissable/réduisible (*capture d'écran ci-dessous*).



Dans cet affichage graphique des définitions, chaque définition est affichée en tant que flèche courbe à deux points de terminaison (un point de terminaison *de* et un autre *vers*). Le type de relations entre les deux éléments de chaque point de terminaison est affiché dans la colonne *Arcrole* de l'élément au point de terminaison *vers*. Par exemple, dans une relation `hypercube-dimension`, la relation (ou arcrole) est recensée avec l'élément qui représente la part `dimension` de la paire d'élément. Les URI Arcrole peuvent donc être saisis dans l'assistant à la saisie Détails.

Pour plus d'informations voir les sections concernant les relations de définitions, voir la section Créer des relations : Partie 1.

Onglet Présentation

Les arcs de présentation ont les mêmes attributs que les arcs de définition, et ils travaillent de la même manière (voir onglets Définitions ci-dessus). Les URI Arcrole sont saisis dans l'assistant à la saisie Détails. Un attribut de présentation important est l'attribut `order`, qui détermine l'ordre dans lequel les éléments enfants d'un seul élément de parent sont présentés. Dans l'onglet de Présentation, ce genre d'élément enfant est affiché dans l'ordre ascendant correct. La valeur de l'attribut d'ordre peut être modifié rapidement en glissant un élément enfant à une autre position dans la liste classée. La valeur de l'attribut `order` peut aussi être changée dans l'assistant à la saisie Détails (*voir capture d'écran ci-dessous*).

Pour plus d'informations on presentation relationships, voir la section Créer des relations : Partie 2.

Onglet Calcul

Les arcs de calcul ont les mêmes attributs que les arcs de définition, et ils travaillent de la même manière (voir onglets Définitions ci-dessus). Les URI Arcrole sont saisis dans l'assistant à la saisie Détails. Il y a deux types d'arcroles:

- ceux qui totalisent les valeurs des éléments vers un autre élément ; et
- ceux qui ne représentent pas une relation de totalisation mais une relation équivalente

Dans ce cas, l'attribut `weight` détermine de combien la valeur de cet élément est totalisée à l'élément agrégateur. Une valeur de `1.0` indique que 100% de la valeur devrait être totalisée. Une valeur négative indique que la valeur doit être soustraite depuis l'agrégateur. La valeur de l'attribut `weight` peut être éditée dans l'onglet Calcul ainsi que dans l'assistant à la saisie Détails.

Onglet Formule

Les formules XBRL peuvent être définies et gérées dans l'onglet Formule. L'onglet Formule est utilisé avec l'assistant à la saisie Aperçu et l'assistant à la saisie Détails pour créer et éditer des formules. Les Définitions et les relations entre les composant de formule peuvent être exécutées dans le diagramme. Pour plus d'informations, voir la section Éditeur de formule XBRL.

Onglet Table

L'onglet Table propose un autre moyen de définir les affichages des concepts définies dans les taxonomies XBRL. Ces définitions sont contenues dans des bases de lien de table que vous pouvez créer et éditer dans l'onglet Table. Voir la section Éditeur de définitions de Table XBRL pour plus d'informations concernant l'édition des bases de lien de table.

Éditer dans le mode graphique

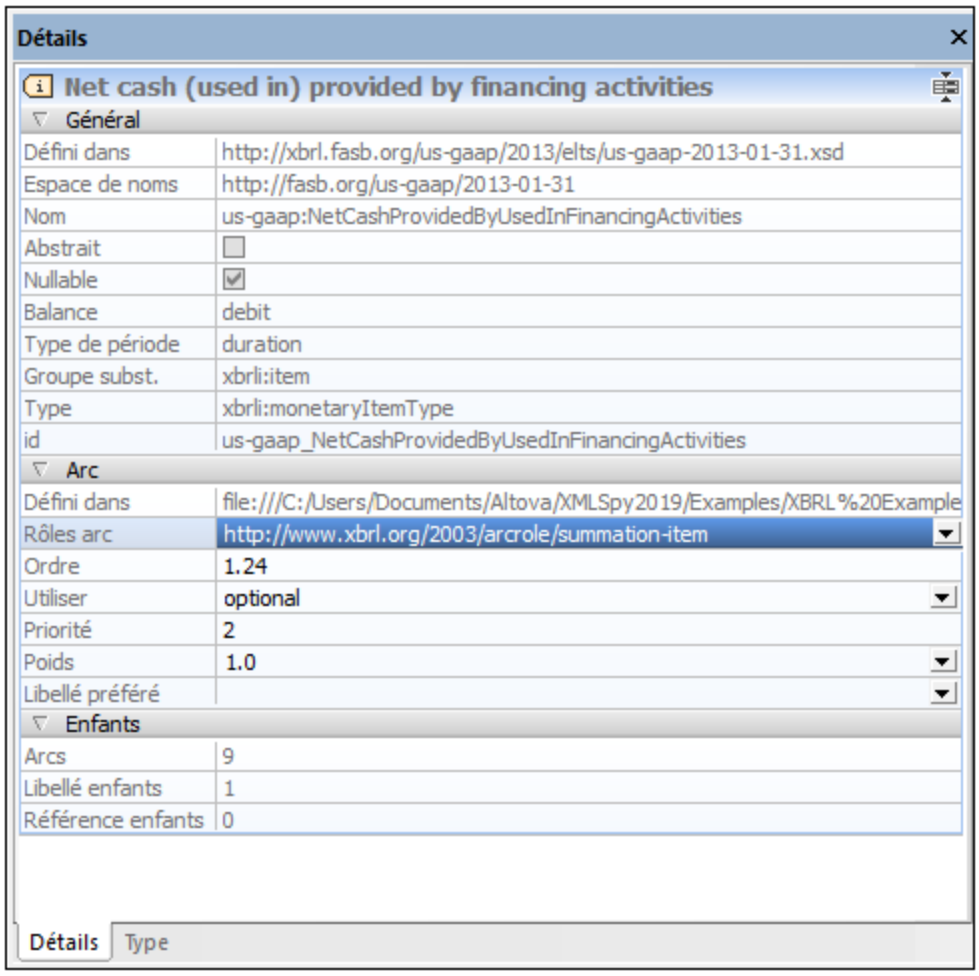
Les différentes possibilités d'édition sont disponibles :

- Les éléments peuvent être glissés depuis l'assistant à la saisie Éléments globaux et déposés dans une relation *vers* dans un élément dans l'arborescence.
- Un arcrole peut être créé ou édité pour un élément en sélectionnant l'arcrole requis (dans la colonne *Arcrole*). La relation définie dans l'arcrole exprime une relation *de–vers*, avec l'élément sélectionné se produisant au point de terminaison *vers* de la relation.
- Les éléments peuvent être glissés vers des emplacements alternatifs dans l'arborescence. Il s'agit d'un moyen rapide pour changer la valeur de l'attribut `order`.
- Les propriétés d'un élément peuvent être éditées en cliquant sur un de ses symboles de propriété et en l'éditant, ou en agrandissant son champ de propriétés et en éditant les propriétés dans ce champ.

Éditer dans l'assistant à la saisie Détails

Lorsqu'un élément est sélectionné dans la Fenêtre principale, les propriétés de son arc de définition sont affichées dans l'assistant à la saisie Détails dans la section Arc (*voir capture d'écran ci-dessous*). Les valeurs de ces propriétés peuvent être éditées en double-cliquant un champ de valeur et en saisissant la valeur requise ou, si disponible, en choisissant une valeur dans la liste déroulante de sa liste de choix.

© 2016-2022 Altova GmbH



De plus, les propriétés de l'arc de l'élément sélectionné sont affichées dans la section *Arc* de l'assistant à la saisie Détails. L'arc sera un arc de définition, un arc de présentation ou un arc de calcul conformément à l'onglet actuellement actif. L'arcrole peut être saisi dans le champ *Arcrole*.

Les relations de libellé et de référence sont recensées sous l'en-tête *Children*. Ces relations peuvent être éditées dans l'onglet Éléments.

Pour plus d'informations, voir la section XBRL.

4.6.3 Assistants à la saisie dans le Mode XBRL

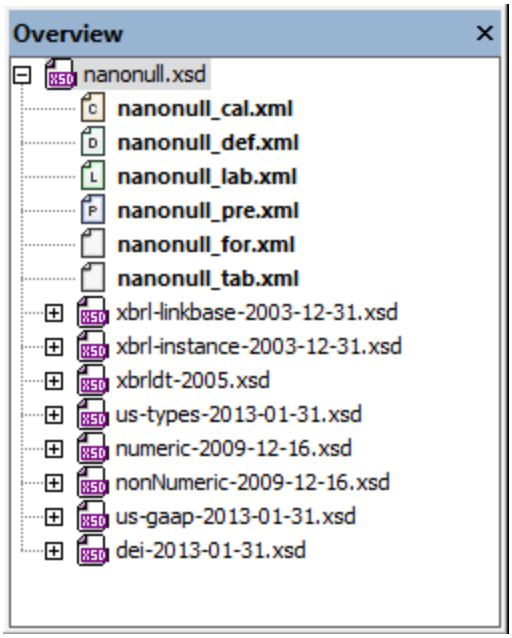
Le Mode XBRL met à votre disposition les assistants à la saisie suivants :

- Un assistant à la saisie Aperçu
- Un assistant à la saisie Éléments globaux
- Un assistant à la saisie Détails

Les informations de validation sont affichées dans la fenêtre Messages. Pour plus d'informations, voir la section XBRL.

Assistant à la saisie Aperçu

L'assistant à la saisie Aperçu affiche les fichiers de taxonomie dans une structure d'arborescence (*capture d'écran ci-dessous*).



L'arborescence est organisée comme suit :

- Le fichier de concepts principaux (un document de Schéma XML) est affiché à la racine de l'arborescence et il s'agit actuellement du fichier actif.
- Les fichiers de linkbase de relation (fichiers XML) ont une icône de fichier colorée arborant un caractère correspondant au caractère initial du type de relation.

  indique un linkbase de définitions ;

  indique un linkbase de libellés ;

  indique un linkbase de calculs ;

  indique un linkbase de présentation ; et

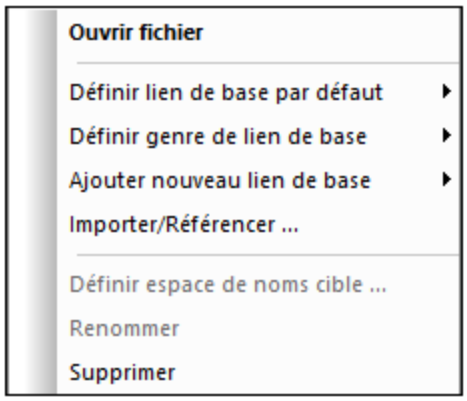
  indique un linkbase de référence ;

  indique un linkbase de formule.
- Les schémas importés sont recensés à des niveaux inférieurs de la hiérarchie. Tous les fichier de Schéma XML (`.xsd`) sont indiqués avec une icône XSD.

© 2016-2022 Altova GmbH

Cette information affichée dans l'assistant à la saisie Aperçu est obtenue à partir de l'élément `/schema/annotation/appinfo` du fichier de concepts principaux. Voir fichiers de taxonomie pour plus d'informations concernant cet élément.

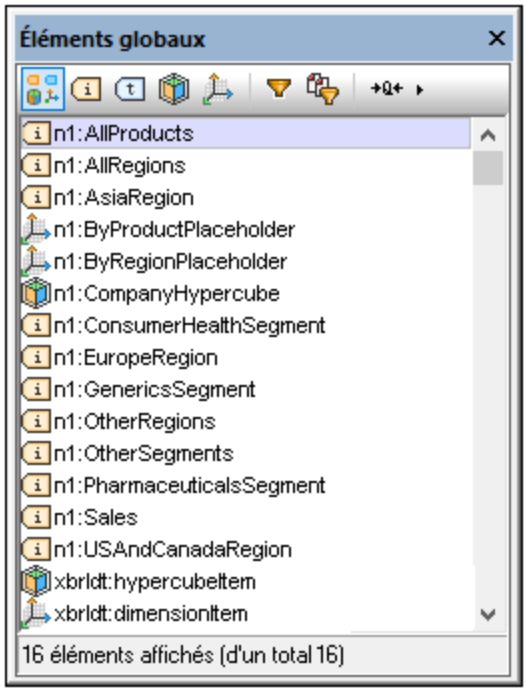
Cliquer avec la touche de droite sur un fichier dans l'assistant à la saisie Aperçu pour ouvrir un menu contextuel (*capture d'écran ci-dessous*) dans lequel les commandes suivantes sont disponibles :



- *Ouvrir fichier :* ouvre le fichier de Schéma XML ou XML sélectionné dans XMLSpy.
- *Définir lien de ressources par défaut :* s'il existe plusieurs fichiers d'un seul genre de relation (par exemple, relations de présentation), alors l'un d'entre eux peut être réglé en tant que le lien de ressources par défaut. Les relations récemment créées seront sauvegardées dans le lien de ressource par défaut récemment créé de ce type de relation particulière. Le lien de ressource par défaut de chaque genre de relation est affiché en gras.
- *Définir genre de lien de ressources :* spécifie le genre de relation du lien de ressource sélectionné. Sélectionner le genre de relation requis à partir du sous-menu qui se déroule. Le genre de relation **Tout** spécifie que le lien de ressource sélectionné sera utilisé pour tous les genres de relation.
- *Ajouter nouveau lien de ressources :* crée un nouveau fichier de lien de ressource pour contenir les relations d'un des cinq genres (définition, présentation, calcul, libellé, ressource). le fichier ajouté peut être renommé en cliquant avec la touche de droite et en sélectionnant la commande **Renommer** dans le menu contextuel. Un nouvel élément `linkbaseRef` est ajouté au fichier de concept ; cet élément référence le lien de ressource récemment ajouté.
- *Importer/Référencer:* Importe une taxonomie standard ou crée une référence à un Schéma XML existant ou un lien de ressource. Si vous choisissez l'option de taxonomie standard, une fenêtre contenant une liste de taxonomies standard s'affiche. Choisir la taxonomie que vous souhaitez importer ; voir Importer une taxonomie pour plus de détails. Si vous créez une référence à un schéma XML ou à un lien de ressource, un nouvel élément `linkbaseRef` contenant la référence est ajouté au fichier de concepts. Cliquer soit sur le Schéma XML Schema soit sur l'option de lien de ressource pour ouvrir un dialogue dans lequel vous pouvez chercher le fichier requis.
- *Définir espace de noms cible :* définit l'espace de noms cible et déclare que cet espace de noms cible est dans l'étendue de l'élément `xs:schema` (donc, pour toute la taxonomie). Voir la description de la commande **XBRL | Définir espace de noms cible**.
- *Renommer :* permet de renommer le fichier sélectionné.
- *Supprimer :* supprime le fichier sélectionné depuis l'Aperçu et son élément `linkbaseRef` du fichier de concepts.

Assistant à la saisie Éléments globaux

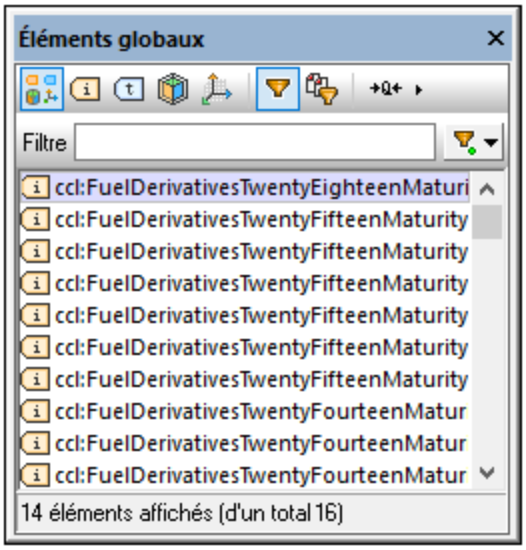
L'assistant à la saisie Éléments globaux (*capture d'écran ci-dessous*) affiche tous les items, tuples, hypercubes et dimensions présents dans un document de taxonomie. Les éléments peuvent être glissés depuis l'assistant à la saisie Éléments globaux dans la fenêtre principale.



Les fonctions suivantes sont disponibles dans l'assistant à la saisie Éléments globaux :

- *Filtrer par type :* l'affichage de chaque type d'élément (item, tuple, hypercube et dimension) peut être activé ou désactivé en cliquant sur ses icônes dans la barre d'outils de l'assistant à la saisie Éléments globaux. Cliquer sur l'icône *Afficher tous les éléments* pour afficher tous les éléments. Le filtre sélectionné est marqué (Afficher tout, *dans la capture d'écran ci-dessus*).
- *Filtrer par texte :* cliquer sur l'icône Clicking the *Filtre texte* pour afficher la barre du Filtre de texte (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH

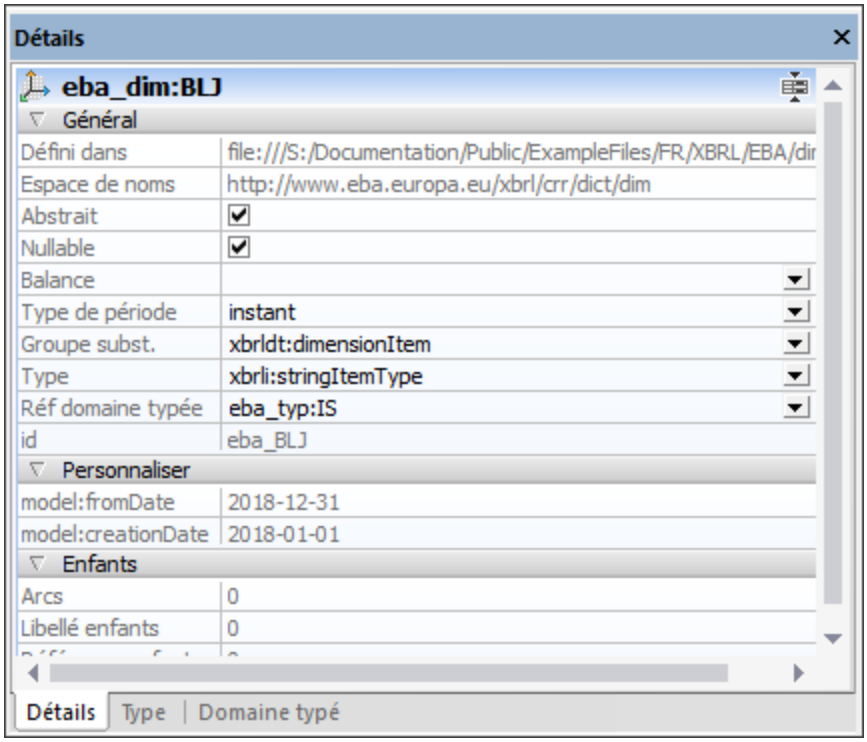


Une fois que le Filtre de texte a été sélectionné, dans la liste de choix située à droite de la barre du Filtre de texte (*sous le curseur fléché dans la capture d'écran ci-dessus*), choisir une condition avec laquelle filtrer. Les options disponibles sont *Contient*, *Ne contient pas*, et *Commence par*. Dans la capture d'écran ci-dessus, la condition *Commence par* a été sélectionnée. Ensuite, saisir le texte pour le filtre dans le champ Filtre de texte. Les éléments affichés seront filtrés. Dans la capture d'écran ci-dessus, la liste est filtrée par les noms commençant avec `n1`. Veuillez noter que : (i) les filtres travaillent sur les noms des entrées en tant que strings de texte, et (ii) les noms des entrées peuvent être modifiées avec l'icône Format (voir ci-dessous) et sont sensibles aux changements dans les formats de nom.

- *Filtrer par sources* : les sources requises peuvent être sélectionnées dans une fenêtre popup, et seuls les éléments dans les sources sélectionnées seront affichées.
- *Format:* il y a trois options de format en ce qui concerne l'affichage des noms des éléments : *Noms qualifiés courts*, *Noms qualifiés longs* et *Libellés*. Le nom qualifié court utilise les préfixes attribués aux espaces de noms cible respectifs ; le nom qualifié long (l'icône est ) utilise tout l'espace de noms cible ; et le libellé utilise les libellés associés avec des éléments. Veuillez noter que la sélection de format influe sur l'option *Filtre de texte*, en ce que le filtre est appliqué aux éléments tels que recensés conformément à l'option *Format* sélectionnée actuellement.
- *Glisser/déposer:* les éléments dans l'assistant à la saisie Éléments globaux peuvent être glissés et développés dans des relations dans tous les modes de relation de la Fenêtre principale (Définitions, Présentations, Calculs)

Assistant à la saisie Détails

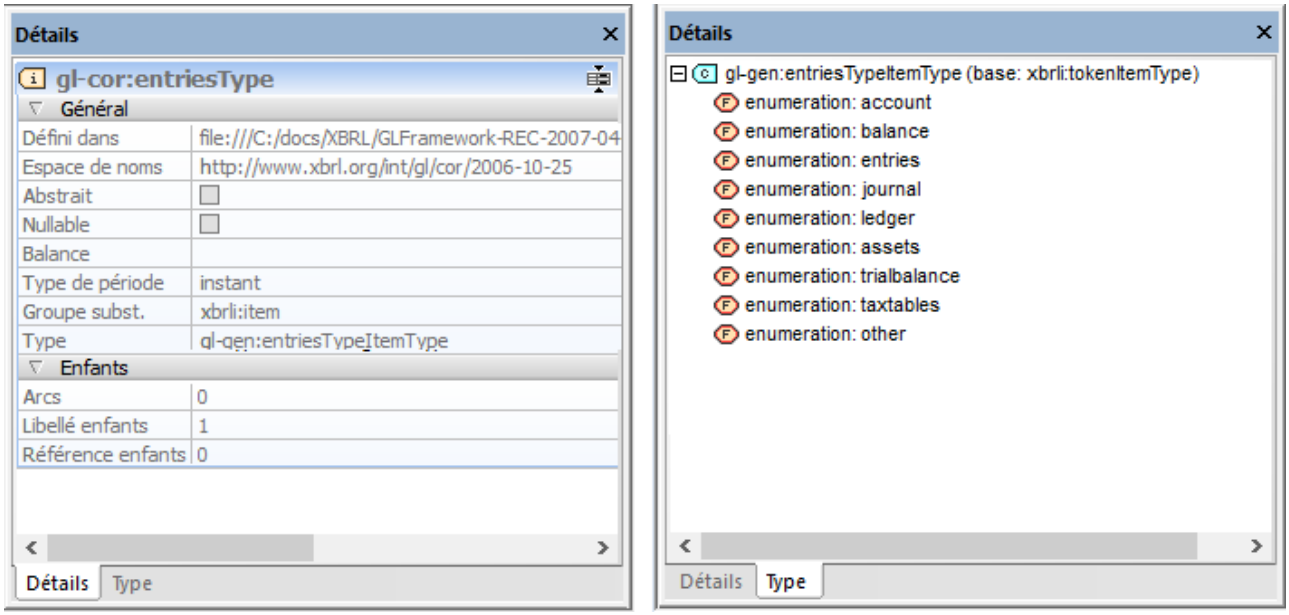
Lorsqu'un élément est sélectionné dans la Fenêtre principale, l'assistant à la saisie Détails (*capture d'écran ci-dessous*) affiche ses propriétés. Si l'élément a des attributs personnalisés, une section nommée *Personnalisation* est affichée dans l'assistant à la saisie et contient les attributs.



Les propriétés de certains éléments peuvent être éditées dans l'assistant à la saisie Détails : par exemple, *Abstract*, *Nillable*, *Balance*, *Period Type*, *Substitution Group* et *Type*.

L'assistant à la saisie Détails contient aussi un onglet Type qui montre le type de l'item sélectionné en tant que le nœud de racine d'un mode d'arborescence (*voir capture d'écran ci-dessous*). Pour des concepts de type `enum:enumerationItemType`, les extensions extensibles comportant des libellés en plusieurs langues, tels que définis dans les Extensible Enumerations Recommendation du 29 octobre 2014 peuvent être spécifiées. Pour des items de ce type, des entrées supplémentaires pour les attributs du type `enum:domain`, `enum:linkrole` et `enum headUsable` sont disponibles.

© 2016-2022 Altova GmbH



Si le type n'est ni un xbrli-item-type ni un type XSD intégré, l'assistant à la saisie montre des informations concernant son type de base entre des crochets. Le mode en arborescence fournit un menu contextuel pour montrer (ou modifier) la définition de type du concept dans le Mode Schéma.

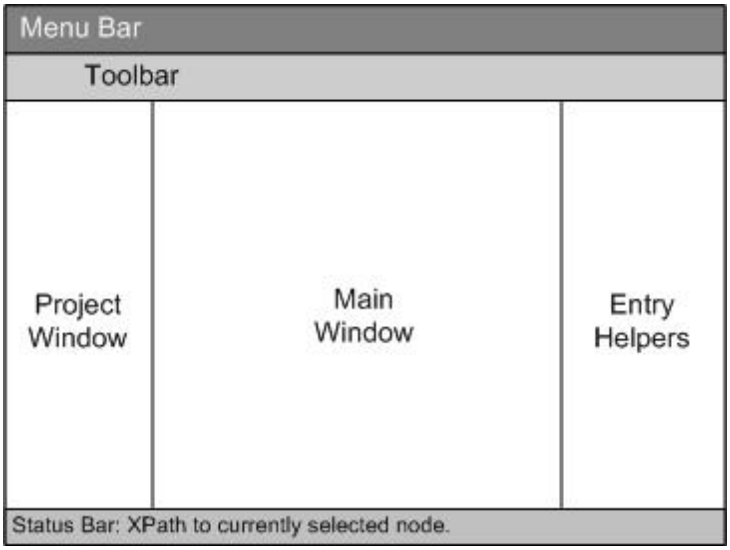
4.6.4 Paramètres du mode XBRL

Il existe deux types de paramètres que vous pouvez configurer dans le Mode XBRL :

- Les polices et les couleurs des composants dans le mode XBRL. Ces paramètres sont accessibles depuis la section Police et couleurs du dialogue d'Options (**Outils | Options**).
- La mise en page et le format du mode, ainsi que les libellés par défaut. Ces paramètres sont disponibles dans le dialogue Paramètres de Mode XBRL, qui est accessible depuis la commande **XBRL | Paramètres de mode XBRL**.

4.7 Mode Authentic

Authentic View présente une barre de menu et une barre d'outils placées au sommet de la fenêtre et celle-ci se réparti en trois volets qui recouvrent le reste de l'interface : la Fenêtre de Projet, la Fenêtre principale, et la Fenêtre d'assistants à la saisie. Vous trouverez ces trois volets ci-dessous.



L'interface du Mode Authentic est décrite en détail dans la section, **Authentic | Interface du Mode Authentic**.

© 2016-2022 Altova GmbH

4.8 Mode Navigateur

Le Mode Navigateur est généralement utilisé pour consulter :

- des fichiers XML qui possèdent un fichier associé XSLT. Lorsque vous passez au Mode Navigateur, le fichier XML est transformé immédiatement à l'aide de la feuille de style XSLT et le résultat est affiché directement dans le Mode Navigateur.
- Les fichiers HTML qui sont créés directement en tant que HTML ou créés par le biais d'une transformation XSLT d'un fichier XML.

Pour consulter les fichiers XML et HTML dans le Mode Navigateur, cliquez sur l'onglet **Navigateur**.

Moteurs de navigation dans le Mode Navigateur

Par défaut, le Mode Navigateur utilise actuellement Microsoft's Internet Explorer comme son moteur de navigation. Si vous souhaitez utiliser le moteur de navigation Microsoft's newer Edge WebView2 pour le Mode Navigateur, vous pouvez sélectionner cette option dans la section Affichage du dialogue des Options.

Note : Puisque Microsoft Edge WebView2 utilise le projet de logiciel Chromium, sur lequel le navigateur Chrome de Google est basé, utiliser WebView2 pour le Mode navigateur fournit également un bon aperçu de l'affichage Chrome de la page web.

Notes relatives à Microsoft Internet Explorer

Le Mode Navigateur requiert Microsoft's Internet Explorer 5.0 ou plus, ou Microsoft Edge WebView2 (*voir ci-dessus*).

Notez les points suivants d'Internet Explorer dans le Mode Navigateur :

- Si vous souhaitez utiliser le Mode Navigateur pour consulter des fichiers XML transformés par une feuille de style XSLT, nous vous recommandons vivement d'utiliser Internet Explorer 6.0 ou plus, qui utilise MSXML 3.0, un parseur XML prenant en charge le standard XSLT 1.0. Vous pouvez également installer MSXML 4.0.
- La prise en charge pour XSLT en IE 5 n'est pas compatible à 100 % avec la recommandation officielle XSLT. Donc si vous rencontrez un problème dans le Mode Navigateur avec IE 5, essayez de passer à IE 6 ou plus.
- En général, Vous devriez contrôler la prise en charge de XSLT de votre version d'Internet Explorer.
- Si vous rencontrez des problèmes avec l'affichage correct de HTML dans Internet Explorer, incluez l'onglet **meta** suivant dans l'élément **head** de votre document HTML :

```
<head>
... <meta http-equiv="X-UA-Compatible" content="ie=edge">...
</head>
```

Outils de développeurs dans le Mode Navigateur

Vous pouvez utiliser les Outils de développeur du navigateur Edge por inspecter, dégoguer et tester votre code HTML. Pour ouvrir les outils, cliquez avec la touche de droite dans le volet du Mode Navigateur et sélectionnez **Open Developer Tools**.

Fonctions Mode Navigateur

Les fonctions suivantes sont disponibles dans le Mode Navigateur. Elles peuvent être accédées via le menu **Navigateur**, le menu **Fichier** et le menu **Éditer**.

- *Ouvrir dans une fenêtre séparée :* Lorsque le Mode Navigateur est une fenêtre séparée, elle peut être positionnée côte à côte avec un mode d'édition du même document. À cette fin, cliquez sur la commande du menu **Navigateur | Fenêtre séparée.** Il s'agit d'une commande à bascule qui bascule le Mode Navigateur entre deux fenêtres : (i) une fenêtre séparée, et (ii) un affichage tabulaire dans la fenêtre principale. Ces commandes sont aussi disponibles dans le menu déroulant du bouton **Mode navigateur** (à la fin de la fenêtre principale). Dans l'onglet Affichage des Options, vous pouvez définir si vous souhaitez que le Mode Navigateur apparaisse par défaut dans une fenêtre séparée.
- *Avant et arrière :* Le navigateur commun envoie la commande de naviguer à travers les pages qui ont été chargées dans le Mode Navigateur. Ces commandes sont dans le menu **Navigateur**.
- *Taille de police :* Peut être ajustée par le biais du menu **Navigateur**.
- *Arrêter, Actualiser, Imprimer :* Plus de commandes de navigateur standard, celles-ci peuvent être trouvées dans les menus **Navigateur** et **Fichier**.
- *Recherche :* Permet des recherches pour des strings de texte. Cette commande se trouve dans le menu **Éditer**.
- *Fenêtre Info :* Il existe des options ici pour afficher la page HTML active avec tout navigateur web installé sur l'appareil et pour ouvrir ou déplacer les navigateurs installés.

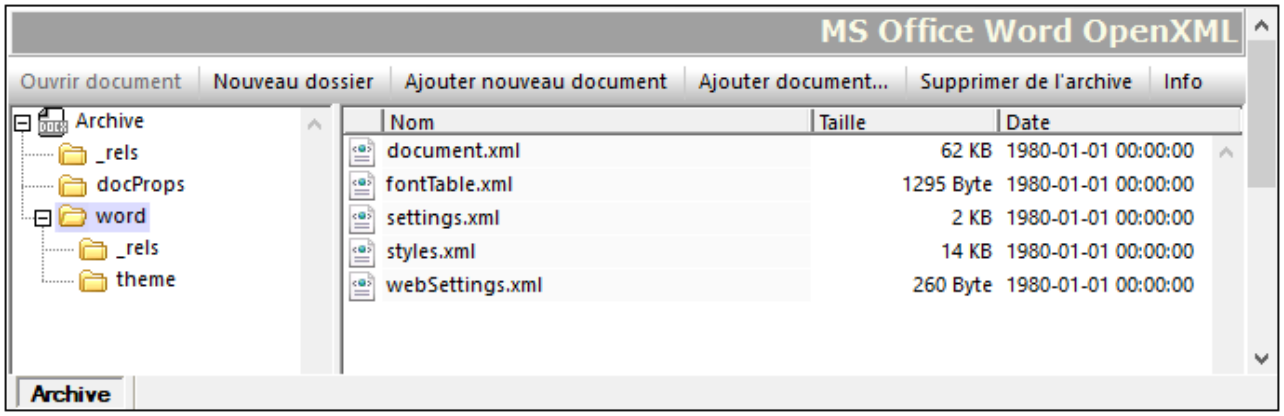
© 2016-2022 Altova GmbH

4.9 Mode Archive

Un fichier Office Open XML (OOXML), un fichier ZIP (par exemple, WinZip ou WinRAR), ou un fichier EPUB peut être ouvert et édité dans Archive View. Non seulement, les archives OOXML, ZIP et EPUB peuvent être modifiées structurellement dans Archive View, mais des fichiers individuels peuvent être ouverts depuis Archive View, édités dans un des modes d'édition de XMLSpy, puis enregistrés directement dans les archives.

Fichiers d'Archive et Mode Archive

Lorsqu'un fichier d'archive (fichier OOXML, ZIP ou EPUB) est créé ou ouvert dans XMLSpy, il est ouvert dans Archive View (*capture d'écran ci-dessous*). Il est possible d'ouvrir plusieurs fichiers d'archive simultanément, chaque fichier d'archive se trouvant dans une fenêtre séparée Archive View. Le type du fichier d'archive apparaît dans le coin situé à droite de Archive View. Dans la capture d'écran ci-dessous, le type du fichier d'archive est MS Office Word Open XML.



Mode Dossier

Le Mode Dossier est situé à gauche de la fenêtre Archive View et affiche la structure de dossiers des archives zippées. À chaque niveau, les dossiers sont classés par ordre alphabétique. Pour consulter les sous-dossiers d'un dossier, cliquer sur le signe Plus se trouvant à gauche du dossier. Si un dossier ne comporte pas de signe plus, c'est qu'il ne contient pas de sous-dossiers. Pour consulter les fichiers de document (que nous appellerons documents) contenus dans un dossier, sélectionner le dossier ; les fichiers seront affichés dans la Fenêtre principale. Dans la capture d'écran ci-dessus, les documents affichés dans la Fenêtre principale se trouvent dans le dossier `word`, qui contient également deux sous-dossiers : `_rels` et `theme`.

Fenêtre principale

La Fenêtre principale recense les documents dans le dossier qui a été sélectionné dans le Mode Dossier. Les documents sont affichés dans l'ordre alphabétique, chacun d'entre eux avec sa taille nom compressée respective et la date et l'heure de la modification. Pour ouvrir un document provenant de Archive View, double-cliquer le document. Il s'ouvre dans une fenêtre XMLSpy séparée.

Boutons de commande

Les boutons de commande sont situés dans la partie supérieure de la fenêtre Archive View.

- ***Ouvrir le document :*** activé lorsqu'un document est sélectionné dans la Fenêtre principale. Cliquer dessus pour ouvrir le document sélectionné. Il est aussi possible d'ouvrir un document en double-cliquant dans la liste des documents dans la Fenêtre principale.
- ***Nouveau dossier :*** permet d'ajouter un nouveau dossier dans le dossier actuellement sélectionné dans le Mode Dossier. Le dossier doit être nommé dès sa création dans le le Mode Dossier. Il n'est pas possible de renommer un dossier par la suite. Le nouveau dossier est enregistré dans l'archive lorsque le fichier d'archive est enregistré.
- ***Ajouter nouveau document :*** permet d'ajouter un nouveau document dans le dossier actuellement sélectionné dans le Mode Dossier. Cliquer sur cette touche pour ouvrir le dialogue Créer nouveau document de XMLSpy. Le document récemment créé s'ouvre dans une fenêtre XMLSpy séparée. Le document doit être nommé dès son apparition dans la liste des document du dossier sélectionné. Le document ne sera enregistré dans les archives que lorsqu'il sera enregistré dans sa propre fenêtre d'édition ou lorsque le fichier d'archive est enregistré.
- ***Ajouter document :*** ouvre un dialogue Chercher avec lequel vous pouvez chercher un document à rajouter. Ce document sera ajouté à la liste dans la Fenêtre principale des documents se trouvant actuellement dans le dossier sélectionné, et le document est ouvert dans une fenêtre XMLSpy séparée. Pour enregistrer le document dans les archives, il doit être soit enregistré dans sa propre fenêtre, soit le fichier d'archive doit être enregistré.
- ***Supprimer des archives :*** supprime le document sélectionné (dans la Fenêtre principale) ou le dossier sélectionné (dans le Mode Dossier) depuis les archives. Le fichier d'archive doit être enregistré afin que la suppression prenne effet.
- ***Info :*** active et désactive la Fenêtre Info. *Voir ci-dessous*.

Fenêtre Info

La Fenêtre Info est activée et désactivée en cliquant sur la commande Info. La Fenêtre Info apporte des informations générales concernant le fichier d'archive, comme le nombre de fichiers qu'il contient, ses tailles comprimées et non comprimées et le rapport de compression.

© 2016-2022 Altova GmbH

4.10 Raccourcis communs

Les raccourcis par défaut sont communément utilisés pour éditer les commandes dans la liste ci-dessous. Vous pouvez changer les raccourcis par défaut dans l'onglet du clavier du dialogue Personnaliser.

Function-key shortcuts (y compris pour la validation et la transformation)

F1	Menu d'aide
F1 + Alt	Ouvrir le dernier fichier
F3	Trouver suivant
F4 + CTRL	Fermer la fenêtre active
F4 + Alt	Fermer XMLSpy
F5	Réinitialiser
F6 + CTRL	Faire défiler les fenêtres ouvertes
F7	Vérifier la bonne formation
F8	Valider
F10	Transformation XSL
F10 + CTRL	Transformation XSL:FO

Commandes de fichier et d'applications

Alt + F1	Ouvrir le dernier fichier
Ctrl + O	Ouvrir le fichier
Ctrl + N	Nouveau fichier
Ctrl + P	Imprimer fichier
Ctrl + S	Enregistrer fichier
CTRL + F4	Fermer la fenêtre active
CTRL + F6	Faire défiler les fenêtres ouvertes
CTRL + TAB	Sauter entre des documents ouverts
Alt + F4	Fermer XMLSpy

Touches diverses

Touches fléchées haut/bas	Déplacer le curseur ou la barre de sélection
Échap	Abandonner les éditions ou fermer le dialogue
Retour	Confirmer la sélection
Suppr	Supprimer le caractère ou la sélection
Shift + Del	Couper

Commandes d'édition

CTRL + A	Sélectionner tout

Ctrl + F	Recherche
Ctrl + G	Aller à la ligne/caractère
Ctrl + H	Remplacer
Ctrl + V	Coller
Ctrl + X	Couper
CTRL + Y	Rétablir
Ctrl + Z	Annuler

© 2016-2022 Altova GmbH

5 XML

Cette section, décrit comment travailler avec des documents XML dans XMLSpy. Elle couvre les aspects suivants :

- Comment créer, ouvrir et enregistrer des documents XML. Dans cette section, certains paramètres XMLSpy importants concernant la création de fichier sont également expliqués.
- Des documents XML peuvent être édités dans les Mode Texte, Mode Grille et Mode Authentic. Vous pouvez sélectionner le mode qui vous est le plus utile et alterner entre les modes pendant l'édition. Chacun de ces modes présente des avantages différents.
- Vous pouvez facilement et rapidement ajouter des fragments XML à vos documents XML depuis des sources externes.
- Comment utiliser les fonctions de validation XML variées de XMLSpy.
- Les assistants de saisie pour les documents XML ont certaines fonctions spécifiques, et celles-ci sont décrites.
- Comment traiter des documents XML avec XSLT et XQuery. Les fonctions XMLSpy variées liées au traitement sont expliquées. Une section concernant les Polices PDF explique comment les polices sont traitées lors de la génération de la sortie PDF.
- D'autres fonctions diverses pour travailler avec des documents XML sont expliquées.

Site web Altova : Éditeur XML

5.1 Créer, ouvrir et enregistrer des documents XML

Lors de la création, de l'ouverture ou de l'enregistrement de documents XML, les questions suivantes peuvent survenir :

- Dans quel mode le document XML s'ouvrira-t-il : Mode Texte, Mode Grille ou Mode Authentic
- Lorsqu'un nouveau document XML est créé, le schéma (Schéma XML ou DTD) sera-t-il attribué automatiquement, manuellement ou pas attribué du tout
- Si un schéma est attribué au document XML, le document sera-t-il validé automatiquement lors de l'ouverture et/ou de l'enregistrement

Mode par défaut

Des paramètres à l'échelle de l'application sont disponibles pour spécifier dans quel mode les documents XML documents (nouveaux et existants) devraient s'ouvrir. Ces paramètres se trouvent dans le dialogue Options (**Outils | Options**).

Dans la section ***Types de fichier*** du dialogue Options, sélectionner un type de fichier de `.xml` et, dans le panneau du Mode Défaut, contrôler le mode d'édition requis (Texte ou Grille). Veuillez noter que : (i) Mode Schéma et Mode WSDL peut être utilisé uniquement pour un Schéma XML et des documents WSDL, respectivement ; et (ii) le Mode Navigateur est un mode d'affichage et pas un mode d'édition.

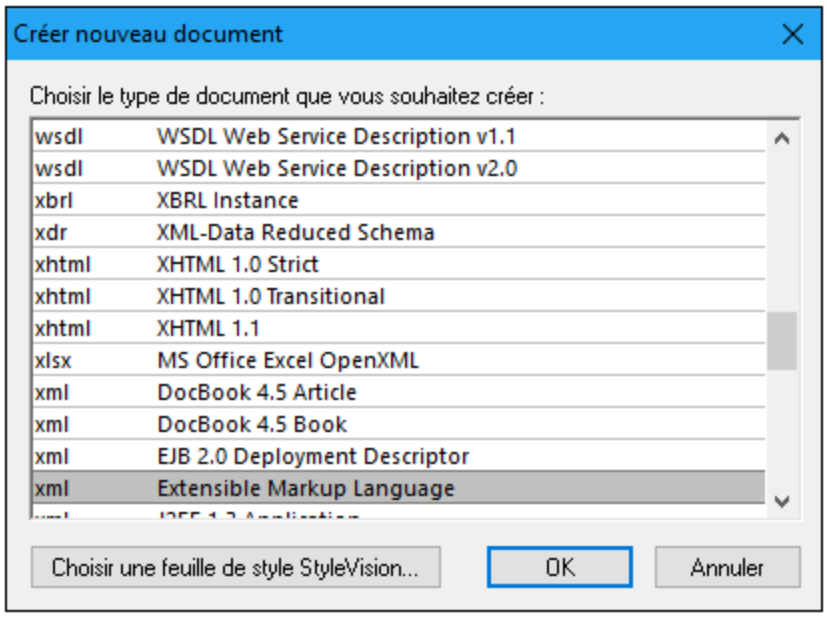
Dans l'onglet ***Types de fichier***, vous pouvez aussi régler XMLSpy en tant que l'éditeur par défaut pour le type de fichier sélectionné.

Un document XML peut être édité dans le Mode Authentic si un StyleVision Power Stylesheet (SPS) lui a été attribué. Lorsqu'un fichier XML avec un SPS associé est ouvert, vous pouvez spécifier qu'il s'ouvre directement dans le Mode Authentic. Pour ce faire, contrôler la section *Toujours ouvrir dans le Mode Authentic* dans l'onglet ***Affichage*** du dialogue Options. Si cette option n'est pas cochée, le fichier s'ouvrira dans le mode par défaut spécifié pour les fichiers `.xml` dans l'onglet ***Types de fichier*** (*voir ci-dessus*).

Attribuer des schémas

Lorsqu'un nouveau fichier XML doit être créé, sélectionner la commande de menu **Fichier | Nouveau**. Le dialogue Créer nouveau document s'ouvre (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Veuillez noter que vous disposez de plusieurs options en ce qui concerne le type de document XML. L'option marquée *Extensible Markup Language* crée un document générique XML. Chacune des autres options est associée à un schéma, par exemple le DocBook DTD. Si vous sélectionnez une de ces options, un document XML est créé dont le (i) schéma correspondant lui est automatiquement attribué, et (ii) une structure de document squelette qui est valide conformément au schéma attribué. Veuillez noter que vous pouvez créer votre propre document squelette XML. Si vous l'enregistrez dans le dossier `Template` du dossier d'application, votre document squelette sera disponible pour une sélection dans le dialogue Créer nouveau document.

Si vous choisissez le type de document Extensible Markup Language générique, vous serez invité à choisir un schéma (DTD ou Schéma XML) à attribuer au document. À ce moment, vous pouvez choisir entre chercher un schéma ou continuer et créer un document XML sans aucun schéma attribué.

Bien entendu, par la suite, vous pourrez attribuer un schéma par le biais du menu **DTD/Schéma** à tout moment pendant l'édition.

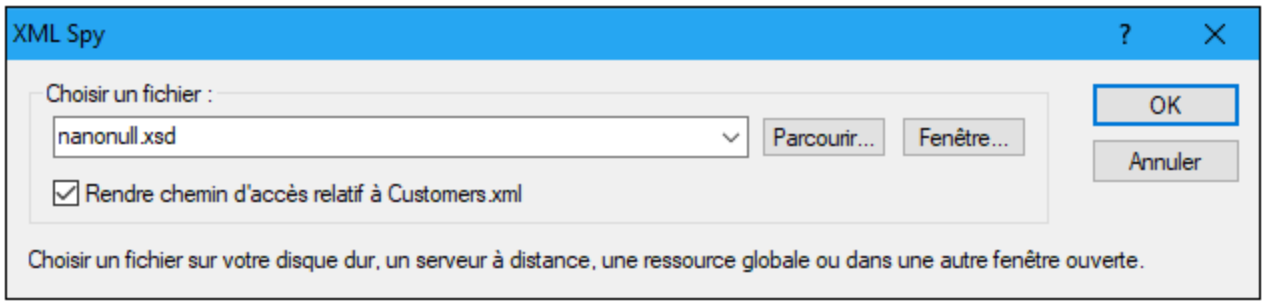
Validation automatique

Si un document XML existant a un schéma qui lui est attribué, il peut être validé automatiquement lors de l'ouverture et/ou de l'enregistrement. Les paramètres se trouvent dans l'onglet ***Fichier*** du dialogue Options (**Outils | Options**).

Les paramètres de validation automatiques de la section ***Fichier*** peuvent être combinés avec un paramètre dans l'onglet Types de fichier pour désactiver la validation automatique pour des types de fichiers spécifiques. L'utilisation des paramètres dans les deux onglets vous permet de spécifier une validation automatique pour des types de fichiers spécifiques.

5.2 Attribuer des schémas et valider

Site web Altova : Valideur XML, Validation XML
Un schéma (DTD ou Schéma XML) peut être attribué à un document XML lors de sa première création. Un schéma peut aussi être attribué, ou modifié ultérieurement grâce aux commandes **Attribuer DTD** ou **Attribuer Schéma** dans le menu **DTD/Schéma**.



Le schéma vers le fichier de schéma qui est inséré dans le document XML peut être rendu relatif en contrôlant la case à cocher du chemin relatif dans le dialogue.

Ressources globales pour les schémas

Une ressource globale est un alias pour un fichier ou un dossier. Le fichier cible ou le dossier peut être modifié dans le cadre de la GUI en changeant la configuration active de la ressource globale (par le biais de la commande de menu **Outils | Configuration active**). Les ressources globales permettent donc au schéma attribué de passer d'un schéma à l'autre, ce qui peut être utile pour les tests. La description des ressources globales est décrite dans la section Ressources globales Altova.

Schéma XML plus DTD

Une fonction très utile de la DTD que Schéma XML ne possède pas est l'utilisation des entités. Néanmoins, si vous souhaitez utiliser des entités dans votre document XML validé par Schéma XML, vous pouvez ajouter une déclaration `DOCTYPE` pour le document XML et y inclure vos déclarations d'entité.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrgChart [
  <!ENTITY name-int "value">
  <!ENTITY name-ext SYSTEM "extfile.xml">
]>
<OrgChart xmlns="http://www.xs.com/org"
          xsi:schemaLocation="http://www.xs.com/org OrgChart.xsd">
  ...
</OrgChart>
```

Une fois avoir déclaré les entités dans la DTD, elles peuvent être utilisées dans le document XML. Le document sera bien formé et valide. Veuillez noter, néanmoins, que les entités parsées externes ne sont pas prises en charge dans le Mode Authentic.

© 2016-2022 Altova GmbH

Se rendre aux définitions de schéma

Une fois le document XML ouvert, vous pouvez ouvrir directement le DTD ou le Schéma XML sur lequel il est basé en cliquant sur la commande Aller au DTD ou Aller au Schéma dans le menu **DTD/Schéma**. De plus, vous pouvez placer le curseur dans un nœud dans le document XML et vous rendre dans la définition de schéma de ce nœud par le biais de la commande **Aller à la définition** dans le menu **DTD/Schéma**.

Valider et contrôler la bonne formation

Pour valider et/ou contrôler la bonne formation, utiliser les commandes **Valider XML (F8)** et **Vérifier la bonne formation (F7)** dans le menu XML ou les commandes correspondantes dans la barre d'outils. Toute erreur sera rapportée dans la fenêtre des Messages. Si un document XML est invalide, le validateur XML propose des smart fixes pour corriger l'erreur sur la base de l'information contenue dans le schéma.

Vous pouvez aussi utiliser un RaptorXML Server pour valider les documents XML.

5.3 XML dans le Mode Texte

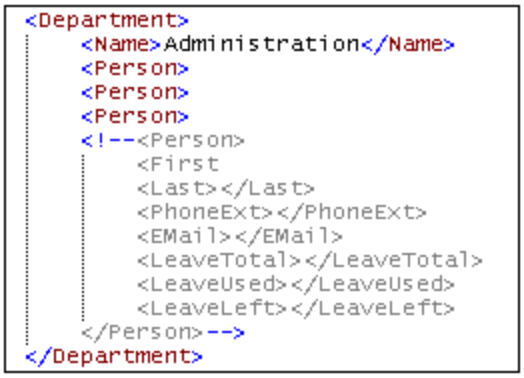
XMLSpy propose des fonctions d'édition de texte XML (décrites ci-dessous) spécialisées outre les fonctions d'édition généralement disponibles dans le Mode Texte (qui sont décrites dans le Mode Texte de la section modes d'édition.

- Activer et désactiver les textes des commentaires
- Échapper et déséchapper les caractères XML
- Insérer des chemins de fichier
- Insérer des fragements XML via XInclude
- Copier des expressions XPath et XPointer dans le presse-papiers
- Enregistrer comme string d'image encodée en Base64 en tant que d'image

Commenter et décommenter du texte

Le texte dans un document XML peut être désactivé en utilisant les délimiteurs de début de commentaire et de fin de commentaire XML, respectivement <!-- et -->. Dans XMLSpy, ces délimiteurs de commentaires peuvent être insérés aisément en utilisant la commande de menu **Édition | Commenter/décommenter.**

Pour désactiver un bloc de texte, sélectionner le texte à désactiver, puis sélectionner la commande **Commenter/décommenter**, soit depuis le menu **Édition** ou le menu contextuel que vous pouvez appeler en cliquant avec la touche de droite sur le texte sélectionné. Le texte commenté sera grisé (*voir capture d'écran ci-dessous*).



Pour retirer les commentaires d'un texte commenté, sélectionner le bloc de texte commenté **à part** les délimiteurs de commentaires et sélectionner la commande **Commenter/décommanter**, soit depuis le menu **Édition** soit depuis le menu contextuel. Pour y accéder, cliquer avec la touche de droite sur le texte sélectionné. Les délimiteurs de commentaire seront supprimés et le texte ne sera plus grisé.

Note concernant les lignes vides

Dans les documents XML, les lignes vides sont éliminées lorsque vous changez de mode ou enregistrez le document. Si vous souhaitez garder des lignes vides, vous pouvez les placer entre des délimiteurs de commentaire.

© 2016-2022 Altova GmbH

Échapper et déséchapper les caractères XML

Les cinq caractères XML spéciaux (*recensés ci-dessous*) peuvent être échappés et déséchappés avec les références d'entité correspondantes (*recensées ci-dessous*) en marquant un bloc de texte et en sélectionnant la commande de menu contextuel **Échapper les caractères XML** ou **Déséchapper les caractères XML** . Les caractères XML spéciaux dans ce bloc de texte seront ensuite échappés ou déséchappés conformément à la commande sélectionnée.

<	`<`
>	`>`
&	`&`
'	`'`
"	`"`

Par exemple :

`<a></a>` peut être échappé avec la commande **Échapper les caractères XML** `<a></a>` et `<a></a>` peut être déséchappé avec la commande **Déséchapper les caractères XML** vers `<a></a>`

Insérer des chemins de fichier

La commande **Édition | Insérer chemin de fichier** vous permet de chercher le fichier en question et d'insérer son chemin de fichier dans l'emplacement sélectionné dans le document XML en cours d'édition. Cette commande vous permet de saisir un chemin de fichier rapidement et précisément. Voir la description de commande pour plus de détails.

Insérer des fragments XML via XInclude

La commande **Édition | Insérer XInclude** vous permet, via XInclude, d'insérer les contenus de tout un document XML ou un fragment d'un, dans le document XML devant être édité. Cette commande vous permet de saisir rapidement et précisément des documents XML entiers (par le biais du mécanisme XInclude) ou des fragments de documents XML (par le biais d'une extension XPointer du mécanisme XInclude). Voir la description de commande pour plus de détails.

Copier les expressions XPath et XPointer dans le presse-papiers

Les expressions XPath et XPointer du nœud sélectionné (exprimant la position du nœud dans le document XML) peuvent être copiées dans le presse-papiers en utilisant les commandes **Édition | Copier XPath** et **Édition | Copier XPointer**, respectivement. Cela vous permet d'obtenir les expressions XPath et XPointer correctes ciblant le nœud sélectionné.

Par exemple, définissez le nœud sélectionné dans le Mode Texte ou Grille être le troisième élément `Office` d'un élément de document appelé `Offices`. Dans ce cas, l'expression XPath copiée sera `/Offices/Ofice[3]`. Et l'expression copiée XPointer, si les éléments `Office` n'ont aucun frère nommé autrement qui se produit avant le troisième élément `Office`, sera `element(/1/3)`.

Les expressions copiées peuvent ensuite être insérées à tout endroit requis. Par exemple, une expression XPath peut être insérée dans une feuille de style XSLT et une expression XPointer dans l'attribut `href` d'un élément `xinclude`.

Pour des descriptions plus détaillées des commandes, voir leur description dans la section Référence de l'utilisateur.

Enregistrer un string encodé en Base64 en tant qu'image

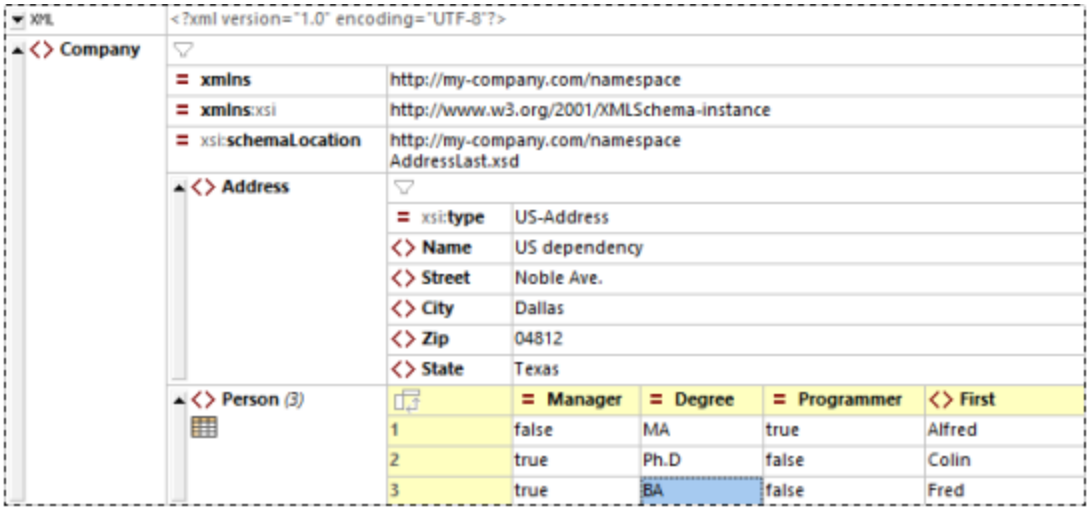
Pour enregistrer un string encodé en Base64 dans son format d'image, cliquez avec la touche de droite sur le texte encodé et sélectionnez la commande **Enregistrer sous Image**. Dans le dialogue qui apparaît, sélectionnez l'emplacement sous lequel vous souhaitez enregistrer l'image et saisissez un nom pour le fichier d'image. L'extension du fichier d'image (`.png`, `.gif`, `.svg`, etc.) sera détectée automatiquement depuis l'encodage Base64 et apparaîtra dans le dialogue Enregistrer. Une fois terminé, cliquer **Enregistrer**.

Cette action peut également être exécutée par le biais de la commande de menu **Editer | Enregistrer sous Image**.

© 2016-2022 Altova GmbH

5.4 XML dans le Mode Grid

Le Mode Grid affiche la structure hiérarchique des **documents XML** par un ensemble de conteneurs imbriqués qui peuvent être étendus et compressés. Cela vous donne une image fidèle de la structure du document. Dans le Mode Grille, la structure du document peut être modifiée facilement et le contenu peut être édité en toute facilité.



Dans la deuxième capture d'écran ci-dessus, notez que le document est affiché comme hiérarchie dans un formulaire de grille. Lorsqu'un nœud peut héberger du contenu, il contient soit directement du contenu (comme dans le cas des nœuds du *Texte*) ou alors il est divisé en deux champs : le nom du nœud et le contenu du nœud (comme dans le cas des nœuds de l' *Élément*). Les noms de nœud sont affichés en gras et le contenu du nœud en police d'écriture normale.

Barre d'outils Affichage Grille

Les commandes liées à l'édition dans le Mode Grille peuvent être accédées rapidement par le biais de la barre d'outils Affichage Grille *(capture d'écran ci-dessous)*—qui est située dans l'espace de la barre d'outils tout en haut de la fenêtre d'application. Pointez sur l'icône de la barre d'outils pour voir ses nom et raccourci.



Note : Ces commandes sont souvent disponibles dans le menu XML.

Aperçu des fonctions du Mode Grille

Pour un aperçu des fonctions du Mode Grille, voir les sections suivantes :

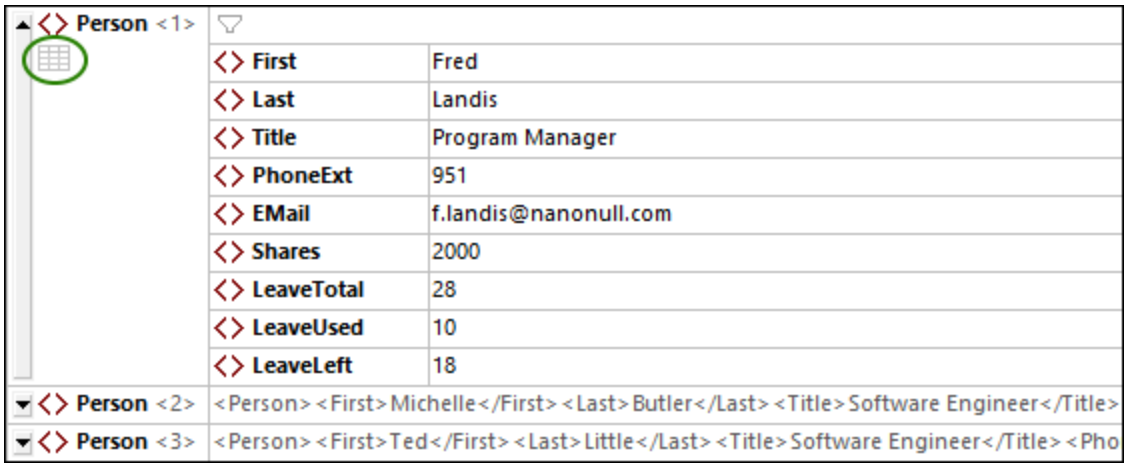
- Affichage du document, qui décrit les fonctions de l'affichage du Mode Grid
- Structure du document, qui décrit comment la structure du document est affichée dans le Mode Grille et explique les fonctions du Mode Grille que vous pouvez utiliser pour modifier la structure du document

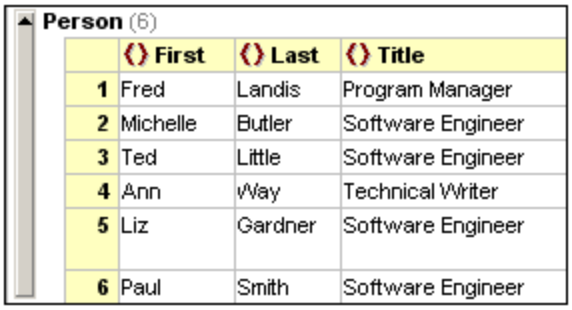
- Contenu du document, qui montre comment éditer et valider le contenu dans le Mode Grille
- Paramètres du Mode Grille, qui listent et décrivent les paramètres qui s'appliquent au Mode Grille

Afficher en tant que Table

Les éléments répétitifs sont affichés en Mode Grille *standard*, l'un après l'autre, en progressant de manière verticale vers le bas dans l'ordre du document (*voir la capture d'écran ci-dessous, à gauche*). Toutefois, afficher des éléments répétitifs comme les *lignes d'une table* fournit des fonctions d'édition additionnelles. Dans la capture d'écran ci-dessous, l'élément **Personne** est l'élément répétitif. La capture d'écran à gauche montre le Mode Grille standard ; l'élément de la première **Personne** est montré en mode élargi, alors que les instances subséquentes sont affichées en mode compressé. La capture d'écran à droite montre les éléments répétitifs de la **Personne** comme lignes d'une table.

Pour passer au Mode Table, cliquez sur l'icône en **Mode Table** *(encerclé en vert dans la capture d'écran ci-dessous, à gauche)*. L'icône est affichée en couleur *(voir la capture d'écran ci-dessous, à droite)* si vous êtes passé à l'affichage de table.





Le Mode Table propose des avantages d'édition uniques dans le sens que les lignes et les colonnes entières peuvent être manipulées par rapport aux autres colonnes et lignes dans la table. Cette fonction permet des opérations comme trier les lignes de la table sur la base des valeurs d'une colonne. Dans la capture d'écran ci-dessus à droite, par exemple, les six éléments **Personne** peuvent être triés sur la base de leurs **Derniers** éléments enfants par le biais d'une seule commande dans la barre d'outils du Mode Grid. Une telle opération est plus simple que d'effectuer une transformation XSLT, qui serait la manière usuelle de trier un nodeset XML.

Pour plus d'informations, voir la page Affichage Table.

© 2016-2022 Altova GmbH

Filtres et formules

Le Mode Grille propose deux fonctions uniques et très utiles :

- Les Filtres vous permettent de filtrer les données qui sont affichées dans le Mode Grid. Un filtre peut être configuré sur un ou plusieurs éléments dans le document.
- Les Formules vous permettent de calculer un résultat ou de générer un nodeset qui peut être enregistré dans le document.

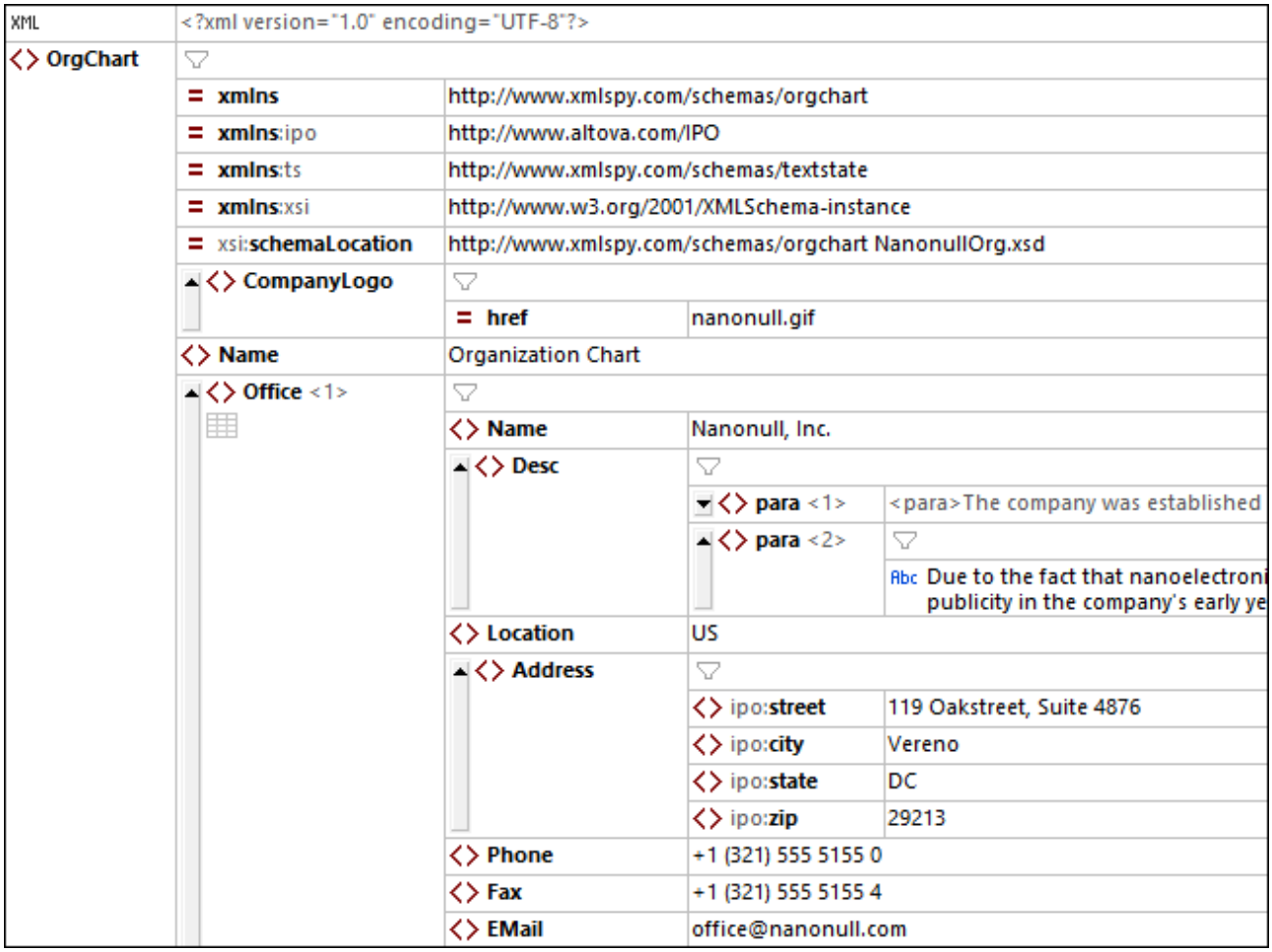
Pour plus d'information, veuillez voir les pages respectives.

Les images et graphiques dans le Mode Grid

Les images et graphiques peuvent être affichés comme graphiques dans le Mode Grid. Voir Affichage de l'image dans le Mode Grid et Graphiques dans le Mode Grille pour des informations sur la procédure à suivre.

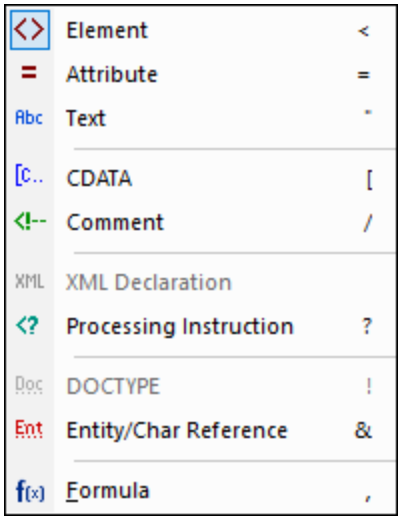
5.4.1 Affichage du document

Dans le Mode Grid, un document XML est affiché de manière hiérarchique à l'intérieur de la grille (*voir la capture d'écran ci-dessous*).



Chaque ligne de la grille affiche un item structurel XML (connu comme nœud), tel un élément, un attribut, un commentaire ou un texte. Les différents types de nœuds qui sont disponibles dans les documents XML sont affichés dans la capture d'écran ci-dessous ensemble avec leurs icônes. Dans le Mode Grid, le type d'un nœud est indiqué par une icône à gauche dans une cellule de grille. Notez que les nœuds du type *Élément, Attribut,* et PI ont une clé et une valeur, alors que les nœuds du type *Texte, CDATA,* et Commenter ont uniquement une valeur. Par exemple, un nœud d'élément aura un nom (sa clé) et un contenu (sa valeur), alors qu'un nœud de texte aura uniquement une valeur.

© 2016-2022 Altova GmbH



Fonctions de l'affichage du Mode Grid

Veuillez noter les fonctions suivantes :

- Vous pouvez zoomer en avant et en arrière de la grille via **Ctrl + Mouse Wheel**, ou **Ctrl+[Plus]** et **Ctrl+[Minus]**.
- Les éléments sibling qui portent le même nom sont numérotés, en commençant par 1. Dans la capture d'écran ci-dessus, par exemple, l'élément `office` qui porte le nombre un est le premier de la séquence d'éléments `office`. Les autres éléments dans la capture d'écran ne sont pas numérotés — parce qu'ils n'ont pas d'éléments sibling du même nom.
- Les nœuds peuvent être étendus ou réduits en cliquant sur l'icône pointe de flèche à gauche du symbole du nœud *(voir la capture d'écran ci-dessus)*. Quand un nœud est réduit, tout contenu y afférent est affiché comme texte dans une seule ligne.
- Si vous sélectionnez de multiples composants au même niveau, vous pouvez tous les élargir/réduire en appuyant sur **Shift** et en cliquant sur n'importe quelle flèche de pointe du composant sélectionné.
- Si word-wrap est activé via la Barre d'outils du Mode Grid, toutes les cellules contenant du texte qui est plus long que la largeur de la cellule seront retournées à la ligne. Vous pouvez désactiver word-wrap en cliquant sur l'icône dans la Barre d'outils du Mode Grid.
- Les Formules sont spécifiques à XMLSpy.

Barre d'outils/boutons de commande dans la cellule

Dans le Mode Grid XML, les boutons dans les cellules de la grille de la barre d'outils du Mode Grid proposent des fonctions qui sont disponibles dans le contexte de la sélection actuelle (cellule ou colonne). L'utilisation de ces commandes est une manière rapide et aisée d'effectuer des opérations utilisées habituellement. Les commandes disponibles sont recensées dans la table ci-dessous. Des commandes liées à l'écran sont affichées avec un fond bleu. Les commandes pour éditer le document (structure et contenu) sont affichées avec un fond beige. Pour les commandes supplémentaires dans un contexte de cellule particulier, utiliser le menu contextuel de la cellule. Note : le nom d'une commande est affiché en tant qu'info-bulle lorsque vous passez sur le bouton de la cellule ou l'icône de la barre d'outils.

	Optimise les largeurs des colonnes de la grille conformément au contenu de la cellule.

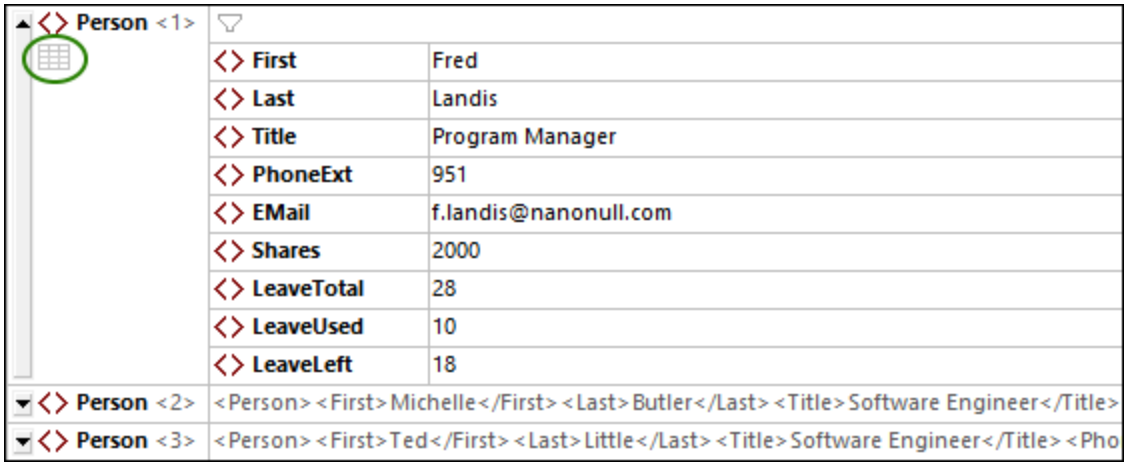
/	Si coloré, Affichage Table est activé, autrement il est désactivé. Cliquer pour passer à l'affichage
/	Une commande bascule en haut à gauche de la cellule de la table. Bascule des lignes vers les colonnes et vice versa
/	Si coloré, un filtre pour la table, un objet, ou un array est activé, autrement le filtre est désactivé. Cliquez pour désactiver/activer. Pour éditer l'expression, double-cliquez sur celle-ci
	Commande bascule pour word-wrapper les cellules. Si sélectionné, word-wrap est activé. Par défaut, seulement les contenus des items sont encapsulés. Si vous voulez en plus encapsuler les noms des items, choisissez cette option depuis la liste déroulante de l'icône.
100%	Niveau de zoom du Mode Grid
	Activé quand une en-tête de colonne est sélectionnée. Les boutons trient les lignes de la table, respectivement par ordre décroissant ou croissant du contenu de la colonne

Note : pour une description d'autres commandes de la barre d'outils, voir la section Menu contextuelo dans le Mode Grille.

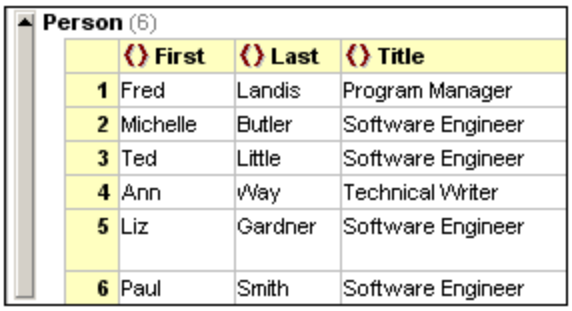
Afficher en tant que Table

Les éléments répétitifs sont affichés en Mode Grille *standard*, l'un après l'autre, en progressant de manière verticale vers le bas dans l'ordre du document (*voir la capture d'écran ci-dessous, à gauche*). Toutefois, afficher des éléments répétitifs comme les *lignes d'une table* fournit des fonctions d'édition additionnelles. Dans la capture d'écran ci-dessous, l'élément **`Personne`** est l'élément répétitif. La capture d'écran à gauche montre le Mode Grille standard ; l'élément de la première **`Personne`** est montré en mode élargi, alors que les instances subséquentes sont affichées en mode compressé. La capture d'écran à droite montre les éléments répétitifs de la **`Personne`** comme lignes d'une table.

Pour passer au Mode Table, cliquez sur l'icône en **Mode Table** *(encerclé en vert dans la capture d'écran ci-dessous, à gauche)*. L'icône est affichée en couleur *(voir la capture d'écran ci-dessous, à droite)* si vous êtes passé à l'affichage de table.



 © 2016-2022 Altova GmbH



Person (6)

	First	Last	Title
1	Fred	Landis	Program Manager
2	Michelle	Butler	Software Engineer
3	Ted	Little	Software Engineer
4	Ann	Way	Technical Writer
5	Liz	Gardner	Software Engineer
6	Paul	Smith	Software Engineer

Le Mode Table propose des avantages d'édition uniques dans le sens que les lignes et les colonnes entières peuvent être manipulées par rapport aux autres colonnes et lignes dans la table. Cette fonction permet des opérations comme trier les lignes de la table sur la base des valeurs d'une colonne. Dans la capture d'écran ci-dessus à droite, par exemple, les six éléments **`Personne`** peuvent être triés sur la base de leurs **`Derniers`** éléments enfants par le biais d'une seule commande dans la barre d'outils du Mode Grid. Une telle opération est plus simple que d'effectuer une transformation XSLT, qui serait la manière usuelle de trier un nodeset XML.

Pour plus d'informations, voir la page Affichage Table.

5.4.2 Structure du document

Dans le Mode Grid, la structure du document XML peut être éditée de manière graphique. Par exemple, vous pouvez réduire et étendre des segments individuels de la structure du document, insérer, ajouter et supprimer des nœuds, glisser-déposer des nœuds vers des emplacements différents, et convertir un type de nœud en un autre.

Ajouter de nouveaux nœuds

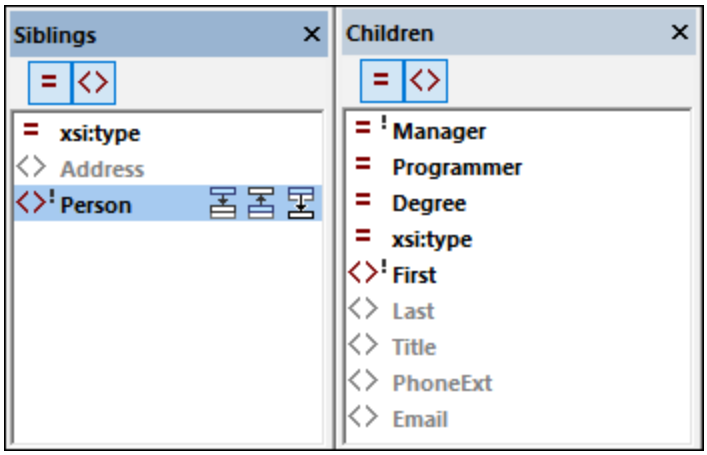
Il y a deux manières d'ajouter de nouveaux nœuds au document :

- Les assistants d'entrée des Siblings et des Enfants saisissent des nœuds spécifiques dans l'emplacement sélectionné. Pour que cela fonctionne, un document XML doit être associé à un schéma (DTD ou Schéma XML). Lorsque vous sélectionnez, par exemple, un nœud dans la fenêtre principale, le sibling et l'enfant qui sont permis à cet endroit (selon le schéma) sont affichés respectivement dans les assistants d'entrée des Siblings et des Enfants. Vous pouvez sélectionner le nœud à ajouter via les assistants d'entrée.
- Vous pouvez ajouter un nouveau nœud qui n'est pas défini dans un schéma comme sibling, enfant ou parent. Les commandes pour cela sont disponibles dans le menu contextuel des cellules du Mode Grid, et également dans le menu XML et la Barre d'outils du Mode.

Assistants d'entrée des Siblings et des Enfants

Si un document XML est associé à un schéma (DTD ou Schéma XML), les siblings et enfants disponibles d'un nœud sélectionné sont affichés dans les assistants d'entrée des Siblings et des Enfants. Dans chacun de ces assistants d'entrée, vous pouvez basculer la visibilité des éléments et des attributs sur on ou off en cliquant sur les icônes respectives de la barre d'outils. Les nœuds obligatoires sont affichés avec un point d'exclamation. Les nœuds grisés ne peuvent pas être ajoutés. Ceci est dû à l'une des raisons suivantes : (i) le nœud a déjà été ajouté et plus aucune instance du nœud n'est permise par le schéma, comme dans le cas du sibling de l'

`adresse` dans la capture d'écran ci-dessous ; (ii) un autre nœud doit être ajouté avant que le nœud grisé puisse être ajouté, comme dans le cas du `dernier` enfant dans la capture d'écran ci-dessous, qui ne peut être ajouté une fois que le `premier` enfant a été ajouté.



Ajouter des nœuds comme suit :

- *Frères :* Dans l'assistant d'entrée, sélectionnez le nœud que vous souhaitez ajouter comme sibling. Puis, cliquez sur l'icône appropriée *(voir la capture d'écran en haut à gauche)*, dépendant du fait si vous souhaitez ajouter le sibling après ou avant le nœud sélectionné dans la grille, ou l'ajouter en tant que dernier des siblings du nœud sélectionné.
- *Enfants :* Double-cliquez sur le nœud que vous souhaitez ajouter comme enfant.

Insérer de nouveaux nœuds

Lorsqu'un nœud est sélectionné dans le document (à savoir, une cellule grille), vous pouvez ajouter un nœud vide comme sibling, enfant ou parent. Les commandes pour ces opérations sont listées dans la table ci-dessous, et sont disponibles dans : (i) le menu contextuel d'une cellule ; (ii) le menu XML; et (iii) la Barre d'outils du Mode Grid.

Commande	Raccourci
Insérer (Sibling) après	Ctr+Enter
Insérer (Sibling) avant	Ctr+Shift+Enter
Ajouter (Sibling)	Ctr+Alt+A
Ajouter enfant	Ctr+Alt+Enter
Emballer dans Élément	Ctr+Alt+W

Veuillez noter les points suivants :

- Le nouveau nœud est créé dans un élément vide par défaut. Vous pouvez changer le type de nœud par la suite.
- La commande **Emballer dans Élément** crée un nœud d'élément autour du nœud actuel. Cet élément devient le nouveau parent du nœud actuel.

© 2016-2022 Altova GmbH

Modifier la structure avec des mécanismes standard Windows

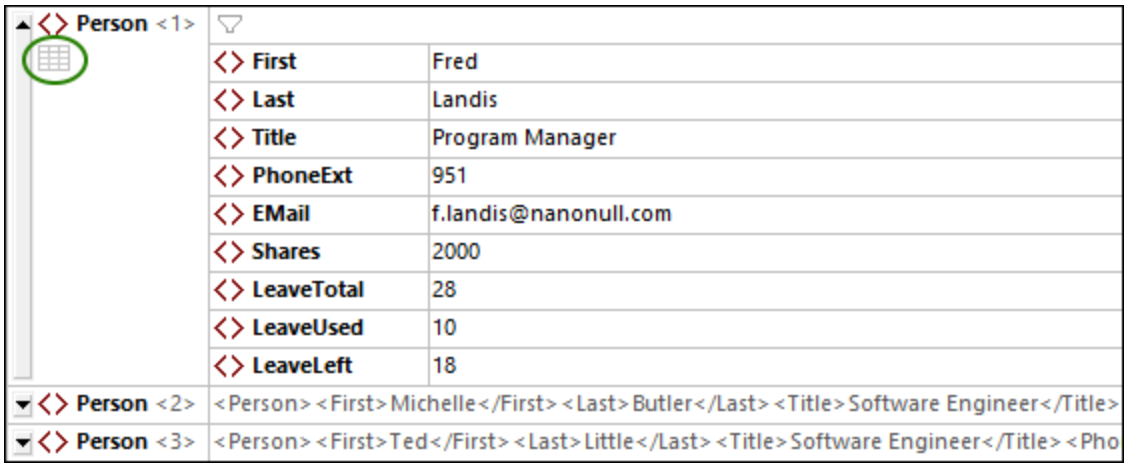
Dans la Mode Grid, vous pouvez aussi modifier la structure du document en utilisant les mécanismes Windows suivants :

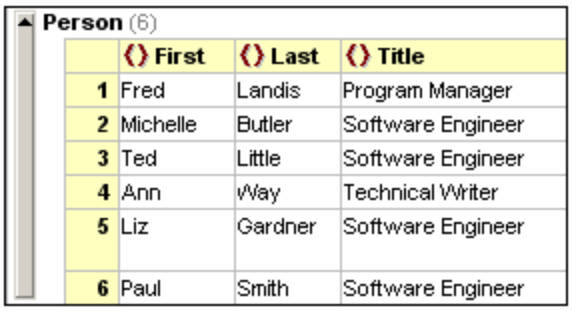
- *Supprimer :* Sélectionnez un composant et le supprimer avec la touche **Delete**.
- *Déplacer :* Sélectionner un composant et glisser-déplacer le vers un nouvel emplacement.

Afficher en tant que Table

Les éléments répétitifs sont affichés en Mode Grille *standard*, l'un après l'autre, en progressant de manière verticale vers le bas dans l'ordre du document (*voir la capture d'écran ci-dessous, à gauche*). Toutefois, afficher des éléments répétitifs comme les *lignes d'une table* fournit des fonctions d'édition additionnelles. Dans la capture d'écran ci-dessous, l'élément **`Personne`** est l'élément répétitif. La capture d'écran à gauche montre le Mode Grille standard ; l'élément de la première **`Personne`** est montré en mode élargi, alors que les instances subséquentes sont affichées en mode compressé. La capture d'écran à droite montre les éléments répétitifs de la **`Personne`** comme lignes d'une table.

Pour passer au Mode Table, cliquez sur l'icône en **Mode Table** *(encerclé en vert dans la capture d'écran ci-dessous, à gauche)*. L'icône est affichée en couleur *(voir la capture d'écran ci-dessous, à droite)* si vous êtes passé à l'affichage de table.





Le Mode Table propose des avantages d'édition uniques dans le sens que les lignes et les colonnes entières peuvent être manipulées par rapport aux autres colonnes et lignes dans la table. Cette fonction permet des opérations comme trier les lignes de la table sur la base des valeurs d'une colonne. Dans la capture d'écran ci-dessus à droite, par exemple, les six éléments **`Personne`** peuvent être triés sur la base de leurs **`Derniers`**

éléments enfants par le biais d'une seule commande dans la barre d'outils du Mode Grid. Une telle opération est plus simple que d'effectuer une transformation XSLT, qui serait la manière usuelle de trier un nodeset XML.

Pour plus d'informations, voir la page Affichage Table.

5.4.3 Document Content

Éditer le contenu XML dans le Mode Grille est simple : Double-cliquer à l'intérieur du champ du contenu et éditer le contenu tel que requis. Le type de nœud peut être changé rapidement en cliquant sur l'icône Type du nœud et en sélectionnant du menu qui apparaît. Les entités peuvent être insérées par le biais des assistants à la saisie des Entités. De plus, le Mode Grille propose une validation et la fonction de recherche et de remplacement.

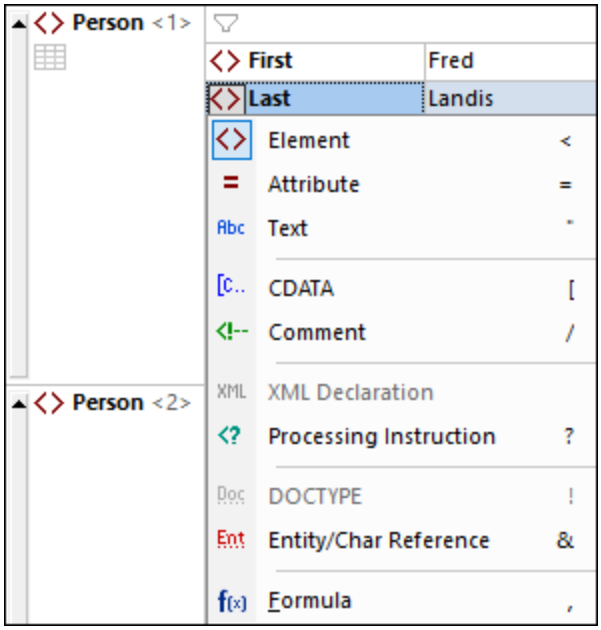
Sélection et conversion du type de nœud

Les cellules dans le mode Grille contiennent des nœuds du document XML. Pour une meilleure compréhension, nous avons regroupé les types comme suit :

- *Types Nom–Valeur :* Élément, attribut, instruction de traitement (PI). Les nœuds de ce type ont des noms et des valeurs.
- *Types de valeur :* Texte, CDATA, commenter, références entité/caractère. Les nœuds de ce type ne prennent que les valeurs.
- *Types de définition :* Déclaration XML, DOCTYPE (interne ou externe DTD). Ces nœuds définissent des propriétés du document XML.
- *Types de conteneurs spéciaux spécifiques à XMLSpy :* Formule.

Le type d'une valeur est indiqué par un symbole qui se trouve devant la valeur *(voir la capture d'écran ci-dessous)*. Pour changer un type, il faut cliquer sur son symbole et sélectionner le type souhaité depuis le menu qui apparaît. De manière alternative, cliquez avec la touche de droite de la souris dans la cellule et, depuis le menu contextuel qui apparaît, sélectionnez un type du sous-menu **Type**. Les symboles et raccourcis de types sont affichés dans la capture d'écran ci-dessous :

© 2016-2022 Altova GmbH



Veuillez noter les points suivants :

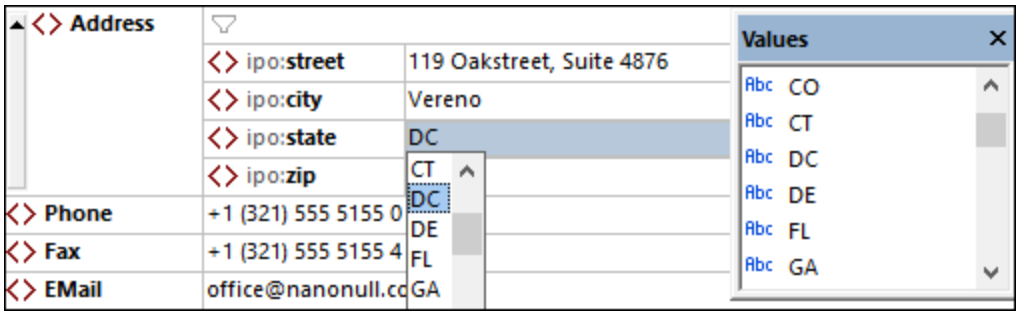
- Les nœuds du type *Élément, Attribut,* et *PI* ont un nom et une valeur, alors que les nœuds du type *Texte, CDATA,* et *Commenter* ont uniquement une valeur. Par exemple, un nœud d'élément aura un nom et un contenu, alors qu'un nœud de texte aura uniquement une valeur.
- Les conversions de type tentent de préserver la clé originale et la valeur. Si, par exemple, vous convertissez un élément en attribut, l'attribut aura (i) le même nom que l'élément, et (ii) une valeur est, dans la mesure du possible, la même chose que le contenu de l'élément.
- Dans la séquence des nœuds enfants d'un élément, les attributs sont toujours listés les premiers. Par conséquent, les conversions de type pourraient entraîner un réarrangement de nœuds.
- Dans le cas d'un contenu mixte (des données caractère parsemées avec des éléments enfants, comme un élément paragraphe qui contient du texte ainsi que des éléments en gras et en italique), les nœuds de texte consécutifs ne sont pas permis et pourraient être ajoutés automatiquement.
- Lorsque des nœuds multiples sont sélectionnés pour la conversion en un type simple, vous pouvez définir une option afin de déterminer si ce changement doit exécuté ou non, ou si vous devriez être averti.
- Vous pouvez éditer du texte brut dans une cellule ou une ligne en sélectionnant la cellule parent ou la ligne parent à éditer de cette manière et en cliquant sur l'icône **Éditer en tant que Raw Text** de la barre d'outils. Lorsque vous éditez du texte en tant que texte brut, les entités et le markup dans cette cellule ne seront pas résolus, respectivement, en glyphes et composants de Mode Grille (et peuvent être édités pour cette raison).

Saisie semi-automatique

La saisie semi-automatique est activée lorsque le document XML qui est en cours d'édition est associé à un schéma.

La saisie semi-automatique vous fournit des options de saisie à l'emplacement du curseur (*voir la capture d'écran ci-dessous*). Ces options, qui sont basées sur les définitions dans le schéma, sont fournies par le biais de (i) pop-ups dans la fenêtre principale, et (ii) via des assistants d'entrée. Les pop-ups et assistants à la saisie affichent chacun des entrées qui sont valides à l'emplacement du curseur. Pour se déplacer d'une entrée

à l'autre dans la liste pop-up, utilisez les touches de direction. Sélectionnez une entrée depuis la fenêtre pop-up ou double-cliquez dans l'assistant d'entrée pour l'insérer.



Barre d'outils/boutons de commande dans la cellule

Dans le Mode Grid XML, les boutons dans les cellules de la grille de la barre d'outils du Mode Grid proposent des fonctions qui sont disponibles dans le contexte de la sélection actuelle (cellule ou colonne). L'utilisation de ces commandes est une manière rapide et aisée d'effectuer des opérations utilisées habituellement. Les commandes disponibles sont recensées dans la table ci-dessous. Des commandes liées à l'écran sont affichées avec un fond bleu. Les commandes pour éditer le document (structure et contenu) sont affichées avec un fond beige. Pour les commandes supplémentaires dans un contexte de cellule particulier, utiliser le menu contextuel de la cellule. Note : le nom d'une commande est affiché en tant qu'info-bulle lorsque vous passez sur le bouton de la cellule ou l'icône de la barre d'outils.

	Optimise les largeurs des colonnes de la grille conformément au contenu de la cellule.
/	Si coloré, Affichage Table est activé, autrement il est désactivé. Cliquer pour passer à l'affichage
/	Une commande bascule en haut à gauche de la cellule de la table. Bascule des lignes vers les colonnes et vice versa
/	Si coloré, un filtre pour la table, un objet, ou un array est activé, autrement le filtre est désactivé. Cliquez pour désactiver/activer. Pour éditer l'expression, double-cliquez sur celle-ci
	Commande bascule pour word-wrapper les cellules. Si sélectionné, word-wrap est activé. Par défaut, seulement les contenus des items sont encapsulés. Si vous voulez en plus encapsuler les noms des items, choisissez cette option depuis la liste déroulante de l'icône.
100%	Niveau de zoom du Mode Grid
	Activé quand une en-tête de colonne est sélectionnée. Les boutons trient les lignes de la table, respectivement par ordre décroissant ou croissant du contenu de la colonne

Note : pour une description d'autres commandes de la barre d'outils, voir la section Menu contextuelo dans le Mode Grille.

© 2016-2022 Altova GmbH

Valider sur modification

Le mode *Valider sur Édition* est désactivée par défaut. Lorsqu'il est activé, des vérifications de bonne formation et de validation sont effectuées pendant que vous éditer un document dans le Mode Grille. Pour que la validation soit effectuée (en plus des vérifications de bonne forme), un DTD ou Schéma XML doit être attribué au document. Les erreurs sont indiquées en affichant du texte erroné en rouge et en marquant l'emplacement avec un point d'exclamation rouge. Si un smart fix est disponible pour une erreur, une ampoule s'affiche sur la ligne qui génère l'erreur. Lorsque vous placez la souris sur l'icône, une fenêtre pop-up apparaît qui recense les smart fixes disponibles. Sélectionner un fix pour l'appliquer immédiatement. Voir Valider des documents XML pour plus d'informations.

Le mode *Valider sur Édition* peut être activé ou désactivé soit (i) via la commande de menu **XML | Valider sur Édition**, (ii) via la touche de la barre d'outils **Valider sur Édition**, ou (iii) via l'option *Sur Édition* des Paramètres de validation du dialogue Options.

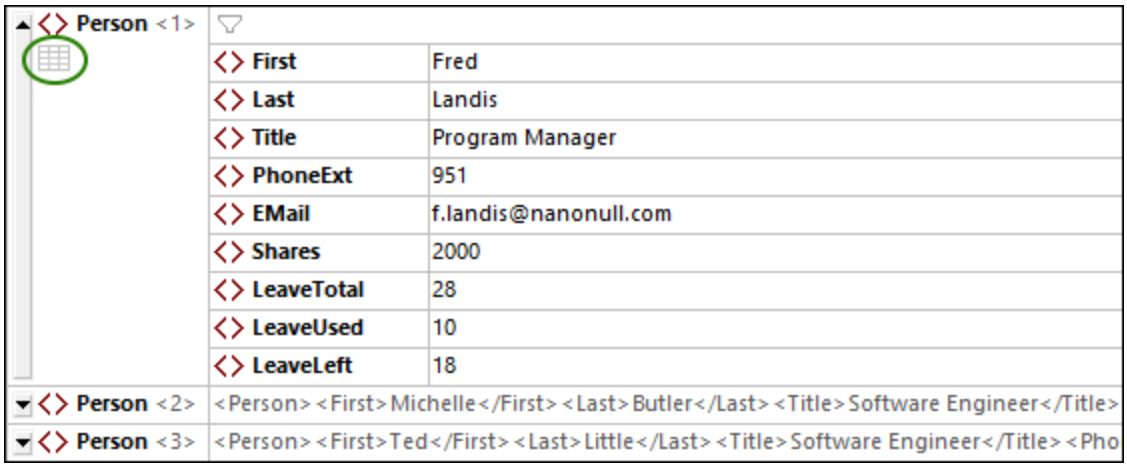
Chercher et Remplacer

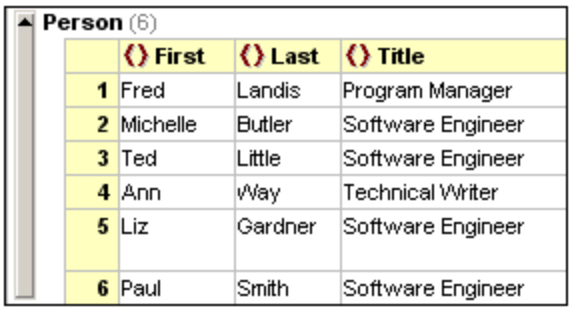
Vous pouvez utiliser les commandes **Recherche (Ctrl+F)** et **Remplacer (Ctrl+H)** du menu **Éditer** pour trouver du texte dans le Mode Grille et le remplacer. Les résultats sont mis en surbrillance orange, et les cellules avec du contenu sont également mises en surbrillance orange.

Afficher en tant que Table

Les éléments répétitifs sont affichés en Mode Grille *standard*, l'un après l'autre, en progressant de manière verticale vers le bas dans l'ordre du document (*voir la capture d'écran ci-dessous, à gauche*). Toutefois, afficher des éléments répétitifs comme les *lignes d'une table* fournit des fonctions d'édition additionnelles. Dans la capture d'écran ci-dessous, l'élément **`Personne`** est l'élément répétitif. La capture d'écran à gauche montre le Mode Grille standard ; l'élément de la première **`Personne`** est montré en mode élargi, alors que les instances subséquentes sont affichées en mode compressé. La capture d'écran à droite montre les éléments répétitifs de la **`Personne`** comme lignes d'une table.

Pour passer au Mode Table, cliquez sur l'icône en **Mode Table** *(encerclé en vert dans la capture d'écran ci-dessous, à gauche)*. L'icône est affichée en couleur *(voir la capture d'écran ci-dessous, à droite)* si vous êtes passé à l'affichage de table.





Person (6)

	First	Last	Title
1	Fred	Landis	Program Manager
2	Michelle	Butler	Software Engineer
3	Ted	Little	Software Engineer
4	Ann	Way	Technical Writer
5	Liz	Gardner	Software Engineer
6	Paul	Smith	Software Engineer

Le Mode Table propose des avantages d'édition uniques dans le sens que les lignes et les colonnes entières peuvent être manipulées par rapport aux autres colonnes et lignes dans la table. Cette fonction permet des opérations comme trier les lignes de la table sur la base des valeurs d'une colonne. Dans la capture d'écran ci-dessus à droite, par exemple, les six éléments **`Personne`** peuvent être triés sur la base de leurs **`Derniers`** éléments enfants par le biais d'une seule commande dans la barre d'outils du Mode Grid. Une telle opération est plus simple que d'effectuer une transformation XSLT, qui serait la manière usuelle de trier un nodeset XML.

Pour plus d'informations, voir le sujet Affichage Table.

5.4.4 Glisser-déposer dans le Mode Grille

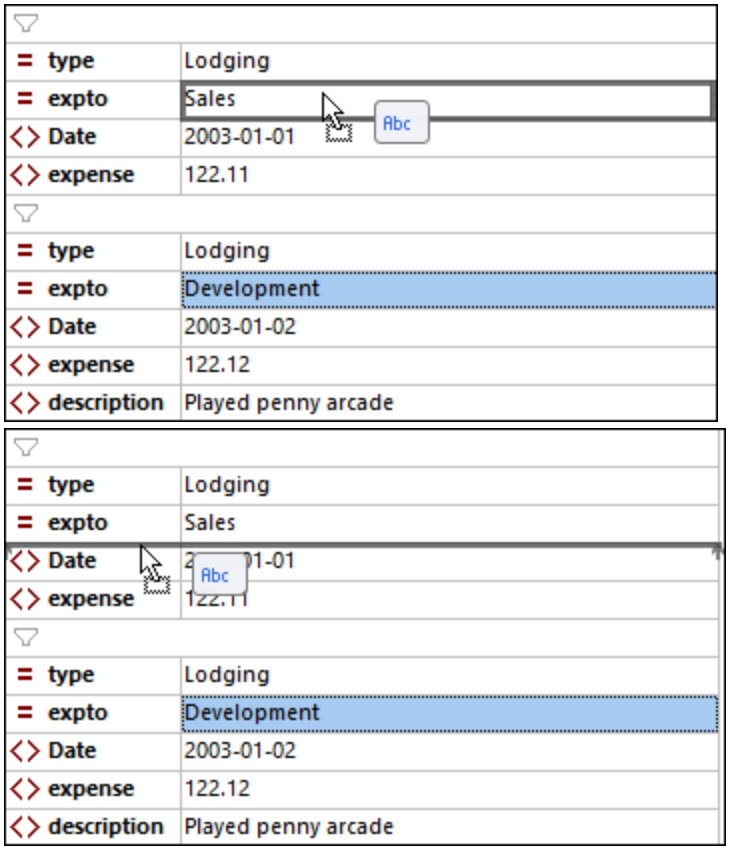
Le Mode Grille propose une superposition pour glisser très utile qui vous permet de glisser un fragment de document XML dans le Mode Grille depuis un document qui est ouvert dans XMLSpy, une application externe, ou même un site web.

Si vous placez le curseur sur un nœud cible, une superposition pour glisser apparait, qui fournit non seulement des informations sur ce qui a été glissé (type et nombre d'item/s) mais également sur la manière dont l'item sera créé lorsqu'il est déposé. Par exemple, dans la capture d'écran ci-dessous, la valeur **`Development`** est glissée (déplacée). Le popup **`Abc`** indique que l'item est une valeur string.

- Si vous passez sur un champ de valeur (comme dans la capture d'écran à gauche), le champ entier est marqué, indiquant que la valeur du champ marqué sera remplacée par la valeur du champ glissé.
- Si vous passez sur un nom de nœud (tel que **`Date`** dans la capture d'écran à droite), une ligne d'insertion apparait, indiquant que le string de valeur sera déposé à cet endroit pour créer un nœud de texte dans l'arborescence.

Note : vous pouvez toujours annuler chaque étape avec **Ctrl+Z** ou la commande menu **Éditer | Annuler**.

© 2016-2022 Altova GmbH

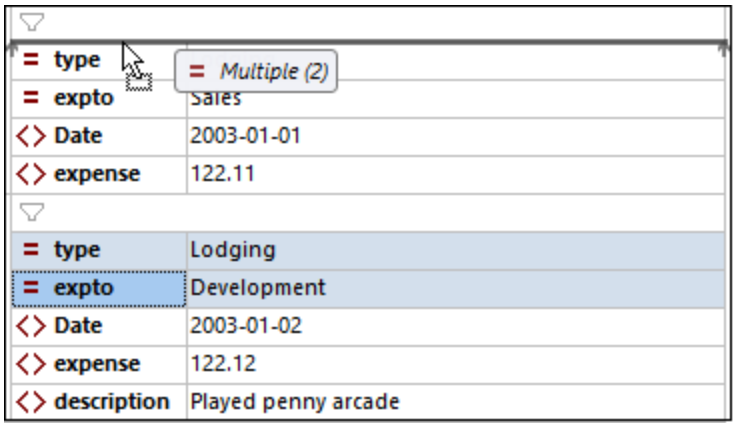


Information contenant une superposition pour glisser

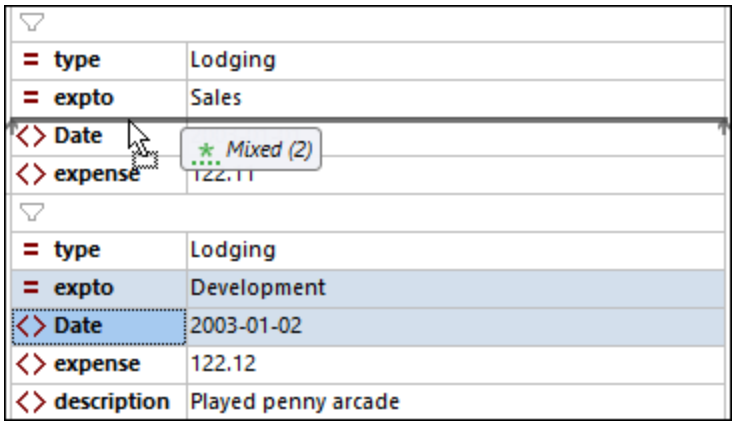
Les informations suivantes concernant une superposition pour glisser seront affichées.

Mode Grille normal

- La valeur glissée dans un champ de valeur fait que la valeur source remplace la valeur dans le champ cible (*voir la capture d'écran ci-dessus gauche*).
- La valeur déposée dans l'arborescence en tant que nœud crée un nœud de texte (*voir la capture d'écran ci-dessus droite*).
- La superposition pour des nœuds multiples d'un seul type indique le type de nœud qui est déposé et le nombre de ces nœuds. Pour sélectionner un nœud (et pas sa valeur), veuillez noter que vous devez cliquer sur le nom du nœud. Dans la capture d'écran ci-dessous, l'information dans la superposition pour glisser indique que les deux attributs seront déposées.

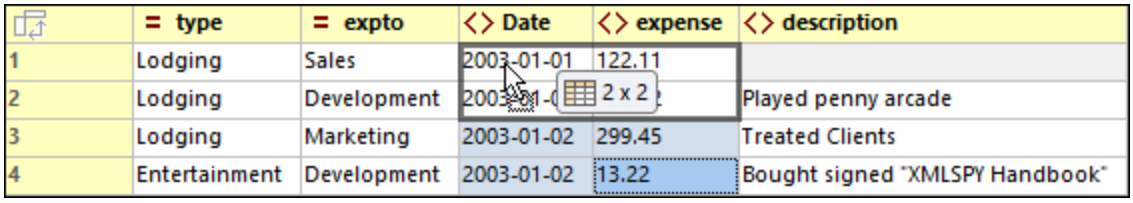


- La superposition pour des nœuds multiples de types différents montre le texte *Mixed* et le nombre de nœuds dans le presse-papiers (*voir la capture d'écran ci-dessous*). Pour sélectionner un nœud (et pas sa valeur), veuillez noter que vous devez cliquer sur le nom du nœud.



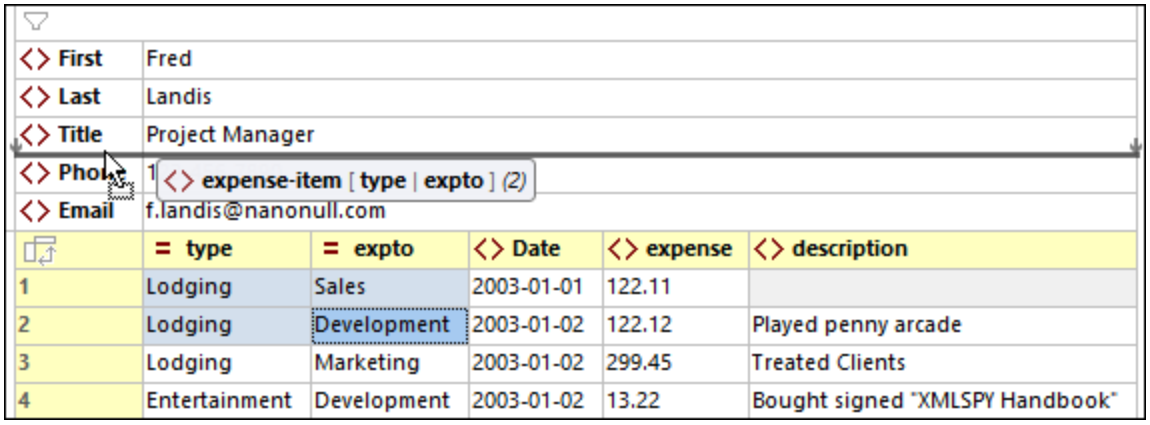
Affichage de la table

- Lorsque les valeurs sont glissées, le nombre de cellules sélectionnées est indiqué par `colonne x lignes` (*voir la capture d'écran ci-dessous*). La matrice des cellules glissées remplacera une matrice correspondante. La cellule cible (la cellule sur laquelle vous déposez la matrice) recevra la valeur de la cellule en haut à gauche de la matrice. Les autres cellules glissées rempliront les cellules vers la droite et vers le bas de la cellule cible. Par exemple, dans la capture d'écran ci-dessous, les cellules bleues sont glissées et déposées sur la cellule `Date`. Ceci a pour conséquence que celles-ci remplacent les cellules qui ont été marquées. Si la matrice des cellules glissées dépasse les limites de la table, le nombre approprié de colonnes et/ou de lignes est ajouté pour accommoder les cellules glissées. Dans ce cas, les limites qui seront élargies sont indiquées par des pointillés.



 © 2016-2022 Altova GmbH

- Lorsque la cible est l'arborescence du nœud —et non la cellule—alors le nœud ajouté est indiqué, ensemble avec (i) les colonnes qui seront ajoutées (dans des crochets), et (ii) le nombre de lignes qui seront ajoutées pour le même élément (*voir la capture d'écran ci-dessous*).

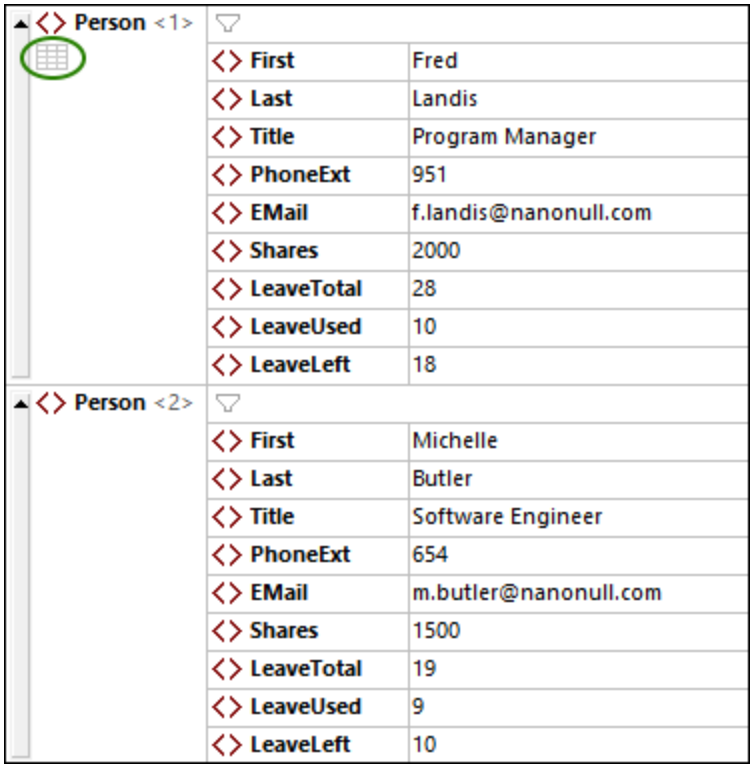


- Une colonne de table peut être déplacée en la sélectionnant (cliquez sur son en-tête à cet égard) et glissée dans l'en-tête de la colonne adjacente à laquelle vous voulez la déplacer. Une ligne d'insertion indiquera de quel côté de la colonne cible la colonne déplacée sera placée.

5.4.5 Affichage de la table

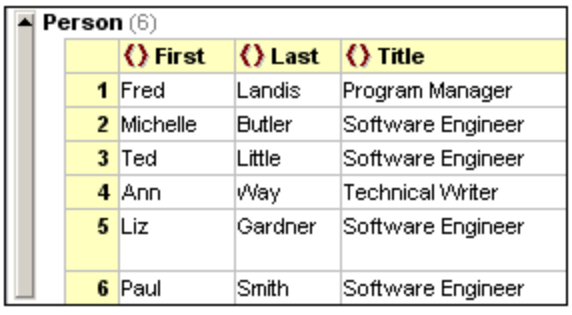
À propos de l'Affichage de la table

Dans le Mode Grid standard, si un élément est répétitif, chaque instance d'élément est affichée, l'une après l'autre dans l'ordre du document. Dans la capture d'écran ci-dessous, par exemple, l'élément **`Personne`** est l'élément répétitif.



Person <1>	
First	Fred
Last	Landis
Title	Program Manager
PhoneExt	951
EMail	f.landis@nanonull.com
Shares	2000
LeaveTotal	28
LeaveUsed	10
LeaveLeft	18
Person <2>	
First	Michelle
Last	Butler
Title	Software Engineer
PhoneExt	654
EMail	m.butler@nanonull.com
Shares	1500
LeaveTotal	19
LeaveUsed	9
LeaveLeft	10

Un élément répétitif, comme l'élément personne **Personne** affichée ci-dessus, peut également être affiché comme table (*capture d'écran ci-dessous*). Dans la représentation de la table, les nœuds enfants de l'élément répétitif forment les colonnes de la table, alors que les instances de l'élément répétitif forment les lignes de la table. Pour basculer vers le Mode Table, cliquez sur l'icône **Mode de la table** qui est affichée dans la première instance de l'élément répétitif *(cercle vert dans la capture d'écran ci-dessus)*. Lorsque vous avez basculé vers l'affichage de la table, l'icône **Mode de la table** est affichée en couleur *(voir la capture d'écran ci-dessous)*.



Person (6)

	First	Last	Title
1	Fred	Landis	Program Manager
2	Michelle	Butler	Software Engineer
3	Ted	Little	Software Engineer
4	Ann	Way	Technical Writer
5	Liz	Gardner	Software Engineer
6	Paul	Smith	Software Engineer

Le Mode Table propose un avantage d'édition unique dans le sens que les lignes et les colonnes peuvent être manipulées par rapport aux autres colonnes et lignes dans la table. Ceci permet des opérations telles que trier des lignes sur les valeurs d'une colonne. Dans la capture d'écran, par exemple, les six éléments `Personne` peuvent être triés sur la base de leurs **Derniers** éléments enfants par le biais d'une simple opération GUI. Une telle opération *(voir ci-dessous pour les détails)* est plus simple que d'effectuer une transformation XSLT, qui est la manière usuelle de trier un nodeset XML.

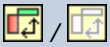
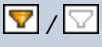
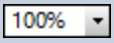
© 2016-2022 Altova GmbH

Note

- L'affichage de la table ne peut être appliquée qu'à une séquence d'éléments du même nom.
- L'affichage de la table est également disponible pour un seul élément. L'icône de l'élément **Mode Table** devient visible lorsque vous cliquez sur l'élément.
- Les couleurs de l'affichage de la table peuvent être définies dans le dialogue des Options : Polices et couleurs | Mode Grid | Couleurs de cellule.

Les icônes pour afficher et éditer dans l'Affichage de la table

Les icônes affichées ci-dessous sont disponibles dans l'Affichage de la table et proposent des fonctions d'affichage et d'édition. Elles sont visibles dans les cellules de table et/ou dans la barre d'outils en Mode Grid.

	Optimise les largeurs des colonnes de la grille conformément au contenu de la cellule.
	Si coloré, Affichage Table est activé, autrement il est désactivé. Cliquer pour passer à l'affichage
	Une commande bascule en haut à gauche de la cellule de la table. Bascule des lignes vers les colonnes et vice versa
	Si coloré, un filtre pour la table, un objet, ou un array est activé, autrement le filtre est désactivé. Cliquez pour désactiver/activer. Pour éditer l'expression, double-cliquez sur celle-ci
	Commande bascule pour word-wrapper les cellules. Si sélectionné, word-wrap est activé. Par défaut, seulement les contenus des items sont encapsulés. Si vous voulez en plus encapsuler les noms des items, choisissez cette option depuis la liste déroulante de l'icône.
	Niveau de zoom du Mode Grid
	Activé quand une en-tête de colonne est sélectionnée. Les boutons trient les lignes de la table, respectivement par ordre décroissant ou croissant du contenu de la colonne

Éditer dans l'Affichage de la table

Dans l'Affichage de la table, vous pouvez réaliser des actions d'édition décrites ci-dessous.

Ajouter une ligne de table (nouvelle instance de l'élément répétitif de la table)
Vous pouvez ajouter une nouvelle ligne—à savoir, une autre instance de l'élément répétitif de la table—comme suit :

1. Cliquez avec la touche droite sur *la cellule qui contient le numéro* de la ligne au-dessus ou en dessous de la ligne que vous souhaitez ajouter.
2. Sélectionnez la commande **Insérer après (Ctrl+Enter)** pour ajouter une ligne en dessous de la ligne sélectionnée, ou **Insérer avant (Ctrl+Shift+Enter)** pour ajouter une ligne au-dessus de la ligne sélectionnée. Ces commandes sont aussi disponibles dans le menu XML et la barre d'outils dans le Mode Grid.

La nouvelle ligne sera créée comme un nœud élément. Vous pouvez changer son type de nœud si vous le souhaitez *(voir ci-dessous pour les détails)*.

Ajouter des éléments sibling ou enfant dans une cellule de table
Si une cellule de table représente un élément enfant d'une ligne de table, vous pouvez attribuer à cet élément enfant un nœud sibling suivant ou un nœud enfant. Cliquez avec la touche droite sur la cellule de table et sélectionnez respectivement la commande **Ajouter (Ctrl+Alt+A)** ou **Ajouter enfant (Ctrl+Alt+Enter)**. Ces commandes sont aussi disponibles dans le menu XML et la barre d'outils dans le Mode Grid. La nouvelle ligne sera créée comme un nœud élément. Vous pouvez changer son type de nœud si vous le souhaitez *(voir ci-dessous pour les détails)*.

Emballer la cellule dans l'élément
Vous pouvez créer un élément autour de la cellule de table. Le nouvel élément sera créé à un niveau entre celui de la cellule et celui du parent de la cellule. À cette fin, cliquez avec la touche droite sur la cellule de table et sélectionnez la commande **Emballer dans Élément (Ctrl+Alt+W)**. Cette commande est également disponible dans le menu XML et la barre d'outils dans le Mode Grid .

Ajouter une colonne de table (nouveau nœud enfant de toutes les instances de l'élément répétitif de la table)
Vous pouvez ajouter une nouvelle colonne—à savoir, un nouveau nœud enfant de toutes les instances de l'élément répétitif de la table—comme suit :

1. Cliquez avec la touche droite sur l'en-tête d'une colonne ou sur une cellule non vide d'une colonne.
2. Sélectionnez une commande **Insérer après (Ctrl+Enter)** pour ajouter une colonne à droite de la colonne sélectionnée, ou **Insérer avant (Ctrl+Shift+Enter)** pour ajouter une colonne à gauche de la colonne sélectionnée. Ces commandes sont aussi disponibles dans le menu XML et la barre d'outils dans le Mode Grid.

La nouvelle colonne sera créée comme un nœud élément. Vous pouvez changer son type de nœud si vous le souhaitez *(voir ci-dessous pour les détails)*.

Changer les types de nœuds et les noms des colonnes
Pour changer un type de nœud de la colonne, cliquez sur l'icône du type de nœud de la colonne et, depuis le menu qui s'affiche, sélectionnez un nouveau type de nœud. Le type de ce nœud sera modifié pour toutes les instances de l'élément répétitif.

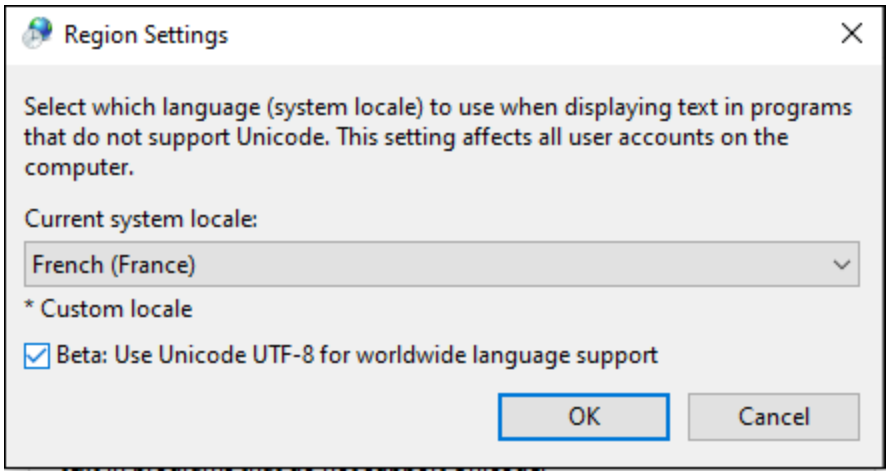
Pour changer le nom d'un en-tête d'une colonne, double-cliquez sur le nom et éditez-le. En conséquence, le nom du nœud sera modifié dans toutes les instances de l'élément répétitif.

Trier les lignes de table sur la base des valeurs d'une colonne sélectionnée
Vous pouvez trier les lignes d'une table sur la base des valeurs relatives d'un de ses nœuds enfant (une colonne). Par exemple, vous pouvez trier les lignes sur la base de la colonne *DernierNom* pour vous donner les éléments répétitifs de la table triée par ordre alphabétique. Pour trier une colonne, sélectionnez l'en-tête de colonne et cliquez sur la commande barre d'outils dans le Mode Grid **Trier par ordre croissant** ou **Trier par ordre décroissant**. Ces commandes sont aussi disponibles dans le menu XML.

Trier l'ordre dans certaines langues, en particulier celles qui n'ont pas d'alphabet latin, peut être bénéfique en activant la prise en charge beta Unicode UTF-8 dans le dialogue des paramètres de la région linguistique de Windows 10 (ou supérieure). Pour ce faire, suivez les étapes suivantes : Allez au dialogue des *Paramètres* de Windows et recherchez les *Paramètres de langue*. Sous *Paramètres associés*, cliquez sur *Paramètres de langue administrative*. Dans le dialogue Région qui apparaît, allez à l'onglet *Administrative* et sous *Language for non-Unicode programs*, cliquez sur *Change system locale*. Dans le dialogue des paramètres de la région qui apparaît (*voir la capture d'écran ci-dessous*), sélectionnez l'option *Beta : Use Unicode UTF-8 for worldwide*

 © 2016-2022 Altova GmbH

language support et cliquez sur **OK**.



Affichage de la table et applications externes

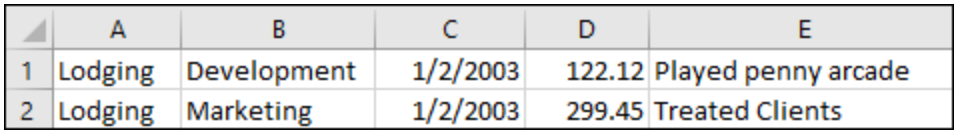
Vous pouvez tirer profit de la structure de la table pour échanger les données entre l'affichage de la table et une application spreadsheet (telle que MS Excel). Pour déplacer les données de l'affichage de la table, suivez les instructions suivantes :

1. Dans l'affichage de la table, sélectionnez les nœuds que vous voulez copier. Procédez en cliquant sur les cellules elles-mêmes, les en-tête colonnes, les en-tête lignes, ou sur la table entière. Si vous cliquez sur la table entière ou les en-tête de colonnes, les en-tête de colonnes sont également copiées ; autrement, elles ne le sont pas. (Dans la capture d'écran ci-dessous, les lignes 2 et 3 sont sélectionnées.)



expense-item (3)		type	expto	Date	expense	description
	1	Lodging	Sales	2003-01-01	122.11	
	2	Lodging	Development	2003-01-02	122.12	Played penny arcade
	3	Lodging	Marketing	2003-01-02	299.45	Treated Clients

2. Sélectionnez la commande du menu contextuel Copier | Copier en tant que texte séparé par tabulation.
3. Collez les données directement dans le programme de spreadsheet.



	A	B	C	D	E
1	Lodging	Development	1/2/2003	122.12	Played penny arcade
2	Lodging	Marketing	1/2/2003	299.45	Treated Clients

L'échange de données fonctionne dans les deux sens. Vous pouvez aussi copier des données d'une spreadsheet à l'affichage de la table. Pour ce faire, procédez comme suit :

1. Sélectionnez une plage dans l'application externe et copiez-la dans le presse-papiers (avec **Ctrl+C**).
2. Sélectionnez une cellule individuelle dans l'affichage de la table.
3. Collez les données copiées avec **Ctrl+V**.

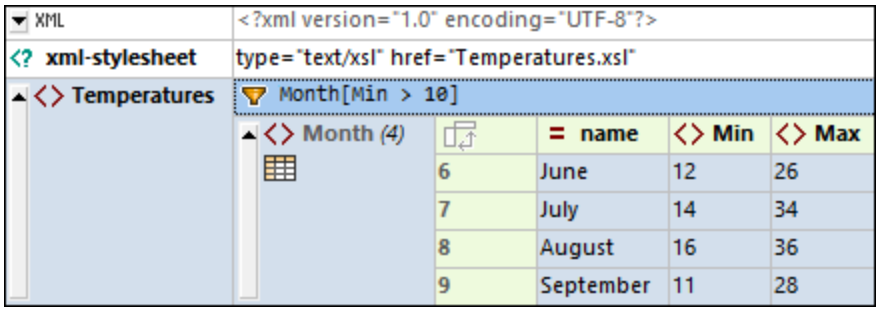
Les données seront collées dans la table avec une structure correspondant à la structure originale et en commençant par la cellule choisie dans le Mode Table. Notons les points suivants :

- Si les données existent déjà dans ces cellules du Mode Table, les nouvelles données écrasent les données originales.
- Si plus de lignes et/ou colonnes sont requises pour tenir compte des nouvelles données, celles-ci sont créées.
- Les données deviennent le contenu des éléments représentés par les cellules respectives.

Pour plus de tâches complexes liées à l'échange de données, utilisez les fonctions uniques importation/exportation de XMLSpy.

5.4.6 Filtres

Un filtre dans le Mode Grille XML peut être appliqué à un nœud d'élément et vous permet de filtrer les descendants de ce nœud. Partout où vous pouvez appliquer un filtre, une icône de filtre grisée est affichée. Une fois qu'un filtre a été défini, cette icône est affichée en couleur *(voir la capture d'écran ci-dessous)*. Un filtre est défini avec une expression XQuery 3.1. Dans la capture d'écran ci-dessous, par exemple, un filtre a été défini pour le nœud **`Températures`** afin de n'afficher que ces éléments **`Mois`** qui ont un élément enfant **`Min`** avec une valeur supérieure à `10`. (Veuillez noter que dans la capture d'écran le numéro de l'index de l'instance élément respective est affichée.)



Pour définir un filtre, cliquez de la touche droite sur l'élément que vous souhaitez filtrer, sélectionnez **Filtre** depuis le menu contextuel qui apparaît, saisissez l'expression XQuery, et cliquez sur **Entrer**. La cellule *Filtre* est indiquée par l'icône  *(voir la capture d'écran ci-dessus)*. Vous pouvez activer/désactiver le filtre en cliquant sur l'icône *Filtre*.

Liste du document XML utilisée dans la capture d'écran ci-dessus

```
<?xml version="1.0" encoding="UTF-8"?>
<Températures>
   <Mois nom="Janvier">
      <Min>-5</Min>
      <Max>3</Max>
   </Mois>
   <Mois nom="Février">
      <Min>-16</Min>
      <Max>1</Max>
   </Mois>
```


© 2016-2022 Altova GmbH

```
    <Mois nom="Mars">
       <Min>-9</Min>
       <Max>7</Max>
    </Mois>
    <Mois nom="Avril">
       <Min>2</Min>
       <Max>16</Max>
    </Mois>
    <Mois nom="Mai">
       <Min>8</Min>
       <Max>21</Max>
    </Mois>
    <Mois nom="Juin">
       <Min>12</Min>
       <Max>26</Max>
    </Mois>
    <Mois nom="Juillet">
       <Min>14</Min>
       <Max>34</Max>
    </Mois>
    <Mois nom="Août">
       <Min>16</Min>
       <Max>36</Max>
    </Mois>
    <Mois nom="Septembre">
       <Min>11</Min>
       <Max>28</Max>
    </Mois>
    <Mois nom="Octobre">
       <Min>10</Min>
       <Max>26</Max>
    </Mois>
    <Mois nom="Novembre">
       <Min>-1</Min>
       <Max>14</Max>
    </Mois>
    <Mois nom="Décembre">
       <Min>-3</Min>
       <Max>9</Max>
    </Mois>
</Températures>
```

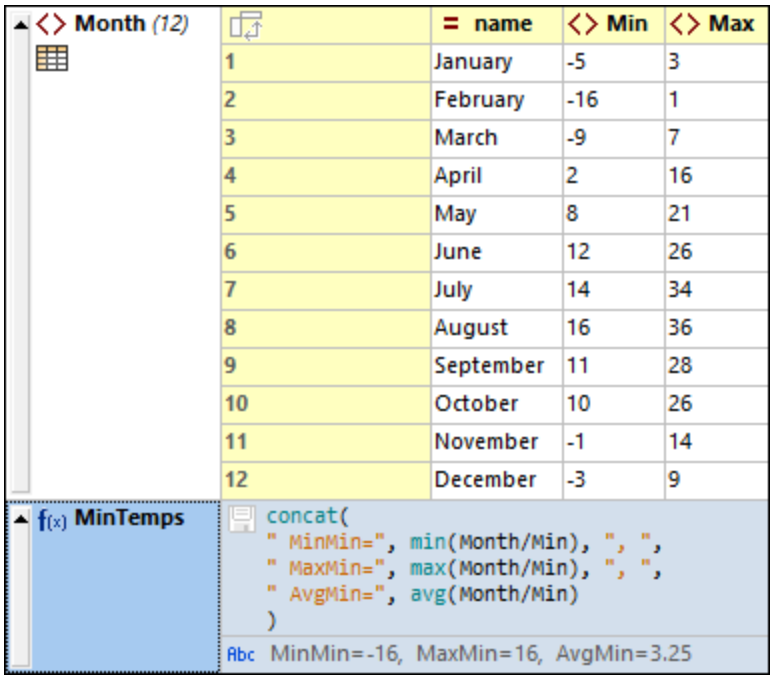
Veuillez noter les points suivants concernant les filtres :

- Les filtres peuvent être appliqués uniquement aux nœuds d'élément.
- Le nœud de contexte de l'expression XQuery du filtre est le nœud actuel. Dans la capture d'écran ci-dessus, par exemple, le nœud de contexte est le nœud **Températures**.
- Les filtres peuvent être imbriqués. Un filtre imbriqué peut être appliqué à un contenu filtré du filtre parent.
- Pour ajouter une nouvelle ligne dans une expression, appuyez sur **Ctrl+Enter**. Ceci est utile si vous voulez afficher une expression sur plusieurs lignes pour une meilleure lecture.

- Le contenu filtré est uniquement un affichage. Le contenu actuel reste inchangé.
- Chaque filtre est exécuté de manière indépendante et n'est pas affecté par d'autres filtres ou par des formules.
- Les filtres ne sont **pas stockés** dans le document XML, mais peuvent être gardés dans un fichier de métadonnées spécial pur une utilisation ultérieure. Si les filtres sont enregistrés, il seront automatiquement appliqués lorsque le document est rouvert dans le Mode Grid. Aller à **Outils | Options | View | Paramètres Mode Grid** pour déterminer s'il faut toujours automatiquement stocker les filtres, s'il faut les stocker uniquement sur demande, ou alors s'il ne faut pas les enregistrer du tout.

5.4.7 Formules

Une formule dans le Mode Grille XML utilise une expression XQuery 3.1 pour calculer un résultat ou générer un nodeset qui peut être stocké dans le document. Une formule est définie avec une expression XQuery 3.1. Par exemple, dans la capture d'écran ci-dessous, une formule appelée `MinTemps` a été créée pour générer un minimum, maximum, et une moyenne d'ensemble de toutes les températures minimum.



Month (12)	name	Min	Max
1	January	-5	3
2	February	-16	1
3	March	-9	7
4	April	2	16
5	May	8	21
6	June	12	26
7	July	14	34
8	August	16	36
9	September	11	28
10	October	10	26
11	November	-1	14
12	December	-3	9

Pour créer un formule, procéder comme suit :

1. Ajouter un nouveau nœud là où vous souhaitez afficher la formule.
2. Le nœud sera créé par défaut comme un élément. Changer le type du nœud vers la formule.
3. Double-cliquez dans la cellule qui contient l'icône `f(x)` et saisissez le nom de la formule *(voir la capture d'écran ci-dessus)*.
4. Vous pouvez cliquer sur l'icône pour enregistrer le résultat de la formule vers le document.
5. Double-cliquez dans la cellule de l'expression et tapez l'expression, puis cliquez sur **Enter**.

Liste du document XML utilisée dans la capture d'écran ci-dessus

© 2016-2022 Altova GmbH

```
<?xml version="1.0" encoding="UTF-8"?>
<Températures>
   <Mois nom="Janvier">
      <Min>-5</Min>
      <Max>3</Max>
   </Mois>
   <Mois nom="Février">
      <Min>-16</Min>
      <Max>1</Max>
   </Mois>
   <Mois nom="Mars">
      <Min>-9</Min>
      <Max>7</Max>
   </Mois>
   <Mois nom="Avril">
      <Min>2</Min>
      <Max>16</Max>
   </Mois>
   <Mois nom="Mai">
      <Min>8</Min>
      <Max>21</Max>
   </Mois>
   <Mois nom="Juin">
      <Min>12</Min>
      <Max>26</Max>
   </Mois>
   <Mois nom="Juillet">
      <Min>14</Min>
      <Max>34</Max>
   </Mois>
   <Mois nom="Août">
      <Min>16</Min>
      <Max>36</Max>
   </Mois>
   <Month name="Septembre">
      <Min>11</Min>
      <Max>28</Max>
   </Mois>
   <Month name="Octobre">
      <Min>10</Min>
      <Max>26</Max>
   </Mois>
   <Month name="Novembre">
      <Min>-1</Min>
      <Max>14</Max>
   </Mois>
   <Month name="Décembre">
      <Min>-3</Min>
      <Max>9</Max>
   </Mois>
</Températures>
```

Liste de l'expression de la formule XML utilisée dans la capture d'écran ci-dessus

```
concat(
" MinMin=", min(//Mois/Min), ", ",
" MaxMin=", max(//Mois/Min), ", ",
" AvgMin=", avg(//Mois/Min)
)
```

Veuillez noter les points suivants :

- Le nœud du contexte de l'expression XQuery de la formule est le nœud parent du nœud de la formule.
- Pour ajouter une nouvelle ligne dans une expression, appuyez sur **Ctrl+Enter**. Ceci est utile si vous voulez afficher une expression sur plusieurs lignes pour une meilleure lecture.
- Les formules (à savoir, les expressions définissant les formules) sont stockées dans les métadonnées du document (situé dans un fichier de métadonnées accessible à toute l'application). Les formules sont également sauvegardées en tant qu'instructions de traitement dans le fichier XML.
- Le résultat généré par une formule est affiché dans la cellule en dessous de l'expression XQuery de la formule et stocké dans les métadonnées du document. Le résultat peut également être stocké dans le document. Pour ce faire, cliquez sur l'icône du disque à côté de l'expression XQuery. Que vous ayez cliquer sur l'icône ou non, le résultat de la formule sera calculé et stocké dans les métadonnées du document.
- Lorsque le résultat de la formule est un calcul, il est stocké comme contenu d'un élément qui porte le nom que vous lui avez assigné. Par exemple, dans la formule `MinTemps` affichée dans la capture d'écran ci-dessus, le résultat sera stocké dans un élément nommé `MinTemps`.
- Aller à **Outils | Options | View | Paramètres Mode Grid** : (i) spécifier si les formules doivent être stockées dans les métadonnées toujours, sur demande uniquement, ou jamais, (ii) spécifier si les formules doivent être stockées dans le document.

Formules dans les tables

Si toutes les cellules d'une colonne de table (in Affichage Table) contiennent la même formule, la formule est affichée une seule fois—dans l'en-tête de la colonne *(voir la capture d'écran ci-dessous)*. Les *résultats* du calcul de la formule sont, toutefois, affichés dans les cellules respectives. La formule dans l'en-tête de la colonne est une représentation du Mode Grid. Dans le document XML (voir Mode Texte), la formule est répétée pour chaque item ligne-table.

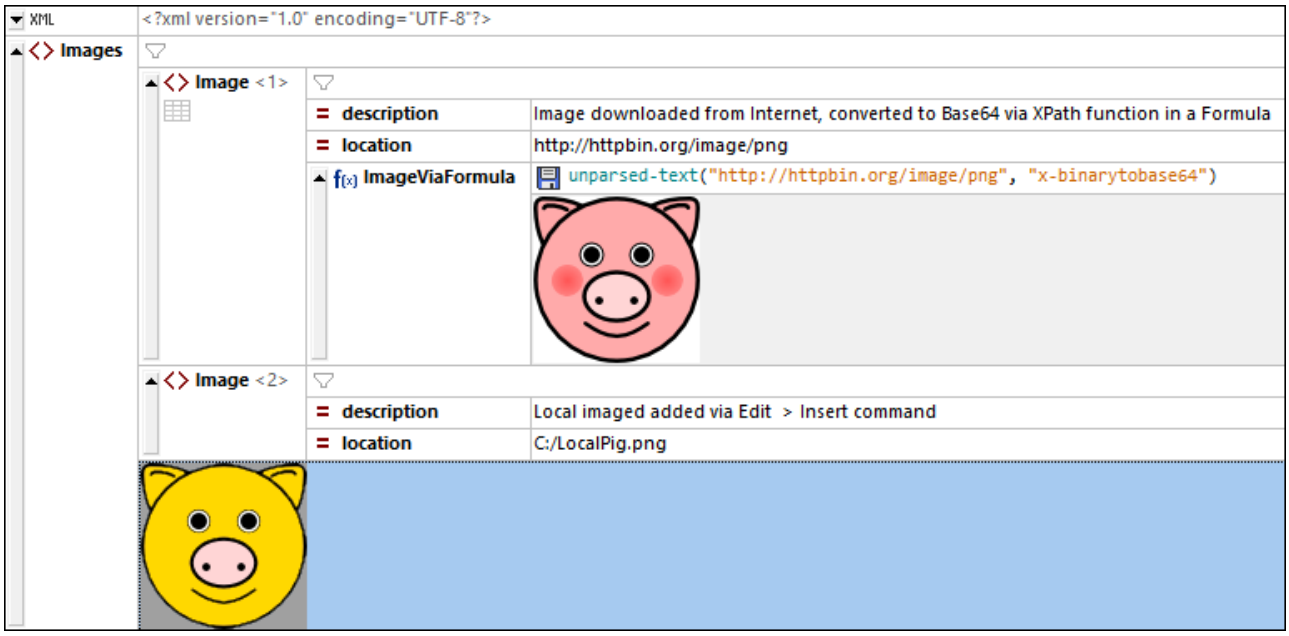
© 2016-2022 Altova GmbH

Month (12)	name	Min	Max	MidRange (Min + Max) div 2
1	January	-5	3	# -1
2	February	-16	1	# -7.5
3	March	-9	7	# -1
4	April	2	16	# 9
5	May	8	21	# 14.5
6	June	12	26	# 19
7	July	14	34	# 24
8	August	16	36	# 26
9	September	11	28	# 19.5
10	October	10	26	# 18
11	November	-1	14	# 6.5
12	December	-3	9	# 3

Même si une seule formule d'une cellule est différente, chaque formule sera affichée dans sa cellule respective. Si toutes les formules d'une colonne de table sont les mêmes pour que la formule apparaisse dans l'en-tête, et vous voulez créer maintenant une formule différente pour une cellule individuelle, éteignez Affichage Table et éditer la formule de cette cellule. Si vous retournez à Affichage Table, les formules seront affichées dans les cellules individuelles (pour toutes les cellules).

5.4.8 Images

Les images peuvent être affichées directement dans le Mode Grid *(voir capture d'écran ci-dessous)*. À cette fin, l'image doit être insérée dans son encodage Base64.



Vous avez deux manières d'insérer l'image encodée en Base64 dans le Mode Grid :

- Créer un nœud de type *Formule*. Dans la cellule d'expression de la formule *(voir la capture d'écran ci-dessus)*, entrer l'expression XPath suivante afin de convertir une image dans son encodage Base64 : `unparsed-text("<Image-URL>", "x-binarytobase64")`. La fonction XPath `unparsed-text` convertit l'image dans son encodage Base64. L'image sera visible sous la cellule qui contient l'expression de la formule.
- Placer le curseur dans la cellule dans laquelle vous voulez ajouter l'image. Par exemple, dans la capture d'écran ci-dessus, l'image est ajoutée en tant que contenu du deuxième élément d'`Image` en sélectionnant l'élément d'`Image`. Ensuite, sélectionner **Éditer | Insérer | Encoder Fichier Externe**. Dans le dialogue qui apparaît : (i) entrez le chemin de l'image souhaitée (local ou internet), (ii) choisissez *Base 64*, (iii) choisissez *Créer Texte*. L'image sera convertie dans son encodage Base64, et le code encodé sera saisi dans la cellule. Vous pouvez voir le texte encodé en mode Texte. Toutefois, dans le mode Grille, vous ne verrez pas l'encodage mais un rendu d'image dans la cellule *(voir la capture d'écran ci-dessus)*.

Des URL d'image peuvent être données sous les formes suivantes, y compris en tant que chemins relatifs :

- `http://httpbin.org/image/png`
- `file:///c:/LocalPig.png`
- `C:/LocalPig.png`
- `LocalPig.png`

La plupart des formats d'image qui sont généralement utilisés sont pris en charge. Ceux-ci incluent les formats PNG, JPEG, BMP, et les GIF animés. Le format SVG est lu comme document XML ; dans le Mode Grid, l'image est affichée en tant que dernier enfant de l'élément **<svg>**.

Enregistrer un string encodé en Base64 en tant qu'image

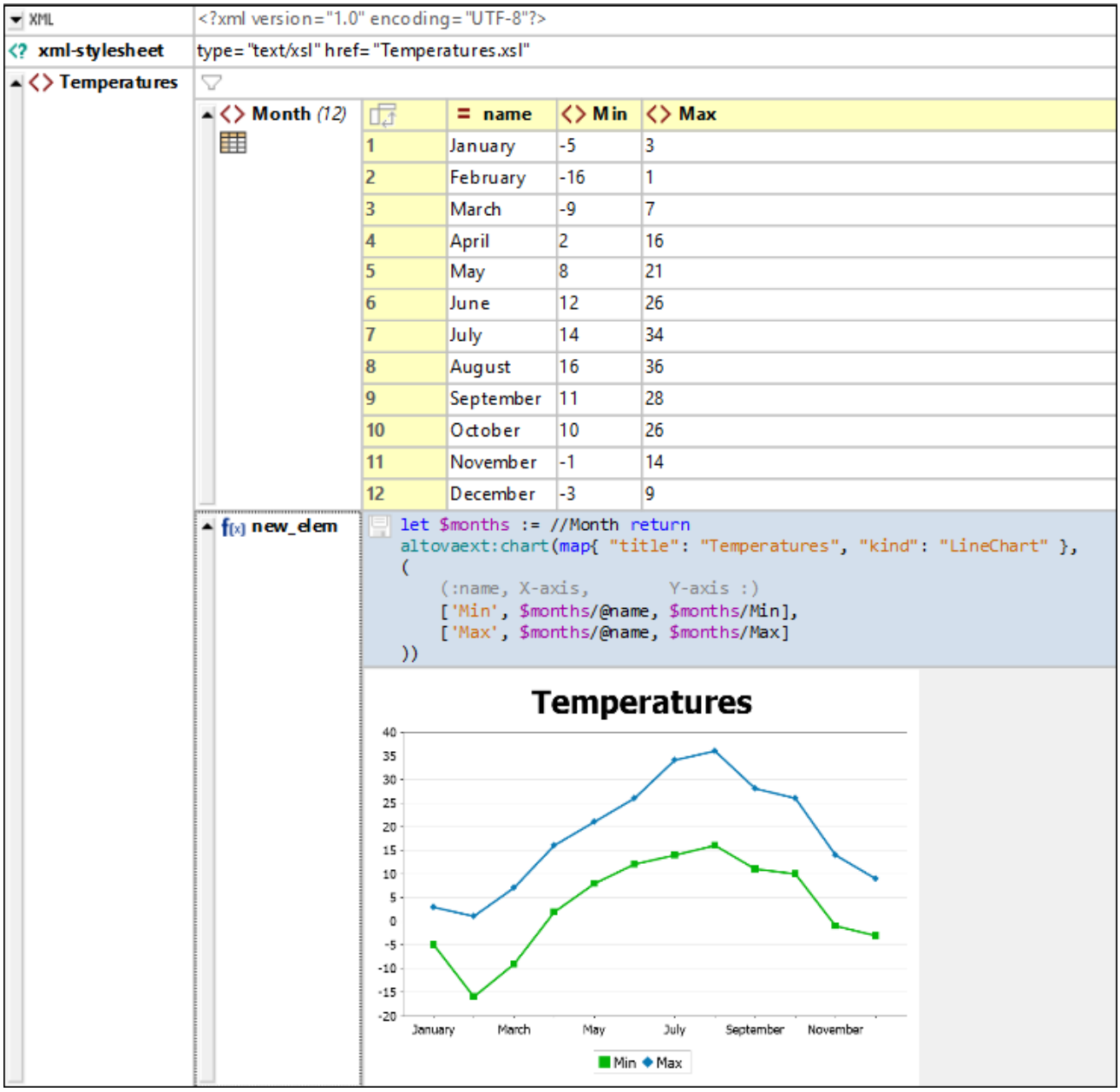
Pour enregistrer un string encodé en Base64 dans son format d'image, cliquez avec la touche de droite sur l'image ou sa cellule et sélectionnez la commande **Enregistrer sous Image**. (Notez que l'image est en fait encodée en tant que string en Base64 dans le fichier actif mais affichée en mode Grille en tant qu'image.) Dans le dialogue qui apparaît, sélectionnez l'emplacement sous lequel vous souhaitez enregistrer l'image et saisissez un nom pour le fichier d'image. L'extension du fichier d'image (`.png`, `.gif`, `.svg`, etc.) sera détectée automatiquement depuis l'encodage Base64 et apparaîtra dans le dialogue Enregistrer. Une fois terminé, cliquer **Enregistrer**.

Cette action peut également être exécutée par le biais de la commande de menu **Editer | Enregistrer sous Image**.

5.4.9 Graphiques

Des graphiques peuvent être créés en Mode XML Grid en utilisant l'extension Altova XPath/XQuery nommée `altovaext: graphique` (*voir capture d'écran ci-dessous*). Cette extension est décrite ci-dessous. Elle est aussi décrite ainsi que d'autres extensions de graphique dans la section Fonctions Graphiques. L'extension `altovaext:chart` affichée dans la capture d'écran ci-dessous est utilisée dans une expression XQuery `Let` qui est définie dans une formule de Mode Grid . Le graphique est affiché en tant qu'une image en-dessous de la formule. (Vous pouvez utiliser la liste de documents XML ci-dessous, ainsi que l'expression XQuery, pour tester la fonction du graphique.)

© 2016-2022 Altova GmbH



Liste du document XML utilisé dans la capture d'écran ci-dessus

```
<?xml version="1.0" encoding="UTF-8"?>
<Températures>
   <Mois nom="Janvier">
      <Min>-5</Min>
      <Max>3</Max>
   </Mois>
   <Mois nom="Février">
      <Min>-16</Min>
```

```
      <Max>1</Max>
   </Mois>
   <Mois nom="Mars">
      <Min>-9</Min>
      <Max>7</Max>
   </Mois>
   <Mois nom="Avril">
      <Min>2</Min>
      <Max>16</Max>
   </Mois>
   <Mois nom="Mai">
      <Min>8</Min>
      <Max>21</Max>
   </Mois>
   <Mois nom="Juin">
      <Min>12</Min>
      <Max>26</Max>
   </Mois>
   <Mois nom="Juillet">
      <Min>14</Min>
      <Max>34</Max>
   </Mois>
   <Mois nom="Août">
      <Min>16</Min>
      <Max>36</Max>
   </Mois>
   <Mois nom="Septembre">
      <Min>11</Min>
      <Max>28</Max>
   </Mois>
   <Mois nom="Octobre">
      <Min>10</Min>
      <Max>26</Max>
   </Mois>
   <Mois nom="Novembre">
      <Min>-1</Min>
      <Max>14</Max>
   </Mois>
   <Mois nom="Décembre">
      <Min>-3</Min>
      <Max>9</Max>
   </Mois>
</Températures>
```

- *Liste des expressions XQuery pour générer le graphique qui est affiché dans la capture d'écran ci-dessus*

```
let $months := //Month return
altovaext:chart(map{ "title":"Temperatures", "kind":"LineChart" },
(
    (:name, X-axis,       Y-axis   :)
    ['Min', $months/@name, $months/Min],
```


© 2016-2022 Altova GmbH

```
        ['Max', $months/@name, $months/Max]
    ))
```

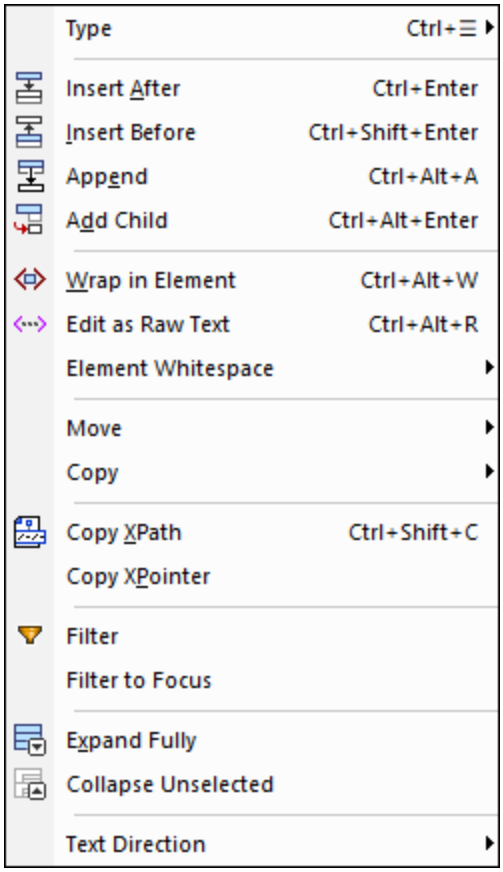
Utiliser l'extension Graphiques Altova

- La fonction d'extension de graphique `altovaext:chart` doit utiliser le préfixe d'espace de noms `altovaext:`.
- La fonction `altovaext:chart` prend deux arguments : (i) information de configuration de graphique et (ii) information de séries de données de graphique.
- L'information de la configuration du graphique est le premier argument du `altovaext:chart`. Il s'agit d'une série non ordonnée de quatre paires de clé-valeur. Ces paires sont pour (i) le titre du graphique (clé est `title`), (ii) le type de graphique, comme des camemberts, des graphiques à lignes, etc (clé est `kind`; voir Fonctions de Graphiques pour les types disponibles), (iii) largeur de graphique en pixels (uniquement entiers ; clé est `width`), et (iv) hauteur de graphique en pixels (uniquement entiers, clé est `height`). Si la largeur ou la hauteur, ou les deux valeurs ne sont pas données, la valeur manquante ou les valeurs sont calculées automatiquement s'appuyant sur les données.
- La série de données de graphique est le second argument de `altovaext:chart`. Chaque série de données est un array de taille 3: (i) le nom de la série, (ii) les valeurs de l'axe x, (ii) les valeurs de l'axe y. Si vous souhaitez créer plusieurs séries (par exemple où chaque série représente une ligne, comme dans l'exemple ci-dessus), créez une séquence de plusieurs arrays. L'exemple ci-dessus contient trois séries de données ; pour les températures minimum, maximum et moyenne. Les données pour les axes X et Y sont référencées depuis la séquence de tous les éléments `Mois`.

Pour plus d'informations, voir la section Fonctions Graphique.

5.4.10 Menu contextuel dans le Mode Grille

Lorsque vous cliquez avec la touche de droite sur un Cellule dans le Mode Grille, un menu contextuel *(capture d'écran ci-dessous)* apparaît qui vous propose des commandes pour le contenu d'édition lié à la Cellule et pour modifier l'affichage. Le menu contextuel peut aussi être accédé en appuyant sur la touche du clavier **Menu**. Les commandes du menu contextuel sont décrites ci-dessous.



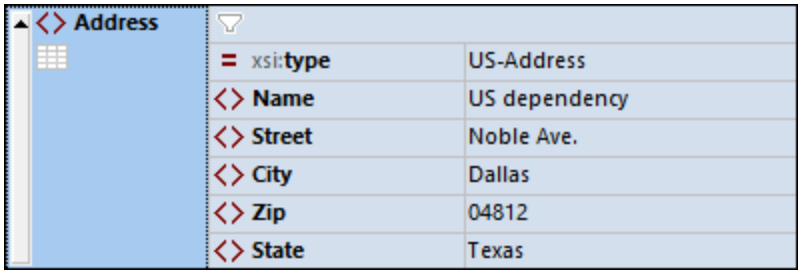
Type
Passer la souris sur la commande **Type** pour afficher un sous-menu dans lequel vous pouvez choisir le type du composant. Vous pouvez aussi directement accéder au sous-menu **Type** avec le Clavier raccourci **Ctrl+ Menu**.

Insérer avant/après, Apposer, Ajouter enfant
Les commandes **Insérer** et **Apposer** ajoutent un item (un élément, par défaut) au même niveau. Le nouvel item est ajouté dans le cas de **Insérer Après** et **Insérer Avant**, respectivement, après et avant l'item sélectionné, et dans le cas d'**Apposer** en tant que le dernier frère de l'item sélectionné. La commande **Ajouter enfant** appose un nouvel item (un élément, par défaut) en tant qu'enfant. Changez le nom du nouvel élément ajouté en double-cliquant sur le nom de la cellule et pour éditer le nom. Changez le type du nœud en cliquant sur l'icône de l'élément (à gauche de son nom), puis en sélectionnant le type de nœud souhaité.

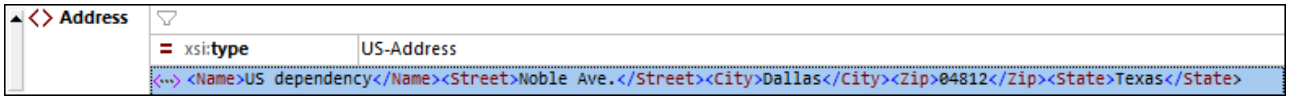
Emballer dans Élément
Un élément parent est attribué à l'item sélectionné avec un élément par défaut, que vous pouvez renommer(en double-cliquant le nom et en l'éditant nœud).

Éditer en tant que Texte brut
Permet d'éditer le contenu du texte de l'item sélectionné en tant que texte brut. Ceci est utile si, par exemple, vous êtes en train d'éditer du contenu complexe tel que du code HTML. Par exemple, la capture d'écran ci-dessous, affiche l'élément `Address` dans le Mode Grille.

© 2016-2022 Altova GmbH



Si vous basculez l'affichage de l'élément **Address** vers le texte brut (*voir la capture d'écran ci-dessous*), la structure de la grille est convertie en un seul item contenant du texte brut, qui peut être entièrement édité dans le Mode Texte.



Espace blanc d'élément
Disponible dans des nœuds d'élément, il ajoute l'attribut **xml:space** à la balise de l'élément et vous donne la valeur à cet attribut que vous choisissez depuis le sous-menu de la commande.

- Toutefois, si vous sélectionnez la valeur *Omettre*, l'attribut **xml:space** ne sera pas ajouté, ce qui a pour conséquence de normaliser l'espace blanc.
- La valeur *Preserve* préserve tous les espaces blancs et désactive pretty printing pour cet élément. Les espaces blancs préservés sont indiqués dans le Mode Grille par une icône points de suspension.
- La valeur *Default* prend la valeur spécifiée pour cet élément, qui peut être utile pour remplacer une valeur **xml:space** héritée d'un élément ancêtre dans le document XML.

Déplacer vers le Haut/Bas/Gauche/Droite
S'il est possible de déplacer une composant vers le haut, le bas, la gauche ou la droite de son emplacement actuel dans la grille, la commande correspondante sera activée. Choisir la commande respective pour effectuer le déplacement.

Copier
Ces commandes peuvent être utilisées pour copier la sélection actuelle. Seules ces options sont activées qui sont permises dans ce composant.

Commande	Description
Copier en tant que texte XML	La sélection actuelle est sérialisée en tant que balisage XML
Copier comme texte séparé par tabulation	La sélection de table actuelle est sérialisée en tant que TSV (valeurs séparées par tab)
Copier en tant qu'Image	L'image actuelle est copiée en tant qu'image

Veuillez noter :

- *Copier en tant qu'Image* copie le string encodé en Base64 de l'image sélectionnée. Si le string est collé dans un document où un string Base64-encoded peut être rendu en tant qu'image (par exemple vers une autre table dans l'Affichage Table), il sera rendu. Autrement, il sera collé en tant que string.
- Pour insérer du texte dans une cellule, copier le texte et le coller dans la cellule.
- Pour importer depuis le fichier, une image par exemple, utiliser la commande **Éditer | Insérer | Fichier externe encodé**. Une image est insérée en tant que string encodé enBase64 , et affiche ce string en tant qu'image dans le Mode JSON Grid. Les formats de fichier d'image qui sont pris en charge pour les importations sont : `PNG`, `JPEG`, `BMP`, `GIF`, `TIFF`.

Copier XPath
Cette commande copie une expression localisatrice XPath 3.1 vers le presse-papier, commençant à la racine du document qui situe le nœud sélectionné.

Copier XPointer
Cette commande copie l'expression XPointer dans le presse-papiers qui situe le nœud sélectionné. Voir **Copy XPointer/JSON-Pointer** pour plus de détails.

Filtre, Filtre pour Focus
La commande **Filtre** est une commande bascule. Elle ajoute un filtre à l'élément sélectionné ou désactive le filtre. Veuillez noter qu'après avoir ajouté un filtre via cette commande, vous devez saisir une expression de filtre ; autrement, le filtre sera supprimé automatiquement. Après avoir ajouté un filtre, vous pouvez le désactiver en sélectionnant la commande une nouvelle fois ou en cliquant sur le filtre. Pour supprimer le filtre, cliquez sur *Supprimer tous les filtres* dans la barre d'outils.

La commande **Filter pour Focus** (i) construit un filtre sur l'élément racine qui contient une expression XPath pour situer le nœud sélectionné, et (ii) filtre l'affichage du Mode Grille pour montrer le nœud sélectionné (et ses descendants) directement sous l'élément racine. Ceci est utile si vous souhaitez focaliser l'affichage uniquement sur le nœud sélectionné. Si vous souhaitez supprimer le filtre, cliquez sur l'icône Filtre dans la cellule du contenu de l'élément racine.

Agrandir complètement
Cette commande est activée si le composant sélectionné ou un de ses descendants est réduit. Elle agrandit le composant et tous les composants du descendant.

Réduire non sélectionné
Cette commande réduit tous les composants sauf le composant sélectionné et ses composants ancêtres.

Direction du Texte
Cette commande est disponible pour les nœuds contenant du texte, et change le sens de la lecture pour commencer soit à gauche ou à droite de la cellule. Ceci est utile lorsque vous utilisez des langues comme l'arabe et le hébreux.

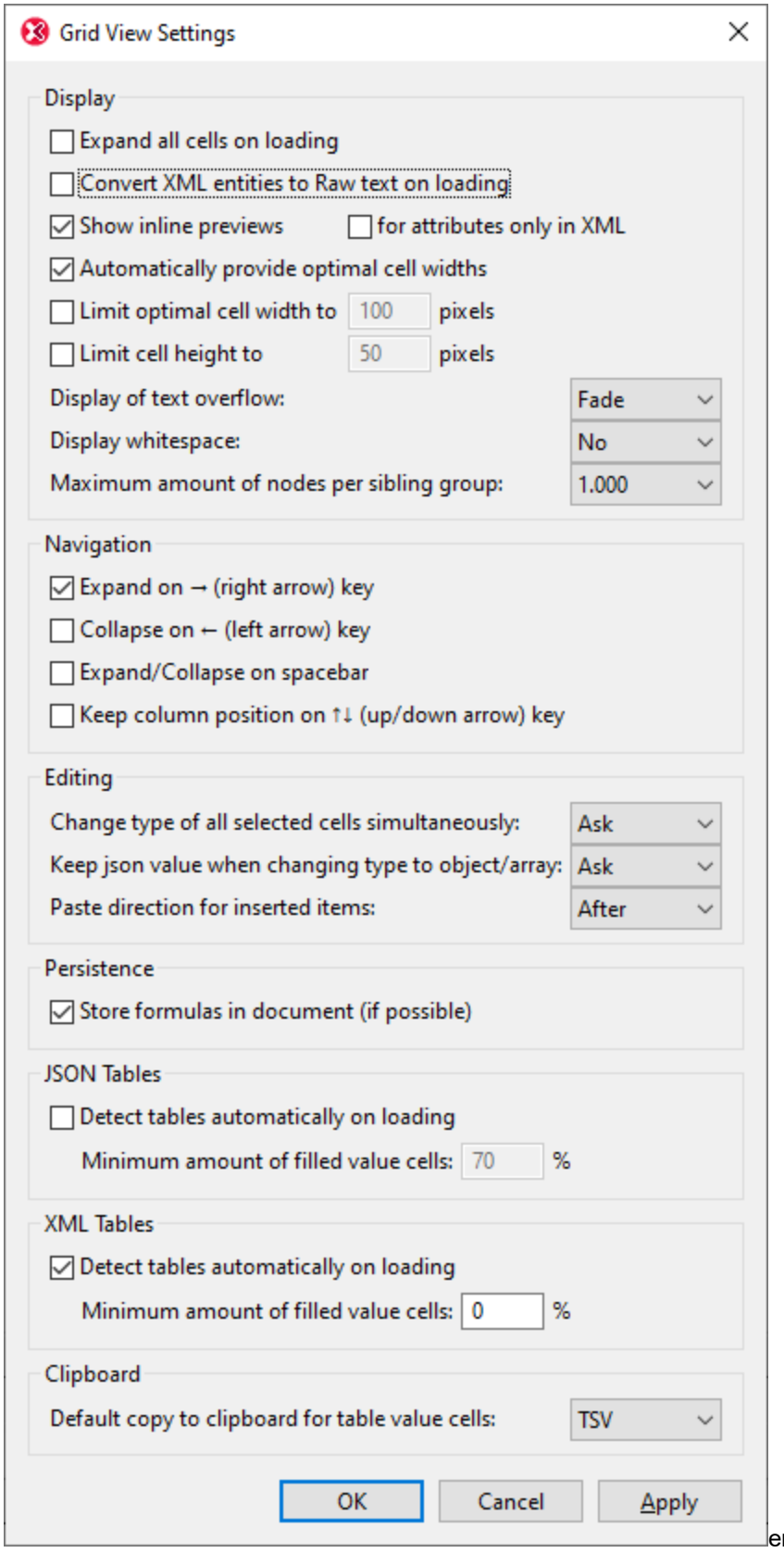
5.4.11 Paramètres du Mode Grilleiew Settings

Les paramètres du Mode Grille sont définis dans le dialogue des Paramètres du Mode Grille *(capture d'écran ci-dessous)*, auquel on accède via l'icône des paramètres dans la Barre d'outils du Mode Grille.

	Les paramètres du Mode Grille dans la Barre d'outils du Mode Grille

© 2016-2022 Altova GmbH

Les paramètres du Mode Grille sont décrits ci-dessous. Notez que ces paramètres s'appliquent à tous les documents du Mode Grille (XML, JSON, DTD).

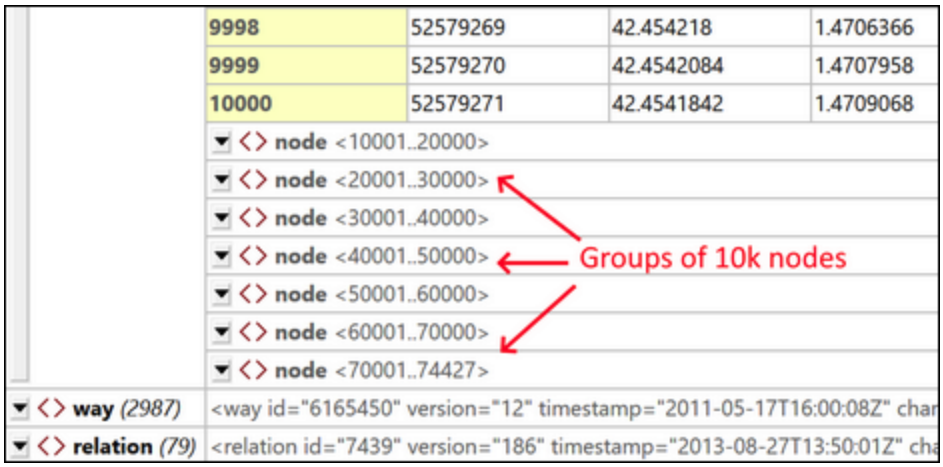


end

© 2016-2022 Altova GmbH

Afficher
Les cases à cocher dans la section *Affichage* sont assez explicites. Veuillez trouver ci-dessous quelques clarifications.

- Si les cellules ne sont pas agrandies lors du chargement, le nœud racine et tous ses descendants sont compressés. Vous devrez étendre chaque nœud au fur et à mesure que vous plongez au cœur du document.
- Si *Convertir les entités XML en texte brut sur chargement* est sélectionné, alors les entités XML seront chargées en Mode Grille comme texte brut de l'entité respective ; elles ne seront pas résolues en leurs représentations de glyphes respectives.
- Si *Afficher aperçus instantanés* n'est pas cochée, alors, au lieu d'un aperçu de la cellule étant affiché, seul le numéro de l'index de l'élément dans la cellule sera affiché. Si des aperçus instantanés sont activés, vous pouvez choisir si vous voulez afficher un aperçu qui contient (i) le contenu et les attributs de l'élément ou (ii) uniquement les attributs. Pour opter pour ce dernier, cochez *pour des attributs uniquement*; pour opter pour le premier, désactivez *pour des attributs uniquement*. Veuillez noter qu'uniquement la première partie du contenu inline d'une cellule sera affichée ; vous pouvez passer sur un onglet de démarrage d'un élément pour voir tout son contenu.
- Lorsque les largeurs optimales sont activées, toute la largeur de la grille est affichée. Pour y parvenir, le texte dans certaines cellules sera renvoyé à la ligne.
- Lorsque du texte dépasse une cellule, ce dépassement peut être affiché soit en tant que texte, soit être indiqué par une ellipse.
- Vous pouvez basculer l'affichage de l'espace blanc dans les cellules de grille on ou off. Un espace est affiché en tant que point centré vertical et un onglet est affiché comme une flèche. Une fin de ligne (end-of-line) est indiquée avec un nouveau saut de ligne à l'intérieur d'une cellule.
- Les nœuds sibling peuvent être affichés en groupes frères de 100 nœuds, 1k ou 10k nœuds *(voir la capture d'écran ci-dessous)*. Ceci est utile pour deux raisons : (i) Si vous souhaitez gagner de l'espace d'affichage dans les fichiers qui ont un grand nombre de nœuds frères, vous pouvez grouper les nœuds frères en grands groupes ; (ii) si vous souhaitez traiter uniquement un sous-ensemble de nœuds frères (gagnant ainsi du temps de traitement), vous pouvez élargir des groupes frères pertinents que vous souhaitez traiter et effectuer l'action de traitement sur ces nœuds uniquement ; par exemple, quand vous utilisez Chercher-et-Remplacer ou quand vous convertissez le type d'un nœud (voir *Édition* ci-dessous).



Navigation
Généralement, vous pouvez utiliser les touches de direction pour naviguer dans la grille. Ces paramètres fournissent des options intelligents pour utiliser les clés.

- *Élargir sur la touche de direction de la droite :* Si un élément de cellule est réduit, appuyer sur la *touche de direction de droite* élargit l'élément dans la cellule. Si l'item de cellule n'est pas réduit, la *touche de direction de droite* vous amène à la prochaine cellule à droite (y compris l'enfant si la prochaine cellule à droite est un enfant). Si l'option n'est pas sélectionnée, la *touche de direction de droite* s'arrête à la cellule réduite. Veuillez noter que la fonction *Étendre avec la touche de direction Flèche droite* ne s'applique pas aux cellules à l'intérieur des tables ; dans les cellules de table, l'action vous amène à la prochaine cellule à droite.
- *Réduire sur la touche de direction de gauche :* Lorsque vous vous déplacez à gauche avec la *touche de direction gauche* alors, à un certain moment, vous devez également déplacer la hiérarchie de document vers le haut. Si cette option est sélectionnée, alors les items qui peuvent être réduits le seront quand vous appuyez sur la *touche de direction de gauche* ; autrement, de tels items ne seront pas réduits bien que le focus basculera vers l'item parent. Veuillez noter que la fonction *Réduire avec la touche de direction de gauche* ne s'applique pas aux cellules des tables ; dans les cellules de table, l'action vous amène à la prochaine cellule à gauche.
- *Élargir/Réduire sur la barre d'espace :* Les fonctions d'espace de barre en tant que bascule pour élargir/réduire un élément. Elle peut donc être utilisée comme touche additionnelle pour naviguer au sein de la grille.
- *Garder la position de la colonne sur les touches de direction Haut/Bas :* Les touches de direction *Haut* et *Bas* vous déplacent, respectivement, vers le haut et le bas dans les cellules de la grille, y compris dans les items parent et enfants - qui sont hiérarchiquement à différents niveaux, et donc dans différentes colonnes. Si cette option est choisie, les niveaux représentés dans les colonnes autres que la colonne actuelle sont ignorés. Ceci fonctionne, par exemple, comme suit. Disons que le curseur se trouve dans la colonne pour l'élément **`subject/course/books/book/title`**. Avec l'option *Garder position de colonne* choisie, vous pouvez utiliser les *touches de direction Haut* et *Bas* pour naviguer uniquement à travers les titres de livres (sans aller dans les colonnes **`book`**, **`books`**, **`course`**, ou **`subject`**, ou toute colonne pour les items descendants de **`Title`**.)

Édition
Les cases à cocher dans la section *Affichage* sont assez explicites. Veuillez trouver ci-dessous quelques clarifications.

- Lorsque vous modifiez le type de cellules multiples sélectionnées, vous avez les options suivantes qui définissent si vous pouvez poursuivre l'action du paramètre : *Toujours, Jamais,* ou *Demander* (pour la décision de l'utilisateur).
- Lorsque vous modifiez un type JSON d'un type atomique à objet ou array, vous avez les options suivantes qui définissent si vous pouvez poursuivre l'action du paramètre : (i) *Demander* (si la valeur du type atomique devrait être maintenue comme valeur d'une paire d'enfants sans nom **`key:value`** ou ignorée), (ii) *Toujours* (maintenir la valeur d'une paire d'enfants sans nom **`key:value`**), (iii) *Jamais* (maintenir la valeur, à savoir ignorer la valeur).
- L'option de la *direction Coller* détermine si une sélection dans le presse-papiers est collée au-dessus ou en dessous de la cellule sélectionnée.

Persistance
Les filtres et formules peuvent être enregistrés dans les méta-données des documents XML et JSON. Les formules sont stockées comme instructions de traitement dans les documents XML et comme commentaires dans certains types de documents JSON. Ces paramètres permettent de spécifier comment procéder avec le stockage des filtres et des formules. Dans les cas des formules, la terminologie fait, *si possible*, référence au fait que les formules ne peuvent pas être enregistrées comme commentaires dans tous les documents JSON (uniquement dans les documents JSON5 et JSONC).

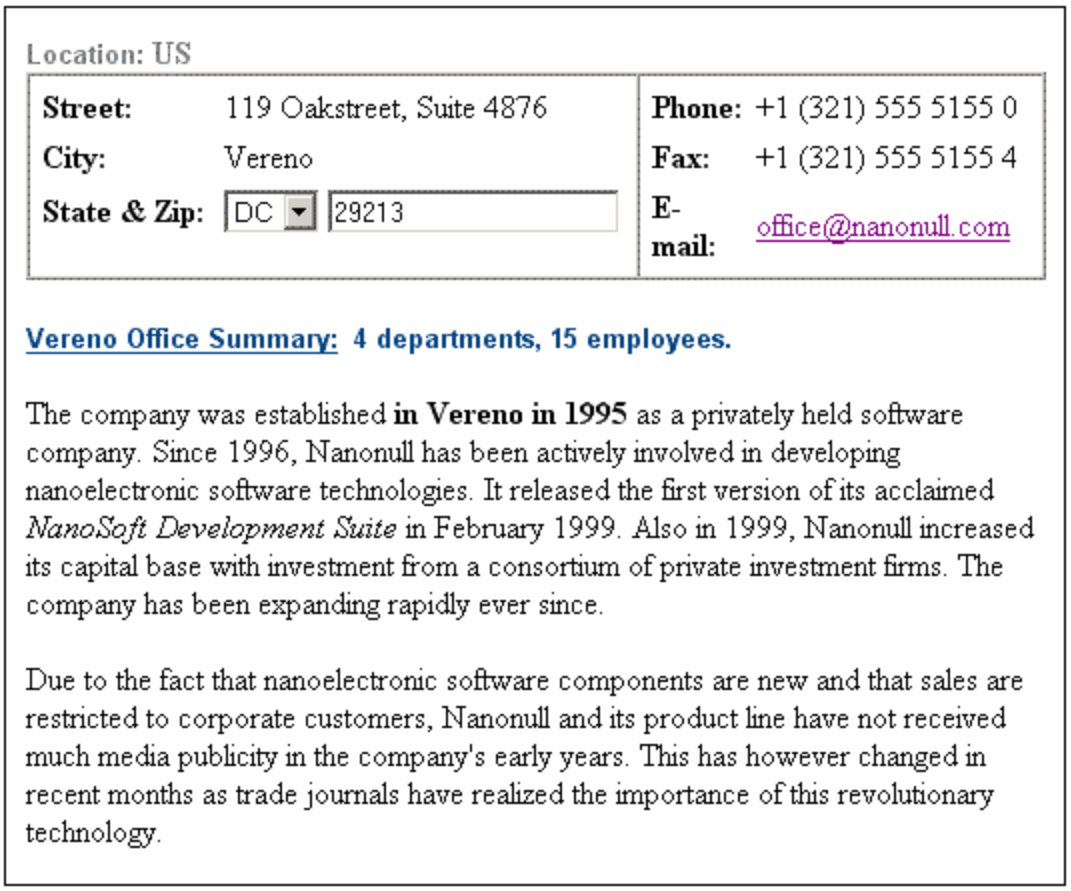
© 2016-2022 Altova GmbH

Tables JSON, tables XML
Si le paramètre pour détecter les tables en Mode Grille est sélectionné automatiquement lors du chargement, vous pouvez sélectionner le pourcentage minimum de cellules de tables remplies qui qualifient les tables à être détectées comme des tables. Si le nombre de cellules de table remplies ne dépasse pas ce niveau, la structure est affichée comme grille normale avec les éléments répétitifs recensés l'un en dessous de l'autre.

Presse-papiers
Vous pouvez aussi choisir si les contenus du presse-papiers doivent être stockés comme valeurs séparées par tabulation (TSV), ou comme XML/JSON (dépendant du type de document). Ceci est une fonction très utile : Si vous souhaitez coller une table depuis le presse-papiers dans un autre document, le paramètre vous permet de choisir si la table copiée est stockée comme TSV ou avec un marquage. (Pour voir la différence, collez une table dans un éditeur de texte après avoir copié la table vers le presse-papiers dans chacun des formats.)

5.5 XML dans le Mode Authentic

Le mode Authentic permet à un utilisateur d'éditer un document XML comme s'il s'agissait d'un document de texte (*capture d'écran ci-dessous*). Les balises XML et tout le texte dépourvu de contenu peuvent être dissimulés de la personne éditant le document. Cela peut être une fonction utile pour les personnes qui ne sont pas familières avec XML, leur permettant de créer des documents XML valides tout en se concentrant sur le contenu du document.



Le Mode Authentic d'un document est activé lorsqu'une StyleVision Power Stylesheet (SPS) est attribuée à un document XML. Une SPS est basée sur la même source de schéma que celle sur laquelle se base le document XML, et elle définit la structure du document XML. La SPS définit aussi la mise en page et le formatage du document dans le Mode Authentic. Par exemple, dans le document affiché dans la capture d'écran ci-dessus, les fonctions de formatage et d'édition suivantes sont utilisées :

- Paragraphe et autre formatage de blocs
- Structures de table
- Formatage de texte, comme des couleurs et familles de police
- Listes de choix (voir les champs État et Code postal) permettent à l'utilisateur d'effectuer un choix à partir d'un groupe de choix valides qui peuvent être prélevés d'énumérations de schémas (voir l'exemple ci-dessus)

© 2016-2022 Altova GmbH

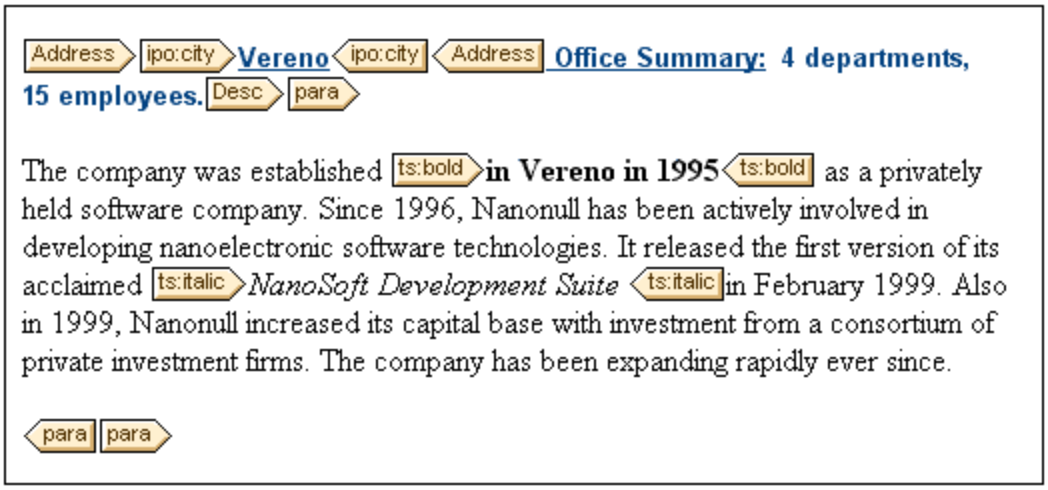
- Des informations supplémentaires peuvent être calculées à partir des données contenues dans le document puis présentées (dans l'exemple ci-dessus, les détails du paragraphe office summary n'ont pas été saisis par l'utilisateur mais ont été calculés à partir des données contenues dans le document)

Les SPS sont créées spécifiquement pour consulter et éditer des documents XML dans le Mode Authentic et pour générer des sorties standard (comme des documents HTML, PDF, RTF et Word 2007) à partir d'un XML. Les SPS sont créées avec Altova StyleVision.

Éditer la structure du document

Les nœuds valides peuvent être ajoutés dans le document à tout moment en sélectionnant un emplacement, puis en ajoutant le nœud requis par le biais des assistants à la saisie (Éléments et Attributs) ou le menu contextuel. Les nœuds disponibles dans n'importe quel emplacement sont limités aux nœuds qui peuvent être ajoutés de manière valide en tant que frères ou enfants dans les emplacements choisis. Par exemple, lorsque le curseur est situé à l'intérieur d'un paragraphe, vous pouvez ajouter un autre paragraphe si cela est autorisé par le schéma.

Lors de l'édition de la structure d'un document XML dans le Mode Authentic, il peut s'avérer utile de visualiser la balise du document. Les balises peuvent donc être activées en tant que balises jaunes (*capture d'écran ci-dessous*) à l'aide de la commande **Authentic | Afficher grande balise** (ou l'icône de la barre d'outils correspondante).



Éditer du contenu

Les contenus sont créés et édités en les saisissant dans les nœuds du document. Les entités et les sections CDATA peuvent être ajoutées par le biais du menu contextuel (les entités peuvent aussi être ajoutées par le biais de l'assistant à la saisie Entités).

En savoir plus à propos de l'édition dans le Mode Authentic

Pour plus de détails concernant comment éditer dans le Mode Authentic, voir la section Mode Authentic.

5.6 Assistants à la saisie (Mode Texte, Mode Authentic)

Pour les documents XML dans le Mode Texte et le Mode Authentic, il existe trois assistants de saisie : Éléments, Attributs et Entités. Lorsqu'un élément est ajouté à l'aide de l'assistant à la saisie Éléments, il peut être ajouté simultanément avec des éléments enfants obligatoires, des attributs obligatoires, tous les éléments enfants ou aucun élément ou attribut enfant, conformément aux paramètres respectifs contenus dans Éditer la section du dialogue des Options. Lorsque des attributs vides sont ajoutés, ils sont ajoutés avec des guillemets.

Veuillez noter que dans les modes différents, les assistants à la saisie sont conçus de manière différente, conformément aux fonctions du mode concerné.

Assistant à la saisie Éléments

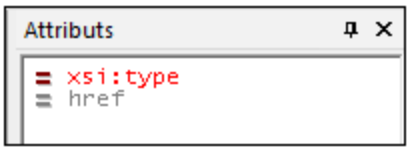
Veuillez noter les points suivants :

- *Mode Texte :* les éléments sont insérés au niveau du point d'insertion du curseur. Les éléments non utilisés sont affichés en rouge, les éléments utilisés en gris. Les éléments obligatoires sont indiqués avec un point d'exclamation "**!**" situé devant le nom de l'élément.
- *Mode Authentic :* les éléments peuvent être insérés avant, après, ou dans l'élément sélectionné. De plus, une arborescence de document vous montre l'emplacement de l'élément sélectionné actuellement dans la structure d'arborescence du document. Pour plus de détails concernant l'édition dans le Mode Authentic, voir la section Mode Authentic.

Assistant à la saisie Attributs

Veuillez noter les points suivants :

- *Mode Texte :* lorsque le curseur est placé dans la balise de démarrage d'un élément et après un espace, les attributs déclarés pour cet élément deviennent visibles. Les éléments non attributs sont affichés en rouge, les éléments utilisés en gris. Les éléments obligatoires sont indiqués avec un point d'exclamation "**!**" situé devant le nom de l' attribut.

  Pour insérer un attribut, double-cliquez sur l'attribut souhaité. Il sera inséré au niveau du point du curseur et sera accompagné d'un signe "égal à" et de guillemets pour délimiter la valeur d'attribut. Le curseur sera placé entre les guillemets, vous pouvez donc directement commencer la saisie de la valeur d'attribut.
- *Mode Authentic :* lorsqu'un élément est sélectionné, les attributs déclarés pour cet élément deviennent visibles. Saisissez la valeur de l'attribut dans l'assistant à la saisie.

Assistant à la saisie Entités

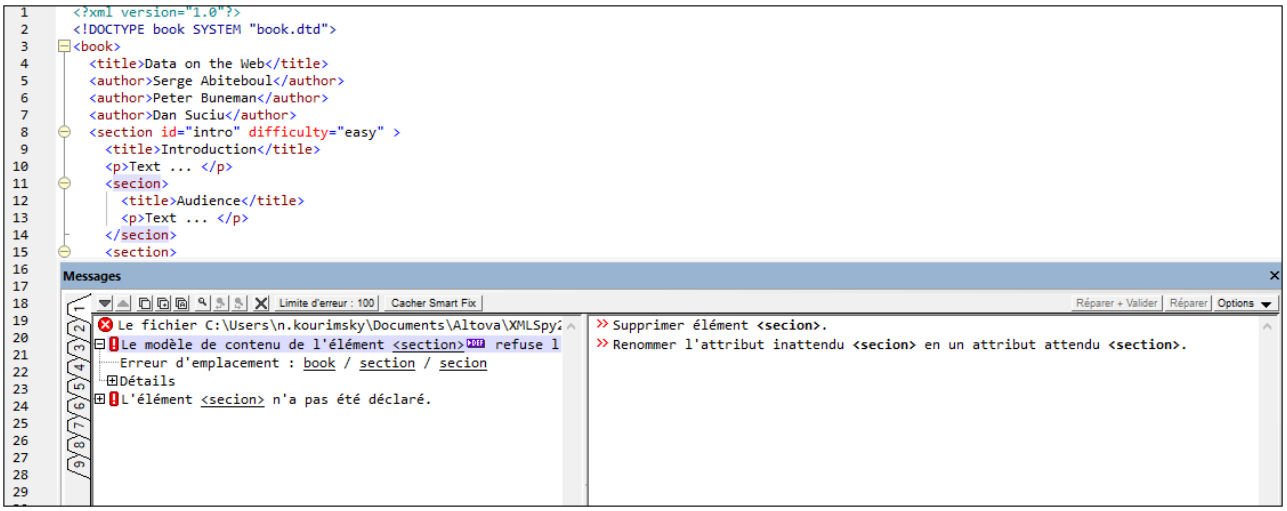
Toute entité parsée ou non qui est déclarée inline (dans le document XML) ou dans un DTD externe, est affichée dans l'assistant à la saisie Entités. Dans les trois modes (Texte, Grille et Authentic), il est possible d'insérer une entité au niveau du point d'insertion du curseur en la double-cliquant. Dans le Mode Grille, les entités sont affichées dans les onglets Apposer, Insérer et ajouter enfant.

© 2016-2022 Altova GmbH

Veuillez noter que si vous ajoutez une entité interne, vous devrez enregistrer et rouvrir votre document avant que l'entité apparaisse dans l'assistant à la saisie Entités.

5.7 Valider un document XML

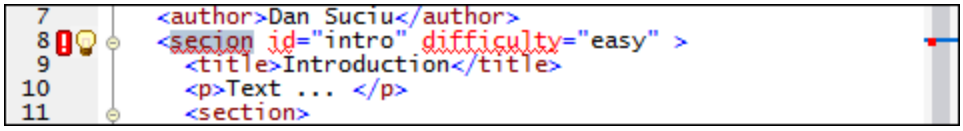
La commande **XML** | **Valider (F8)** valide un document XML par rapport a un DTD, Schéma XML ou un autre schéma associé. Si un document est valide, un message de validation réussi est affiché dans la fenêtre Messages. Dans le cas contraire, les causes de l'erreur sont affichées dans le panneau de gauche (*voir capture d'écran ci-dessous*). Si une cause est sélectionnée dans le panneau de gauche, des smart fixes, si disponibles, sont affichés dans le panneau de droite. Les suggestions de smart fix sont basées sur des information dans le schéma associé. Pour appliquer un smart fix, soit le (i) double-cliquer, soit (ii) le sélectionner et cliquer soit sur l'option **Fix** ou **Fix + Valider** (*voir capture d'écran ci-dessous*).



Indicateurs d'erreur et smart fixes

Indicateurs d'erreur

Dans le Mode Texte, vous trouverez deux indicateurs supplémentaires d'une erreur de validation (*voir capture d'écran ci-dessous*): (i) un point d'exclamation rouge dans la marge numérotée et (ii) un carré de marquage rouge dans la barre de défilement (à la droite de la fenêtre).

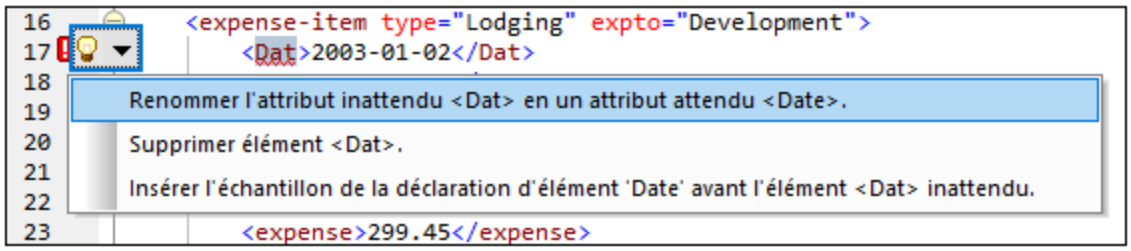


Veuillez noter que le carré rouge apparaît du côté gauche de la barre de défilement (située du côté droit de la fenêtre ; *voir capture d'écran ci-dessus*). Nous mentionnons ce fait ici parce que la barre de défilement affiche deux autres types de carrés de marquage: (i) pour les occurrences de texte marqué (brun, côté gauche de la barre de défilement ; *voir Parcourir le document*) ; (ii) Trouver des occurrences (brun, côté droit de la barre de défilement ; *voir Chercher et remplacer*).

Smart fixes

Si un smart fix est disponible pour une erreur, une ampoule s'affiche sur la ligne qui génère l'erreur (*voir capture d'écran ci-dessous*). Lorsque vous placez la souris sur l'icône, une fenêtre popup apparaît qui recense les smart fixes disponibles (*voir capture d'écran ci-dessous*). Sélectionner un fix pour l'appliquer immédiatement.

© 2016-2022 Altova GmbH



Veuillez noter les points suivants :

- Les indicateurs d'erreur de validation et les smart fixes sont disponibles pour les types de document qui peuvent être validés dans XMLSpy, par exemple des documents JSON.
- Les indicateurs d'erreur de validation et les smart fixes décrits ci-dessus sont réinitialisés uniquement lorsque la commande **XML** | **Valider (F8)** est exécutée ; ils ne sont pas mis à jour en arrière-plan. Donc, si vous corrigez une erreur, vous devrez exécuter la commande **Valider (F8)** à nouveau pour vous assurer que l'erreur a bien été réparée.

Pour plus d'informations concernant la validation d'un document XML, voir la description de la commande **Valider**.

Valider sur Édition

Quand le mode *Valider sur Édition* est activé, des vérifications de bonne formation et de validation sont effectuées pendant que vous modifiez un document en mode Texte (et aussi dans le mode JSON Grid). Pour que la validation soit effectuée (en plus des contrôles de bonne formation), un DTD ou un schéma XML doit être attribué au document XML (un schéma JSON doit être attribué à un document JSON). Les erreurs sont indiquées par des indicateurs d'erreur (voir ci-dessus) dans la marge de gauche et sur les lignes contenant les erreurs.

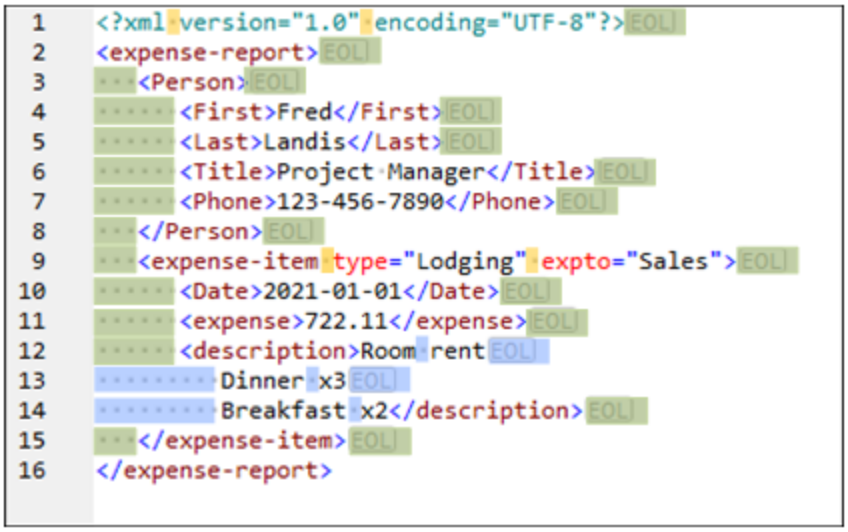
Le mode *Valider sur Édition* peut être activé ou désactivé soit (i) via la commande de menu **XML | Valider sur Édition**, (ii) via la touche de la barre d'outils **Valider sur Édition**, ou (iii) via l'option *Sur Édition* des Paramètres de validation du dialogue Options.

5.8 Whitespace

Les caractères d'espace blanc sont les espaces, l'onglet, le retour de chariot et le saut de ligne. Vous pouvez activer la bascule sur l'affichage des marqueurs d'espace blanc (marqueurs d'espace, onglet et end-of-line EOL) dans le dialogue des paramètres du Mode texte (**Mode | Paramètres Mode texte**).

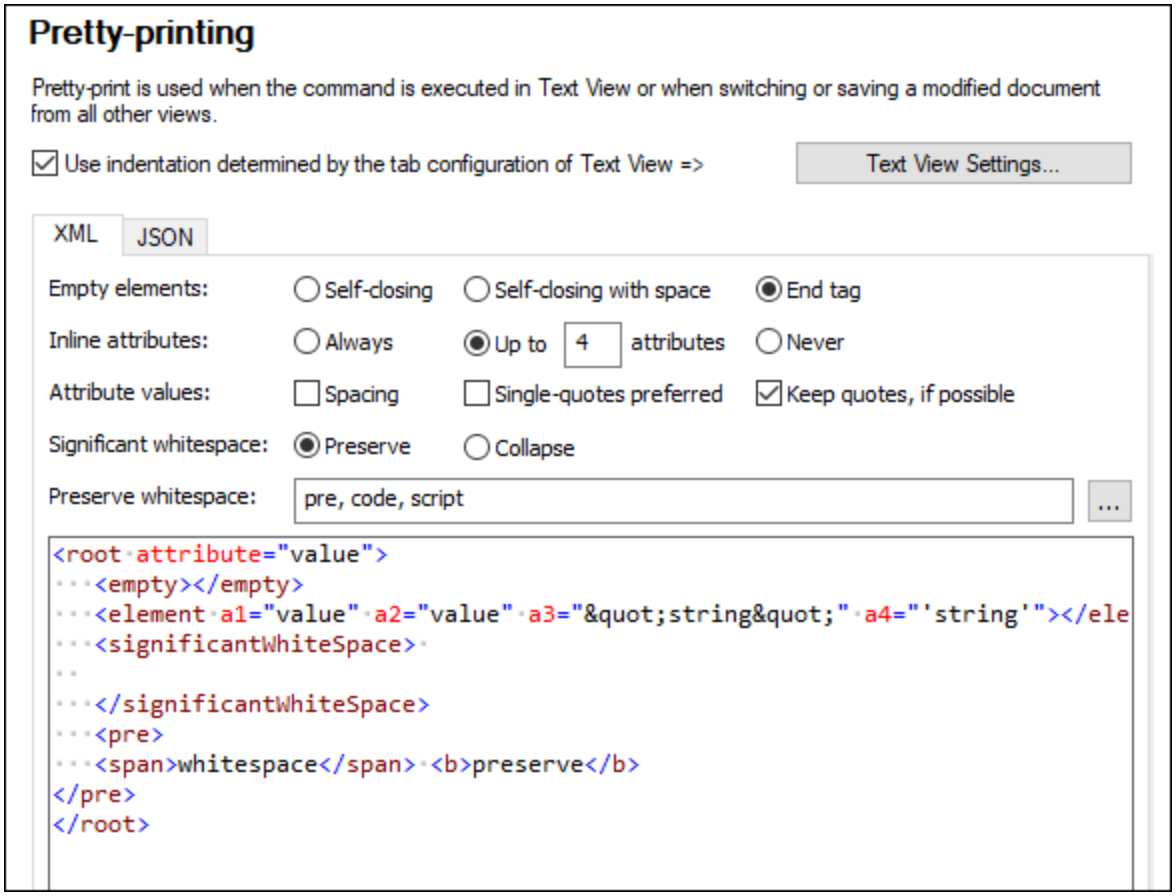
Dans un document XML, les caractères d'espace blanc apparaissent pour les raisons suivantes :

- Pour des raisons de syntaxe XML, normalement pour délimiter les constructions XML. Un tel espace blanc est marqué en jaune dans la capture d'écran ci-dessous.
- Un espace blanc significatif, qui apparait à l'intérieur d'un élément, attribut ou instruction de traitement ne doit pas être ignoré car il a une signification. Il est marqué en bleu dans la capture d'écran ci-dessous.
- Un espace blanc insignifiant apparait entre deux éléments qui n'ont pas de nœuds texte frère. Un espace blanc insignifiant apparaitrait donc uniquement dans des éléments qui ne sont pas du contenu mixte. Il est normalement utilisé à des fins de formatage et n'a aucune signification. Un tel espace blanc insignifiant est marqué en vert dans la capture d'écran ci-dessous.



Dans XMLSpy, un espace blanc est ajouté quand vous vous trouvez en mode pretty-print avec votre document (**Éditer | Pretty-Print**). L'action pretty-print ajoute un espace blanc insignifiant afin de formater un document de telle manière que la structure du document soit affichée clairement. Pretty-printing peut aussi réduire l'espace blanc significatif dépendant des options qui sont actuellement définies pour l'action pretty-printing (*voir la capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Dans les options de pretty-printing, les paramètres suivants affectent la façon dont l'espace blanc est géré :

- Un *espace blanc significatif* peut être préservé ou réduit. Si cette option est définie pour être *réduite*, vous pouvez néanmoins préserver un espace blanc dans des éléments spécifiques en ajoutant ces éléments à la liste *Préserver l'espace blanc*.
- L'option *Préserver l'espace blanc* vous permet de créer une liste d'éléments au sein de laquelle tout espace blanc (significatif et insignifiant) est préservé.
- Si un espace blanc significatif existe dans un élément vide, il sera supprimé si l'*espace blanc significatif* a été défini pour être *réduit*. Le paramètre de l'option *Éléments vides* déterminerait comment l'élément vide est affiché quand son espace blanc significatif a été supprimé.

Note : quand vous changez de mode entre les Modes Texte et Grille, tout changement qui résulte de pretty-printing sera maintenu. Dans le cas d'un changement vers/depuis un autre mode, les changements ne s'appliqueront pas.

5.9 Insérer des fragments XML

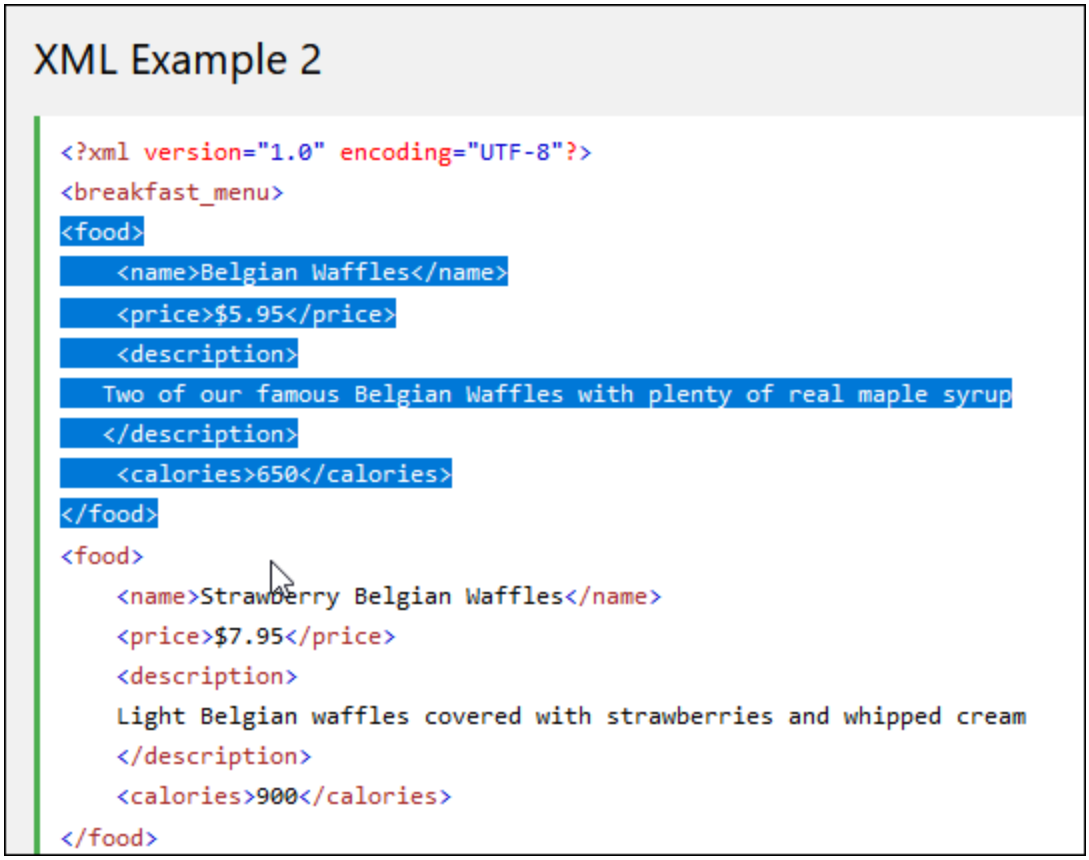
Vous pouvez insérer des fragments XML d'autres applications et pages web. Ces fragments peuvent être insérés des deux manières suivantes :

- En utilisant glisser-déposer dans le Mode Texte ou le Mode Grille. Si vous glissez-déposez dans le Mode Grille, l'information intelligente disponible dans les superpositions pour glisser peut vous aider à décider où vous souhaitez déposer le fragment.
- En utilisant copier-coller dans le Mode Texte ou le Mode Grille.

Exemple

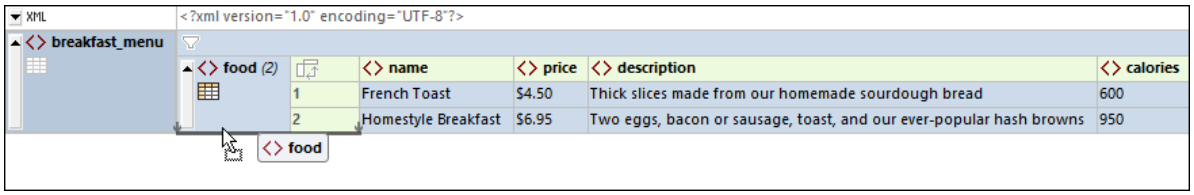
L'exemple suivant montre comment un fragment peut être ajouté rapidement et dans le bon emplacement d'un document XML.

1. Le fragment en surbrillance ci-dessous (du tutoriel XML sur w3schools.com) est sélectionné. Il s'agit d'un élément nommé **food** qui contient un nombre d'éléments enfant.

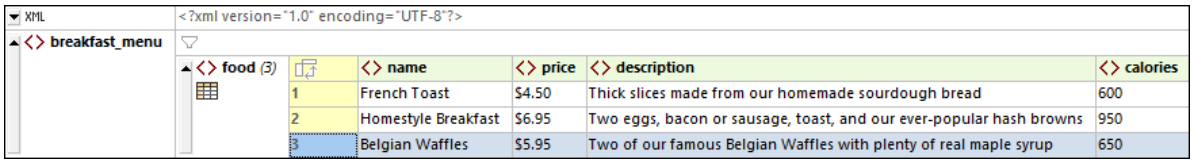


2. La capture d'écran ci-dessous affiche le Mode Grille d'un document XML, où deux éléments **food** sont affichés comme lignes d'une table. Lorsque le fragment de la page web est glissé dans la table **food**, une superposition pour glisser apparait contenant des informations comme quoi le fragment XML sera déposé en tant qu'élément **food** dans la table **food** en tant que sa dernière ligne.

© 2016-2022 Altova GmbH



3. Lorsque le fragment est déposé, il est placé exactement où il doit être placé—en tant que dernier élément enfant **`food`** de **`breakfast_menu`** (*voir la capture d'écran ci-dessous*).



5.10 Traiter un document avec XSLT et XQuery

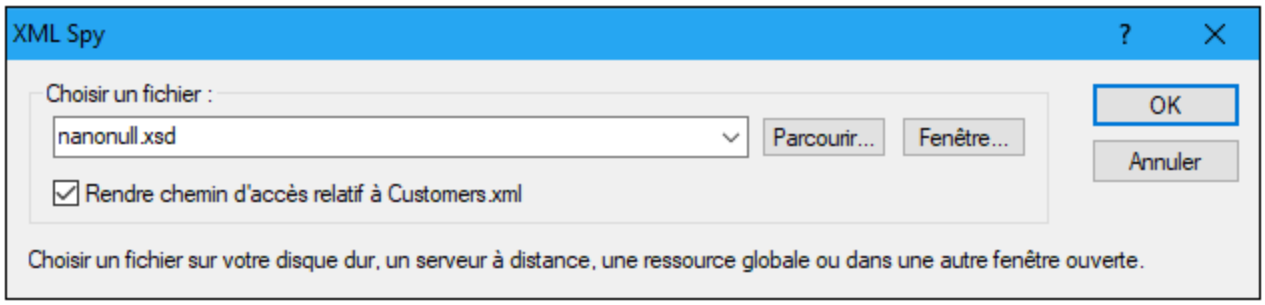
Les Documents XML peuvent être traités avec des documents XSLT ou XQuery pour produire des documents de sortie. XMLSpy contient des processeurs XSLT 1.0, XSLT 2.0, XSLT 3.0, XQuery 1.0 et XQuery 3.0 intégrés. Les fonctions suivantes concernant le traitement sont disponibles dans la GUI :

- Attribuer des feuilles de style XSLT
- Aller au XSLT
- Paramètres XSLT et variables XQuery
- Transformations XSLT
- Exécutions XQuery
- Automatiser les tâches XML avec RaptorXML

Attribuer les feuilles de style XSLT

Vous pouvez attribuer une feuille de style XSLT à un document XML via la commande **XSL/XQuery | Assigner XSL** (chercher le fichier dans le dialogue (*capture d'écran ci-dessous*) qui s'affiche). L'assignation est saisie dans le document XML en tant qu'instruction de traitement (PI) et dont la cible XSLT standard est définie par le W3C: `xml-stylesheet`. Cette assignation est utilisée lorsqu'une transformation XSLT est invoquée (**XSL/XQuery | Transformation XSL**).

De plus, une feuille de style XSLT-pour-FO peut être assignée avec la commande **XSL/XQuery | Assigner XSL:FO** (le fichier dans le dialogue (*capture d'écran ci-dessous*) qui s'affiche). assignation est saisie dans le document XML en tant qu'instruction de traitement (IT) et dont la cible définie par Altova est : `altova_xslfo`. Cette assignation est utilisée lorsqu'une transformation XSLT-pour-FO est invoquée (**XSL/XQuery | Transformation XS:FO**).



Vous pouvez également sélectionner une ressource globale pour spécifier le fichier XSLT. Une ressource globale est un alias pour un fichier ou un dossier. Le fichier ou le dossier cible peut être modifié dans le cadre de la GUI en modifiant la configuration active de la ressource globale (via la commande de menu **Outils | Configuration active**). Les ressources globales permettent donc de faire passer le fichier XSLT assigné de l'une à l'autre ce qui peut être utile pour des tests. Pour plus de détails concernant l'utilisation des ressources globales, consulter la section Ressources globales Altova.

S'il existe une des assignations précédentes utilisant une de ces cibles PI, vous devrez choisir si vous souhaitez écraser l'assignation existante.

Aller au XSLT

La commande **XSL/XQuery | Aller au XSL** ouvre le fichier XSLT qui a été assigné au document XML.

© 2016-2022 Altova GmbH

Paramètres XSLT et variables XQuery

Les paramètres XSLT et les variables XQuery peuvent être définis, édités et supprimés dans le dialogue qui apparaît après avoir cliqué sur la commande **XSL/XQuery | Paramètres XSLT / Variables XQuery**. Les valeurs de paramètres/variables définies ici sont utilisées pour toutes les transformations XSLT et les exécutions XQuery dans XMLSpy. Néanmoins, ces valeurs ne seront pas passées à des processeurs externes comme un MSXML. Pour plus de détails concernant l'utilisation de cette fonction, voir la section Référence d'utilisateur.

Transformations XSLT

Deux types de transformation XSLT sont disponibles :

- Transformation XSLT standard (**XSL/XQuery | Transformation XSL**) : la sortie de la transformation est affichée dans une nouvelle fenêtre ou, si spécifié dans la feuille de style, elle est enregistrée dans un emplacement de fichier. Le processeur utilisé pour la transformation est spécifié dans l'onglet XSL du dialogue Options (**Outils | Options**).
- Transformation XSL-pour-FO (**XSL/XQuery | Transformation XSL-FO**) : le document XML est transformé en PDF dans un processus en deux étapes. Dans la première étape, le document XML est transformé en un document FO grâce au processeur XSLT spécifié dans l'onglet XSL du dialogue Options (**Outils | Options**) ; veuillez noter que vous pouvez aussi sélectionner (situé en bas de l'onglet) le processeur XSLT qui accompagne certains Processeurs FO comme le FOP. Dans la seconde étape, le document FO est traité par le processeur FO spécifié dans l'onglet XSL du dialogue Options (**Outils | Options**) pour produire une sortie PDF.

Note : Un document FO (il s'agit là d'un type particulier de document XML) peut être transformé en un PDF en cliquant sur la commande de transformation XSL:FO. Si le document de source est un document FO, la seconde étape du processus en deux étapes pour cette commande sera exécutée directement.

Exécutions XQuery

Un document XQuery peut être exécuté dans le document XML actif en cliquant sur la commande **XSL/XQuery | Exécution XQuery**. Vous serez invité à exécuter le fichier XQuery, et le document de résultat est affiché dans une nouvelle fenêtre dans la GUI.

Automatiser les tâches XML avec RaptorXML

Altova RaptorXML est une application qui propose une validation XML, des transformations XSLT et des exécutions XQuery. Elle peut être utilisée depuis la ligne de commande, par le biais d'une interface COM, dans des programmes Java et dans des applications .NET. Des tâches comme des transformations XSLT peuvent donc être automatisées grâce à l'utilisation de RaptorXML. Par exemple, vous pouvez créer un fichier batch qui appelle RaptorXML pour transformer un ensemble de documents. Voir la documentation RaptorXML pour plus de détails.

5.11 Polices PDF

Comment le programme de formatage et le PDF Viewer utilisent les polices
Le programme de formatage (par exemple, FOP) crée le PDF et le PDF Viewer (généralement Adobe Reader d'Adobe) le lit.

Afin de pouvoir procéder à la mise en page du PDF, le programme de formatage doit connaître les détails des polices d'écriture utilisées dans le document, en particulier les largeurs de tous les glyphes utilisés. Il lui faut cette information pour calculer les longueurs de ligne, la ponctuation, la justification, etc. Ces informations sont appelées Métriques des polices et elles sont stockées avec chaque police de caractères. Certains programmes de formatages peuvent lire les métriques directement depuis le dossier de polices du système. D'autres (comme le FOP) nécessitent les métriques dans un format spécial qu'il peut comprendre. Lorsque les métriques d'une police sont disponibles pour le programme de formatage, celui-ci peut mettre en page correctement le PDF. Vous devez vous assurer que les fichiers de métriques de police de toutes les polices d'écriture que vous utilisez dans votre document sont disponibles pour le programme de formatage que vous utilisez.

Le programme de formatage peut soit référencer une police d'écriture ou l'intégrer dans le fichier PDF. Si la police est référencée, le PDF Viewer (par exemple, Adobe Reader) cherchera généralement d'abord cette police dans son propre dossier de ressource de polices (qui contient les 14 polices standard), puis, il élargira sa recherche au dossier de polices du système. Si la police est disponible, elle sera utilisée lorsque le PDF sera affiché. Dans le cas contraire, le Viewer utilisera une alternative depuis son dossier de ressources ou bien il générera une erreur. Une police alternative peut avoir des métriques différentes et pourrait donc générer des erreurs d'affichage.

Si le programme de formatage intègre une police dans le fichier PDF, le PDF Viewer utilisera la police intégrée. Le programme de formatage peut intégrer un ensemble de fontes ou de polices entier ou bien seulement un sous-ensemble contenant les glyphes utilisés dans le document. Ce facteur influe sur la taille du fichier PDF et entraîne éventuellement des problèmes de droits d'auteur (copyright) concernant l'utilisation de la police d'écriture (voir note ci-dessous). Vous devriez pouvoir influer sur le choix entre ces deux options en configurant les options de votre programme de formatage.

XMLSpy et polices PDF

Dans XMLSpy, un document PDF est généré depuis un document XSL-FO (ci-après nommé document FO) en traitant le document XSL-FO avec un processeur FO externe comme le FOP. (Dans le dialogue d'Options, vous pouvez spécifier l'emplacement du processeur FO. Cela permet de lancer le traitement FO depuis la GUI XMLSpy.)

Le document XSL-FO lui-même est généré en traitant un document XML avec une feuille de style XSLT. (Vous pouvez utiliser soit le processeur XSLT d'Altova (qui est intégré dans XMLSpy) ou un moteur XSLT externe.)

Le formatage du document PDF, y compris les propriétés de police du texte entier, est spécifié dans le document XSL-FO. Si le programme de formatage que vous utilisez peut lire les métriques de la police d'écriture désirée directement depuis la police, tout ce qu'il vous restera à faire sera de configurer le programme de formatage pour accéder à la police. Si, néanmoins, vous utilisez FOP en tant que votre programme de formatage, vous devrez lui fournir les fichiers de métrique corrects pour les polices différentes des 14 polices standard.

© 2016-2022 Altova GmbH

Rendre les polices disponibles pour le programme de formatage

La plupart des programmes de formatages (y compris FOP) disposent d'ores et déjà des 14 polices standard. Il est important de connaître les noms par lesquels le programme de formatage détecte ces polices de manière à ce qui vous puissiez les indiquer correctement au programme de formatage. Il s'agit de la prise en charge de base des polices d'écriture fournit par le programme de formatages. Vous pouvez, néanmoins, augmenter le nombre de polices disponibles dans le programme de formatage en exécutant quelques étapes particulières au programme de formatage que vous utilisez. Ces étapes nécessaires pour FOP sont indiquées ci-dessous.

Procédure générale nécessaire pour configurer une prise en charge supplémentaire dans FOP

Afin de rendre les polices supplémentaires disponibles dans FOP, veuillez procéder comme suit :

1. Générer un fichier de métrique de police pour la police désirée à partir des fichiers de police PostScript ou TrueType. FOP propose des outils PFM Reader et TTF Reader pour convertir des polices PostScript et TrueType, respectivement, en un fichier de métriques de police XML. Pour plus de détails concernant la procédure, voir la page FOP: polices.
2. Définir le fichier de configuration FOP pour utiliser les fichiers de métrique de police désiré. Pour ce faire, vous pouvez saisir des informations concernant les fichiers de police dans un fichier de configuration FOP. Voir FOP: polices.
3. Dans le fichier `fop.bat`, modifier la dernière ligne :

   ```
   "%JAVACMD%" […] org.apache.fop.cli.Main %FOP_CMD_LINE_ARGS%
   ```

   pour inclure l'emplacement du fichier de configuration :

   ```
   "%JAVACMD%" […] org.apache.fop.cli.Main %FOP_CMD_LINE_ARGS% -c conf\fop.xconf
   ```

Une fois que les fichiers de métrique sont inscrits dans FOP (dans un fichier de configuration FOP) et que le programme d'exécution FOP a été défini pour lire le fichier de configuration, les polices d'écriture supplémentaires sont disponibles pour la création de PDF.

Définir le fichier de configuration FOP

Le fichier de configuration FOP est nommé `fop.xconf` et se trouve dans le dossier `conf` dans le dossier d'installation FOP. Ce fichier, qui est un document XML, doit être édité de manière à ce que FOP lise les fichiers de métrique de police correctement. Pour chaque police que vous souhaitez rendre avec FOP, ajouter un nouvel élément `font` à l'emplacement marqué par l'espace réservé élément `font` dans le document :

```
<font metrics-url="arial.xml" kerning="yes" embed-url="arial.ttf">
          <font-triplet name="Arial" style="normal" weight="normal"/>
          <font-triplet name="ArialMT" style="normal" weight="normal"/>
</font>
```

Dans l'exemple ci-dessus,

`arial.xml`	est l'URL du fichier métrique ; il vaut mieux utiliser un chemin absolu.
`arial.ttf`	est le nom du fichier TTF (généralement situé dans `%WINDIR%\Fonts`).

`Arial`	spécifie que les métriques ci-dessus et les fichiers TTF seront utilisés si la famille de police est définie en tant qu'`Arial`.
`style="normal"`	spécifie que les métriques ci-dessus et les fichiers TTF seront utilisés si le style de police est défini en tant que `normal` (et non, par exemple, `italic`).
`weight="normal"`	spécifie que les métriques ci-dessus et les fichiers TTF seront utilisés si le poids de la police est défini en tant que `normal` (et non, par exemple, `bold`).

Note à propos du copyright de police : l'utilisation des polices est soumise aux lois de copyright et les conditions d'utilisation diffèrent. Avant d'intégrer une police, en particulier si vous intégrez toute la police, veuillez vous assurer que la licence que vous avez acquise pour cette police vous y autorise.

Jeux de caractères

Veuillez noter que les jeux de caractères des polices diffèrent l'un de l'autre. Les 14 polices standard couvrent les caractères ISO-8859-1 plus les glyphes dans les polices Symbol et Zapf Dingbats. Si votre document contient un caractère qui n'est pas couvert par les 14 polices standard, alors vous devrez utiliser une police qui contient ce caractère dans ce jeu de caractères. Certaines polices, comme Arial Unicode, proposent des caractères couverts par Unicode.

© 2016-2022 Altova GmbH

5.12 Graphiques

Quand un document XML est ouvert dans le Mode Texte ou le Mode Grille, un graphique (camembert, graphique à barres, etc.) représentant les données sélectionnées dans le document XML peut être généré dans la Fenêtre Graphiques (une des Fenêtres Sorties). Le camembert peut ensuite être exporté en tant qu'un fichier d'image ou en tant qu'un fragment XSLT ou XQuery dans le presse-papiers. La fonction Graphiques est utile pour représenter rapidement graphiquement des données numériques sélectionnées dans un document XML.

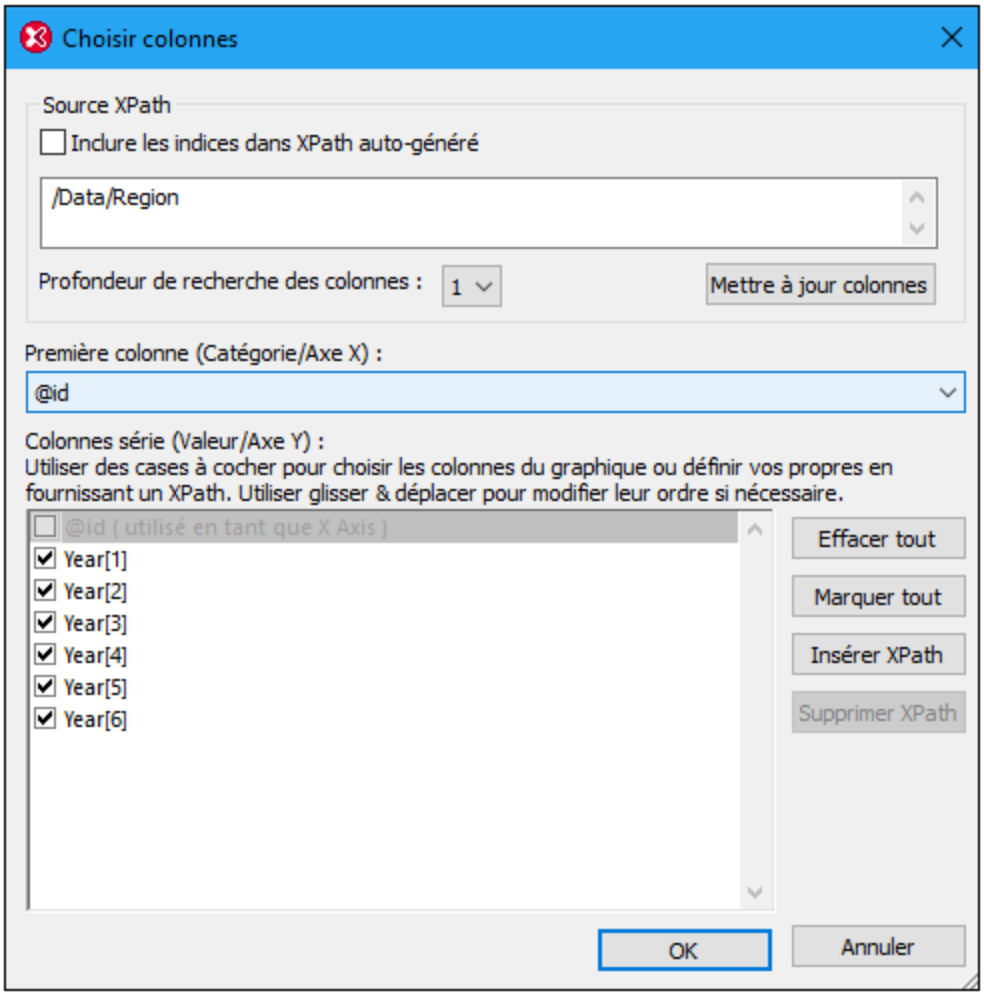
Les types de graphiques suivants sont disponibles :

- Camemberts (2D, 3D)
- Graphiques à barres, graphiques à barre unique (2D, 3D)
- Graphiques à barres, graphiques à barres regroupées (2D, 3D)
- Graphiques à barres superposées
- Graphiques à ligne de catégorie
- Graphiques à ligne de valeur
- Graphiques à surface et graphiques à surface superposées
- Graphiques à chandelles
- Graphiques à jauge (rondes et barres)
- Graphiques à superpositions

Aperçu : de la création à l'exportation

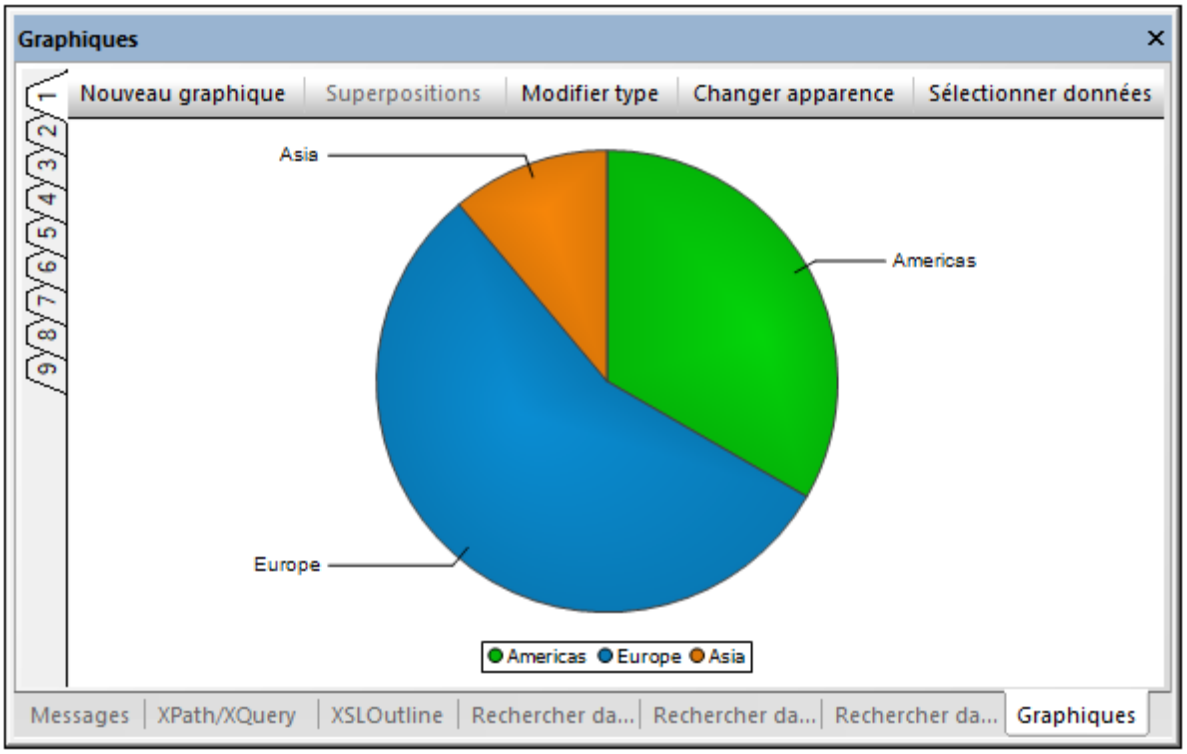
Les étapes permettant de créer un graphique sont décrites dans leurs grandes lignes ci-dessous. Pour une description plus détaillée, veuillez consulter les sous-sections de cette section.

1. Dans le Mode Texte ou le Mode, sélectionner le nœud que vous souhaitez utiliser en tant que le nœud contextuel pour la sélection de données. Vous pouvez aussi sélectionner une série de nœuds. Les implications des différentes méthodes de sélection sont expliquées dans la section XPath de source.
2. Cliquer avec la touche de droite et, à partir du menu contextuel qui apparaît, sélectionner la commande **Nouveau graphique**. En alternative, dans la fenêtre de sortie de Graphiques, cliquer sur le bouton **Nouveau graphique**. Le dialogue Choisir colonnes (*capture d'écran ci-dessous*), dans laquelle les données de l'axe X et de l'axe Y seront sélectionnées et dans lequel le XPath de source pourra être modifié.



3. Cliquer sur **OK** pour créer le graphique dans la Fenêtre Graphiques (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



4. La sélection du graphique et les autres paramètres pourront être édités par la suite. Non seulement il est possible d'éditer son XPath de source et la sélection de colonnes mais aussi son type et son apparence . La sélection de données pour les axes du graphique peuvent être éditées en cliquant sur la touche **Sélectionner les données**. Le type et l'apparence du graphique peut être modifié en cliquant sur le bouton **Modifier type** et **Changer apparence**, respectivement.
5. Le graphique peut être exporté en tant que fichier d'image ou en tant qu'un fragment XSLT ou XQuery sur le presse-papiers.

Autres fonctions

Les fonctions suivantes vous aident à utiliser les graphiques :

- ***Onglets multiples :*** si vous souhaitez créer un nouveau graphique sans supprimer le graphique actuel, vous pouvez créer le nouveau graphique dans un des autres onglets numérotés de 1 à 9 (*voir capture d'écran ci-dessus*). Veuillez noter que même lorsqu'un document XML est fermé, les graphiques générés depuis ce document resteront ouverts dans leurs onglets respectifs dans la Fenêtre Graphiques.
- ***Auto rechargement :*** si le bouton **Auto** (*voir capture d'écran ci-dessus*) est activé, le graphique sera automatiquement rechargé à chaque fois que les données dans le document XML sont modifiées. Dans le cas contraire, le graphique devra être remis a jour manuellement en cliquant sur le bouton **Recharger**.

Fichier d'exemple

Dans cette section et sous-section, les explications concernant le fonctionnement des graphiques s'appuient sur un fichier XML `YearlySales.xml`. Ce fichier est disponible dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
       <Region id="Asia">
              <Year id="2005">10000</Year>
              <Year id="2006">25000</Year>
              <Year id="2007">70000</Year>
              <Year id="2008">110000</Year>
              <Year id="2009">125000</Year>
              <Year id="2010">150000</Year>
       </Region>
</Data>
```

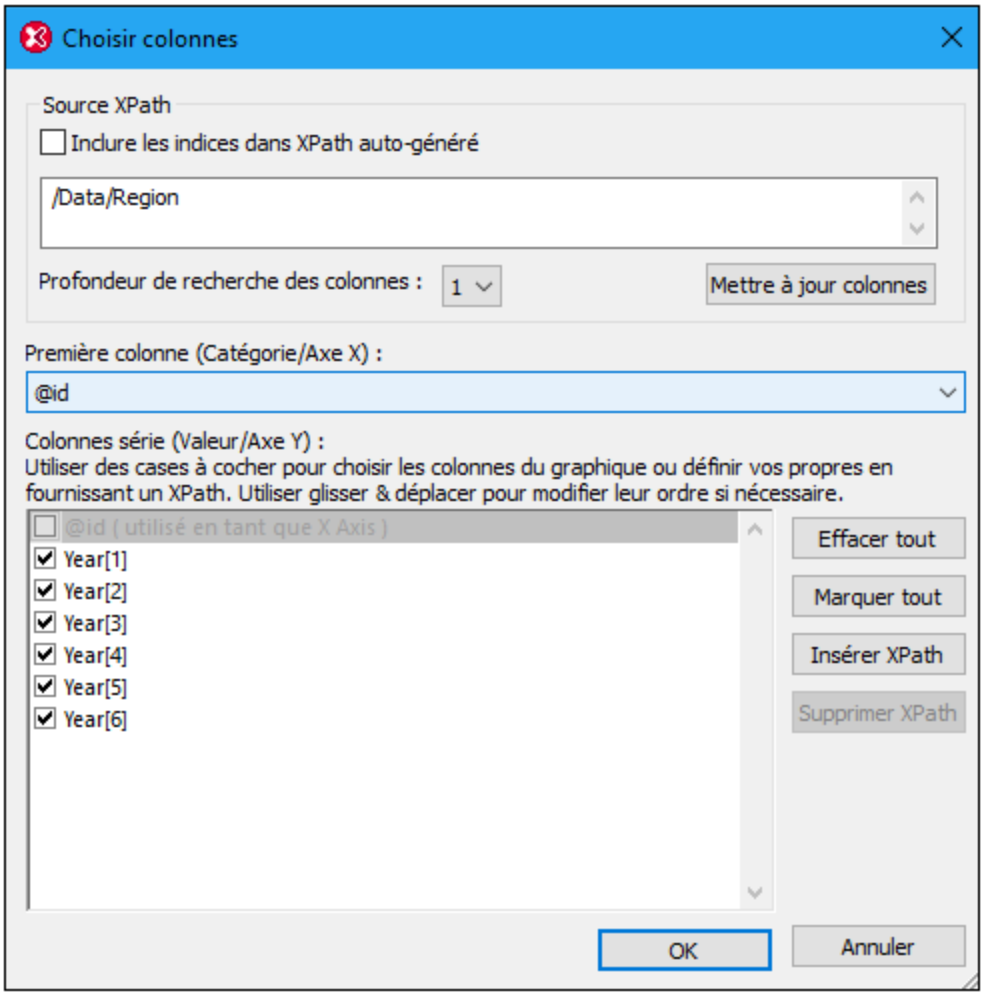
5.12.1 Créer un graphique

Le bouton **Nouveau graphique** entraîne l'affichage du dialogue Choisir colonnes (*capture d'écran ci-dessous*), contenant trois paramètres de sélection de données essentiels pour le graphique. Ces paramètres (voir ci-dessous) sont utilisés pour constituer la table de données de graphique.

- *XPath de source* : une expression XPath est automatiquement saisie lors de l'ouverture du dialogue. Elle sélectionne le nœud dans le document XML qui a été sélectionné au moment de l'accès au dialogue Choisir colonnes. Il peut être édité dans le dialogue à l'aide du clavier. La case à cocher *Inclure les Indices* détermine si les filtres de prédicat dans le XPath seront utilisés ou pas (voir XPath de source pour plus de détails). Des nœuds descendants du/des nœud/s sélectionné/s par le XPath de source seront disponibles pour une sélection en tant que colonnes de données Axe X et Axe Y. La liste de choix Profondeur de recherche des colonnes détermine combien de niveaux de descendants seront cherchés pour retourner des nœuds qui peuvent être utilisés pour la sélection de données de l'Axe X et Axe Y. Une fois que le XPath de source a été édité, cliquer sur **Mettre à jour colonnes** pour que le changement prenne effet et pour que les listes Axe X et Axe Y dans le dialogue soient réinitialisées.
- *Axe X* : la sélection dans cette liste de choix spécifie quel nœud sera utilisé en tant que l'Axe X. La séquence retournée pour cette sélection produira les libellés qui se produisent sur l'Axe X. L'option *Énuméré auto* de la liste de choix fournit des libellés numérotés pour l'Axe X. Veuillez noter que les expressions XPath créées pour l'Axe Y sont aussi disponibles pour la sélection dans la liste de choix de l'Axe Y.

© 2016-2022 Altova GmbH

- *Axe Y* : les entrées qui sont cochées dans ce panneau seront les nœuds, dont les valeurs numériques seront représentées dans l'axe numérique Y. Le bouton **Effacer tout** désélectionne tous les items et le bouton **Marquer tout** sélectionne tous les items dans le panneau Axe Y. Le bouton **Insérer XPath** permet de générer une série qui n'est pas disponible parce qu'elle n'est pas un descendant du nœud que le Source XPath retourne. Le nœud ou l'expression XPath sélectionné pour l'Axe X n'est pas disponible pour la sélection de l'Axe Y et est grisé.



Comment la table de données du graphique est créée

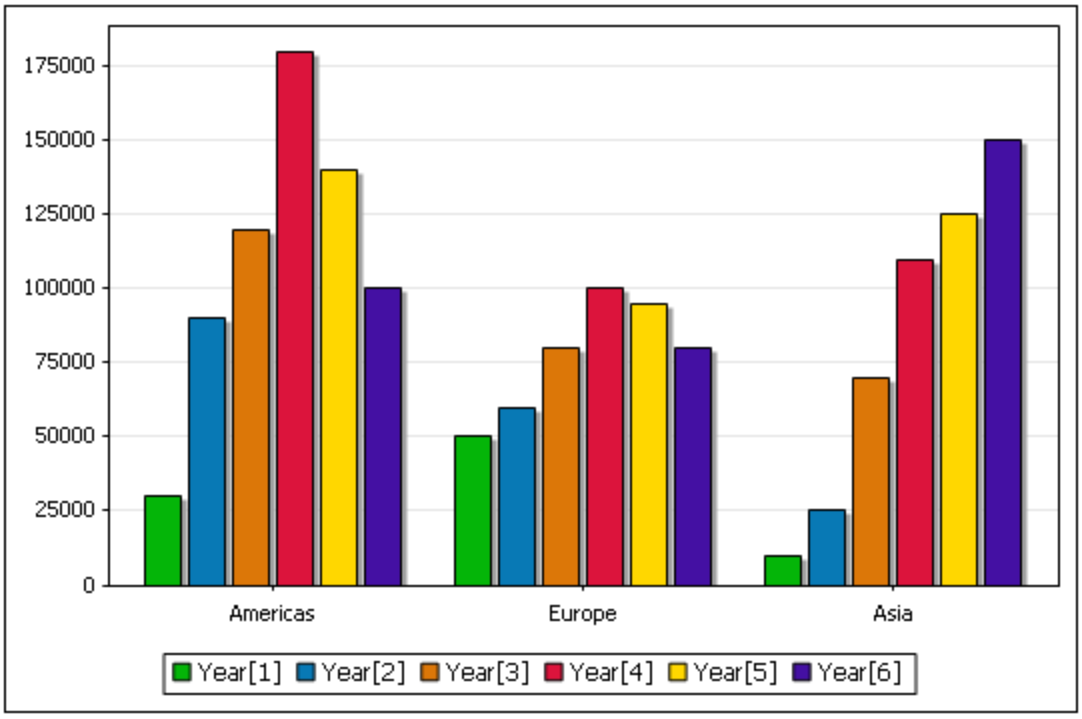
Les données qui sont utilisées pour le graphique sont déterminées par la sélection réalisée dans le dialogue Choisir colonnes. Nous allons expliquer comment les données du graphique sont sélectionnées à l'aide d'un exemple. Puisque le document XML (*capture d'écran plus bas ci-dessous*) contient trois éléments `Region`, le XPath de source `/Data/Region` sélectionne l'un d'entre eux chacun à son tour. Chaque élément `Region` constitue un nœud contextuel, ensuite, les colonnes des données sont générées. Pour chaque élément `Region` sélectionné en raison de l'XPath de source, l'étape suivante est effectuée :

1. L'expression de l'Axe X génère la première colonne (par défaut, cette colonne sera la colonne utilisée pour les libellés de l'Axe X).
2. Pour chaque série (sélections de l'axe Y) une colonne est générée.

Les données de graphique générées pour le dialogue Choisir colonnes affichées ci dessus peuvent être visualisées à l'exemple de la table suivante.

XPath source	Axe X	Axe Y (colonnes de séries)					
Region[1]	@id	Year[1]	Year[2]	Year[3]	Year[4]	Year[5]	Year[6]
Region[2]	@id	Year[1]	Year[2]	Year[3]	Year[4]	Year[5]	Year[6]
Region[3]	@id	Year[1]	Year[2]	Year[3]	Year[4]	Year[5]	Year[6]

Un graphique à barres généré depuis ces données pourrait ressembler à l'exemple suivant :



Les points suivants importants devraient être notés :

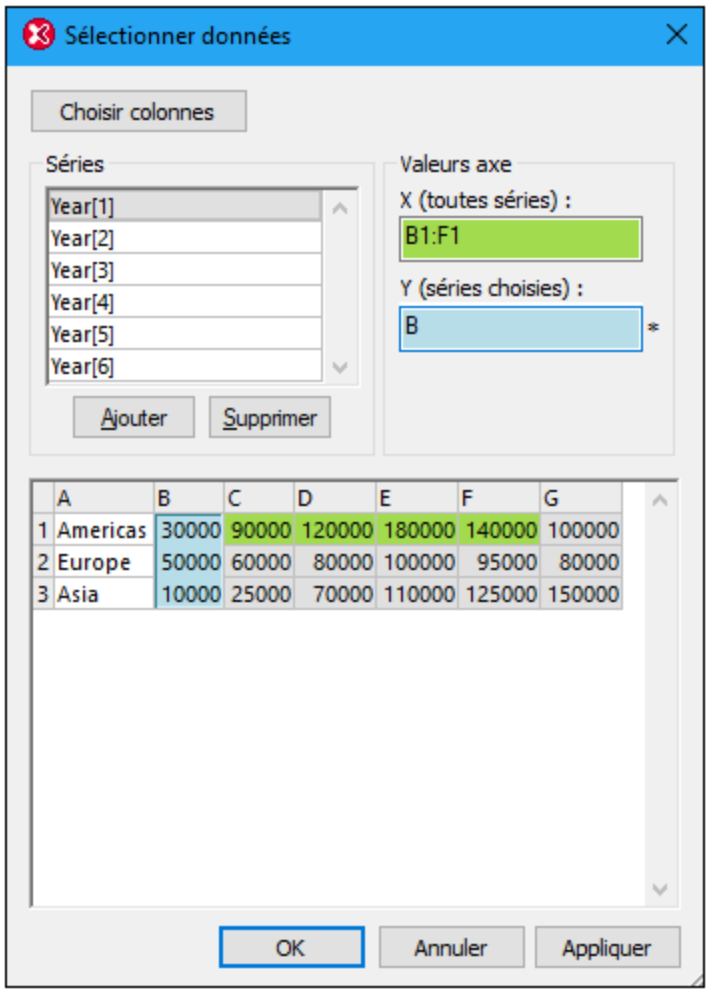
- Le nombre de graduations sur l'Axe X est déterminé par la taille de la séquence retournée par l'expression de XPath de source (3, dans ce cas).
- Les nœuds retournés par le XPath de source seront des nœuds contextuels, respectivement, pour générer deux ensembles de données pour chaque graduation sur l'Axe X : (i) le libellé de graduation Axe X (effectué avec la sélection de l'Axe X), et (ii) toutes les séries à dessiner pour cette graduation (ces séries sont choisies avec la sélection de l'Axe Y). Les expressions XPath saisies pour l'Axe X et l'Axe Y seront évaluées en tant qu'expressions XPath dans le contexte de ces nœuds (XPath Source).
- La séquence retournée par la sélection Axe X sera, respectivement, le libellé pour chaque coche. S'il y a moins de libellés que de graduations, certaines graduations demeureront sans libellé.
- Chaque série (par exemple `Year[1]`) est évaluée une fois pour chaque nœud contextuel. Pour certains graphiques, comme des camemberts ou des graphiques à barre unique, seule une série peut être utilisée.
- Les légendes sont obtenues à partir des noms des items de série.

© 2016-2022 Altova GmbH

Le document XML utilisé pour l'exemple ci-dessus est indiqué ici en guise de référence. Il est nommé `YearlySales.xml` et est disponible dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
       <Region id="Asia">
              <Year id="2005">10000</Year>
              <Year id="2006">25000</Year>
              <Year id="2007">70000</Year>
              <Year id="2008">110000</Year>
              <Year id="2009">125000</Year>
              <Year id="2010">150000</Year>
       </Region>
</Data>
```

La sélection de données indiquée dans le dialogue Choisir colonnes ci-dessus peut être consultée dans la table du dialogue Choisir colonnes (*capture d'écran ci-dessous*). Le dialogue Choisir colonnes est accessible en cliquant sur le bouton **Sélectionner données** dans la fenêtre de Sortie de graphiques.

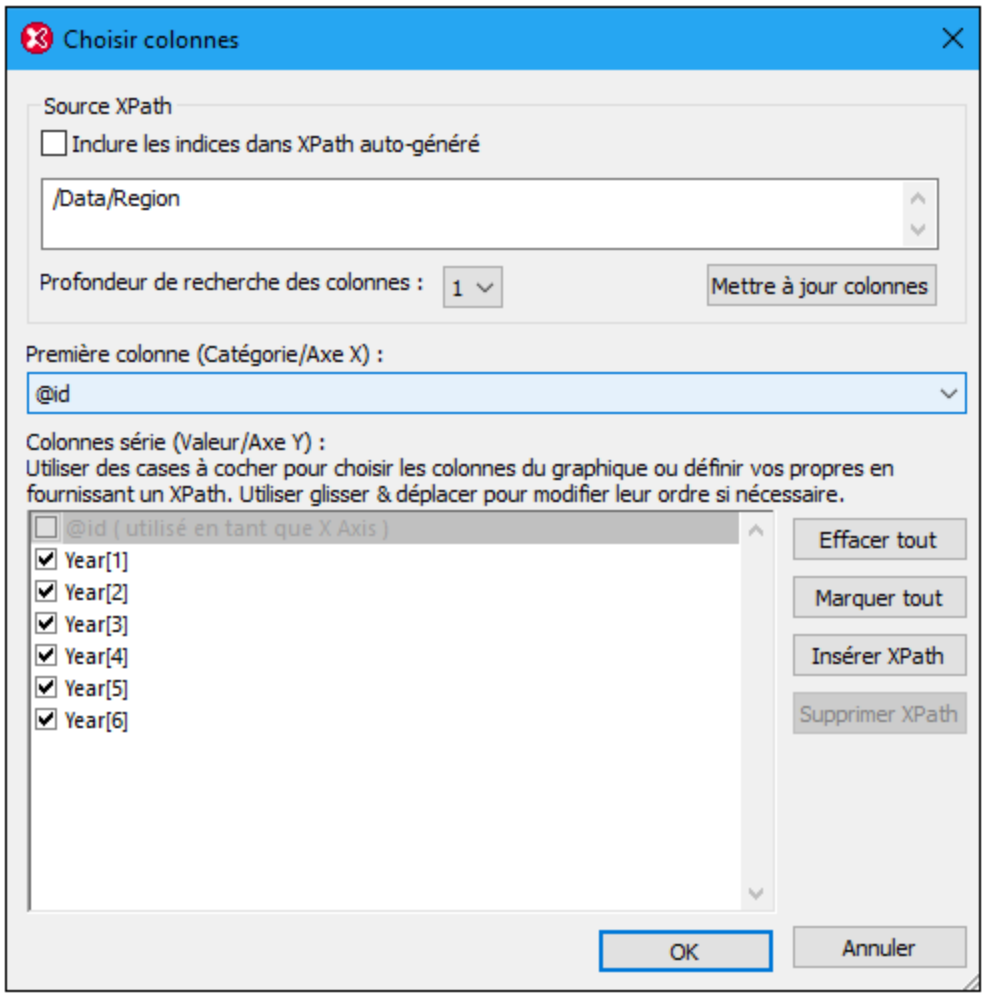


Pour plus de détails concernant les paramètres individuels du dialogue Choisir colonnes, voir les sections : XPath de source, Sélection Axe X, Sélection Axe Y et Données de graphique.

5.12.2 XPath de source

Le XPath de source est spécifié dans le dialogue Choisir colonnes. Il détermine quels nœuds dans le document sont disponibles pour la sélection en tant que données de l'Axe Y et de l'Axe Y. La liste de choix Profondeur de recherche des colonnes détermine combien de niveaux descendants seront cherchés pour retourner des nœuds qui peuvent être utilisés pour les données d'Axe X et d'Axe Y.

© 2016-2022 Altova GmbH



Après avoir sélectionné un nouveau XPath de source, cliquer sur **Mise à jour des colonnes** pour réinitialiser les sélections disponibles (dans le dialogue) pour l'Axe X et l'Axe Y. Si des prédicats sont inclus dans l'expression XPath (par exemple, `/Data/Region[1]` utilise le `[1]` prédicat pour sélectionner le premier élément `Region`), alors, il faudra cocher la case Inclure les indices. Si vous modifiez l'expression de XPath de source, n'oubliez pas de cliquer sur **Mettre à jour les colonnes**.

XPath de source depuis l'emplacement du curseur

Afin d'expliquer quel XPath de source est sélectionné lors de l'ouverture du dialogue Choisir colonnes, nous allons utiliser le document XML affiché ci-dessous. Il est nommé `YearlySales.xml` et il se trouve dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.

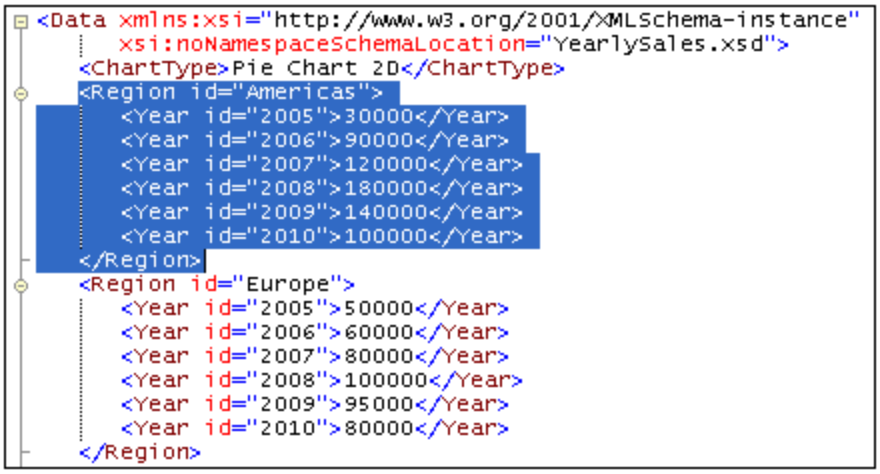
```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
```

```
            <Year id="2008">180000</Year>
            <Year id="2009">140000</Year>
            <Year id="2010">100000</Year>
      </Region>
      <Region id="Europe">
            <Year id="2005">50000</Year>
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
      </Region>
      <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
      </Region>
</Data>
```

Les cas suivants sont possibles :

- Le curseur est placé n'importe où dans la balise de démarrage (y compris dans un attribut-valeur) ou dans la balise de fin de l'élément `Data`, ou ailleurs dans l'élément `Data` mais pas dans un nœud descendant, le XPath de source sera : `/Data`
- Le curseur est placé n'importe où dans la balise de démarrage (y compris dans un attribut-valeur) ou dans la balise de fin de tout élément `Region` element, ou ailleurs dans un élément `Region` mais pas dans un nœud descendant, le XPath de source sera : `/Data/Region`
- Le curseur est placé n'importe où dans la balise de démarrage (y compris dans un attribut-valeur) ou dans la balise de fin d'un élément `Year` element, ou ailleurs dans un élément `Year`, le XPath de source sera : `/Data/Region[N]/Year`. Le filtre de prédicat `[N]` sélectionne l'élément `Region` particulier qui est l'élément `Year` sélectionné. Donc les données de sélection de l'Axe X et de l'Axe Y seront limitées aux éléments `Year` de cet élément `Region` particulier.
- Vous souhaitez sélectionner juste un élément `Region`, par exemple : `/Data/Region[1]`, alors il faudra marquer cet élément, c'est à dire, le premier élément `Region` tel que montré dans la capture d'écran ci-dessous.

© 2016-2022 Altova GmbH



- De même, le fait de marquer les deux éléments `Region` générera une expression XPath qui sélectionne uniquement ces deux éléments `Region`.

L'expression de XPath de source pourra être éditée par la suite dans le dialogue Choisir colonnes.

Dans le Mode Grille, la sélection s'effectue en cliquant sur un nœud ou en marquant une plage. Le XPath de source qui est généré depuis la sélection du Mode Grille est telle que décrite ci-dessus pour le Mode Texte.

Inclure les indices dans XPath

La case *Inclure les Indices* détermine si des filtres de prédicat sont utilisés dans l'expression XPath, si ces filtres de prédicats sont saisis automatiquement au moment où le dialogue Choisir colonnes est appelé ou si ils sont saisis manuellement. Par exemple, si le curseur est placé dans un élément descendant du premier élément `Region` du document et que la case *Inclure les Indices* est cochée, alors l'expression XPath saisie automatiquement sera par exemple : `/Data/Region[1]/Year`. Si la case *Inclure les Indices* n'est pas cochée, alors l'expression sera : `/Data/Region/Year`.

La case *Inclure les Indices* détermine aussi si un prédicat saisi manuellement est retenu. C'est pourquoi, si vous souhaitez utiliser des prédicats dans l'expression XPath de source, vous devrez cocher la case *Inclure les Indices*.

Implications des sélection de XPath de source

Veuillez noter les implications suivantes de la sélection du XPath de source.

- Le nombre des items dans la séquence retourné par le XPath de source détermine le nombre de graduations dans l'Axe X. Le nombre des graduations dans l'Axe X peut être modifié d'une seule autre manière (à part la modification du XPath de source) : en sélectionnant un nombre de libellés pour chaque série, mais le nombre doit être supérieur au nombre de graduations. Voir Sélection de l'Axe X pour plus d'informations concernant ce scénario.
- Le XPath de source nœud est le nœud ancêtre pour tous les nœuds disponibles pour la sélection de données de l'Axe X et de l'Axe Y et pour toutes les expressions XPath que vous pourriez saisir.
- En conséquence des deux autres points, veuillez noter que tout changement à l'expression du XPath de source affecte non seulement le nombre de graduations sur l'Axe X mais aussi le contexte pour toute expression XPath liée au graphique.

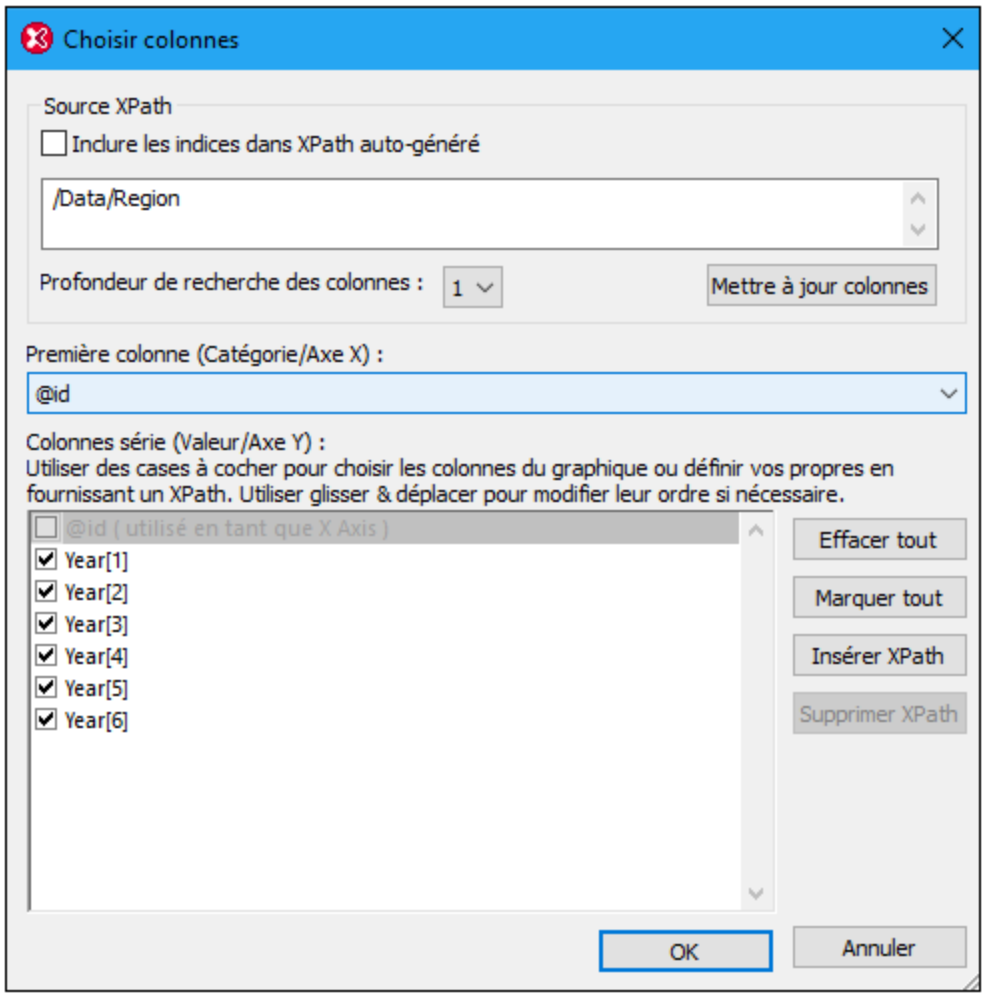
Par exemple, voici les implications de certaines expressions XPath par rapport au document XML affiché ci-dessus.

- /Data/Region: retourne les trois éléments `Region`, d'où les trois graduations sur l'Axe X. Chaque élément `Region` sera, à son tour, le nœud contextuel pour les expressions XPaths.
- /Data/Region/Year: retourne 18 éléments `Year`, d'où les 18 graduations sur l'Axe X. Chaque élément `Year` sera, à son tour, le nœud contextuel pour les expressions XPaths.
- /Data/Region[1]/Year: retourne les six enfants élément `Year` du premier élément `Region`, d'où les six graduations sur l'Axe X. Chaque élément `Year` du premier élément `Region` element sera, à son tour, le nœud contextuel pour les expressions XPaths.
- distinct-values(//Year/@id): retourne six items (les valeurs distinctes de l'attribut `Year/@id` : `2005`, `2006`, `2007`, `2008`, `2009`, `2010`). Néanmoins, puisque cette expression XPath ne retourne aucun item de nœud, elle ne peut pas être un nœud contextuel pour une expression XPath. Si les items dans cette séquence doivent être utilisées pour cibler des nœuds dans le document XML (à l'aide par exemple de la fonction `current()` (en abrégé : .)), alors l'expression XPath utilisant l'item actuel doit être lancée depuis la racine de document afin que le contexte soit établi. Par exemple : `/Data/Region[1]/Year[@id eq .]`.

5.12.3 Sélection de l'Axe X

La sélection de l'Axe X est spécifiée dans le dialogue Choisir colonnes (*capture d'écran ci-dessous*). Cette sélection détermine les libellés qui vont apparaître sur l'Axe X. Les libellés peuvent ensuite être édités dans le dialogue Choisir données (*voir ci-dessous*).

© 2016-2022 Altova GmbH



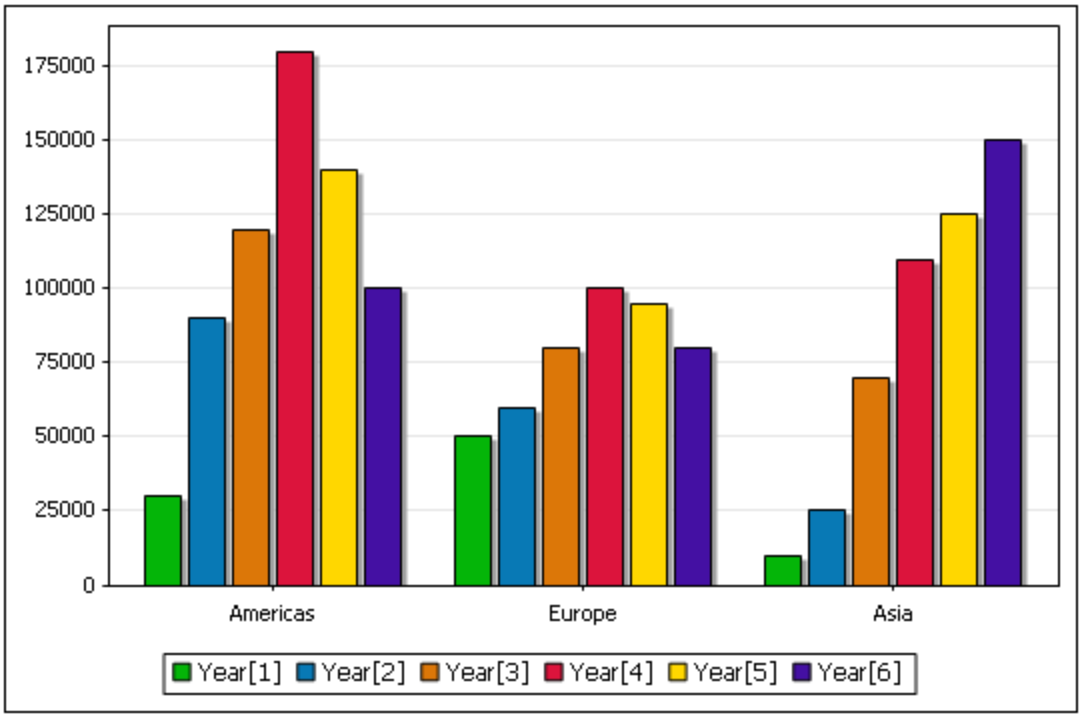
Prenons le document XML suivant en guise d'exemple. (Il est nommé `YearlySales.xml` et est disponible dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.) Placer le curseur dans la balise de démarrage du premier élément `Region` et cliquer sur le bouton **Nouveau graphique** de la fenêtre de sortie Graphiques. Le dialogue Choisir colonnes apparaît, avec le XPath de source : `/Data/Region` (*voir capture d'écran ci-dessus*).

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
```

```
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
        </Region>
        <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
        </Region>
</Data>
```

Comme expliqué dans la section XPath de source, cet XPath de source configure un graphique avec trois graduations sur l'Axe X (parce que le XPath de source retourne trois items : les trois éléments `Region`). Étant donné que nous souhaitons que les libellés de ces trois graduations dans le graphique soient les noms des trois régions, nous choisissons l'attribut `@id` dans la liste de choix de la sélection de l'Axe X (*voir capture d'écran du dialogue Choisir colonnes ci-dessus*).

Afin de produire les données de graphique pour chaque graduation, chaque élément `Region` est évalué à son tour. Pour chaque élément `Region`, l'attribut `id` génère le libellé correct pour l'Axe X. L'Axe X ressemblera donc à l'exemple ci-dessous.



Si une autre expression XPath est sélectionnée dans la liste de choix de l'Axe-X, cette expression sera évaluée dans le cadre du contexte de l'élément `Region` respectif et le résultat évalué sera le libellé de la graduation

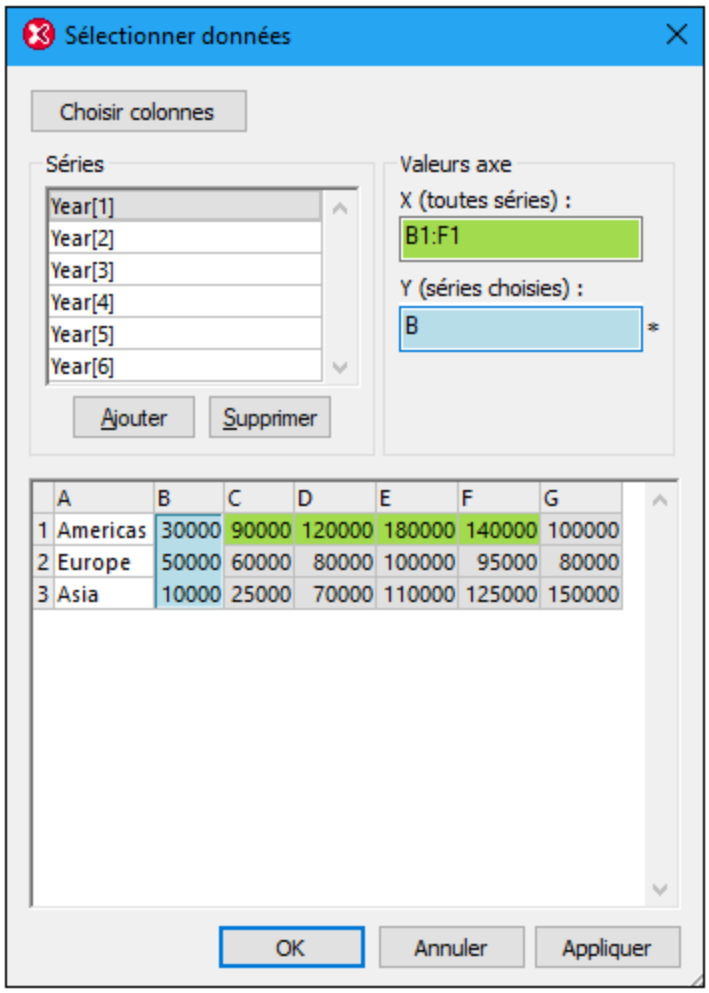
© 2016-2022 Altova GmbH

respective. L'option *Énuméré auto* génère une séquence de numéro qui correspond au nombre de graduations : la première graduation affichera le numéro 1, la deuxième 2, etc.

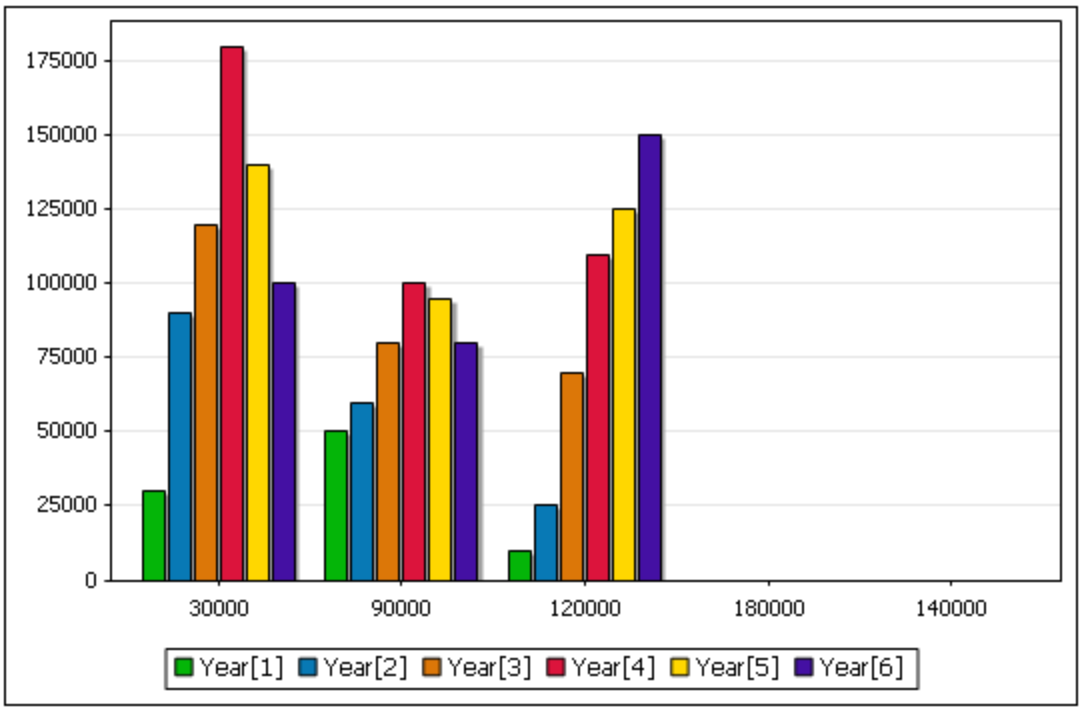
Modifier les libellés de l'Axe X et la numérotation de l'Axe X

La sélection des libellés pour l'Axe X peut être modifiée dans le dialogue Sélectionner données (accessible en cliquant sur le bouton **Sélectionner données** de la sortie Graphiques).

Dans le dialogue Sélectionner données affiché dans la capture d'écran ci-dessous, par exemple, cliquer dans le champ de texte de l'Axe X du panneau Valeurs axe pour modifier la sélection de l'Axe X. À présent, cliquer sur le champ `B1` et glisser la souris sur `F1` pour procéder à la sélection `B1:F1`. Cliquer sur **OK** pour voir le nouveau graphique.



Cette sélection contient les libellés pour les graduations, tel que montré dans la capture d'écran ci-dessous. Veuillez également noter que puisque la nouvelle sélection contient cinq items, cinq graduations ont été générées. Néanmoins, seule les trois premières ont été remplies. Cela s'explique par le fait que le XPath de source retourne trois nœuds et qu'il s'agit des nœuds qui ont été traités pour les graphiques. Ces trois nœuds correspondent aux lignes de la table affichés dans le dialogue Sélectionner les données. Veuillez noter que le nombre de lignes dans la table peut uniquement être modifié en changeant le XPath de source.

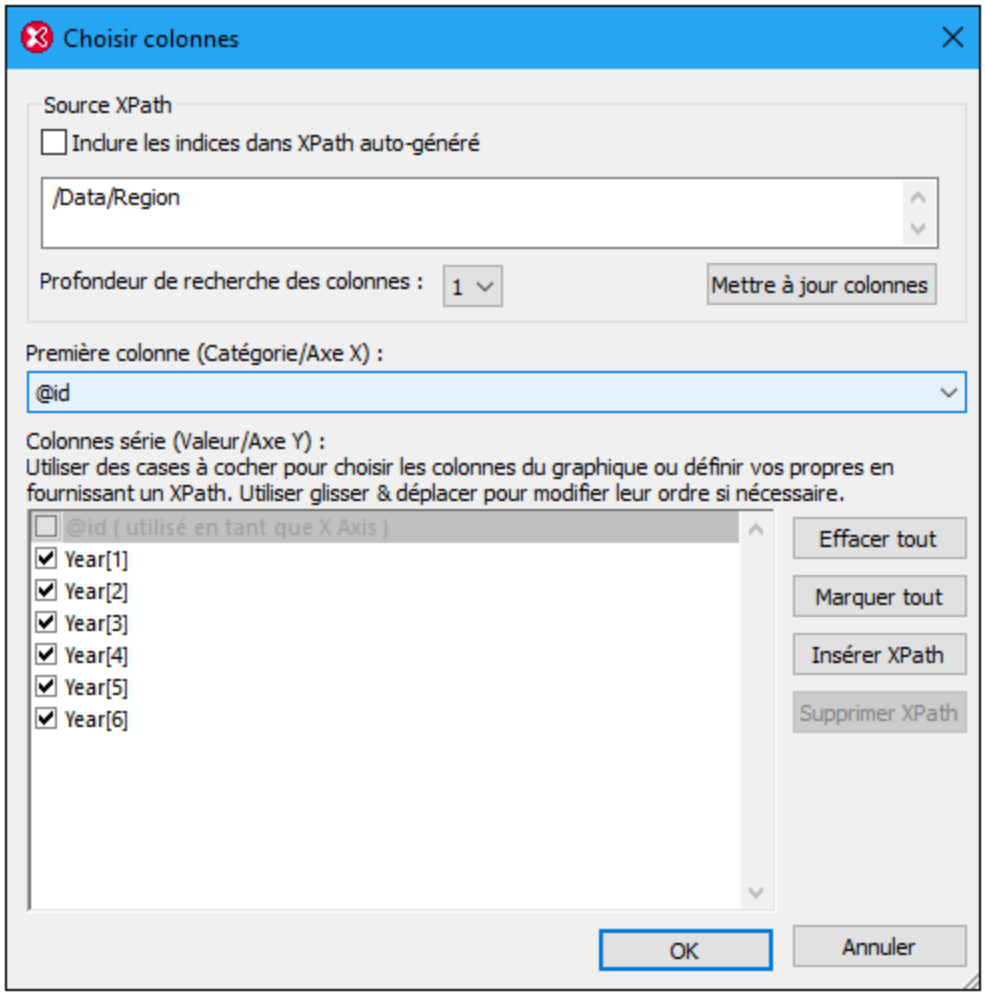


Pour plus d'informations concernant les interactions des sélections XPath de source et Axe X, voir Comment créer les données de graphique.

© 2016-2022 Altova GmbH

5.12.4 Sélection de l'Axe Y

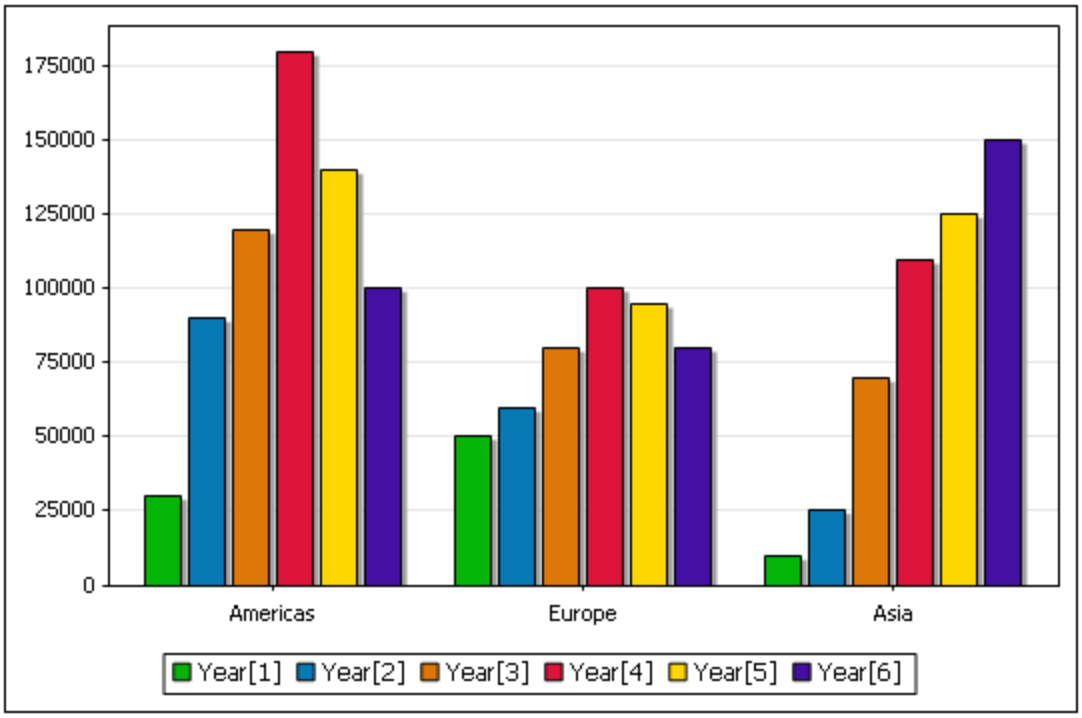
L'axe Y The axe Y(*voir capture d'écran ci-dessous*) est également appelé l'Axe de séries.



La sélection que vous effectuez pour cet axe détermine combien de séries seront calculées pour chaque graduation de l'Axe X. Si une seule série est sélectionnée, à chaque graduation de l'Axe X, la valeur retournée par l'expression XPath pour cette série-là sera calculée. Si plusieurs séries sont sélectionnées (comme indiqué dans la capture d'écran ci-dessous, dans laquelle six séries sont sélectionnées), les données de la sélection du graphique seront identiques à la table ci-dessous. (Le nœud contextuel pour la sélection de l'Axe Y est l'élément `Region` respectif.)

XPath de Source	Axe X	Axe Y (colonnes Séries)					
`Region[1]`	`@id`	`Year[1]`	`Year[2]`	`Year[3]`	`Year[4]`	`Year[5]`	`Year[6]`
`Region[2]`	`@id`	`Year[1]`	`Year[2]`	`Year[3]`	`Year[4]`	`Year[5]`	`Year[6]`
`Region[3]`	`@id`	`Year[1]`	`Year[2]`	`Year[3]`	`Year[4]`	`Year[5]`	`Year[6]`

Le graphique ressemblera donc à :



Il y a trois graduations de l'Axe X, libellés avec la valeur des attributs `Region/@id` respectifs. à chaque graduation Axe X, l'expression XPath est évaluée pour chaque série. Dans notre exemple, pour chaque graduation Axe X, chacune des six séries est évaluée et calculée. Par exemple, la première série (`Year[1]`) est calculée pour les trois régions, donc aussi `Year[2]` à `Year[6]`.

Note : Certains graphiques, comme les camemberts et les graphiques à barre unique ne présentent qu'un axe. Par exemple, dans un graphique à barre unique, chaque graduation de l'Axe X n'aura qu'une seule barre : celle représentant la série unique. Dans un **camembert**, les valeurs de la série unique totaliseront jusqu'à 100% du camembert, chaque valeur étant attribuée à une graduation de l'Axe X.

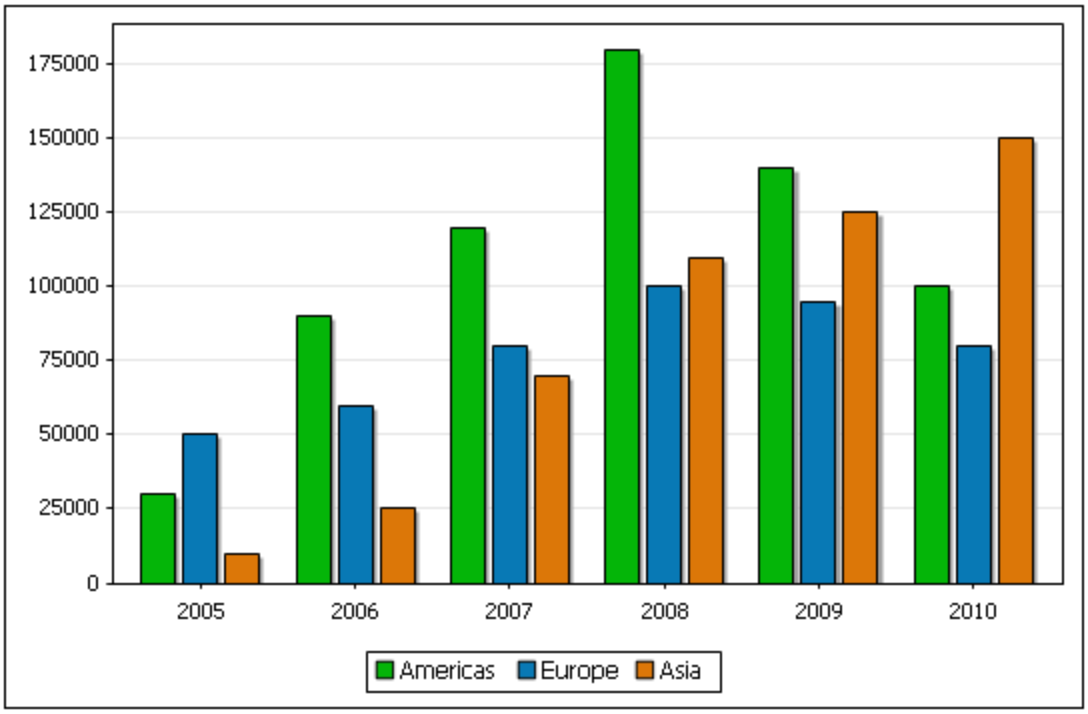
Légendes de l'Axe Y

Les légendes qui sont affichées dans la partie inférieure du graphique sont les noms des séries. Ces noms peuvent être modifiés dans le dialogue Sélectionner données.

Changer les sélections de l'Axe X et de l'Axe Y

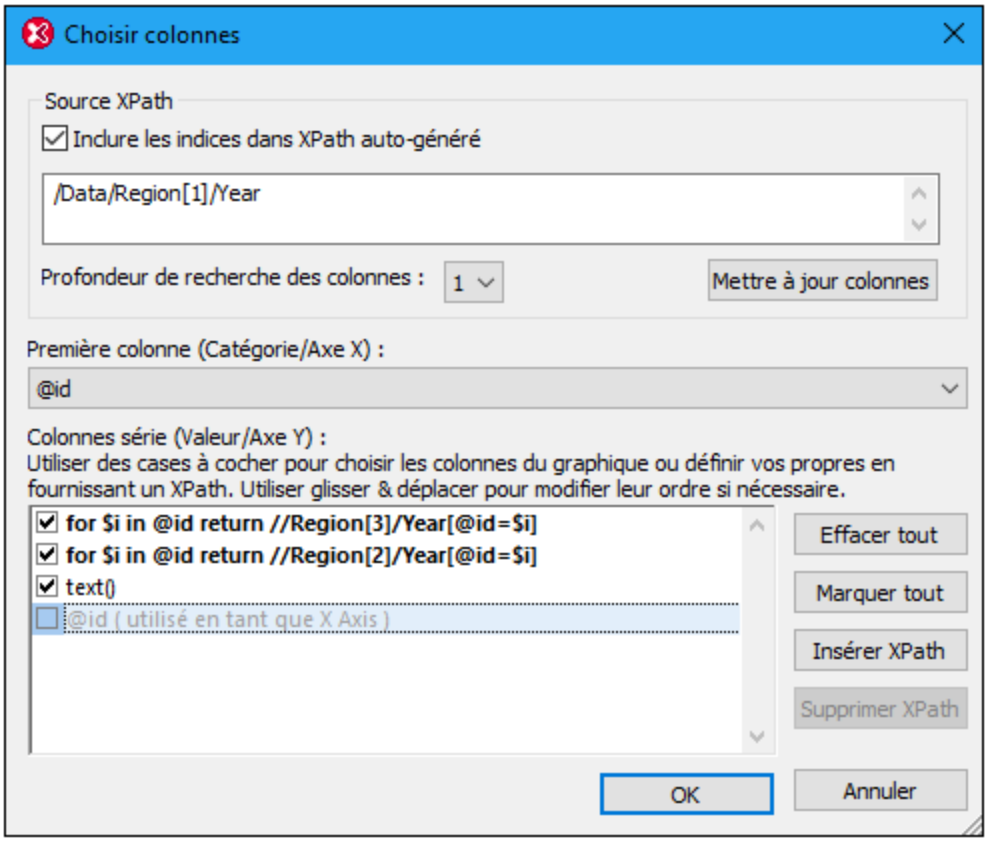
Dans l'exemple ci-dessus, les régions se trouvent sur l'Axe X et les ventes annuelles se trouvent sur l'Axe Y pour chaque région ; les éléments `Year` sont la série. Mais que se passe-t-il si nous essayons de calculer les années sur l'Axe X et de comparer les ventes régionales pour chaque année dans le graphique à barres ci-dessous ? Il nous faudrait six graduations de l'Axe X (obtenus par le biais de la sélection du XPath de source), il faut ensuite libellé les graduations de l'Axe X avec les années respectives, et enfin, sélectionner trois séries (pour les régions), qui seront toutes représentées à chaque graduation de l'Axe X. Le dialogue Choisir colonnes dans la capture d'écran ci-dessous, montre comment cette sélection de données peut être obtenue.

© 2016-2022 Altova GmbH



Sélectionner la série

Pour faire d'un nœud une série pour le graphique, cocher la case de ce nœud. Vous pouvez modifier le XPath de Source et la profondeur de recherche de colonnes pour rendre le nœud disponible dans le panneau Séries. En alternative, vous pouvez ajouter une expression XPath pour choisir un nœud, comme dans la capture d'écran ci-dessous. Voir Exemple de graphique : Avancé pour une description de ce scénario.



Référence

Le document XML utilisé pour illustrer l'exemple dans cette section est indiqué ici pour référence. Il est nommé `YearlySales.xml` et vous le trouverez dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
```

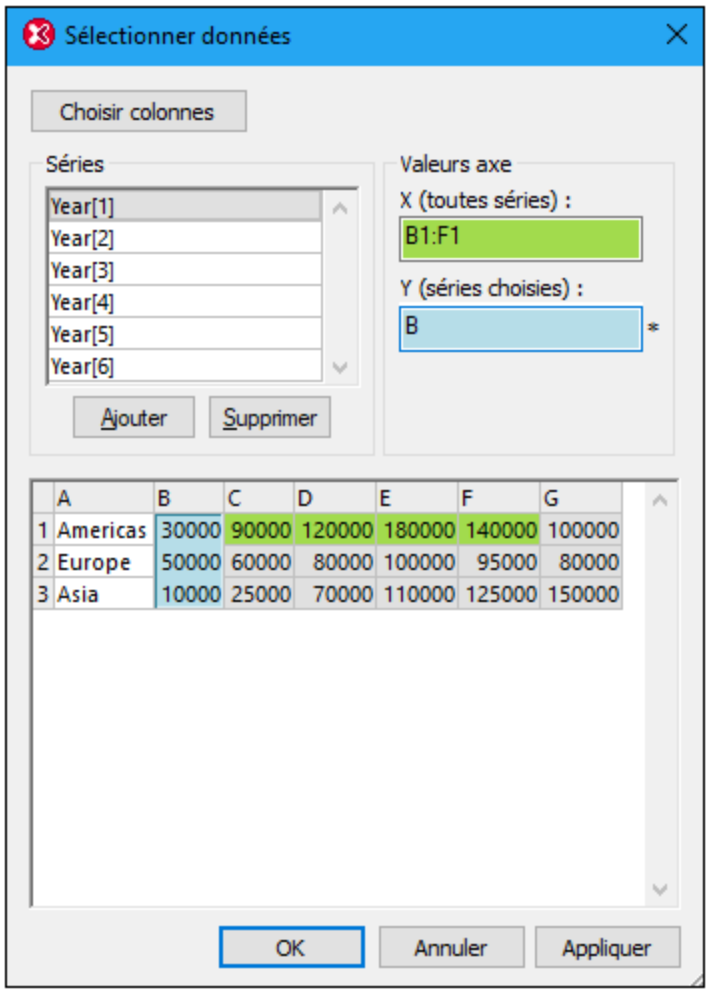

© 2016-2022 Altova GmbH

```
        <Region id="Asia">
                <Year id="2005">10000</Year>
                <Year id="2006">25000</Year>
                <Year id="2007">70000</Year>
                <Year id="2008">110000</Year>
                <Year id="2009">125000</Year>
                <Year id="2010">150000</Year>
        </Region>
</Data>
```

5.12.5 Données de graphique

Cliquer sur le bouton **Sélectionner données** pour ouvrir le dialogue Sélectionner données (*capture d'écran ci-dessous*), qui consiste en trois volets : (i) la volet Séries, (ii) le volet Valeurs d'axe et (iii) la table de données de graphique. Chacun d'entre eux est décrit ci-dessous.

Le bouton **Choisir colonnes** ouvre le dialogue Choisir colonnes, dans lequel vous pouvez modifier le XPath de Source et la sélection des données pour l'Axe X et l'Axe Y.



Volet Séries

Les séries contenues à l'origine dans le volet Séries sont celles qui ont été sélectionnées dans le dialogue Choisir colonnes. Lorsque vous cliquez sur **OK**, les séries contenues dans ce volet seront les séries qui apparaissent dans le graphique. Le volet Séries vous permet d'effectuer trois opérations :

- *Ajouter et supprimer des séries :* permet de contrôler le nombre de séries qui apparaissent dans le graphique.
- *Éditer le nom des séries :* les noms des séries sont les légendes qui apparaissent dans le graphique.
- *Sélectionner des données pour les séries individuelles :* une fois une série sélectionnée dans le volet Séries, les donnés de l'Axe X et de l'Axe Y peuvent être spécifiées dans le volet Valeurs de l'axe. Pour ce faire, suivre les instructions ci-dessous.

Volet Valeurs d'axe

Les données de l'Axe X et de l'Axe Y peuvent être définies dans les champs de texte respectifs dans le volet Valeurs de l'axe. Cliquer dans un des champs de texte, pour éditer la valeur ; cela est indiqué par un astérisque situé à la droite du champ. Les données peuvent être sélectionnées en tant que plage depuis la

© 2016-2022 Altova GmbH

table des données de la table. Une plage peut être soit (i) une colonne ou une ligne entière, soit (ii) faire partie d'une colonne ou d'une ligne. En alternative, les données peuvent être saisies soit par le biais du clavier (par exemple, `A` ou `3` ou `B1:F1`). Pour marquer une plage, choisir la première cellule dans la plage et glisser le curseur jusqu'à la dernière cellule dans la plage. Pour marquer tout une colonne ou une ligne, sélectionner l'en-tête de ligne ou de colonne.

La sélection de l'Axe X détermine les libellés des nœuds de l'Axe X et s'applique à toute la série. Elle ne modifie pas le nombre de graduations de l'Axe X.

La sélection de l'Axe Y détermine quelle plage de cellules doit être utilisée pour les séries sélectionnées. Si le nombre de cellules sélectionné est inférieur au nombre de graduations de l'Axe X, alors cette série sera sous-représentée pour les graduations de l'Axe X ultérieures. Si le nombre de cellules sélectionné est supérieur au nombre de graduations de l'Axe X, alors des graduations supplémentaires seront créées. Les graduations supplémentaires seront égales au nombre des cellules sélectionnées en plus. Les valeurs supplémentaires seront représentées pour cette série sur le nombre supplémentaire de graduations.

Table de données de graphique

La structure de la table des données du graphique (située en bas du dialogue Sélectionner les données) est obtenu à partir des sélections dans le dialogue Choisir colonnes.

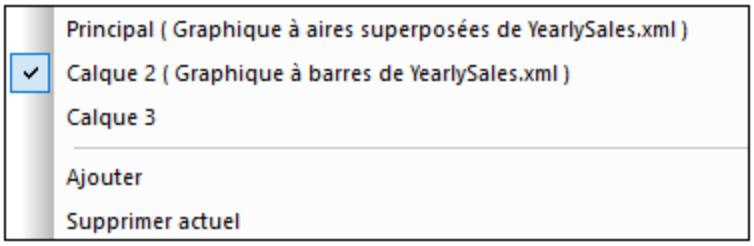
- Le nombre de lignes dans la table est égal au nombre d'items se trouvant dans la séquence retournée par le XPath de Source.
- Les colonnes sont nommées en commençant pas A. Le but de ce nom est de permettre une sélection dans le volet Valeurs de l'Axe (par exemple `B1:F1`).
- La première colonne est obtenue en évaluant la sélection de l'Axe X dans le dialogue Choisir colonnes dans le contexte des nœuds retournés par l'expression XPath de source.
- Toutes les autres colonnes sauf la première sont obtenues en évaluant les sélections de l'Axe Y dans le dialogue Choisir colonnes. Chaque série dans la sélection de l'Axe Y du dialogue Choisir colonnes correspond à une colonne dans la table de données de graphique.

La table de données de graphique peut être consultée en tant que surensemble de données sélectionnées à l'aide des paramètres dans le dialogue Choisir colonnes. À partir de ce surensemble, vous pouvez sélectionner les plages des données que vous désirez (dans le volet valuers d'Axe) pour les séries individuelles.

5.12.6 Superpositions

Une superposition est un graphique qui se superpose au graphique de base. Pour ajouter une superposition, procéder comme suit :

1. Cliquer sur le bouton **Superpositions** pour afficher le menu Superpositions (*capture d'écran ci-dessous*).



2. Cliquer sur **Apposer**. Un nouveau calque sera ajouté.
3. Une fois le nouveau calque sélectionné, cliquer sur le bouton **Nouveau graphique** et sélectionner les données comme décrit dans la section Créer un graphique.
4. Pour modifier le type de graphique et son apparence, cliquer sur le bouton **Modifier type** et **Changer apparence**, respectivement.

Vous pouvez ajouter autant de superpositions que vous le souhaitez. Chaque nouveau calque sera ajouté aux calques existants et sera surimposé dans le diagramme. Si vous souhaitez modifier l'ordre des calques, vous devrez les recréer dans le bon ordre. Vous pouvez supprimer le calque sélectionné actuellement en cliquant sur **Supprimer actuel**.

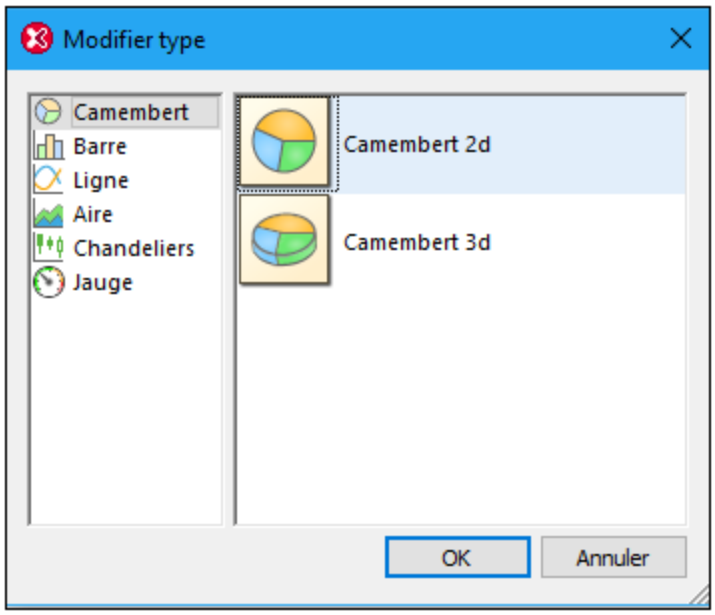
Note : Chaque superposition recouvrira les calques du dessous. Puisque seules des surfaces de graphique peuvent être rendues transparentes, certaines dispositions de calques peuvent s'avérer non-optimales. Par exemple, un graphique de barres placé sur un graphique à ligne dissimulera des parties de la ligne. Il faudra y penser pendant la planification de l'ordre des calques.

5.12.7 Configurations du graphique : référence rapide

Type de graphique

Pour sélectionner le type de graphique dans le dialogue Modifier Type (*capture d'écran ci-dessous*), choisir le type de graphique que vous souhaitez et cliquez sur **OK**. Le dialogue Modifier Type est accessible en cliquant sur le bouton **Modifier type**.

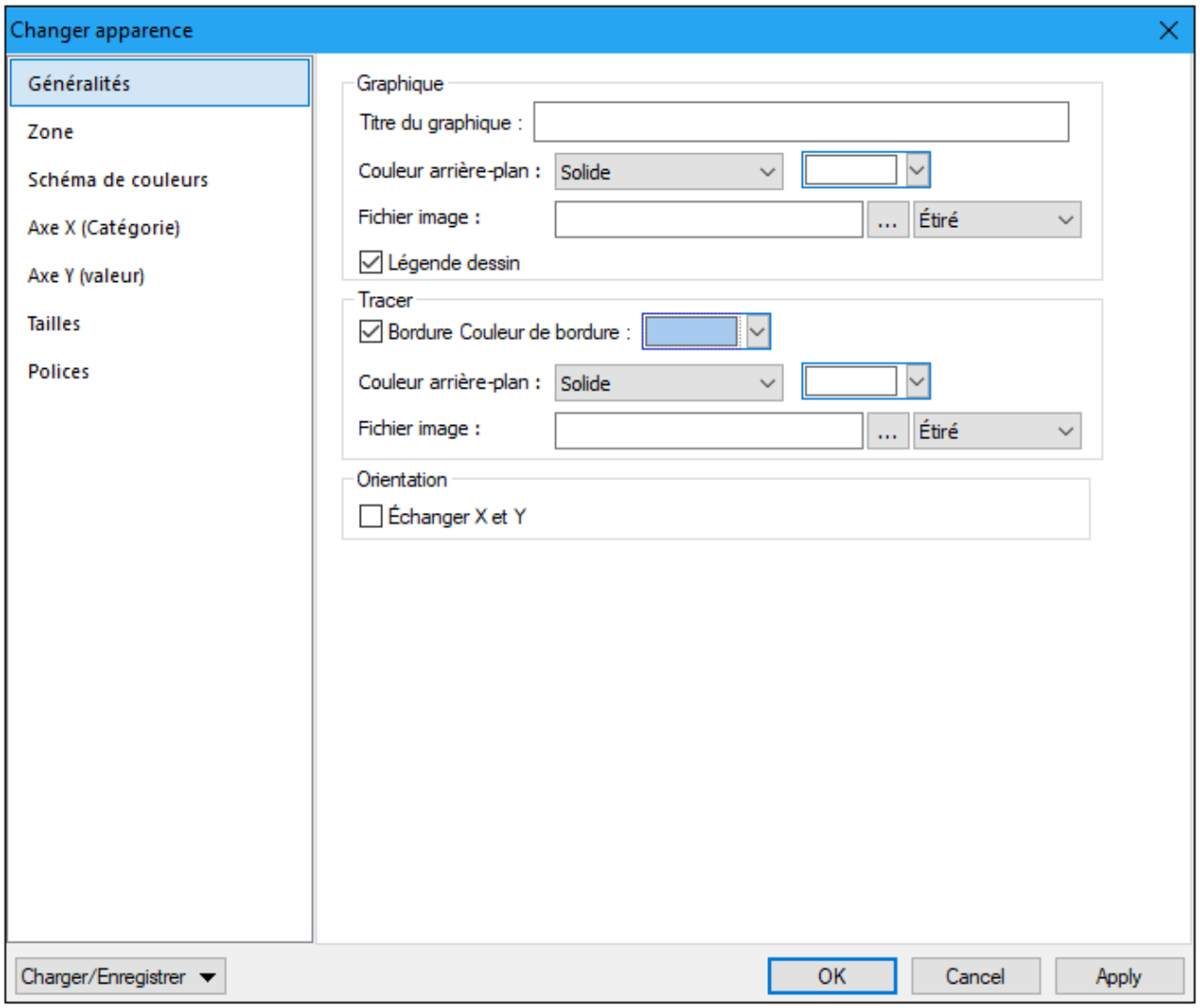
© 2016-2022 Altova GmbH



Une fois que le type de graphique a été sélectionné, les paramètres de graphique (comme le titre, la hauteur et la largeur) doivent être effectués dans le dialogue Changer apparence (*capture d'écran ci-dessous*) et les données de graphique doivent être spécifiées. La sélection des données pour chaque type de graphique est décrite dans les sections, Créer un graphique et Données de graphique.

Apparence du graphique

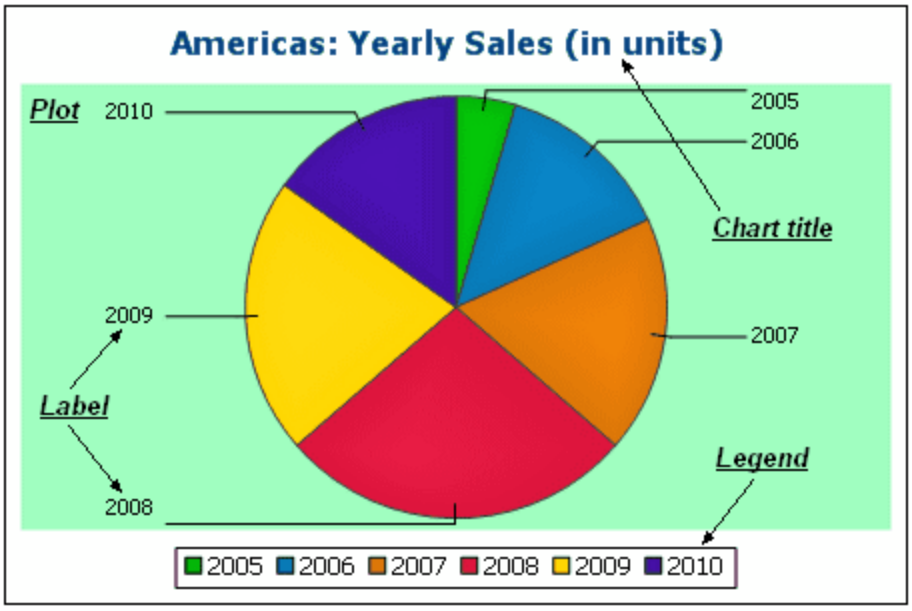
Les paramètres du graphique (comme le titre du graphique, le schéma de couleurs et la taille des polices) sont définis dans le dialogue Changer apparence (*capture d'écran ci-dessous, qui montre le dialogue Paramètres d'un graphique à barres*). Cliquer sur le bouton **Changer apparence** pour accéder au dialogue. Les paramètres diffèrent selon le type de graphique.



Les paramètres variés sont organisés dans les onglets Généraux suivants :

- ***Général :*** le titre de graphique (*voir capture d'écran ci-dessous*) peut être édité dans cet onglet, ainsi que la couleur d'arrière-plan du graphique et le rebord du canevas et la couleur d'arrière-plan. Dans la capture d'écran ci-dessous, le fond du canevas a été coloré en vert pâle. Les légendes se trouvent en bas du graphique ; elles expliquent les codes de couleur du graphique et peuvent être allumées ou cachées dans le dialogue.

© 2016-2022 Altova GmbH



- ***Schéma de couleurs :*** quatre schémas de couleur prédéfinis sont disponibles plus un schéma à définir par l'utilisateur. Vous pouvez modifier tous les schémas de couleur en ajoutant ou en supprimant des couleurs. Le schéma de couleur sélectionné dans l'onglet Schéma de couleur sera utilisé dans le graphique.
- ***Tailles :*** les tailles des différents aspects du graphique peuvent être définies, soit selon les pixels, soit en pourcentage.
- ***Polices d'écriture :*** les propriétés de police pour le titre, les légendes et les libellés du graphique peuvent être spécifiées dans cet onglet. Les tailles peuvent être définies en tant que pourcentage de la taille du graphique ou en tant que pixels.

En outre, chaque type de graphique a des paramètres spécifiques conformément à son type. Voir la liste ci-dessous :

- ***Camemberts :*** paramètres pour : (i) l'angle à partir duquel le premier secteur sera dessiné ; (ii) la direction dans laquelle les secteurs seront dessinés ; (iii) la couleur du pourtour ; (iv) une éventuelle surbrillance pour les couleurs (dans les camemberts 3D : utilisation éventuelle d'ombre portée et de transparence) ; (v) dessiner éventuellement des libellés ; et (vi) l'ajout éventuel de valeurs et de pourcentages aux libellés et combien de chiffres après la virgule doivent être ajoutés aux pourcentages.
- ***Graphiques à barres :*** paramètres pour : *(Général)* l'échange des axes X et Y génère un graphique horizontal (uniquement pour les graphiques à barres 2D) ; *(Barre)* les contours des barres et les ombres portées (ombres portées uniquement dans les graphiques 2D) ; *(Axe X)* libellé et couleur de l'axe x, et lignes verticales ; *(Axe Y)* libellé et couleur de l'axe y, lignes horizontales, la plage des valeurs à afficher, et les marques de graduation sur l'axe y ; *(Axe Z, uniquement 3D)* libellé et couleur de l'axe z ; *(3D)* l'inclinaison verticale, la rotation horizontale et la largeur de la vue.
- ***Graphiques à lignes :*** paramètres pour : *(Général)* l'échange des axes X et Y ; *(Ligne)* incluant les points du rebord ou pas ; *(Axe X)* libellé et couleur de l'axe x, et lignes verticales ; *(Axe Y)* libellé et couleur de l'axe y, lignes horizontales, la plage des valeurs à afficher, et les marques de graduation sur l'axe y.
- ***Jauge :*** paramètres pour : (i) l'angle auquel la jauge commence et la portée angulaire sur le cadran (*Jauge ronde uniquement*); (ii) la plage des valeurs à afficher (*Jauges rondes et à barre*); (iii) l'intervalle

et les couleurs des graduations principales et mineures (*Jauges rondes et à barre*); (iv) couleurs du cadran, de l'aiguille et du pourtour.

5.12.8 Paramètres et apparence des graphiques

Les paramètres de graphique sont organisés comme suit :

- Paramètres du graphique de base, les paramètres les plus basiques sont ceux du type de graphique. Pour sélectionner le type de graphique, cliquer sur **Modifier type** dans la barre outils de la fenêtre du graphique. Le dialogue Modifier type s'affiche.
- Paramètres du graphique avancés, qui vous permettent de changer l'apparence d'un graphique (titre, légende, couleurs, polices, etc.). Les paramètres avancés sont regroupés dans le dialogue Changer apparence. Pour accéder à ce dialogue, cliquer sur **Changer apparence** dans la barre outils de la fenêtre du graphique.

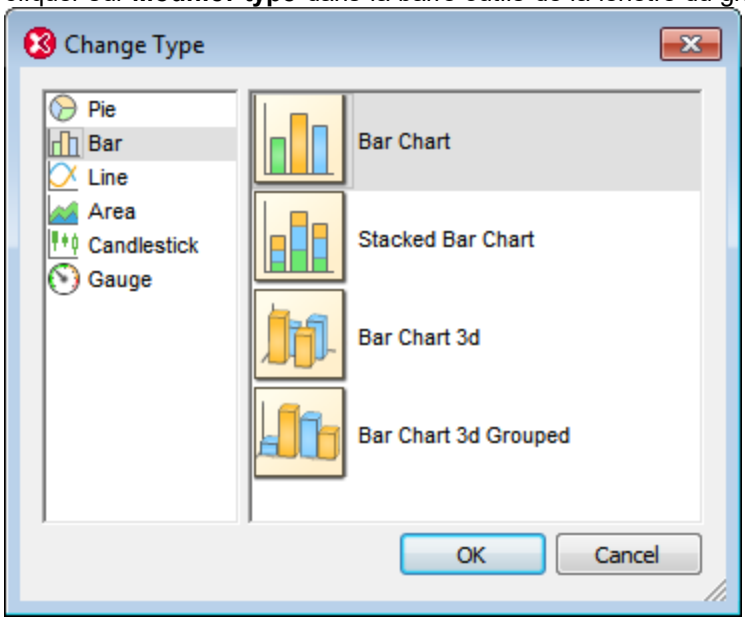
5.12.8.1 Paramètres graphiques de base

Cette section :

- Configurer le type de graphique
- Liste des types de graphiques
- Autres paramètres de base

Configurer le type de graphique

Les paramètres les plus basiques sont ceux du Type de graphique. Pour sélectionner le type de graphique, cliquer sur **Modifier type** dans la barre outils de la fenêtre du graphique.



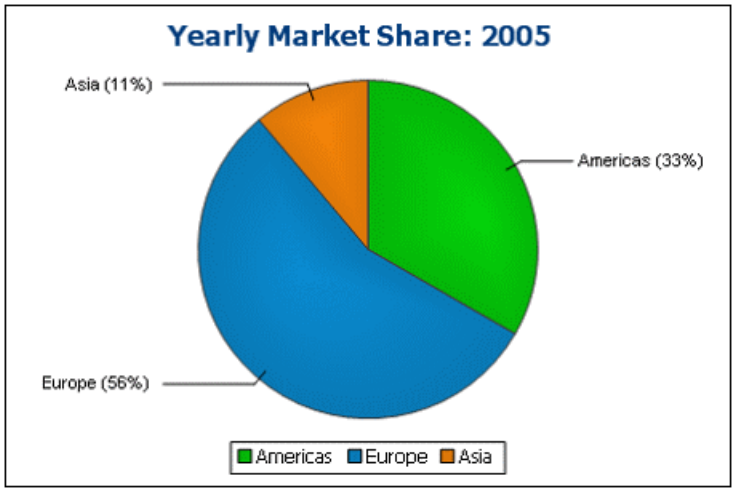
© 2016-2022 Altova GmbH

Types de graphiques

Les différents types de graphiques disponibles sont regroupés ci-dessous. Dans le dialogue Modifier type (*capture d'écran ci-dessus*), sélectionner le type de graphique que vous souhaitez et cliquer sur **OK**.

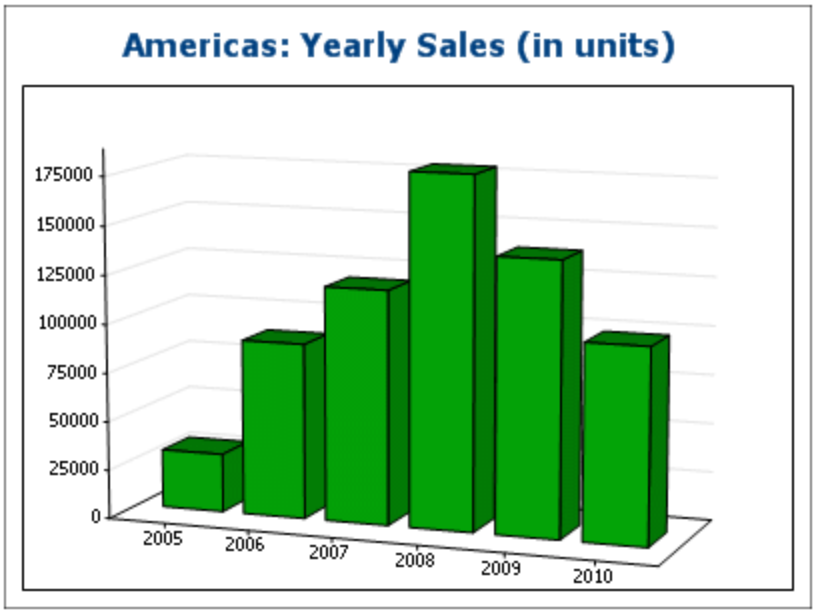
- *Camemberts*

  Dans les camemberts, une colonne/axe fournit les valeurs, une autre colonne/axe fournit les libellés pour ces valeurs. La colonne/l'axe de libellés peut prendre des valeurs non-numériques.

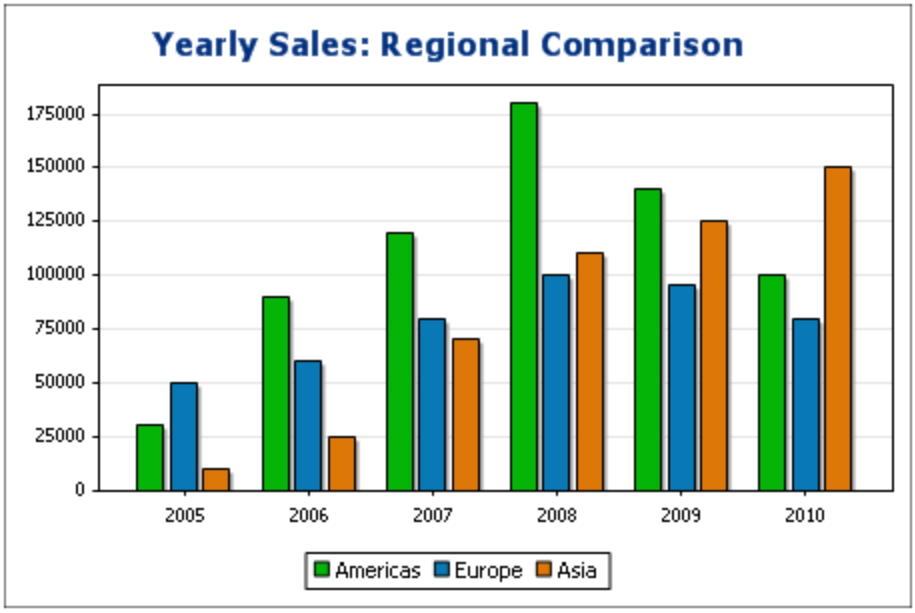


- *Graphique à barres*

  Les graphiques à barres peuvent présenter deux ensembles de valeurs réparties sur deux axes (*ci-dessous*).

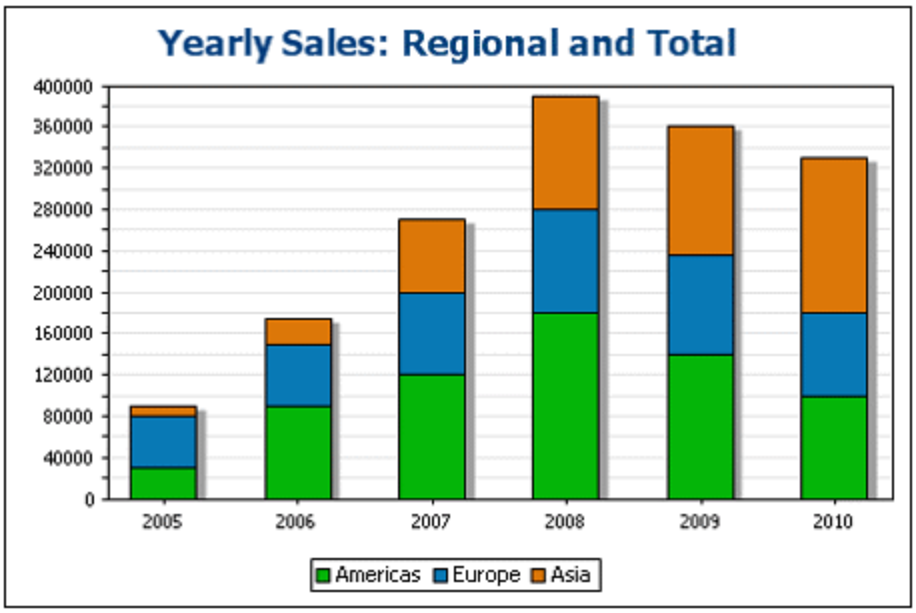


Ils peuvent aussi utiliser trois ensembles de valeurs, comme montré dans l'exemple ci-dessous : (i) continent, (ii) année, (iii) volume des ventes. Les graphiques à barres peuvent être affichés en 2D (*ci-dessous*) ou 3D (*ci-dessus*).



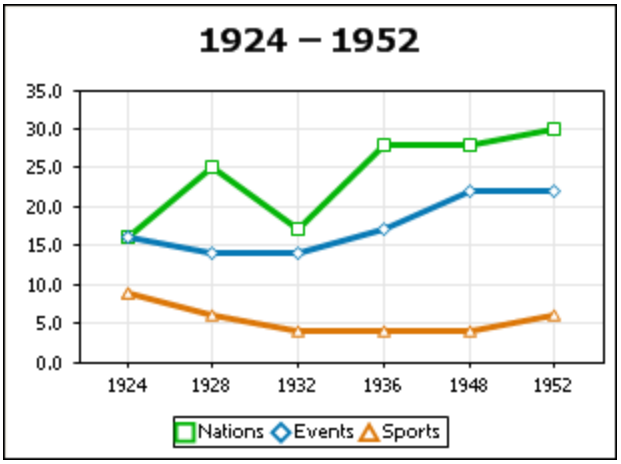
Un graphique à trois axes peut aussi être superposé si vous souhaitez présenter des totaux. Comparer le graphique superposé ci-dessous avec le graphique ci-dessus. Le graphique superposé montre le total des ventes sur tous les continents.

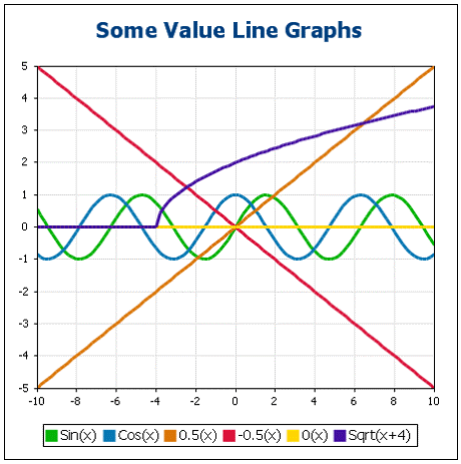
© 2016-2022 Altova GmbH



- *Graphiques à lignes*

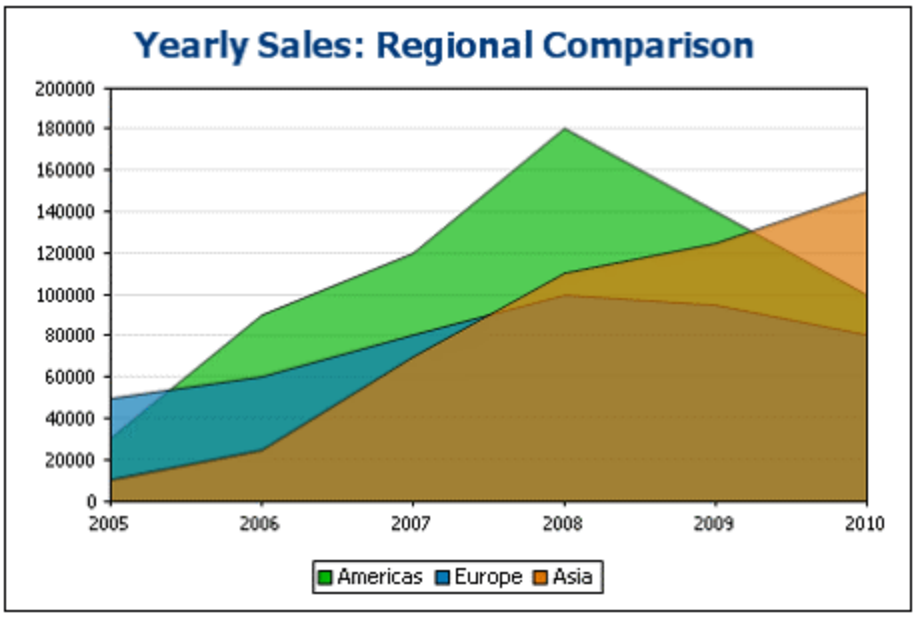
  La différence entre un graphique à lignes (*ci-dessous à gauche*) et un graphique à valeur de mesure (*ci-dessous à droite*) est que les graphiques à lignes de mesure ne prennent que des valeurs numériques pour l'axe X. Si vous souhaitez afficher des graphiques à ligne avec les valeurs de texte sur l'axe X, utiliser les graphiques à lignes.





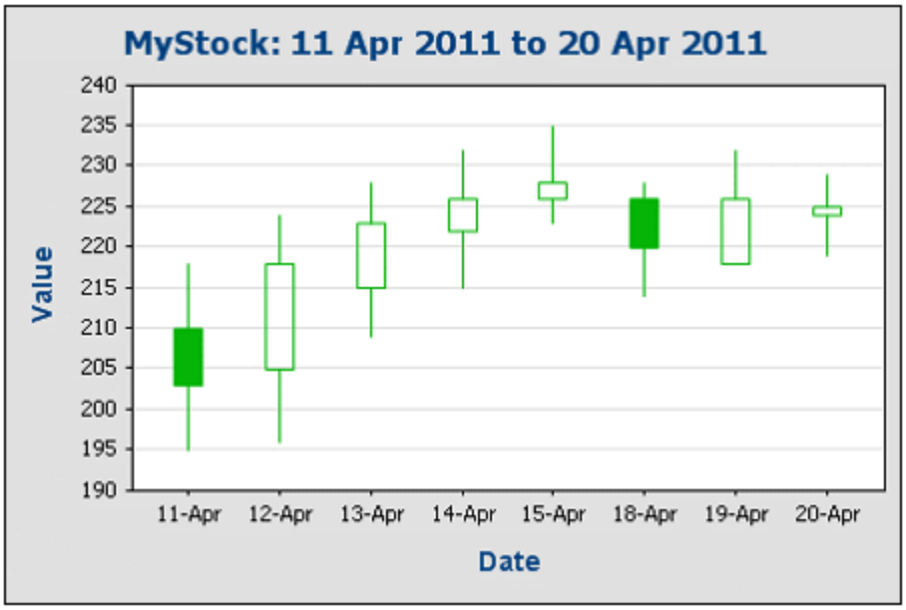
- *Graphiques à aires*

  Les graphiques à aires constituent une variation des graphiques à lignes, dans lesquels les aires situées en-dessous des lignes sont colorées. Les graphiques à aires peuvent être superposés (*voir graphique à barre ci-dessus*).



- *Graphiques à chandeliers*

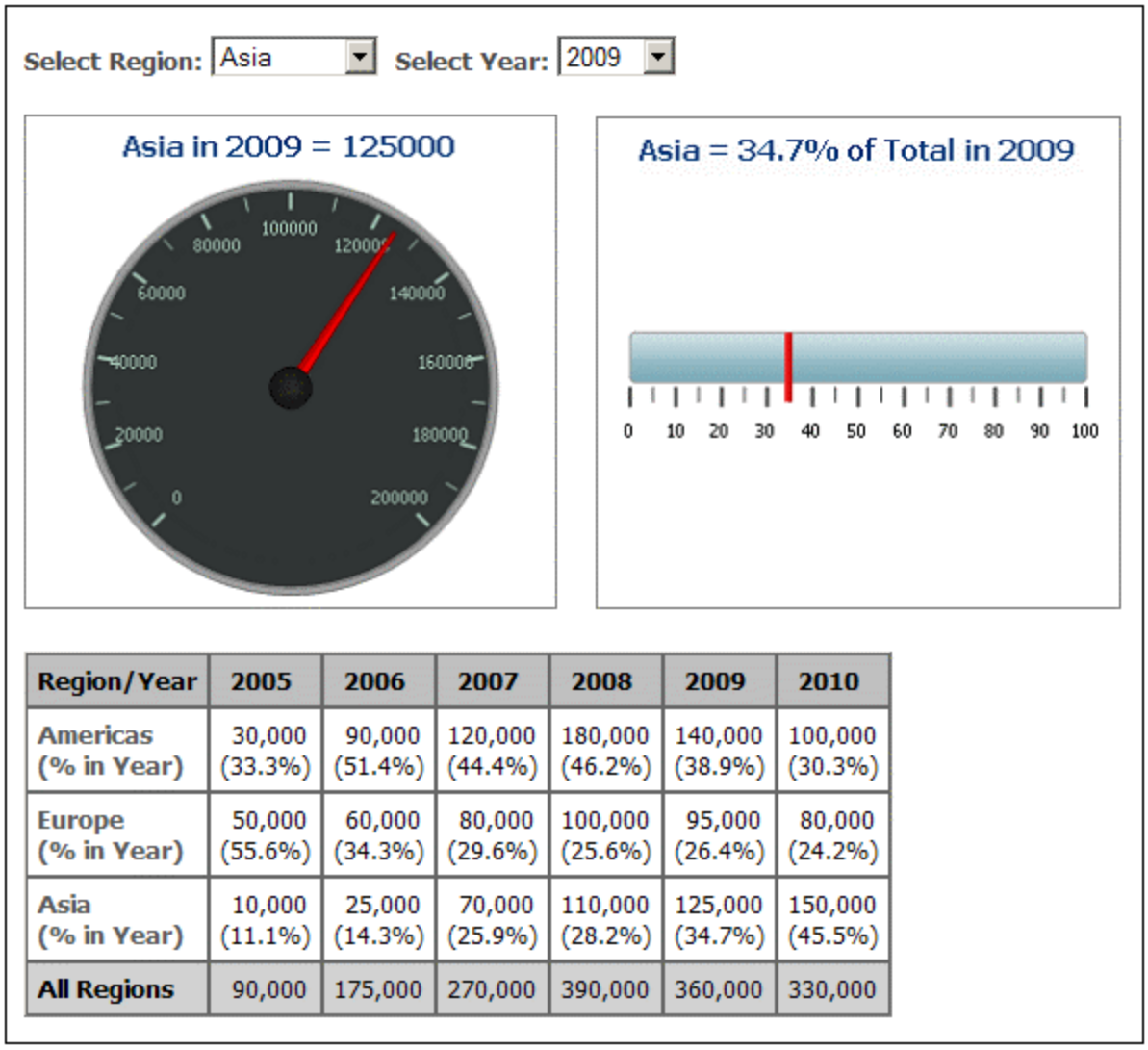
  Un graphique à chandeliers peut être utilisé pour illustrer l'évolution des prix de titres, de marchandises, de devises, etc. sur une certaine période. Le graphique indique non seulement la manière dont les prix se sont développés au fil du temps, la clôture, les prix les plus élevés et les plus bas de la journée et (éventuellement) l'ouverture. L'axe Y prend trois ou quatre séries (fermeture, haut, bas, ouverture (en option)). La capture d'écran ci-dessous montre un graphique à chandeliers à quatre séries.



© 2016-2022 Altova GmbH

- *Graphiques à jauge*

  Les graphiques à jauge sont utilisés pour illustrer une seule valeur et montrer sa relation par rapport à des valeurs minimum et maximum.



Region/Year	2005	2006	2007	2008	2009	2010
Americas (% in Year)	30,000 (33.3%)	90,000 (51.4%)	120,000 (44.4%)	180,000 (46.2%)	140,000 (38.9%)	100,000 (30.3%)
Europe (% in Year)	50,000 (55.6%)	60,000 (34.3%)	80,000 (29.6%)	100,000 (25.6%)	95,000 (26.4%)	80,000 (24.2%)
Asia (% in Year)	10,000 (11.1%)	25,000 (14.3%)	70,000 (25.9%)	110,000 (28.2%)	125,000 (34.7%)	150,000 (45.5%)
All Regions	90,000	175,000	270,000	390,000	360,000	330,000

Autres paramètres de base

Dans le panneau Paramètres du graphique, vous pouvez aussi définir le titre du graphique (*voir capture d'écran ci-dessous*).

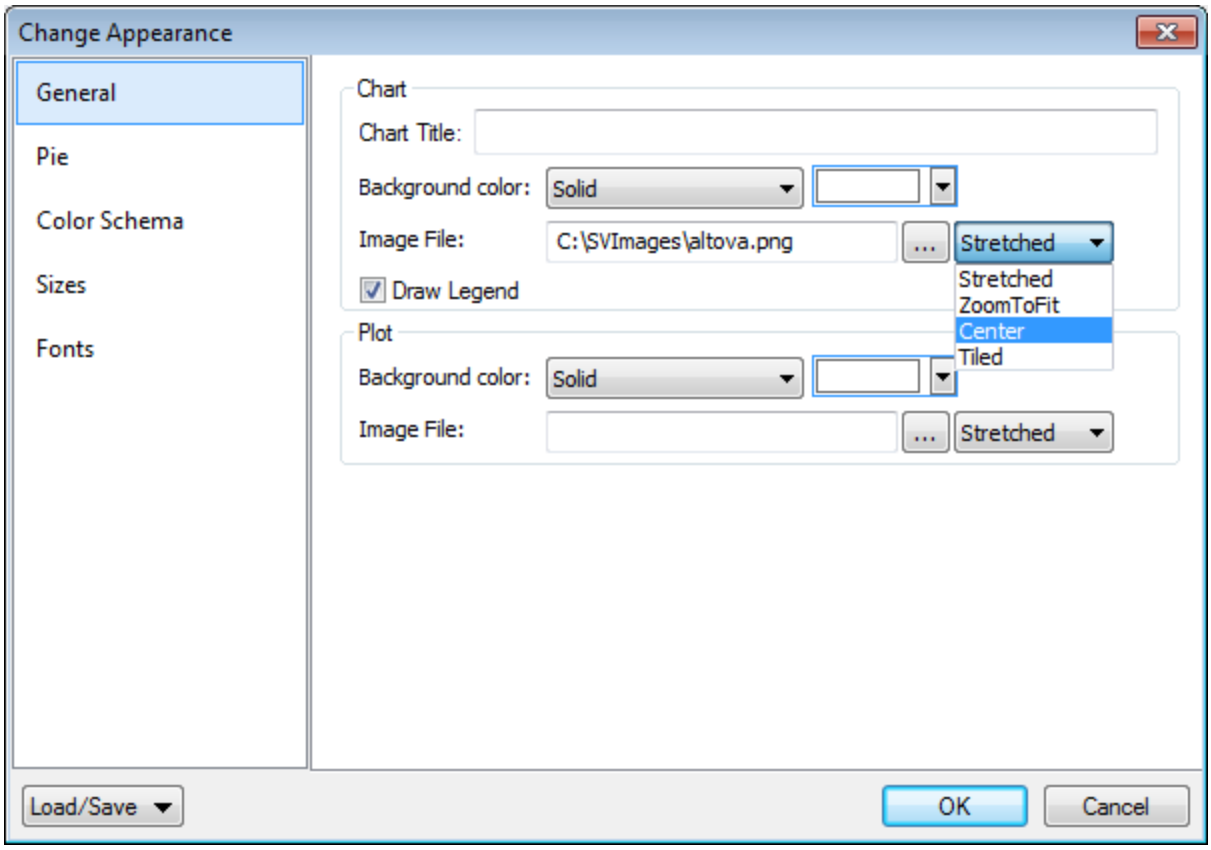
5.12.8.2 Paramètres graphiques avancés

Cette section :

- Accéder aux paramètres avancés
- Aperçu des paramètres avancés
- Charger, enregistrer, réinitialiser les paramètres de graphique

Accéder aux paramètres avancés

Pour accéder aux paramètres avancés de graphique, procéder comme suit : Cliquer sur **Changer apparence** dans la barre d'outils de la fenêtre du graphique. Le dialogue Changer apparence pour ce type de graphique s'affiche (*la capture d'écran montre le dialogue Changer apparence d'un camembert*).



Aperçu des paramètres avancés

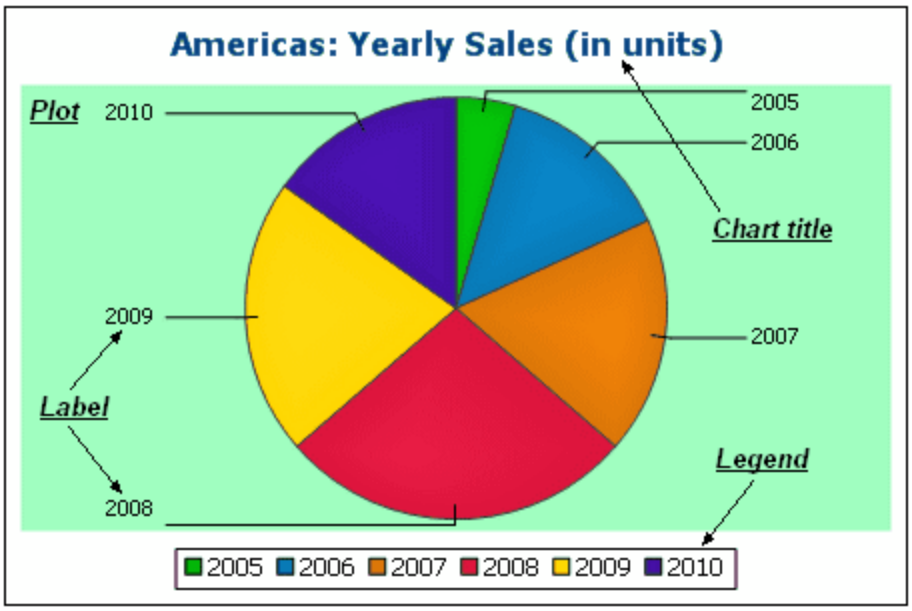
Les paramètres avancés sont répartis en onglets. Il y a le groupe des onglets communs pour tous les types de graphiques et les onglets spécifiques à un seul type de graphique.

Paramètre de graphiques communs

▼ *Général*

© 2016-2022 Altova GmbH

Le titre du graphique (*voir capture d'écran ci-dessous*) est le même que dans les paramètres de base (*voir ci-dessus*) et peut aussi être modifié en tant qu'un paramètre avancé. D'autres paramètres dans ce dialogue sont la couleur d'arrière-plan du graphique et le canevas. Dans la capture d'écran ci-dessous, le canevas a reçu une couleur d'arrière-plan vert clair. Un fichier d'image peut aussi être défini en tant qu'image d'arrière-plan du graphique et/ou du canevas. Cette image peut être étirée pour recouvrir toute la surface du graphique ; elle peut être zoomée à la taille de la fenêtre pour que le zoom corresponde à une des deux dimensions (du graphique/canevas); elle peut être centrée ou mise en mosaïque. La légende est la clé du code de couleurs dans le graphique, et elle peut être allumée ou éteinte.



- *Schéma de couleurs*

  Quatre schémas de couleur prédéfinis sont disponibles ainsi qu'un schéma de couleur défini par l'utilisateur. Vous pouvez modifier les couleurs de la palette en ajoutant et/ou en supprimant des couleurs d'un schéma. Le schéma de couleur sélectionné dans cet onglet sera utilisé dans le graphique.

- *Tailles*

  La taille du graphique peut être réglée, soit sur la base de pixels soit sur la base d'un pourcentage.

- *Police*

  Les propriétés de polices du titre du graphique et des légendes et des libellés peuvent être spécifiées dans cet onglet. Les tailles peuvent être définies en tant que pourcentage du graphique ou absolument sur la base de points.

- *Bouton Charger/Enregistrer*

  Les paramètres peuvent être enregistrés dans un fichier XML et peuvent être chargés depuis un fichier XML si celui-ci a une structure correcte. Pour voir la structure, enregistrer les paramètres d'un graphique

et puis ouvrir le fichier XML. Cliquer sur ce bouton pour accéder à l'option de réinitialisation des paramètres de graphiques sur le mode défaut.

Paramètres de graphique spécifiques au type

- *Camemberts*

  Paramètres pour : (i) l'angle à partir duquel la première tranche doit être tracée ; (ii) la direction dans laquelle les tranches doivent être tracées ; (iii) la couleur du canevas ; (iv) si les couleurs reçoivent des zones de surbrillance (dans les camemberts 3D : si des ombres portées et la transparence sont utilisés); (v) si des libellés doivent être tracés ; et (vi) si les valeurs et les pourcentages doivent être ajoutés aux libellés et combien de chiffres après la virgule doivent être ajoutés aux pourcentages.

- *Graphiques à barres*

  Paramètres pour : *(Général)* L'inversion du tracé des axes X et Y génère un graphique à barres horizontales ; *(Barres)* Tracé des barres et ombres portées (ombres portées dans les graphiques à barres 2D uniquement) ; *(axe X)* Libellé et couleur de l'axe X et lignes de grille verticales ; *(axe Y)* Libellé et couleur de l'axe Y, lignes de grille verticales, la plage des valeurs à afficher et les graduations sur l'axe Y ; *(axe Z, 3D uniquement)* Libellé et couleur de l'axe Z ; *(3D)* l'inclinaison verticale, la rotation horizontale et la largeur de la vue.

- *Graphiques à lignes*

  Paramètres pour : *(Général)* L'inversion du tracé des axes X et Y; *(Ligne)* comprenant les points de tracé ou pas; *(axe X)* Libellé et couleur de l'axe X, et lignes de grille verticales ; *(axe Y)* Libellé et couleur de l'axe Y, lignes de grille verticales, la plage des valeurs à afficher et les graduations sur l'axe Y.

- *Graphiques à jauge*

  Paramètres pour : (i) l'angle sous lequel la jauge commence et l'étendue angulaire de l'échelle ; (ii) la plage des valeurs affichées ; (iii) l'intervalle et les couleurs des graduations principales et mineures ; (iv) les couleurs du cadran, de l'aiguille et de la bordure.

- *Graphiques à aires*

  La transparence des zones peut être définie en tant qu'une valeur allant de 0 (aucune transparence) à 255 (transparence maximum). Dans le cas des graphiques à aires non superposées, la transparence permet de rendre la partie des aires se trouvant sous d'autres aires visibles pour l'utilisateur. Il est également possible de spécifier des bordures pour les aires.

- *Graphiques à chandeliers*

  La couleur de remplissage peut être spécifiée pour les deux situations : (i) lorsque la valeur de fermeture est plus importante que la valeur d'ouverture, et (ii) si les valeurs d'ouverture sont plus importantes que les valeurs de fermeture. Dans le deuxième cas, la couleur de série est également disponible en option. La couleur de série est spécifiée dans l'onglet Schéma de couleur du dialogue Changer apparence.

© 2016-2022 Altova GmbH

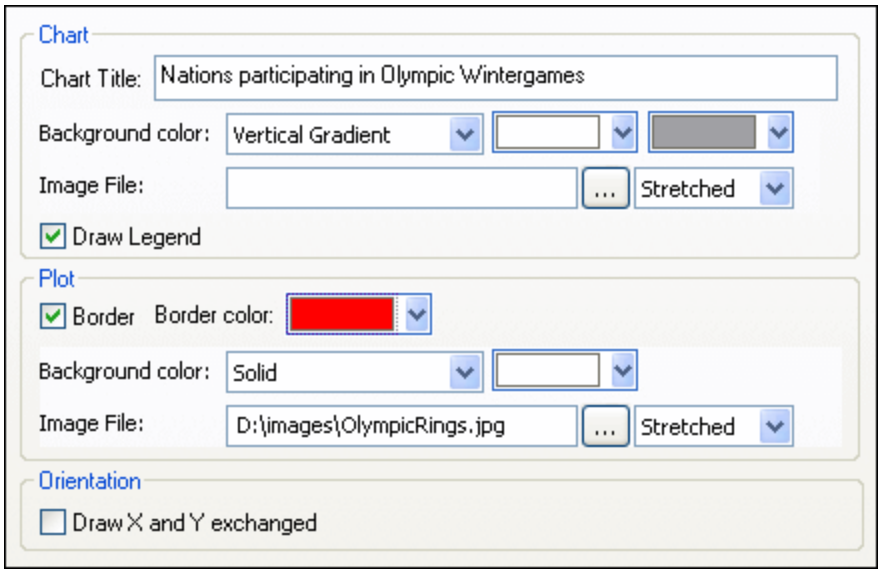
Charger, enregistrer, réinitialiser les paramètres de graphique

Les paramètres de graphique qui sont différents des paramètres par défaut peuvent être enregistrés dans un fichier XML. Ces paramètres peuvent ensuite être chargés en tant que les paramètres d'un graphique ; cela peut vous aider à économiser du temps et des efforts. Le bouton **Charger/Enregistrer** (voir première capture d'écran dans cette section) fournit les options suivantes sur clic :

- **Définir sur défaut** : rejette les modifications effectuées aux paramètres et restaure les paramètres par défaut de **toutes** les sections de paramètres.
- **Charger à partir du fichier** : permet d'importer des paramètres précédemment enregistrés dans un fichier XML (*voir commande suivante*). La commande affiche le dialogue Ouvrir, dans lequel vous pouvez saisir l'emplacement du fichier requis.
- **Enregistrer sous fichier** : ouvre le dialogue Enregistrer sous. Vous pouvez spécifier un fichier XML dans lequel vous pouvez enregistrer les paramètres. Ce fichier recense les paramètres qui diffèrent depuis les paramètres par défaut.

5.12.8.2.1 Général

La section Générale du dialogue Changer apparence vous permet de définir le titre du graphique, d'ajouter ou de supprimer une légende et de définir des images et des couleurs d'arrière-plan et l'orientation du graphique pour les graphiques à barres, à lignes, à aires et à chandelles.



Graphique

Saisir un titre descriptif pour votre graphique dans le champ `Titre du graphique` et choisir une couleur d'arrière-plan pour tout le graphique depuis la liste déroulante. Vous pouvez choisir un arrière-plan solide, un

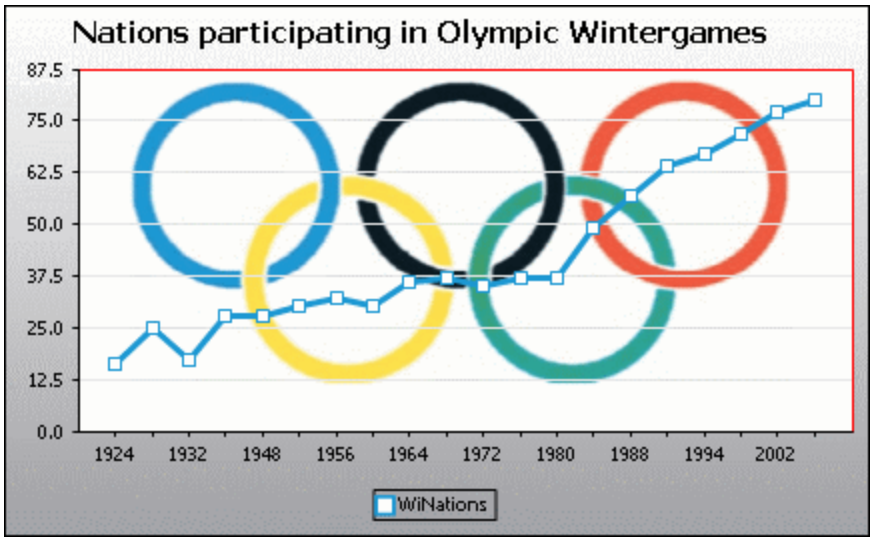
dégradé vertical ou horizontal et définir les couleurs de début et de fin pour le dégradé, le cas échéant. De plus, au lieu de prendre un arrière-plan coloré, vous pouvez définir une image d'arrière-plan et choisir une des options d'affichage disponibles à partir de la liste déroulante :

- Étiré : l'image sera étirée à la hauteur et la largeur du graphique
- Ajusté à la fenêtre : l'image sera ajustée au cadre du graphique et le rapport de l'aspect de l'image sera conservé
- Centré : l'image sera affichée à sa taille originale au centre du graphique
- Mosaïque : si l'image est plus petite que le graphique, des copies de l'image seront affichées pour remplir le fond de l'arrière-plan

La case à cocher `Légende dessin` est activée par défaut, décocher la case si vous ne souhaitez pas afficher de légende dans votre graphique.

Canevas

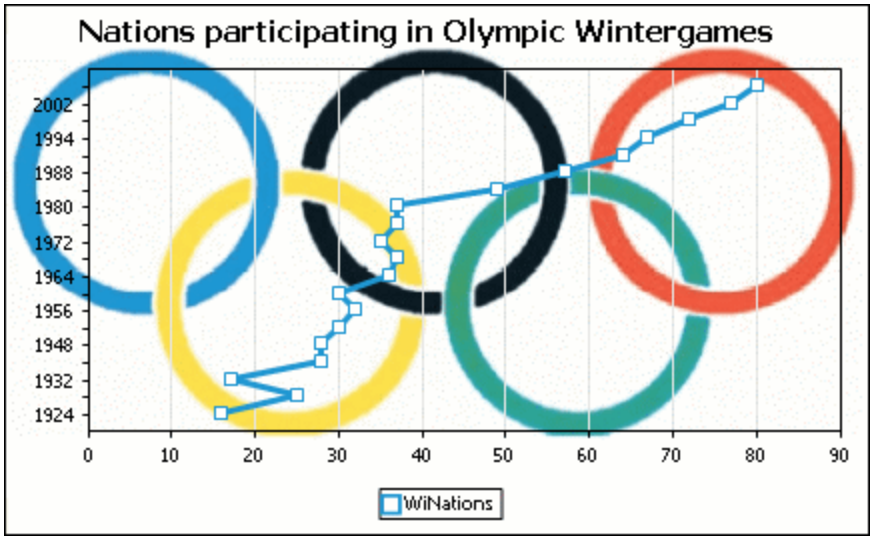
Le Canevas est la surface sur laquelle les données du graphique sont affichées. Vous pouvez tracer une bordure autour du canevas et spécifier une couleur d'arrière-plan différente et/ou une image pour la surface du canevas. Dans la capture d'écran ci-dessous, la couleur d'arrière-plan du graphique a été modifiée en gris (dégradé vertical). Le canevas, quant à lui, reste blanc. Il est délimité par une bordure rouge et une image d'arrière-plan a été ajoutée.



Orientation

Si vous détenez une petite série de grandes valeurs, il peut s'avérer avantageux d'échanger les axes X et Y pour une meilleure illustration. Veuillez noter que dans la capture d'écran ci-dessous, la couleur d'arrière-plan a été définie sur "Transparent" et l'image d'arrière-plan a été appliquée à tout le graphique.

© 2016-2022 Altova GmbH



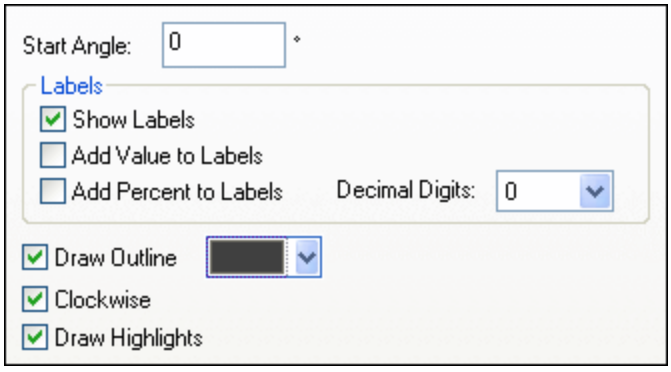
Veuillez noter que cette option n'est pas disponible pour les camemberts et les graphiques à jauge.

5.12.8.2.2 Fonctions spécifiques au type

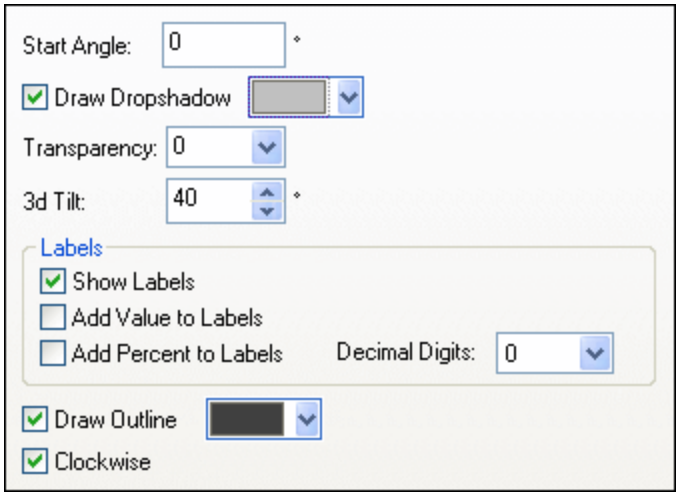
Pour chacun des types de graphiques, même pour les sous-types différents, le dialogue Changer apparence contient une section dans laquelle vous pouvez définir les fonctions spécifiques au type du graphique.

Camembert

La plupart des paramètres sont identiques pour les versions 2D et 3D. Dans les camemberts 2D, vous pouvez rajouter des surbrillances.

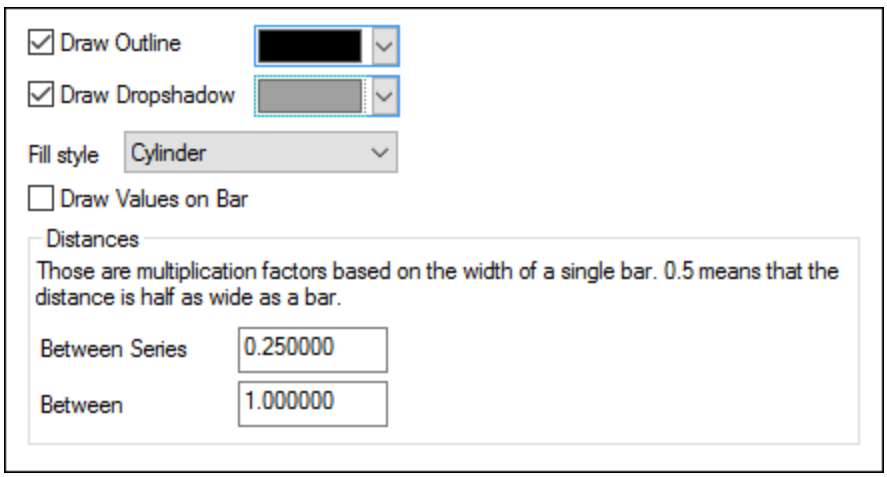


Dans les camemberts 3D, vous pouvez rajouter des ombres portées, ajouter une transparence et définir l'inclinaison 3D.



La valeur `Angle de début` définit l'endroit où la première ligne des colonnes sélectionnées sera affichée dans le graphique. Un angle de 0 degrés correspond à midi/minuit sur une montre.
À la place des légendes, ou en plus d'elles, vous pouvez afficher des libellés, ajouter des valeurs et/ou des pourcentages aux libellés et définir le nombre de chiffres décimaux à afficher.
La couleur que vous pouvez sélectionner à côté de la case à cocher `Dessiner bordure` est utilisée pour tracer en option une bordure autour du graphique et des segments individuels du camembert. La case à cocher `Dans le sens des aiguilles d'une montre` vous permet de spécifier si les lignes devraient être classées dans le sens des aiguilles d'une montre ou dans le sens inverse.
Dans les camemberts 3D, vous pouvez intégrer une ombre portée et définir sa couleur, ajouter une transparence au graphique et définir l'inclinaison 3D. Dans les graphiques 2D, l'option `Dessiner des surbrillances` ajoute une structure supplémentaire au graphique.

Graphique à barres

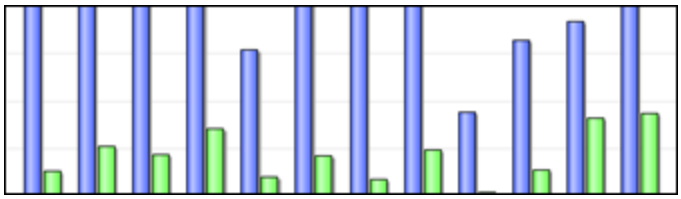


Pour les graphiques à barres, vous pouvez procéder aux configurations suivantes :

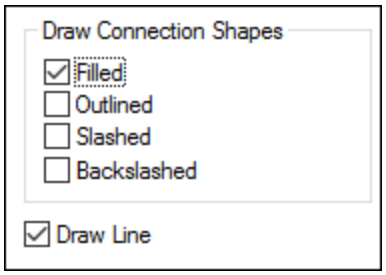
- Ajouter une bordure aux barres et définir sa couleur.

 © 2016-2022 Altova GmbH

- Dans les graphiques à barres 2D, vous pouvez également dessiner une ombre portée et définir sa couleur (cette option n'est pas disponible pour les graphiques à barres 3D).
- Par défaut, la forme des barres ressemble à un cylindre, mais vous pouvez choisir les options "Dégradé vertical" ou "Solide" depuis la liste de choix `Style de remplissage` (cette option est uniquement disponible pour les graphiques à barres 2D).
- Les valeurs d'une barre (correspondant à la hauteur de la barre sur l'axe Y) peuvent être dessinées sur la barre. La police des valeurs peut être spécifiée dans les paramètres *Polices* (cette option est uniquement disponible pour les graphiques à barres 2D, pas empilés).
- La distance entre les séries d'un groupe de barre et entre les groupes de barre peut être spécifiée en tant qu'une fraction décimale de la largeur d'une seule barre. Par exemple, dans la capture d'écran ci-dessous, qui montre les groupes de barre qui consistent chacune en une série bleue et une série verte, la distance entre les série a été définie à 25% (`=0.25`) de la largeur de la barre ; la distance entre les groupes de barre a été réglée à 100% (`=1.0`) de la largeur de la barre. (cette option est uniquement disponible pour les graphiques à barres 2D).



Graphique à lignes



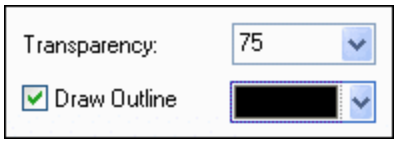
Pour dessiner les marques de connexion qui représentent les valeurs dans les graphiques de ligne, il vous faut activer au moins une case à cocher dans la fenêtre Dessiner des marques de connexion. Vous pouvez utiliser cinq formes différentes pour marquer une série : carré, rhombe, triangle, triangle inversé et cercle. Si votre graphique contient plus de cinq séries, vous pouvez combiner les marques de connexion en sélectionnant plus d'une option dans la fenêtre Dessiner des marques de connexion. Dans la capture d'écran ci-dessous, les options `Remplie` et `Oblique` ont été sélectionnées et le type `Oblique` est utilisé pour la sixième série et plus.

L'option *Dessiner la ligne* permet au graphique d'être tracé avec (i) uniquement des formes de connexion, ou (ii) avec des formes de connexion reliées par une ligne.

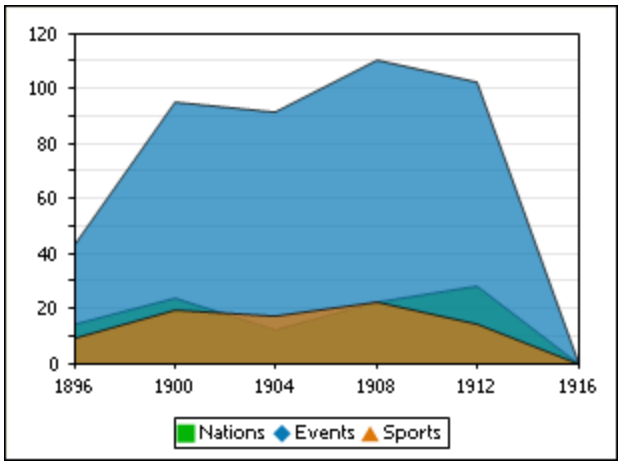


Les marques de connexion sont disponibles pour les graphiques de lignes et les graphiques de ligne de valeur.

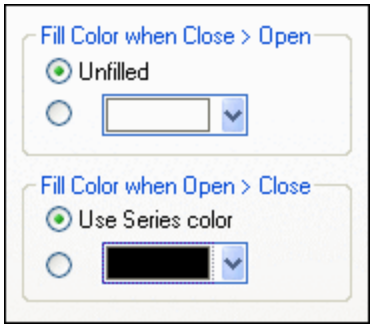
Graphique à aires



Parmi les propriétés que vous pouvez modifier pour les graphiques à aires, vous pouvez jouer avec la transparence ; cela vous permet d'empêcher qu'une série en dissimule une autre dans le graphique. De plus, vous pouvez ajouter une bordure aux aires individuelles et définir sa couleur (*voir capture d'écran ci-dessous*).



Graphique à chandeliers



Si les valeurs d'ouverture et de fermeture sont définies en tant que séries, vous pouvez choisir les couleurs et si vous souhaitez ou non que les chandeliers soient remplis si la valeur de fermeture est plus importante que la valeur d'ouverture.

© 2016-2022 Altova GmbH

Graphique à jauge



La valeur `Début` de la fenêtre des Angles définit la position de la marque 0 et la valeur `Étendue` est l'angle utilisé pour l'affichage. Dans la fenêtre Plage de valeur, vous pouvez définir les valeurs minimum et maximum à afficher. Les marques de graduation sont affichées avec (graduations majeures) ou sans (graduations mineures) la valeur correspondante ; elles peuvent présenter des couleurs séparées. Dans la fenêtre Couleurs, vous pouvez définir des couleurs pour le remplissage du cadran, de l'aiguille, de la base de l'aiguille (dissimulée par première partie de l'aiguille au centre du graphique) et la bordure qui entoure le graphique. La valeur actuelle et un libellé supplémentaire peut être affiché sous n'importe quel angle.

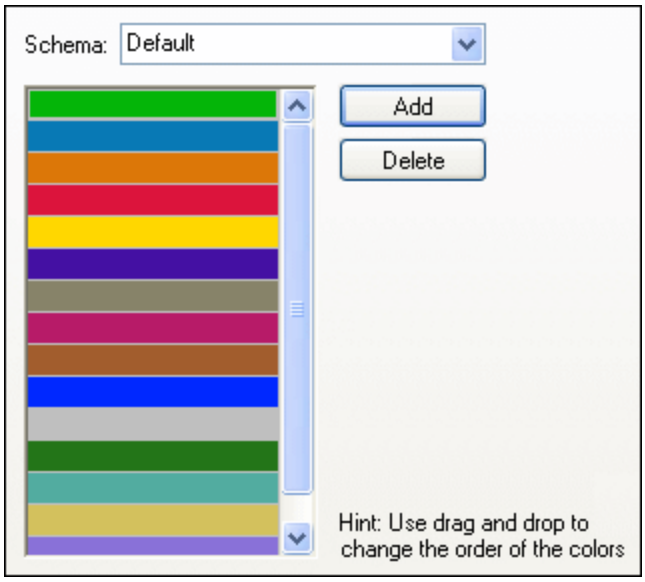
5.12.8.2.3 Couleurs

Selon le type de graphique que vous avez sélectionné, XMLSpy propose deux sections différentes pour la définition des couleurs à utiliser dans les graphiques :

- Schéma de couleurs pour les camemberts, les graphiques à barres, à lignes, à aires et à chandeliers
- Plage de couleurs pour les graphiques à jauge

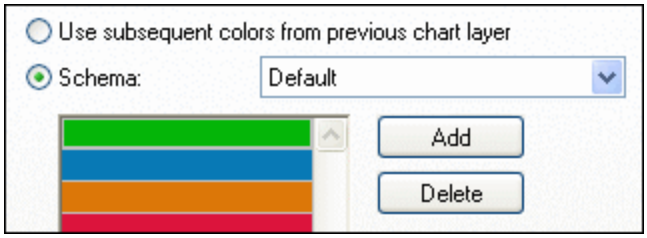
Schéma de couleurs

La section Schéma de couleur du dialogue Changer apparence offre quatre schémas de couleurs prédéfinis (défaut, nuances de gris, coloré et pastel) qui peuvent être personnalisés ; vous pouvez également entièrement définir votre propre schéma de couleurs.



La couleur supérieure sera utilisée pour la première série, la deuxième couleur pour la deuxième série, etc. Vous pouvez changer l'ordre des couleurs en sélectionnant un coloris et en le glissant sur sa nouvelle position avec la souris. Vous pouvez changer l'ordre des couleurs en sélectionnant une couleur et en la glissant sur sa nouvelle position avec la souris. Pour ajouter une nouvelle couleur ou retirer une couleur indésirable, cliquer sur le bouton correspondant. Dans les graphiques à chandeliers, seule la première couleur sera utilisée.

Si vous avez ajouté une ou plusieurs couches de graphiques superposés dans une fenêtre de Graphiques, la section Colorier schéma du dialogue Changer apparence contient le bouton radio supplémentaire `Utiliser les couleurs suivantes du calque de graphique précédent` qui est activé par défaut.



Lorsque le bouton radio est activé, le schéma de couleurs de la couche précédente sera utilisé et vous ne pourrez pas choisir un schéma de couleurs séparé pour la superposition. La série de la couche active sera affichée en utilisant les couleurs suivantes à partir du schéma de couleurs de la couche précédente. De cette manière, toutes les séries de la fenêtre Graphiques auront une couleur différente et pourront donc être distinguées plus facilement.

© 2016-2022 Altova GmbH

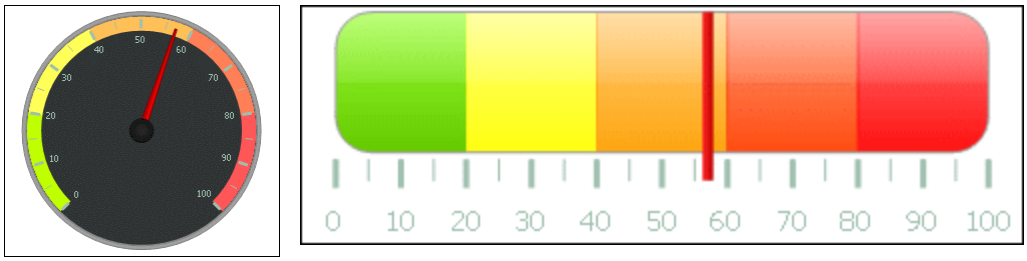
Vous pouvez interrompre ce lien pour tous les niveaux supplémentaires que vous ajoutez et choisir des schémas de couleurs différents qui peuvent ensuite être réutilisés dans des couches ultérieures.

Plage de couleurs

Dans les graphiques à jauge, vous pouvez personnaliser l'apparence de la jauge en appliquant des couleurs à certaines plages de valeur.

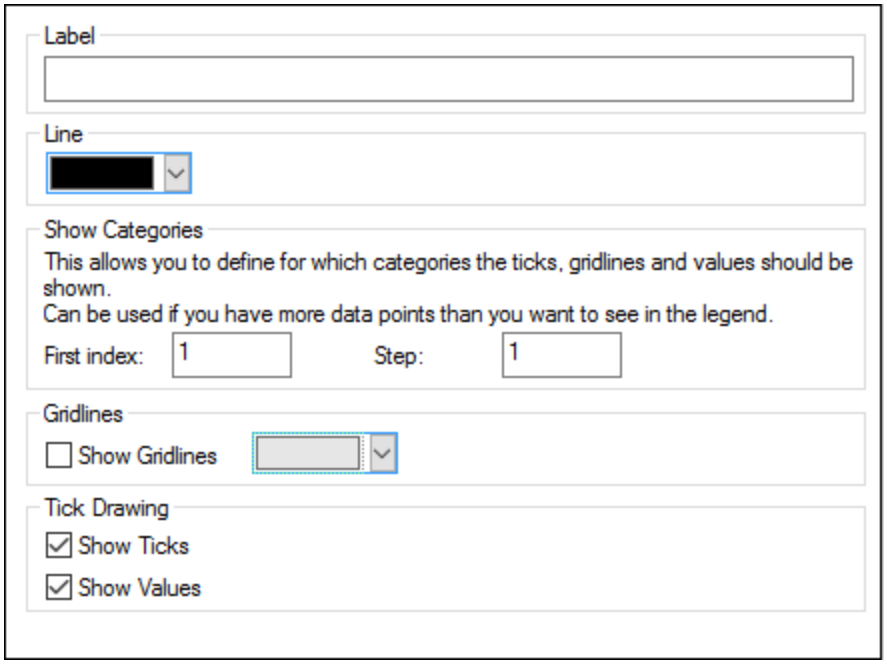


La définition affichée dans la capture d'écran ci-dessus apparaîtra dans les graphiques à jauge suivants :



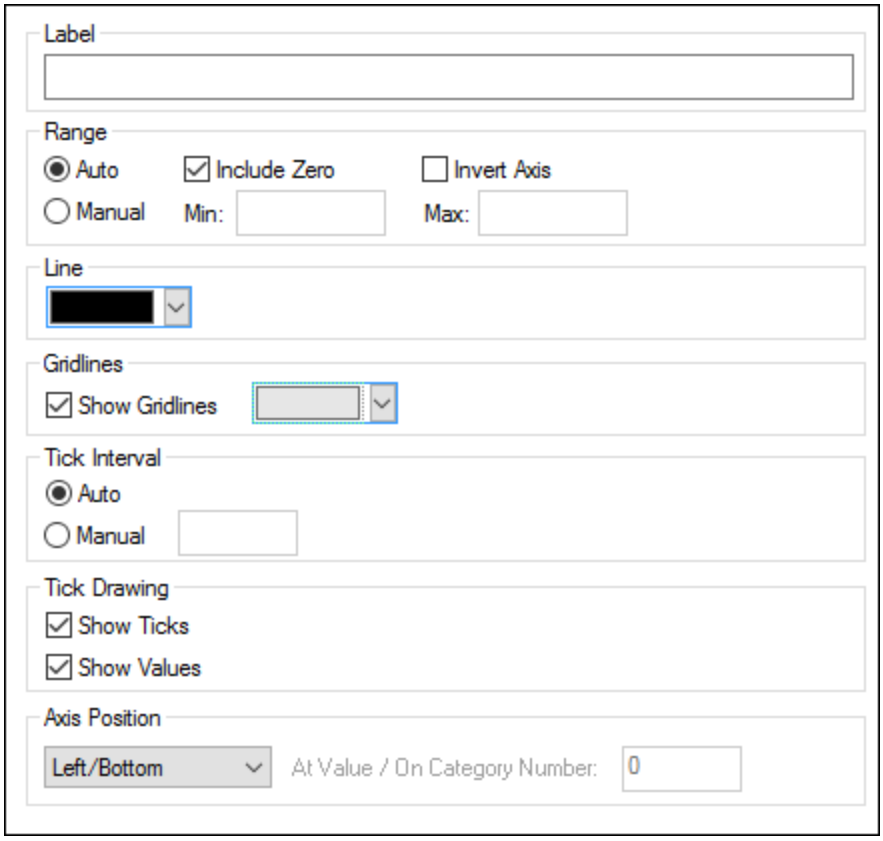
5.12.8.2.4 Axe X

Dans la section de l'Axe X du dialogue Changer apparence, vous pouvez saisir un libellé pour l'axe, et définir les couleurs pour l'axe et les quadrillages (si affiché). Vous pouvez également définir si vous voulez afficher les graduations et les valeurs d'axe ou non. Cette section est identique pour tous les graphiques à barres, à ligne, à aires et à chandeliers. Les options *Afficher catégories* vous permettent de spécifier qu'un seul sous-ensemble de toutes les catégories (valeurs Axe X) sont affichées, c'est à dire que seules les coches, les lignes de grillage et les valeurs des catégories sélectionnées seront affichées. Créer le sous-ensemble des catégories affichées en saisissant (i) l'index de la première valeur à afficher, et (ii) le nombre de graduations par étapes. Par exemple, si vous disposez de 101 catégories, en partant de `1900, 1901, 1902 ... 1999, 2000`, vous pourrez afficher un an sur dix de `1900` à `2000` en configurant *Première graduation sur* `1` et *Étape* sur `10`.



Néanmoins, dans les Graphiques à valeur de mesure, vous pouvez aussi définir la plage de valeur et définir à quel intervalle les graduations doivent être placées.

© 2016-2022 Altova GmbH



Libellé
Le texte saisi dans le champ `Libellé` sera imprimé sous l'axe en guise de la description de l'axe X.

Plage
Par défaut, le bouton radio `Auto` est sélectionné dans la fenêtre Plage. Si vous souhaitez afficher un fragment du graphique avec plus de détails, activer le bouton `Manuel` et saisir les valeurs minimum et maximum dans les champs respectifs.
Si la colonne utilisée pour l'axe X n'inclut pas zéro, vous pouvez désactiver la case à cocher `Inclure zéro` et l'axe X commencera avec la valeur minimum disponible dans la série. L'option `Axe inversé` vous permet d'invertir les valeurs de l'axe X. Par exemple, si les valeurs sont exécutées de `0` à `360`, la sélection de cette option générera l'axe X de manière à ce que `360` est à l'origine et que les valeurs tendent vers `0` lorsque l'axe X monte.

Ligne
L'axe est affiché dans la couleur que vous choisissez depuis la liste déroulante `Ligne`. Vous pouvez utiliser une des couleurs présélectionnées ou cliquer sur le bouton **Autre couleur...** pour choisir une couleur standard ou définir une couleur personnalisée. Cliquer sur le bouton **Choisir...** sur l'onglet de personnalisation et utiliser la pipette pour choisir une couleur qui s'affichera à un endroit de votre écran.

Quadrillages
Si la case à cocher `Afficher quadrillage` est activée, vous pouvez choisir une couleur à partir de la fenêtre de la liste déroulante correspondante.

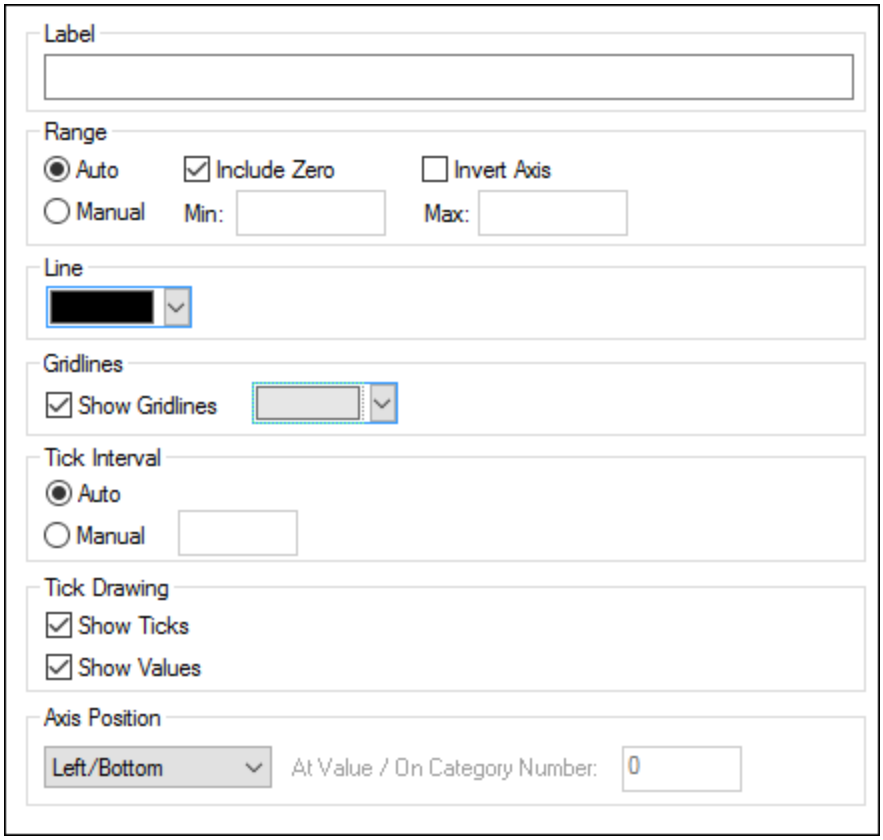
Intervalle des graduations
Si vous n'êtes pas satisfait des graduations par défaut, vous pouvez activer le bouton radio `Manuel` dans la fenêtre Intervalle des graduations et saisir la différence entre les graduations individuelles dans le champ correspondant.

Dessin graduations
Vous pouvez faire passer l'affichage des graduations sur l'axe et/ou les valeurs de l'axe sur marche ou arrêt.

Position de l'axe
À partir de la liste déroulante, vous pouvez choisir la position où vous souhaitez afficher l'axe. En sélectionnant "à la valeur / au numéro de catégorie", vous pouvez aussi positionner l'axe à un autre endroit sur le canevas.

5.12.8.2.5 Axe Y

Dans la section de l'Axe Y du dialogue Changer apparence, vous pouvez saisir un libellé pour l'axe, définir les couleurs pour l'axe et le quadrillage (si affiché), définir la plage de valeur. Vous pouvez également décider si des graduations doivent être affichées et où et si vous voulez afficher les valeurs d'axe ou non. Cette section est identique pour tous les graphiques à barres et à ligne.



Libellé
Le texte saisi dans le champ `Libellé` sera imprimé à gauche de l'axe en tant que description de l'axe Y.

© 2016-2022 Altova GmbH

Plage
Par défaut, le bouton radio `Auto` est sélectionné dans la fenêtre Plage. Si vous souhaitez afficher un fragment du graphique avec plus de détails, activer le bouton `Manuel` et saisir les valeurs minimum et maximum dans les champs respectifs.
Si la colonne utilisée pour l'axe Y n'inclut pas zéro, vous pouvez désactiver la case à cocher `Inclure zéro` et l'axe Y commencera avec la valeur minimum disponible dans la série. L'option `Axe inversé` vous permet d'invertir les valeurs de l'axe Y. Par exemple, si les valeurs sont exécutées de `0` à `360`, la sélection de cette option générera l'axe Y de manière à ce que `360` est à l'origine et que les valeurs tendent vers `0` lorsque l'axe Y monte.

Ligne
L'axe est affiché dans la couleur que vous choisissez depuis la liste déroulante `Ligne`. Vous pouvez utiliser une des couleurs présélectionnées ou cliquer sur le bouton **Autre couleur...** pour choisir une couleur standard ou définir une couleur personnalisée. Cliquer sur le bouton **Choisir...** sur l'onglet de personnalisation et utiliser la pipette pour choisir une couleur qui s'affichera à un endroit de votre écran.

Quadrillages
Si la case à cocher `Afficher quadrillage` est activée, vous pouvez choisir une couleur à partir de la fenêtre de la liste déroulante correspondante.

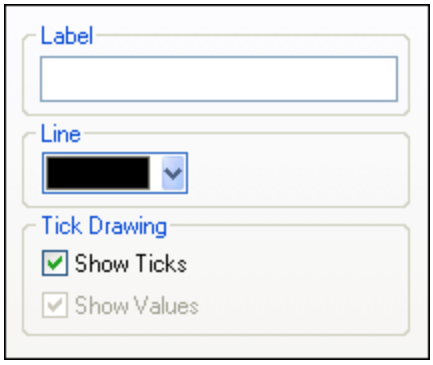
Intervalle des graduations
Si vous n'êtes pas satisfait des graduations par défaut, vous pouvez activer le bouton radio `Manuel` dans la fenêtre Intervalle des graduations et saisir la différence entre les graduations individuelles dans le champ correspondant.

Dessin graduations
Vous pouvez faire passer l'affichage des graduations sur l'axe et/ou les valeurs de l'axe sur marche ou arrêt.

Position de l'axe
À partir de la liste déroulante, vous pouvez choisir la position où vous souhaitez afficher l'axe. En sélectionnant "à la valeur / au numéro de catégorie", vous pouvez aussi positionner l'axe à un autre endroit sur le canevas.

5.12.8.2.6 Axe Z

Dans la section de l'axe Z du dialogue Changer apparence, vous pouvez saisir un libellé pour l'axe, définir les couleurs pour l'axe et décider si vous voulez ou non afficher des graduations sur l'axe. Cette section est identique pour tous les graphiques à barres 3D (graphiques à barres 3D et graphiques à barres 3D groupé).



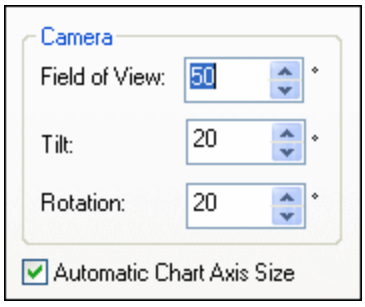
Libellé
Le texte saisi dans le champ `Libellé` sera imprimé à droite de l'axe en tant que description de l'axe Z.

Ligne
L'axe est affiché dans la couleur que vous choisissez depuis la liste déroulante `Ligne`. Vous pouvez utiliser une des couleurs présélectionnées ou cliquer sur le bouton **Autre couleur...** pour choisir une couleur standard ou définir une couleur personnalisée. Cliquer sur le bouton **Choisir...** sur l'onglet de personnalisation et utiliser la pipette pour choisir une couleur qui s'affichera à un endroit de votre écran.

Dessin graduations
Vous pouvez faire passer l'affichage des graduations sur l'axe sur marche ou arrêt

5.12.8.2.7 Angles 3D

Dans les graphiques à barres 3D, vous pouvez personnaliser l'apparence 3D du graphique dans la section Angles 3D du dialogue Changer apparence.



Avec l'option **Champ de vue** le diagramme apparaît comme s'il était observé depuis une petite ou une grande distance. Les valeurs valides s'étendent de 1 à 120. Des valeurs plus élevées donneront l'impression que le diagramme est observé depuis une distance plus grande.

La valeur **Inclinaison** détermine la rotation autour de l'axe X, alors que la valeur **Rotation** définit la rotation autour de l'axe Y. Vous pouvez adapter automatiquement la taille de l'axe du graphique à la largeur de la fenêtre Graphique en activant la case à cocher correspondante.

© 2016-2022 Altova GmbH

Si la case à cocher **Taille automatique des axes du graphique** est activée, XMLSpy calculera automatiquement la taille idéale de l'axe X ainsi que l'axe Y pour la taille de la fenêtre Graphique actuelle. La largeur et la hauteur du graphique changeront dynamiquement lorsque vous redimensionnerez la fenêtre Graphique.

5.12.8.2.8 Tailles

Dans la section Tailles du dialogue Changer apparence, vous pouvez définir des marges différentes ainsi que la taille des axes et des graduations de jauge. Veuillez noter que toutes les propriétés regroupées ci-dessous ne sont pas toutes disponibles pour tous les types de graphique.

Généralité	
Marge extérieure	Espace entre le canevas et le rebord de la fenêtre Graphique.
Titre à Canevas	Espace entre le titre de graphique et le rebord supérieur du canevas.
Légende à Canevas	Espace entre le rebord inférieur du canevas et la légende.
Camembert	
Canevas à Libellé	Dans les camemberts, l'espace entre l'extrémité gauche et droite du rebord du camembert et de ses libellés.
Hauteur de Camembert	Dans les camemberts 3D, la hauteur du camembert.
Camembert Ombre portée	Dans les camemberts 3D, la longueur de l'ombre (si elle est activée dans la section du camembert).
Axe X	
Axe X à Libellé d'axe	Dans les graphiques à barres et à lignes, l'espace entre l'axe X et son libellé.
Axe X à Canevas	Dans les graphiques à barres et à lignes 2D, l'espace entre l'axe X et le canevas.
Axe X Taille de graduation	Dans les graphiques à barres et à lignes, la longueur des graduations sur l'axe X.
Axe Y	
Axe Y à Libellé d'axe	Dans les graphiques à barres et à lignes, l'espace entre l'axe Y et son libellé.
Axe Y à Canevas	Dans les graphiques à barres et à lignes 2D, l'espace entre l'axe Y et le canevas.
Axe Y Taille de graduation	Dans les graphiques à barres et à lignes, la longueur des graduations sur l'axe Y.
Axe Z	
Axe Z à Libellé d'axe	Dans les graphiques à barres et à lignes 3D, l'espace entre l'axe Z et son libellé.
Axe Z Taille de graduation	Dans les graphiques à barres et à lignes 3D, la longueur des graduations sur l'axe Z.
Tracé de ligne	
Taille de marque de connexion	Dans les graphiques à lignes, la taille des carrés qui marquent les valeurs dans le graphique.
Largeur de ligne	Dans les graphiques à lignes, la largeur des lignes
Tailles axe 3D	
Axe X manuelle Taille de la base	Dans les graphiques à barres 3D, définit la relation entre la longueur de l'axe X et la taille de la fenêtre Graphique. Veuillez noter que la case à cocher `Taille automatique des axes du graphique` dans la section Angles 3D doit être désactivée, sinon la taille sera tout de même calculée automatiquement.
Axe Y manuelle Taille de la base	Dans les graphiques à barres 3D, définit la relation entre la longueur de l'axe Y et la taille de la fenêtre Graphique. Veuillez noter que la case à cocher `Taille automatique des axes du graphique` dans la section Angles 3D doit être désactivée, sinon la taille sera tout de même calculée automatiquement.
Axe Z marge de la série	Dans les graphiques à barres 3D, la distance sur l'axe Z entre les séries individuelles.

Jauge

Largeur de bordure	Dans les graphiques à jauge ronde, la largeur de la bordure autour de la jauge.

Graduation de jauge

Distance de la bordure à la graduation	Dans les graphiques à jauge ronde, l'espace entre le rebord intérieur de la bordure et les graduations qui marquent les valeurs.
Longueur de la graduation principale	Dans les graphiques à jauge ronde, la longueur des graduations principales (les graduations qui affichent un libellé).
Largeur de la graduation principale	Dans les graphiques à jauge ronde, la largeur des graduations principales (les graduations qui affichent un libellé).
Longueur de la graduation intermédiaire	Dans les graphiques à jauge ronde, la longueur des graduations dont la valeur n'est pas affichée.
Largeur de la graduation intermédiaire	Dans les graphiques à jauge ronde, la largeur des graduations dont la valeur n'est pas affichée.

Aiguille de jauge

Longueur de l'aiguille	Dans les graphiques à jauge ronde, la longueur de l'aiguille. (Veuillez noter que le pourcentage est calculé à partir du diamètre de la jauge, donc si vous choisissez une valeur supérieure à 50%, l'aiguille pointera à un endroit en-dehors de la jauge !)
Largeur de l'aiguille à la base	Dans les graphiques à jauge ronde, la largeur de l'aiguille au centre de la jauge.
Rayon de la base de l'aiguille	Dans les graphiques à jauge ronde, le rayon de la base qui recouvre le centre de la jauge.

Plage de couleurs de la jauge

Distance entre la bordure et la plage de couleur	Dans les graphiques à jauge ronde, l'espace entre le rebord intérieur de la bordure et le rebord extérieur de la plage de couleurs.
Largeur de la plage de couleur	Dans les graphiques à jauge ronde, la largeur de la plage de couleurs personnalisable. (Veuillez noter que le pourcentage est calculé à partir du diamètre de la jauge !)

Valeur de jauge

Déport vers Centre	Distance depuis le centre à laquelle la valeur de la jauge est affichée.

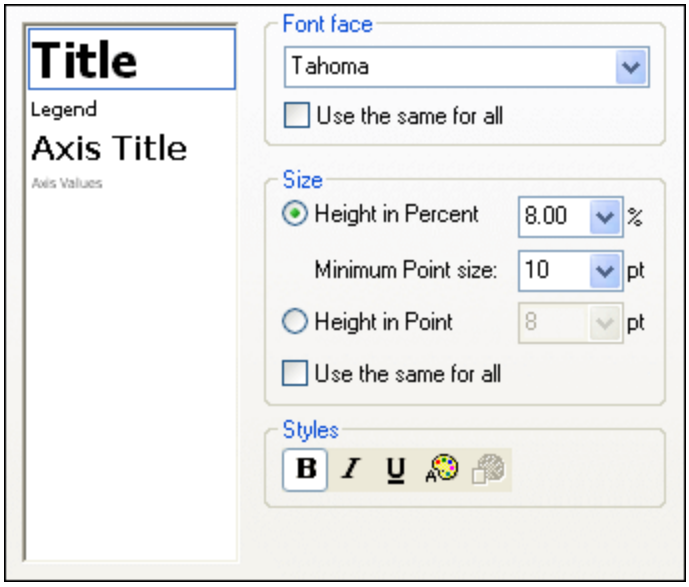
Valeur supplémentaire

Déport vers Centre	Distance depuis le centre à laquelle le libellé supplémentaire (défini dans les Paramètres de Graphique de jauge) est affiché.

5.12.8.2.9 Polices d'écriture

La section Police du dialogue Changer apparence vous permet de configurer les polices d'écriture pour les objets dans la fenêtre Graphique.

© 2016-2022 Altova GmbH



Paramètres de polices

Vous pouvez choisir la police, la taille, le style pour les éléments individuels affichés dans la fenêtre Graphique. Vous pouvez définir la taille en tant que pourcentage de la taille du graphique et définir une taille minimum en points, ou spécifier une valeur absolue (en points). Pour appliquer la même police d'écriture et/ou taille à tous les éléments de texte, activer la case à cocher respective `Utiliser la même police pour tous`.
Les noms d'élément dans la liste sont définis comme suit :

- **Titre** : le nom d'un graphique
- **Légende** : la clé des couleurs utilisées dans le graphique
- **Libellés** : la désignation des parts dans un graphique de camembert
- **Titre d'axe** : le nom des axes X, Y et Z dans un graphique à barres ou à lignes
- **Valeurs d'axe** : les unités affichées sur un axe dans un graphique à barres ou à ligne
- **Valeurs de graduations** : les unités affichées dans un graphique à jauge

5.12.9 Exporter

Cliquer sur le bouton **Exporter** pour obtenir les options suivantes :

- *Enregistrer le graphique en tant qu'une image dans le fichier:* les formats d'image disponibles sont PNG, GIF, BMP et JPG.
- *Copier l'image à la taille actuelle ou d'une autre taille dans le presse-papiers :* permet de copier ultérieurement le graphique depuis le presse-papiers dans un rapport dans une autre application.
- *Imprimer le graphique :* envoie l'image vers une imprimante sur votre réseau. La hauteur et la largeur de l'image peuvent être spécifiées en tant que pourcentage de la taille de l'image.
- *Copier le code XSLT ou XQuery sur le presse-papiers :* crée un fragment XSLT ou un fragment XQuery. Chacun est essentiellement la fonction d'extension Altova `CreateChart`. Cette fonction peut être utilisée avec d'autres fonctions d'extension Altova et traitées avec XMLSpy pour générer des graphiques. Pour vous aider à utiliser la fonction d'extension `CreateChart`, un exemple d'utilisation commenté est aussi créé avec chaque fragment.

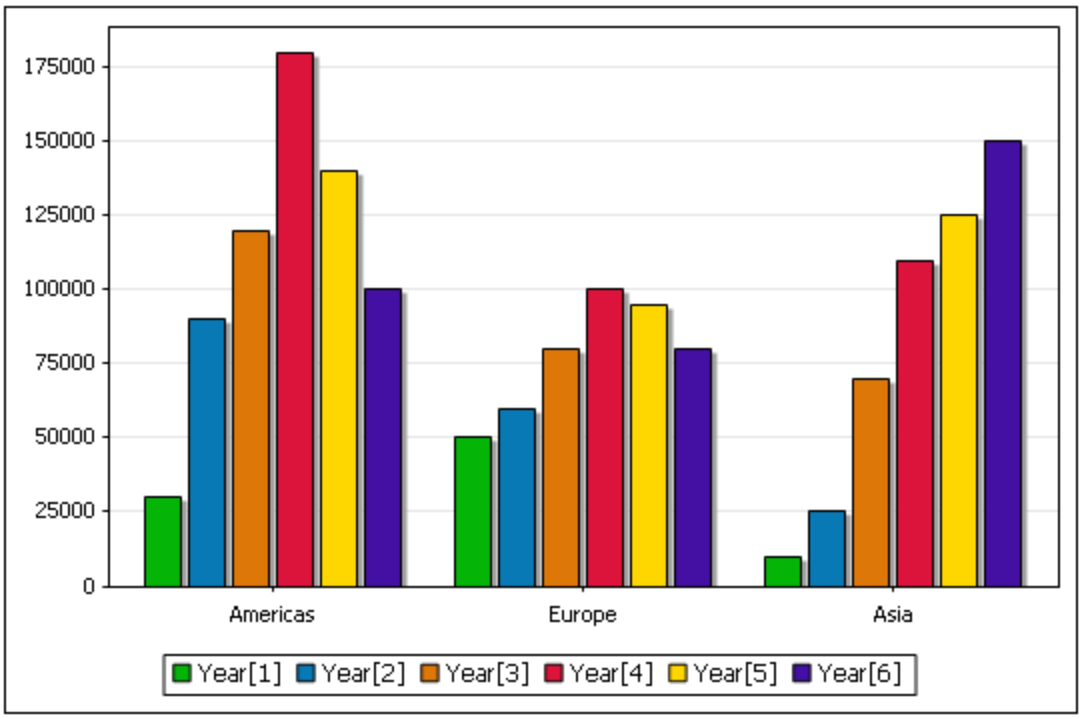
5.12.10 Exemple de graphique : simple

Examinons le document XML suivant. (Il est nommé `YearlySales.xml` et est disponible dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.)

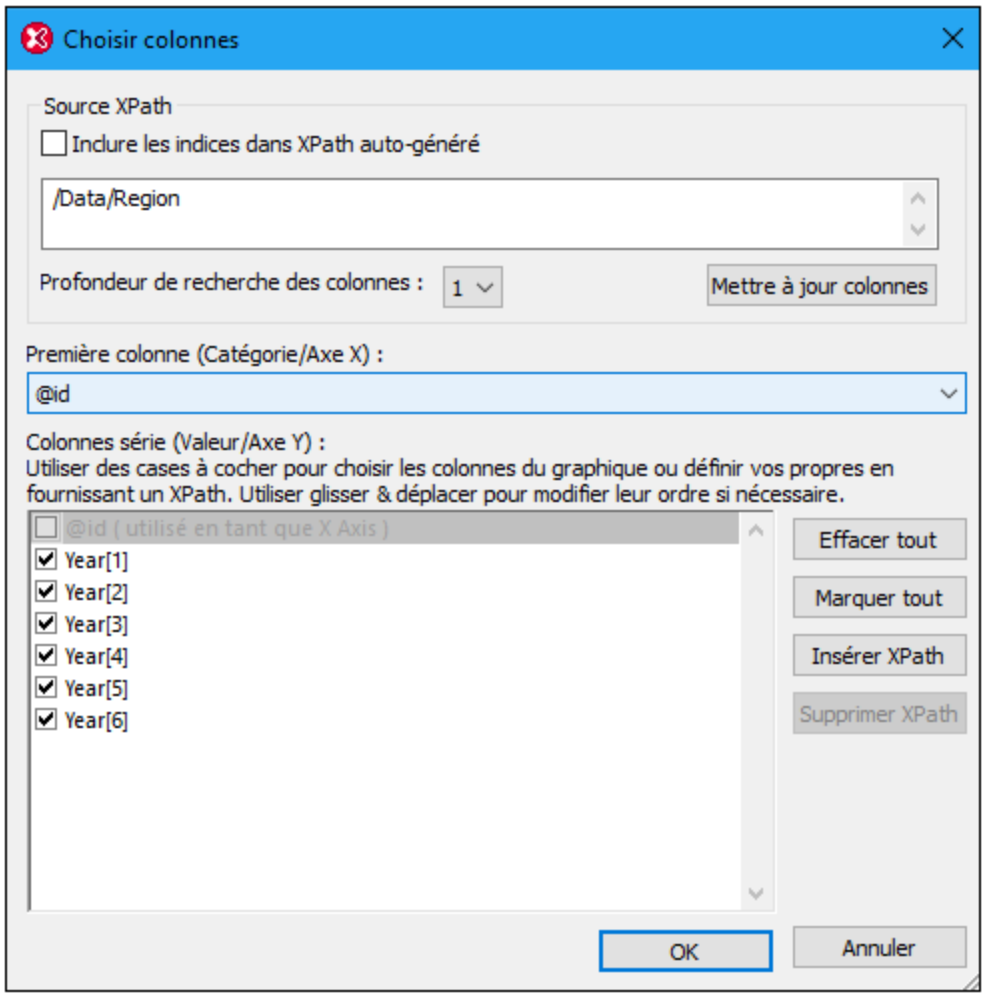
```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
              <Year id="2006">60000</Year>
              <Year id="2007">80000</Year>
              <Year id="2008">100000</Year>
              <Year id="2009">95000</Year>
              <Year id="2010">80000</Year>
       </Region>
       <Region id="Asia">
              <Year id="2005">10000</Year>
              <Year id="2006">25000</Year>
              <Year id="2007">70000</Year>
              <Year id="2008">110000</Year>
              <Year id="2009">125000</Year>
              <Year id="2010">150000</Year>
       </Region>
</Data>
```

Nous souhaitons produire un graphique qui affiche les trois régions sur l'Axe X et indique les ventes annuelles pour chaque région. Notre graphique devrait ressembler au graphique à barres ci-dessous.

© 2016-2022 Altova GmbH



Il s'agit d'un graphique simple parce que nous pouvons sélectionner l'élément `Region` en tant que le XPath de source. L'expression du XPath de source retourne une séquence de trois items : les trois éléments `Region`. Chaque élément `Region` sera, à son tour, le nœud contextuel pour les sélections de données de l'Axe X et de l'Axe Y.



Pour les séries, nous nécessitons les éléments `Year` de chaque région, donc une profondeur de recherche de 1 niveau suffira. Nous choisissons l'attribut `id` de l'élément `Region` pour l'Axe X. Les valeurs de l'attribut `id` seront donc utilisées en tant que les libellés des trois graduations de l'Axe X. Toutes les séries `Year` sont cochées parce que nous souhaitons inclure tous les éléments `Year` dans la table des données de graphique.

Cliquer sur la touche **OK** pour générer le graphique. Pour des graphiques plus avancés, voir la section, Exemple de graphique : avancé.

5.12.11 Exemple de graphique : avancé

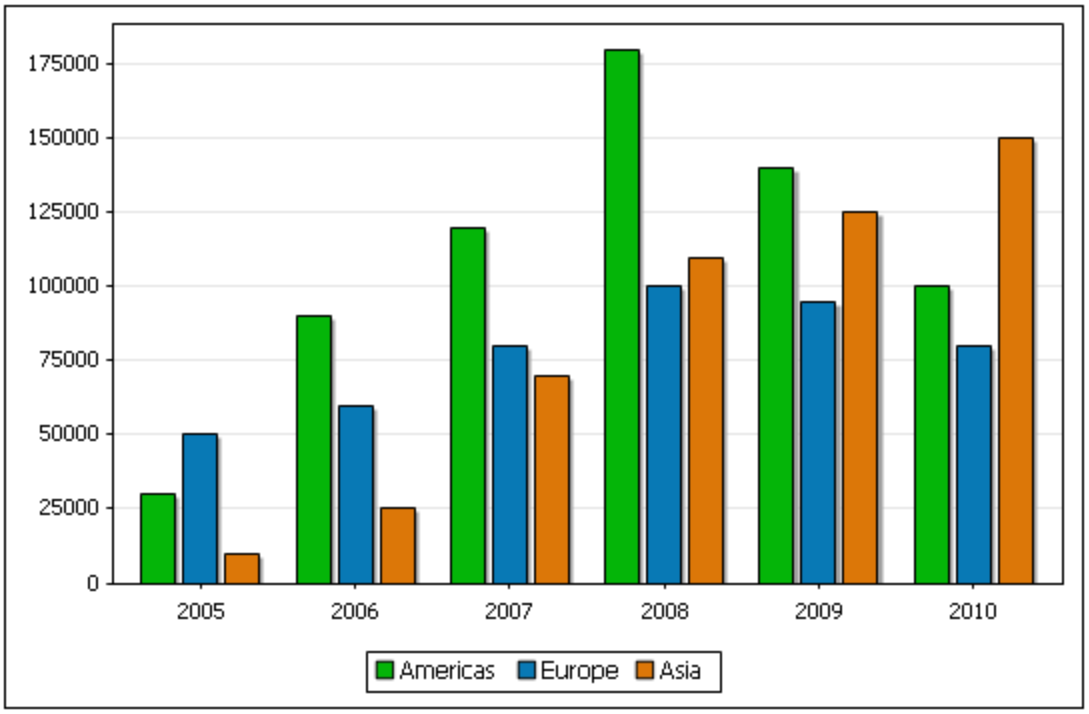
Examinons le document XML suivant. (Il est nommé `YearlySales.xml` et est disponible dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`.)

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <Region id="Americas">
              <Year id="2005">30000</Year>
```


© 2016-2022 Altova GmbH

```
            <Year id="2006">90000</Year>
            <Year id="2007">120000</Year>
            <Year id="2008">180000</Year>
            <Year id="2009">140000</Year>
            <Year id="2010">100000</Year>
        </Region>
        <Region id="Europe">
            <Year id="2005">50000</Year>
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
        </Region>
        <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
        </Region>
</Data>
```

Nous souhaitons produire un graphique qui affiche les années sur l'Axe X et compare les ventes annuelles pour chaque année. Notre graphique devrait ressembler au graphique à barres ci-dessous.

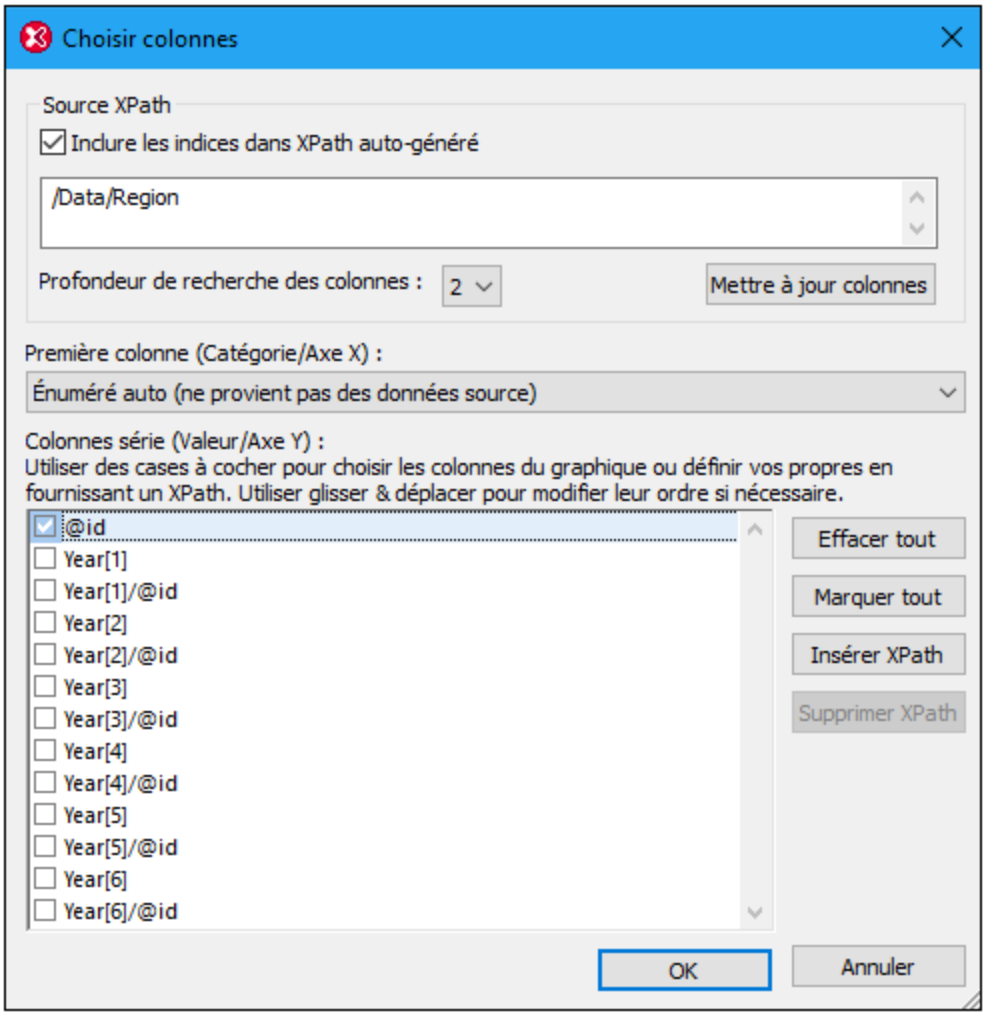


Nous allons vous montrer deux méthodes différentes. Ces deux méthodes indiquent comment les paramètres de sélection des données différents peuvent être combinés pour produire les résultats requis.

Méthode 1 : Modifier les valeurs d'axe

Dans la première méthode, les axes qui ont été sélectionnés dans le dialogue Choisir colonnes sont modifiés dans le dialogue Sélectionner les données. Il suffit de s'assurer que toutes les données requises sont accessibles à la sélection dans la table de données de graphique contenue dans le dialogue Sélectionner les données.

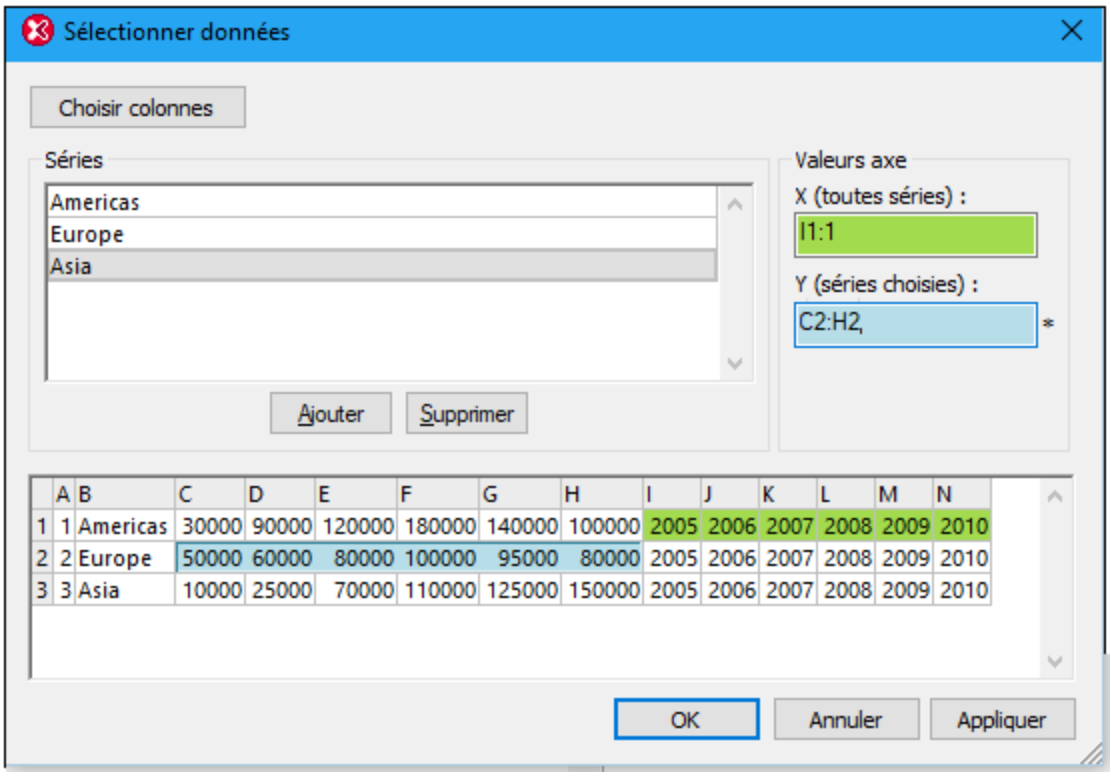
1. Dans le dialogue Choisir colonnes, veuillez vous assurer que tous les nœuds requis seront disponibles pour la sélection de l'Axe X et de l'Axe Y. Dans la capture d'écran ci-dessous, veuillez noter que la Profondeur de recherche de colonne a été définie à `2` de manière à ce que les attributs `Year/@id` sont aussi sélectionnés.



2. Dans le dialogue Sélectionner les données (*capture d'écran ci-dessous*), la table de données de graphique comporte les colonnes suivantes : la première colonne est la sélection de l'Axe X (qui est la sélection d'énumération automatique), les colonnes restantes sont les colonnes de la série (Axe Y), qui sont les attributs `Region/@id`, les contenus d'élément `Year` et les attributs `Year/@id`. Veuillez

© 2016-2022 Altova GmbH

aussi noter que : (i) il y a uniquement trois lignes, donc trois graduations de l'Axe X ; (mais nous nécessitons six graduations de l'Axe X pour les six ans) ; (ii) il y a 13 colonnes de série.



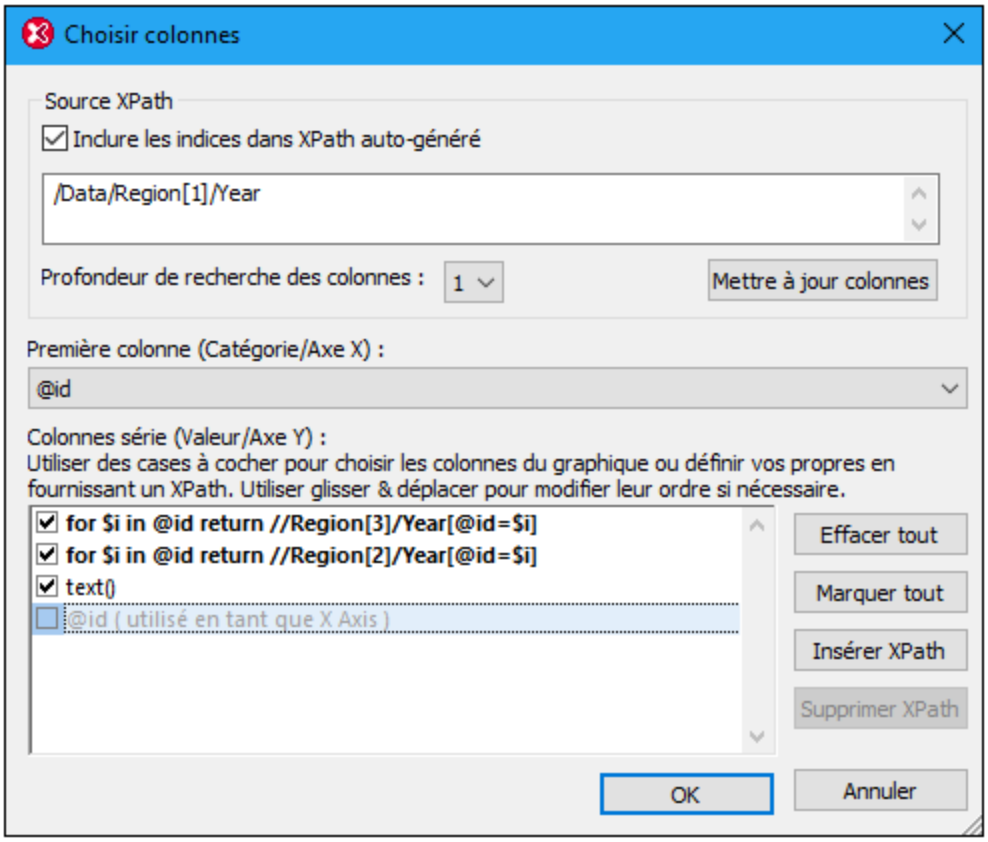
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	Americas	30000	90000	120000	180000	140000	100000	2005	2006	2007	2008	2009	2010
2	2	Europe	50000	60000	80000	100000	95000	80000	2005	2006	2007	2008	2009	2010
3	3	Asia	10000	25000	70000	110000	125000	150000	2005	2006	2007	2008	2009	2010

3. Dans le volet Séries, nous supprimons 10 des 13 lignes de la série et renommons les trois séries restantes en `Americas`, `Europe`, et `Asia`, comme indiqué dans la capture d'écran ci-dessus. L'ordre sélectionné ici sera l'ordre du libellé de la graduation de l'Axe X.
4. Dans le volet Séries, sélectionner la série `Americas`. Dans le volet Valeurs de l'axe, cliquer sur le champ de l'Axe X pour permettre une modification. Ensuite cliquer sur la cellule `I1` dans la table des données du graphique et la glisser dans la cellule `N1`. Dans le champ de texte de l'Axe Y, saisir soit `C1:H1` ou effectuer la sélection en glissant de `C1` en `H1`.
5. Pour les séries Europe et Asie, sélectionner `C2:H2` et `C3:H3`, respectivement pour l'Axe Y. La sélection de l'Axe X peut être la même que celle pour la série `Americas`.
6. Cliquer sur **OK**. Le graphique requis est généré.

Note : le nombre de graduations de l'Axe X (défini par défaut pas le nombre de lignes dans la table des données du graphique) est augmenté de trois à six parce que le nombre des libellés de l'Axe X est six.

Méthode 2 : Générer des séries avec des expressions XPath

Dans la seconde méthode, les expressions XPath sont insérées pour générer des séries. Cela est nécessaire parce que le XPath de source (*voir capture d'écran ci-dessous*) ne possède pas en tant que ses descendants, les nœuds choisis pour la série. Néanmoins, le XPath de source génère six graduations d'Axe X (en choisissant les éléments `Year` du premier élément `Region`). Afin de pouvoir sélectionner le premier élément `Region` avec le prédicat `[1]`, la case à cocher *Inclure indices* doit être cochée et cliquer le bouton **Mettre à jour colonnes**.



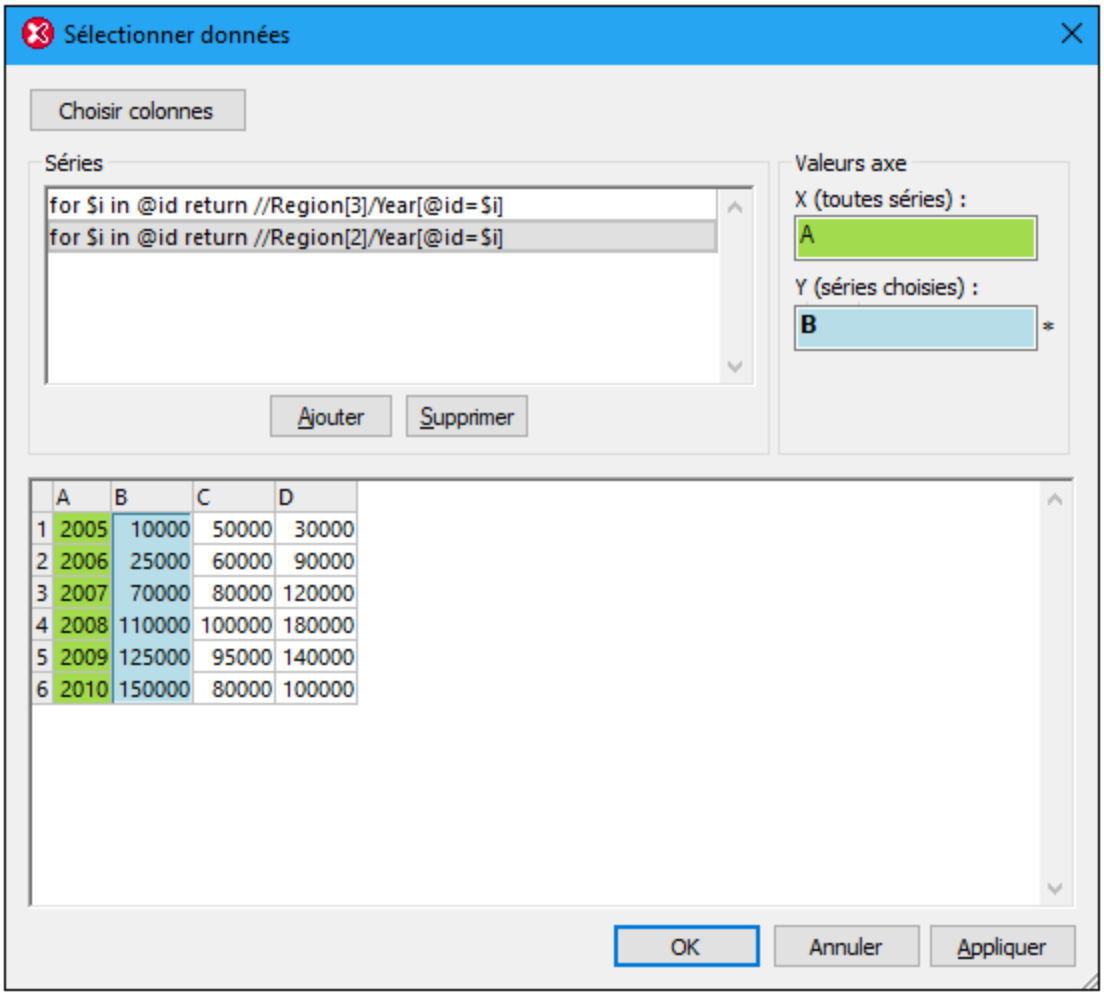
Dans le dialogue Choisir colonnes, l'élément `Region[1]/Year` a uniquement deux descendants : `@id` et `text()`. L'attribut `@id` est sélectionné pour l'Axe X, générant ainsi le libellé de l'Axe X correct pour chacune des six graduations de l'Axe X. La table de données de graphique serait évaluée comme suit.

Source XPath	X-Axis	axe Y(Series columns)		
Region[1]/Year[1]	@id	text()	XPath-1	XPath-2
Region[1]/Year[2]	@id	text()	XPath-1	XPath-2
Region[1]/Year[3]	@id	text()	XPath-1	XPath-2
Region[1]/Year[4]	@id	text()	XPath-1	XPath-2
Region[1]/Year[5]	@id	text()	XPath-1	XPath-2
Region[1]/Year[6]	@id	text()	XPath-1	XPath-2

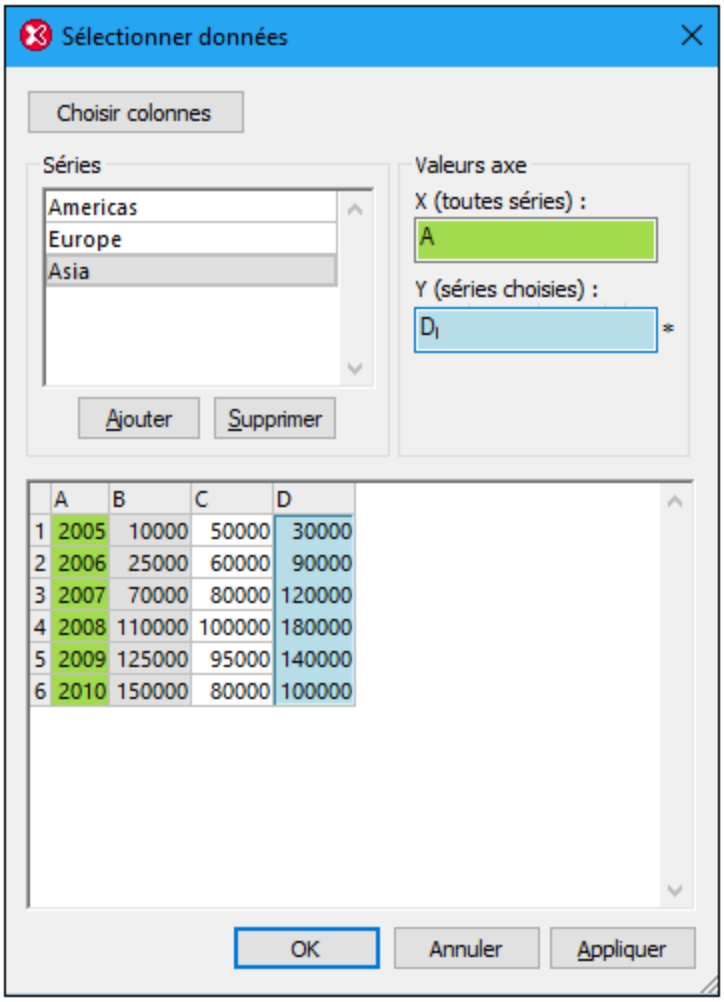
Veuillez noter que le nœud contextuel est un des six éléments `Region[1]/Year` à son tour. La première expression XPath cherche la valeur d'attribut actuelle `Year/@id` et retourne l'élément `Region[2]/Year` qui a la même valeur `Year/@id` que la valeur `@id` du `Region[1]/Year` actuel. La seconde expression XPath fait la même chose pour les éléments `Region[3]/Year`. Donc, pour chacune des six années : Les trois séries de l'Axe Y sont les enfants d'élément `Year`, respectivement, de chacun des trois éléments `Region`. (Le nœud `text()` retourne les contenus des éléments `Region[1]/Year`.)

La table de données du graphique dans le dialogue Sélectionner les données ressemblera à :

© 2016-2022 Altova GmbH



Le nom des séries dans le dialogue Sélectionner les données peut être changé pour passer des expressions XPath (*comme dans la capture d'écran ci-dessus*) à des légendes lisibles (*capture d'écran ci-dessous*). Pour chaque série, la colonne de données correcte peut être assignée dans le volet de Valeurs d'Axe (en cliquant dans le champ de saisie Axe Y puis en sélectionnant la colonne requise dans la table de données du graphique).



Les deux méthodes affichées ci-dessus génèrent des graphiques identiques. Les différentes approches sont censées montrer comment les paramètres de sélection des données doivent être utilisés.

5.12.12 Exemple de graphique : chandelier

Les graphiques à chandeliers sont généralement utilisés pour représenter le mouvement des prix de titres en bourse. Il existe deux types de graphiques à chandeliers :

- Graphiques à chandeliers à quatre séries, représentant les prix d'ouverture, maximum, minimum et le prix de clôture de la journée.
- Graphiques à chandeliers à trois séries, représentant les prix maximum, minimum et le prix de clôture de la journée.

Le fichier `Candlestick.xml`, qui est disponible dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Tutorial`, est un exemple d'une structure de document XML qui pourrait être utilisée pour des graphiques à chandeliers. La liste ci-dessous montre la structure essentielle du fichier.

© 2016-2022 Altova GmbH

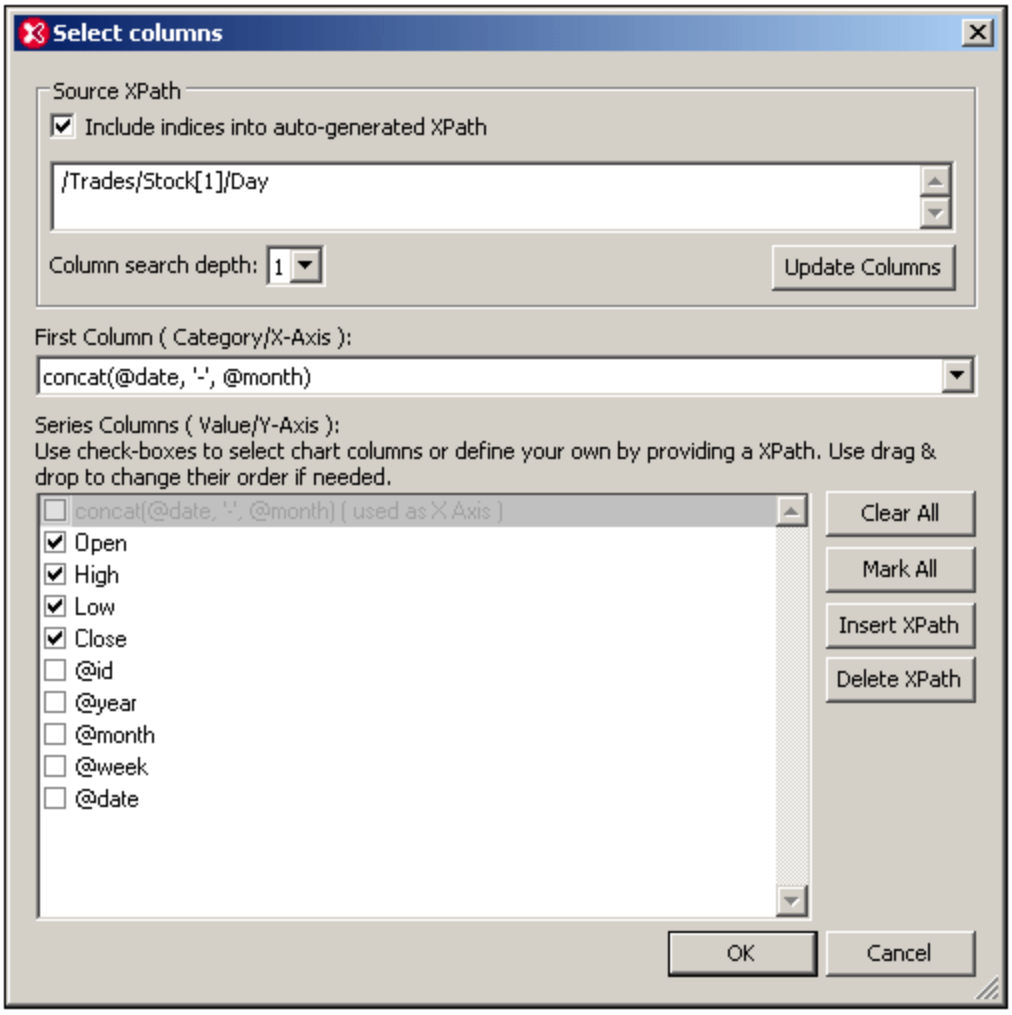
```
<Trades>
  <Stock name="MyStock">
    <Day id="20110103" year="2011" month="Jan" week="01" date="03">
      <Open>90</Open>
      <High>110</High>
      <Low>88</Low>
      <Close>105</Close>
    </Day>

    .....

  </Stock>
</Trades>
```

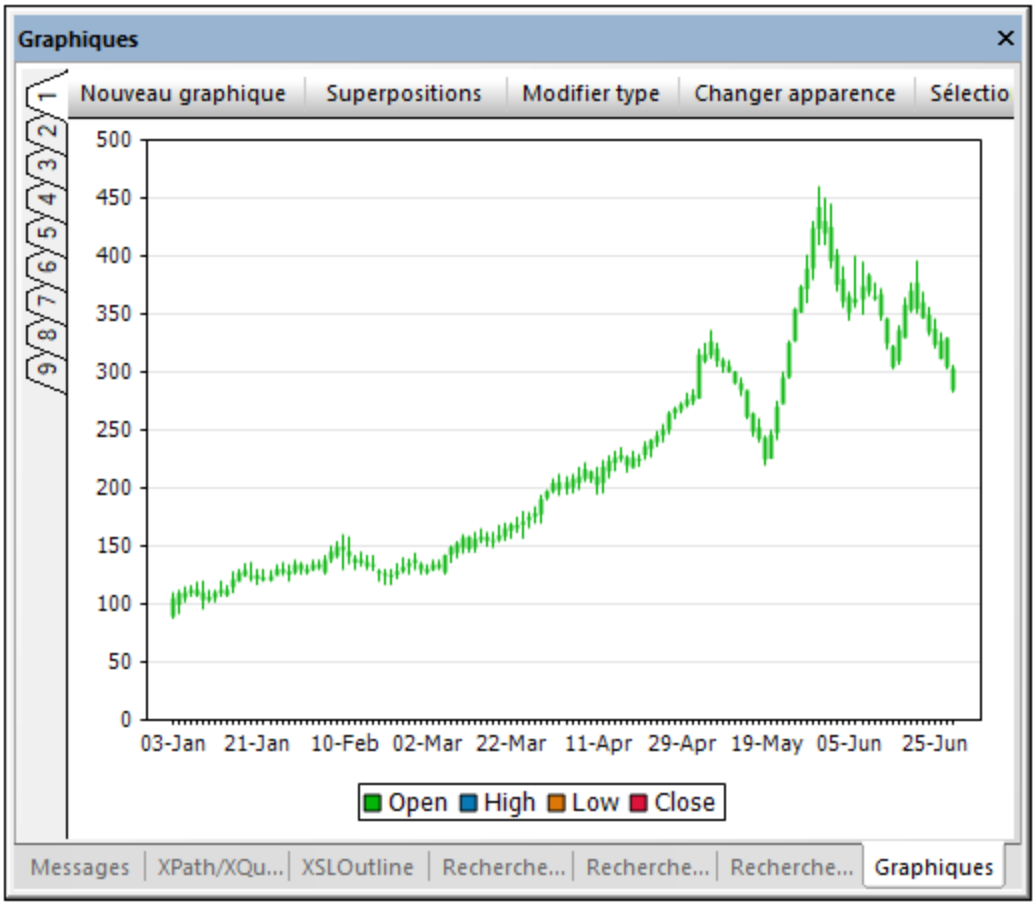
Un graphique à chandeliers peut être créé pour le fichier mentionné ci-dessus comme suit :

1. Placer le curseur dans la balise d'élément Day, cliquer sur le bouton **Nouveau graphique** de la fenêtre Graphiques. Le dialogue Choisir colonnes s'ouvre (*capture d'écran ci-dessous*). Si la case à cocher *Inclure des Indices* n'est pas cochée, la cocher et cliquer sur **Mettre à jour les colonnes**.



2. Cliquer sur le bouton Insérer XPath et insérer l'expression XPath : `concat(@date, '-', @month)`.
3. À partir de la liste déroulante de la liste de choix Première colonne, sélectionner l'expression XPath que vous venez de saisir. Cela créera la date et le mois de chaque élément `Day` en tant que les libellés des graduations de l'Axe X.

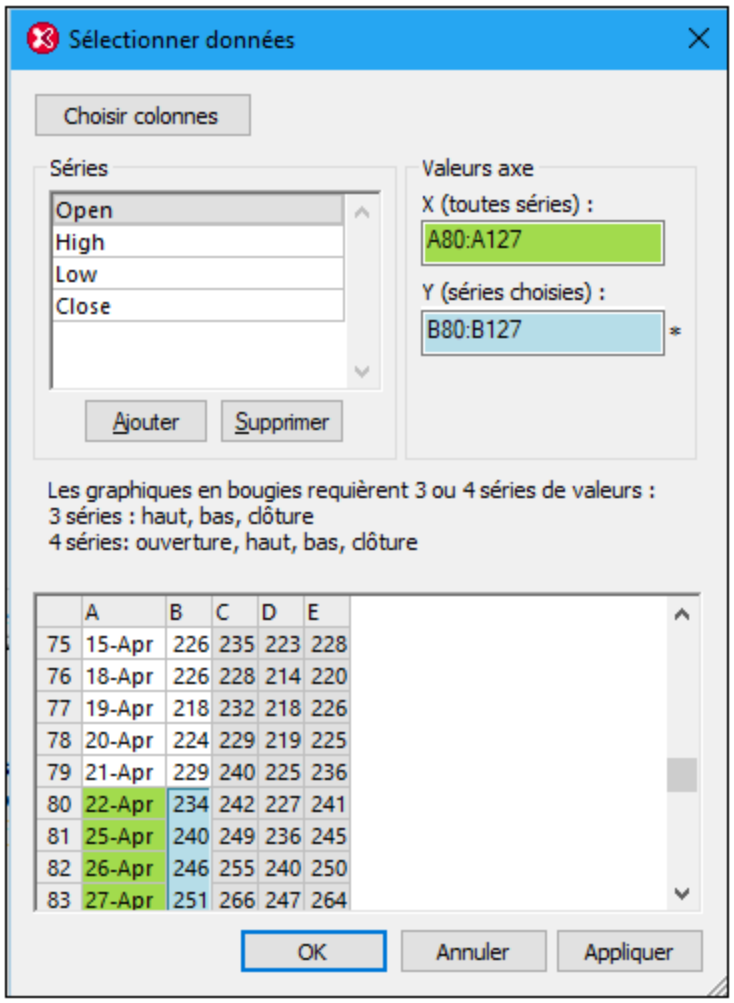
4. Pour les séries de l'Axe Y, choisir les cases à cocher Open, High, Low et Close dans le volet Colonnes de série.
5. Cliquer sur **OK**.
6. Cliquer sur le bouton **Modifier Type** de la fenêtre Graphiques et changer le type de graphique en Chandelles.
7. Cliquer sur **OK**. Un graphique à chandeliers sera créé comme celui contenu par la capture d'écran ci-dessous.



Choisir un sous-ensemble de données

Si vous souhaitez consulter un sous-ensemble des données de graphique sélectionnées dans le dialogue Choisir colonnes, par exemple, si vous souhaitez consulter un ensemble de dates dans le cadre de la sélection des données du graphique, cliquer sur le bouton **Sélectionner données** de la fenêtre Graphiques. Le dialogue Sélectionner les données s'ouvre (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Dans le volet Valeurs d'axe, saisir la plage de l'Axe X, par exemple, les cellules `A80:A127`, comme affiché dans la capture d'écran ci-dessus. Pour les séries différentes, tout d'abord cliquer dans le champ de texte de l'Axe Y, puis sélectionner la série dans le volet Séries, puis saisir la plage pour cette série. Répéter l'opération pour chacune des quatre séries. Par exemple, la capture d'écran ci-dessus montre la plage de l'Axe Y sélectionnée pour la série Open. Pour plus d'informations concernant le dialogue Sélectionner les données, voir la section, Données de graphique.

5.13 Signatures XML

Un fichier XML peut être signé numériquement et la signature peut être vérifiée par la suite. Si le fichier a été modifié après la signature, la vérification échouera. XMLSpy prend en charge la création et la vérification des signatures XML.

Les signatures XML dans les modes XMLSpy

Des signatures XML peuvent être créées pour tous les types de fichiers XML, y compris pour les fichiers de Schéma XML, WSDL et XBRL. Les commandes **XML | Créer une signature Signature** et **XML | Vérifier la signature XML** sont donc disponibles dans tous les modes XMLSpy : Mode Texte, Mode Grille, Mode Schéma, Mode WSDL et Mode XBRL.

Comment fonctionnent les signatures XML

Le processus, de la création de la signature jusqu'à la vérification de la signature, fonctionne de la manière suivante :

1. Le fichier XML est signé soit avec la clé privée d'un certificat, soit avec un mot de passe. Dans XMLSpy, vous pouvez créer une signature en utilisant la commande the **XML | Créer une signature Signature**. La signature est obtenue en traitant : (i) le document XML et (ii) la clé privée d'un certificat, ou un mot de passe.
2. La signature peut soit être incluse dans le fichier XML soit stockée dans un fichier séparé.
3. La signature du fichier XML est vérifiée à l'aide de la clé publique du certificat ou du mot de passe. Le processus de vérification fonctionne de la manière suivante : tout d'abord en traitant (i) le document XML, et (ii) la clé privée du certificat, ou un mot de passe, utilisé pour la signature, ensuite, en comparant ce résultat avec la signature. Si le fichier XML est modifié après la signature, la vérification échouera. Dans XMLSpy, vous pouvez vérifier une signature avec la commande **XML | Vérifier la signature**.

Les détails de la création et de la vérification des signatures dans XMLSpy sont décrits dans les sous-sections de cette section :

- Créer des signatures XML
- Vérifier des signatures XML

Comment les certificats sont utilisés dans les signatures XML

Pour pouvoir être utilisés avec des signatures XML, les certificats doivent posséder une clé privée et une clé publique. La clé privée sert à créer la signature XML, la clé publique a pour but de vérifier la signature XML.

Dans un scénario typique, l'expéditeur d'un document XML a accès à la clé privée d'un certificat et crée la signature XML avec cette clé. Le destinataire du document aura accès à la clé publique du certificat. Cet accès peut être de deux types : (i) l'expéditeur envoie une information de clé publique avec la signature ; (ii) le destinataire aura accès à une clé de version publique du certificat utilisé par l'expéditeur.

Pour plus de détails concernant les certificats, voir la sous-section, Travailler avec des certificats.

Note : La fonction Signature XML de XMLSpy prend en charge des certificats de type RSA-SHA1, DSA-SHA1 et SHA-256. Les signatures qui sont basées sur des mots de passe utilisent l'algorithme HMAC-SHA256 (XMLSpy 2018 et plus ; *voir Créer les signatures XML*).

© 2016-2022 Altova GmbH

Validité du document XML

Si une signature XML est intégrée dans le document XML , un élément `Signature` dans l'espace de noms `http://www.w3.org/2000/09/xmldsig#` est ajouté au document XML. Pour que le document reste valide, conformément à un schéma, le schéma doit contenir les déclarations d'élément appropriées. XMLSpy intègre les signatures de deux manières :

- *Enveloppée :* l'élément `Signature` est créé en tant que le dernier élément enfant de l'élément racine (ou document).
- *Enveloppante :* l'élément `Signature` est créé en tant que l'élément racine (ou document), et l'élément de document XML original est placé dans un élément enfant de l'élément de signature nommé `Object`.

Si vous ne souhaitez pas modifier le schéma du document XML, la signature XML peut être créée dans un fichier externe. Pour plus de détails, voir la description des options de placement dans la section Créer des signatures XML.

Ci-dessous, vous trouverez des extraits du Schéma XML qui montrent comment l'élément `Signature` d'une signature enveloppée peut être permise. Vous pouvez utiliser ces exemples en tant que guides pour modifier vos propres schémas.

Dans la première des deux listes ci-dessous, le Schéma de signatures XML est importé dans le schéma de l'utilisateur. Le Schéma de signatures XML est situé sous l'adresse Internet :
http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xsig="http://www.w3.org/2000/09/xmldsig#"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
              schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd"/>
   <xs:element name="Root">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="FirstChildOfRoot"/>
            <xs:element ref="SecondChildOfRoot" minOccurs="0"/>
            <xs:element ref="ThirdChildOfRoot" minOccurs="0"/>
            <xs:element ref="xsig:Signature" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   ...
</xs:schema>
```

Une seconde option (*liste ci-dessous*) est d'ajouter un élément de caractère générique général qui fait correspondre tout élément depuis d'autres espaces de noms. La configuration de l'attribut `processContents` sur `lax` fait passer le validateur sur cet élément, parce qu'aucune déclaration de l'élément correspondant n'a été trouvé. Par conséquent, l'utilisateur n'a pas besoin de référencer le Schéma de signatures XML. L'inconvénient de cette option, néanmoins, est que tout élément (pas seulement l'élément `Signature`) peut être ajouté à l'emplacement spécifique dans le document XML sans invalider le document XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
            elementFormDefault="qualified"
            attributeFormDefault="unqualified">
   <xs:element name="Root">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="selection"/>
            <xs:element ref="newsitems" minOccurs="0"/>
            <xs:element ref="team" minOccurs="0"/>
            <xs:any namespace="##other" minOccurs="0" processContents="lax"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   ...
</xs:schema>
```

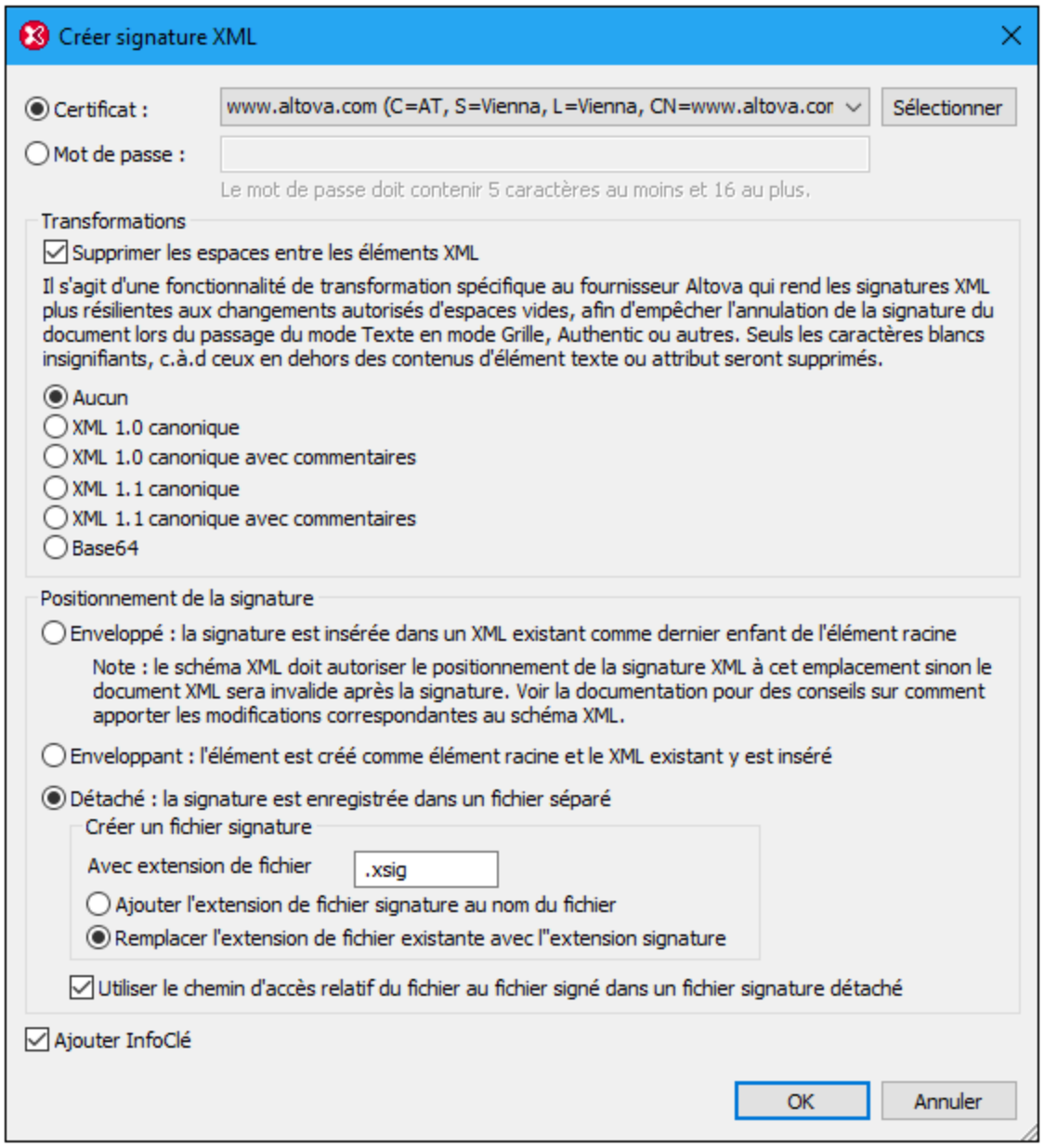
Spécifications W3C

Pour plus de détails concernant les signatures XML, voir les spécifications W3C pour les signatures XML sous https://www.w3.org/TR/xmldsig-core1/.

5.13.1 Créer des signatures XML

Afin de créer une signature XML pour un document XML, ouvrir le document XML pour lequel vous souhaitez créer une signature. Ensuite, cliquer sur la commande de menu **XML | Créer Signature XML**. Le dialogue Créer Signature XML s'ouvre (*capture d'écran ci-dessous*), les paramètres sont expliqués ci-dessous.

© 2016-2022 Altova GmbH



Méthode d'authentification : certificat ou mot de passe

La signature peut se baser sur un certificat ou sur un mot de passe. Choisir le bouton radio de la méthode que vous souhaitez utiliser.

- *Certificat :* cliquer sur le bouton **Sélectionner** et chercher le certificat que vous souhaitez. Le certificat que vous sélectionnez doit avoir une clé privée. La signature est générée à l'aide de la clé privée du certificat. Pour vérifier la signature, il est nécessaire d'accéder au certificat (ou à une version à clé publique). La clé publique du certificat est utilisée pour vérifier la signature. Pour plus de détails concernant les certificats, voir la section Travailler avec des certificats.

- *Mot de passe :* Saisir un mot de passe d'une longueur de 16 caractères. Ce mot de passe sera ensuite nécessaire pour vérifier la signature.

Note : À partir de XMLSpy 2018, l'algorithme qui est utilisé pour signer des documents basés sur des mots de passe a été mis à jour depuis HMAC-SHA1 vers HMAC-SHA256. Les versions précédentes de XMLSpy ne seront pas en mesure de vérifier les signatures en utilisant les mots de passe générés avec XMLSpy 2018.

Transformations

Les données XML sont transformées et le résultat de la transformation est utilisé pour la création de la signature. Vous pouvez spécifier l'algorithme de canonicalisation à appliquer dans les données du fichier XML (le contenu `SignedInfo`) avant d'effectuer les calculs de la signature. Les différences les plus importantes entre les algorithmes sont notées ci-dessous :

- *XML canonique avec ou sans commentaires :* si les commentaires sont inclus pour un calcul de la signature, tout changement des commentaires dans les données XML entraînera un échec de la vérification. Sinon, les commentaires peuvent être modifiés ou ajoutés au document XML après que le document ait été signé et la signature sera toujours vérifiée en tant qu'authentique.
- *Base64 :* l'élément de racine (ou document) du document XML est considéré être encodé en Base64 et est lu sous sa forme binaire. Si l'élément racine n'est pas en Base64, une erreur est retournée ou l'élément est lu en tant que vide.
- *Aucun :* aucune transformation n'est effectuée et les données XML provenant du fichier binaire enregistré sur le disque sont passées directement pour la création de la signature. Toute modification ultérieure entraînera un échec de la vérification de la signature. Néanmoins, si l'option *Supprimer les espaces* est cochée, tous les espaces blancs seront supprimés et les modifications des espaces blancs seront ignorées. Une différence considérable entre l'option *Aucun* et l'option *Canonicalisation* est que cette dernière produit un flux de données XML dans lequel certaines différences, comme l'ordre des attributs sont normalisées. En conséquence, une transformation de canonicalisation normalisera tous les changements comme par exemple l'ordre des attributs (donc la vérification réussira), alors que la non transformation réfléchira un tel changement (la vérification échouera). Veuillez noter, néanmoins, qu'une canonicalisation par défaut est effectuée si la signature est intégrée (enveloppée ou enveloppante). Ainsi, les données de XML seront utilisées telles qu'elle (donc sans transformation), lorsque la signature est détachée, *Aucun* est sélectionné, et la case *Supprimer les espaces* n'est pas cochée.

Position de la signature XML

La signature XML peut être placée dans le fichier XML ou être créée en tant que fichier séparé. Les options suivantes sont disponibles :

- *Enveloppée :* l'élément `Signature` est créé en tant que le dernier élément enfant de l'élément racine (ou document).
- *Enveloppante :* l'élément `Signature` est créé en tant que l'élément racine (ou document), et le document XML est inséré en tant qu'un élément enfant.
- *Détachée :* la signature XML est créée en tant que fichier séparé. Dans ce cas, vous pouvez spécifier l'extension de fichier du fichier de signature et si le nom de fichier est créé avec : (i) l'extension apposée au nom du fichier XML (par exemple, `test.xml.xsig`), ou (ii) l'extension remplaçant l'extension XML du fichier XML (par exemple, `test.xsig`). Vous pouvez aussi spécifier si, dans le fichier de signature, la référence au fichier XML est un chemin relatif ou absolu.

Note : Les signatures XML pour les fichiers de Schéma XML (`.xsd`) peuvent être créées depuis le Mode Schéma en tant que fichiers de signature détachés (pas intégrés). Les signatures XML pour les fichiers

© 2016-2022 Altova GmbH

XBRL peuvent être créées depuis le Mode XBRL en tant que fichiers de signature détachés (non intégrés). Les signatures XML pour les fichiers WSDL peuvent être créées à partir du Mode WSDL en tant que fichiers de signature détachés ou ils peuvent être "enveloppés" dans le fichier WSDL.

Note : Si la signature XML est créée en tant que fichier détaché (séparé), le fichier XML et le fichier de signature sont associés l'un à l'autre par le biais d'une référence dans le fichier de signature. Par conséquence, la vérification de la signature dans les cas où la signature se trouve dans un fichier externe doit être effectuée avec un fichier de signature actif, et non avec le fichier XML actif.

Apposer des informations-clé

L'option *Apposer info clé* est disponible lorsque la signature est basée sur un certificat. Elle n'est pas disponible si la signature se base sur un mot de passe.

Si l'option est choisie, l'information de clé publique est placée dans la signature, sinon, l'information de clé n'est pas incluse dans la signature. L'avantage d'inclure l'information de clé est que le certificat lui-même (en particulier l'information de clé publique contenue) ne sera pas nécessaire pour le processus de vérification (puisque l'information de clé est contenue dans la signature).

5.13.2 Vérifier des Signatures XML

Une signature XML sera vérifiée correctement si le fichier XML n'a pas été modifié depuis sa signature. Sinon la vérification échouera. Les signatures XML peuvent être vérifiées dans XMLSpy dans les circonstances suivantes telles que décrit ci-dessous :

- le fichier XML contient une signature basée sur certificat, les informations clé sont contenues dans la signature
- le fichier XML contient une signature basée sur certificat, les informations clé ne sont pas contenues dans la signature
- signature basée sur certificat dans un fichier externe, les informations clé sont contenues dans la signature
- signature basée sur certificat dans un fichier externe, les informations clé ne sont pas contenues dans la signature
- le fichier XML contient une signature basée sur mot de passe
- signature basée sur mot de passe dans le fichier externe

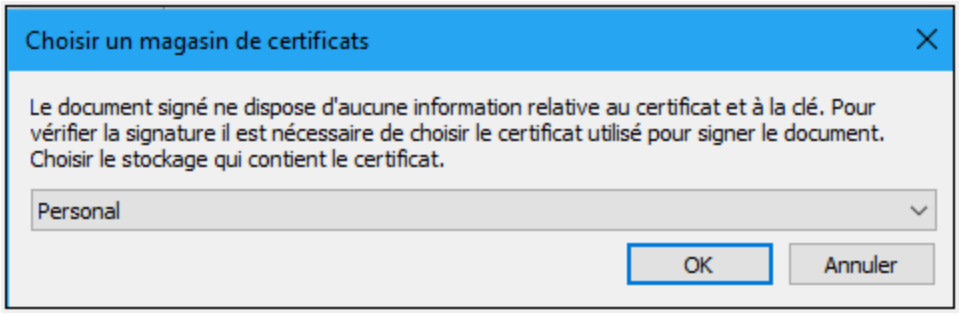
Le fichier XML contient une signature basée sur certificat, les informations clé sont contenues dans la signature

Pour vérifier la signature XML dans ce scénario, activer le fichier XML dans XMLSpy. Cliquer sur la commande **XML | Vérifier la Signature XML**, le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

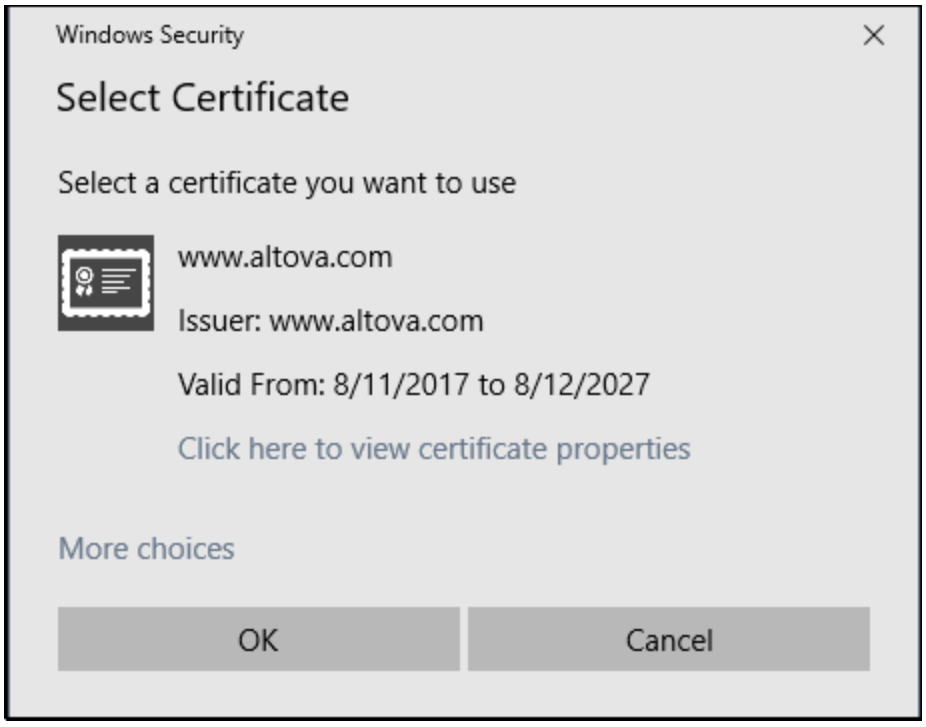
Le fichier XML contient une signature basée sur certificat, les informations clé ne sont pas contenues dans la signature

Si aucune information clé n'est contenue dans la signature basée sur certificat, XMLSpy vous invitera à choisir le certificat à partir duquel l'information de clé publique pour la vérification peut être lue. La vérification est effectuée avec le fichier XML activé dans XMLSpy. Cliquer sur la commande **XML | Vérifier la Signature**

XML, vous serez invité à choisir le magasin de certificat dans lequel le certificat est stocké (*capture d'écran ci-dessous*).



Sélectionner un magasin de certificat et cliquer sur **OK** pour faire apparaître un dialogue affichant les certificats dans ce magasin (*capture d'écran ci-dessous*). Choisir le certificat nécessaire pour la vérification et cliquer sur **OK**.



Le processus de vérification est exécuté et le résultat est affiché dans la fenêtre Messages.

Signature basée sur certificat dans un fichier externe, les informations clé sont contenues dans la signature

Si une signature XML basée sur certificat se trouve dans un fichier externe, la signature est vérifiée avec le fichier XML actif dans XMLSpy. Cliquer sur la commande **XML | Vérifier la Signature XML**, le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

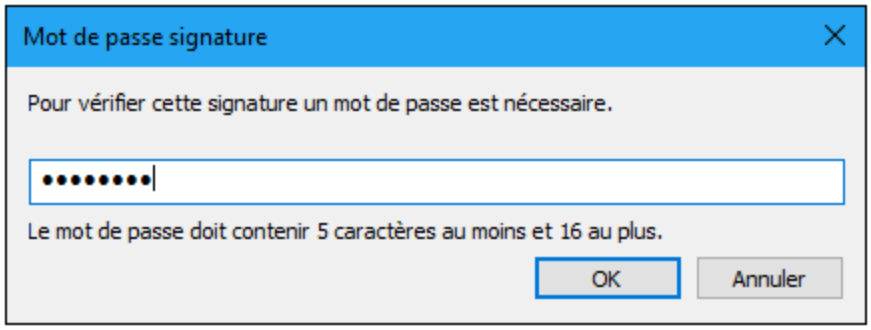
© 2016-2022 Altova GmbH

Signature basée sur certificat dans un fichier externe, les informations clé ne sont pas contenues dans la signature

Si une signature XML basée sur certificat se trouve dans un fichier externe, la signature est vérifiée avec le fichier XML actif dans XMLSpy. Cliquer sur la commande **XML | Vérifier la Signature XML**, XMLSpy vous invitera à choisir le certificat à partir duquel l'information de clé publique pour la vérification peut être lue. Sélectionner le certificat tel que décrit dans la section : *le fichier contient une signature basée sur certificat, les informations clé ne sont pas contenues dans la signature*. Le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (vérification a réussi ou a échoué).

Le fichier XML contient une signature basée sur mot de passe

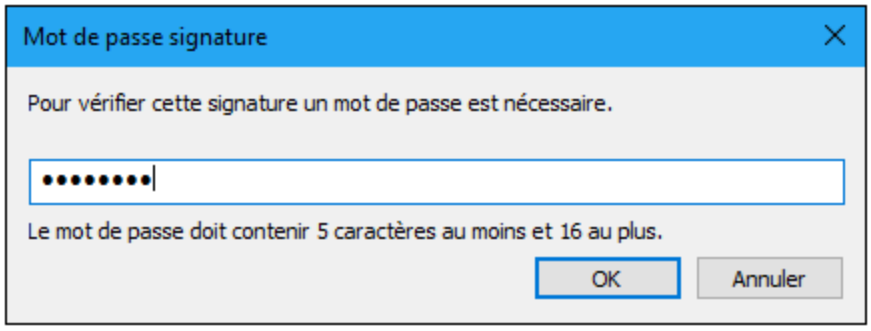
Si le fichier XML contient une signature XML basée sur mot de passe, la signature est vérifiée avec le fichier XML actif dans XMLSpy. Cliquer sur la commande **XML | Vérifier la Signature XML**, un dialogue s'ouvre vous invitant à saisir le mot de passe (*capture d'écran ci-dessous*).



Saisir le mot de passe qui doit contenir entre cinq et seize caractères, puis cliquer sur **OK**. Le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

Signature basée sur mot de passe dans le fichier externe

Si une signature XML sur base de mot de passe se trouve dans un fichier externe, la signature est vérifiée avec le fichier XML actif dans XMLSpy. Cliquer sur la commande **XML | Vérifier la Signature XML**, un dialogue s'ouvre vous invitant à saisir le mot de passe (*capture d'écran ci-dessous*).



Saisir le mot de passe qui doit contenir entre cinq et seize caractères, puis cliquer sur **OK**. Le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

5.13.3 Travailler avec des certificats

Les certificats d'autorisation sont souvent utilisés pour créer et vérifier des signatures XML. Cette section contient des informations concernant l'obtention, l'importation et l'exportation de certificats. Elle est organisée dans les sous-sections suivantes :

- Obtenir un certificat avec une paire de clé publique-privée
- Importer un certificat de clé publique-privée
- Les magasins de certificat sur un appareil Windows
- Exporter un certificat de clé publique

Obtenir un certificat avec une paire de clé publique-privée

Un certificat peut être obtenu des manières suivantes :

- *Depuis une autorité de certificat.* L'autorité de certificat vérifie l'identité du propriétaire du certificat. Les certificats obtenus de cette manière diffèrent des certificats auto-signés qui peuvent être créés par n'importe qui possédant un outil de création de certificats.
- *En créant un certificat auto-signé.* Ce type de certificats n'est pas vérifié par une autorité, mais fournit souvent une sécurité adéquate. Un certain nombre d'outils de création de certificats est disponible, comme Visual Studio de Microsoft.

Pour une utilisation avec des signatures XML signatures, vous devrez utiliser un certificat avec une paire de clé privée-publique.

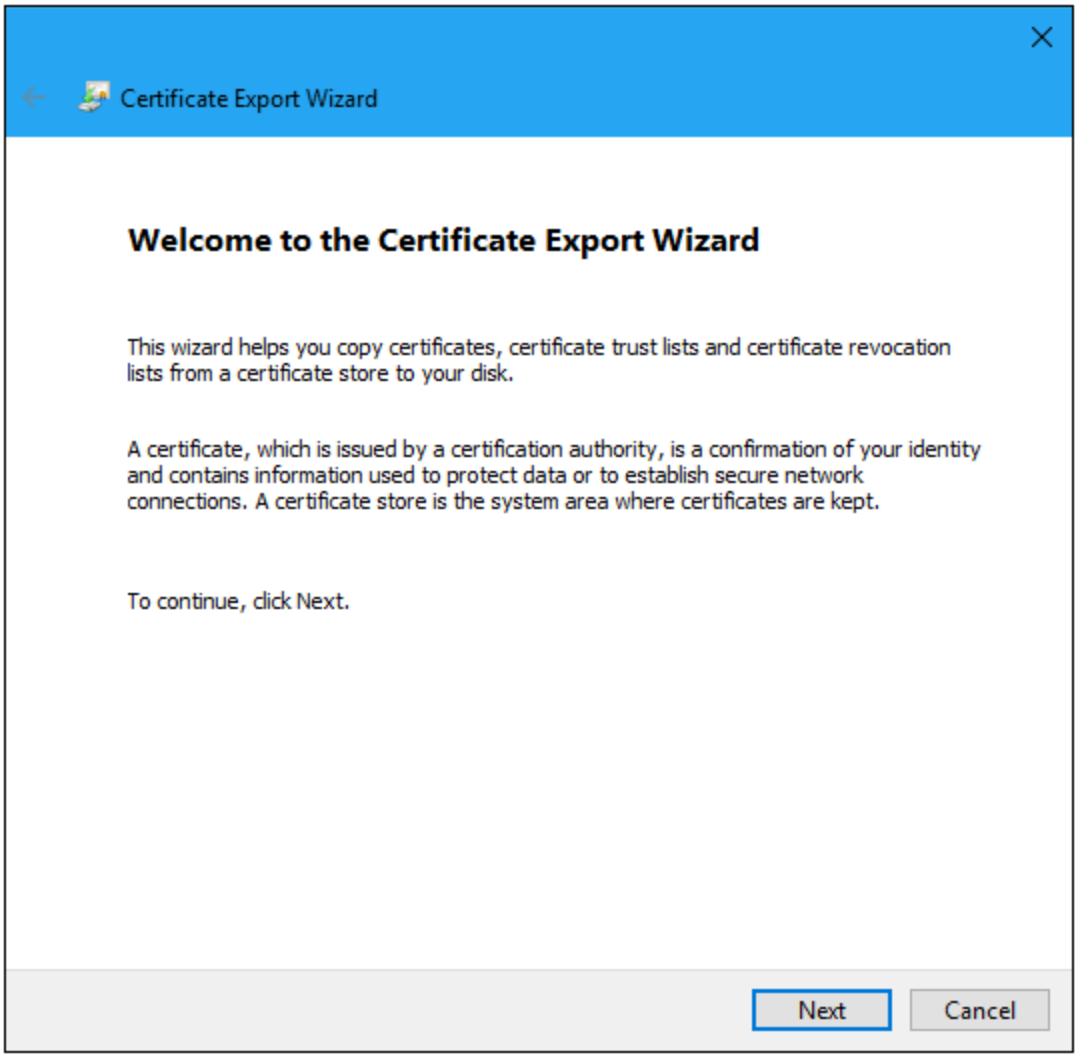
Note : La fonction de signature XML XMLSpy prend en charge uniquement des certificats du type RSA-SHA1, DSA-SHA1 et SHA-256.

Importer un certificat de clé publique-privée

Une fois qu'un certificat de clé privée-publique a été obtenu, vous devrez l'importer dans votre magasin de certificats Windows. Procéder comme suit :

1. Double-cliquer le fichier de certificat pour ouvrir l'Assistant d'importation du certificat (*capture d'écran ci-dessous*), et cliquer sur **Suivant**.

© 2016-2022 Altova GmbH

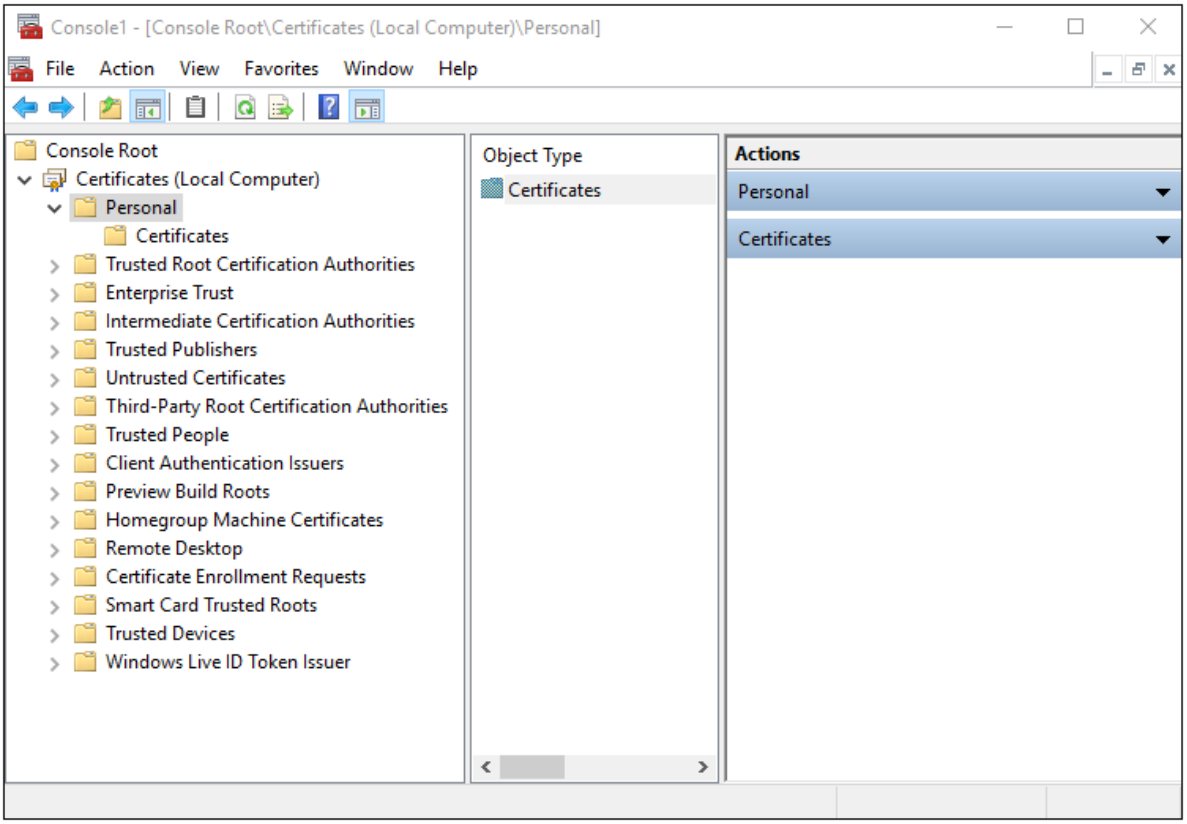


2. Dans la fenêtre Fichier à importer, veuillez vous assurer que le fichier de certificat a été sélectionné, puis cliquer sur **Suivant**.
3. Saisir le mot de passe pour la clé privée. Vous devez connaître le mot de passe si vous voulez utiliser la clé privée pour créer une signature XML. Le mot de passe pour la clé privée vous sera fourni lorsque vous obtiendrez le certificat. Une fois avoir saisi le mot de passe, cliquer sur **Suivant**.
4. Vous pouvez permettre l'assistant de sélectionner automatiquement le magasin dans lequel placer le certificat, conformément au type de certificat, ou vous pouvez sélectionner le magasin vous-même. (Il vaut mieux choisir le magasin vous-même pour connaître l'emplacement du certificat.) Cliquer sur **Suivant** une fois terminé.
5. Cliquer sur **Terminer** pour clore le processus.

Les magasins de certificat sur un appareil Windows

Pour accéder à un magasin de certificat sur un appareil Windows XP :

1. Dans le menu Démarrage, choisir **Exécuter**.
2. Saisir `mmc` et cliquer sur **OK**. Une fenêtre Console s'ouvre (*capture d'écran ci-dessous*).



3. Dans la fenêtre Console, sélectionner la commande **Fichier | Ajouter/Supprimer Snap-in**.
4. Dans l'onglet *Standalone* du dialogue Ajouter/Supprimer Snap-in qui s'ouvre, cliquer sur **Ajouter**.
5. Dans le dialogue Ajouter Standalone Snap-in qui s'ouvre, choisir Certificats et cliquer sur **Ajouter**.
6. Fermer le dialogue Ajouter Standalone Snap-in.
7. Dans le dialogue Ajouter Standalone Snap-in, cliquer sur **OK**.
8. La Racine Console dans la fenêtre Console contient maintenant un item de Certificat (*voir capture d'écran ci-dessus*). Celui-ci contient les magasins de certificat de votre appareil.
9. Enregistrer la Console en tant qu'un fichier Microsoft Management Console (`.msc` file) par le biais de la commande **Fichier | Enregistrer** de la fenêtre Console. Vous pourrez par la suite utiliser ce fichier MSC (via la commande **Fichier | Ouvrir** d'une fenêtre Console) pour accéder aux magasins de certificat sur votre appareil.

Exporter un certificat de clé publique

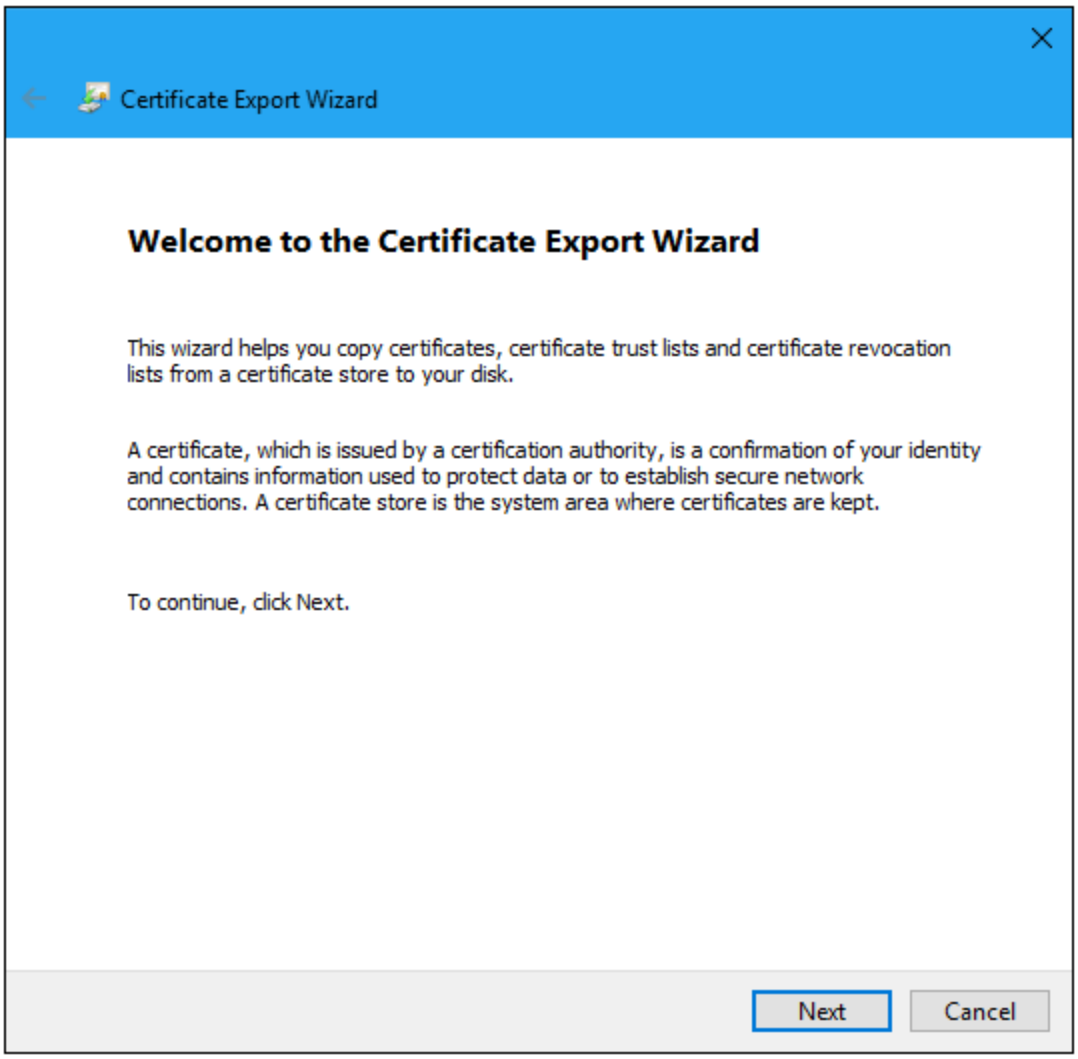
Si vous disposez d'un certificat avec une clé privée-publique, vous souhaiterez éventuellement exporter ce certificat avec uniquement une clé publique. Ce certificat peut ensuite être envoyé aux destinataires pour une utilisation dans des signatures vérifiées créées avec la clé privée du certificat.

Un certificat de clé publique peut être exporté depuis un certificat à clé privée-publique existant comme suit :

1. Ouvrir les magasins de certificat dans une fenêtre Console. Pour ce faire : (i) Saisir `mmc` dans la ligne de commande **Lancer** du menu **Démarrer** ; (ii) Dans la fenêtre Console qui apparaît, choisir **Fichier | Ouvrir**, et sélectionner le fichier MSC dans lequel les magasins de certificat ont été enregistrés (*voir la section juste au-dessus de cette section*).

© 2016-2022 Altova GmbH

2. Chercher le certificat que vous souhaitez exporter en tant que certificat de clé publique et cliquer dessus avec la touche de droite .
3. Dans le menu contextuel qui s'ouvre, sélectionner **Toutes les tâches | Exportation**. L'Assistant d'exportation de certificat s'ouvre (*capture d'écran ci-dessous*).



4. Sélectionner **Suivant**.
5. Dans la fenêtre Exporter clé privée, sélectionner *Non, ne pas exporter la clé privée*, et cliquer sur **Suivant**.
6. Dans la fenêtre Exporter format de fichier, sélectionner le format souhaité (ne pas modifier le format par défaut DER si vous n'êtes pas sûr de ce que vous faites), et cliquer sur **Suivant**.
7. Dans la fenêtre Fichier à exporter, chercher l'emplacement où vous souhaitez enregistrer le fichier et fournir un nom pour le fichier (sans extension de fichier, qui sera ajouté automatiquement). Click sur **Suivant** une fois que vous avez terminé.
8. Cliquer sur **Terminer** pour achever l'exportation.

Un certificat de clé publique sera créé à l'emplacement que vous avez spécifié. Ce certificat peut être envoyé à des destinataires de fichiers XML signés avec la clé privée correspondante. Le destinataire peut ensuite importer ce certificat de clé publique dans un magasin de certificat sur son appareil et utiliser la clé publique de ce certificat pour une vérification.

© 2016-2022 Altova GmbH

5.14 Fonctions supplémentaires

Les fonctions supplémentaires pour travailler avec des fichiers XML sont regroupées ci-dessous.

- Encoder
- Générer des DTD et des Schémas XML
- Chercher et remplacer
- Évaluer un XPath
- Importer et exporter du texte

Encoder

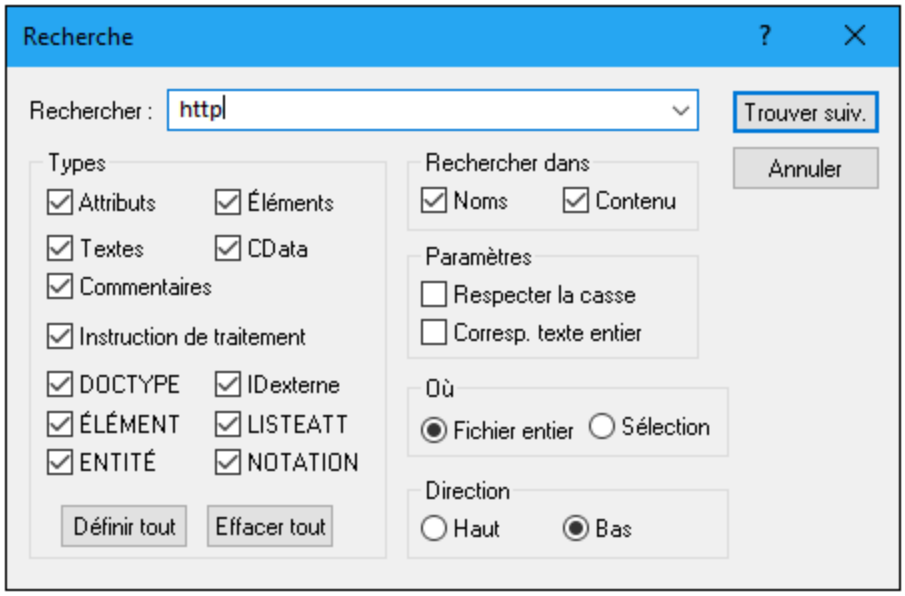
L'encodage des fichiers XML (et d'autres types de documents) peut être configuré par le biais de la commande de menu **Fichier | Encodage**. L'encodage par défaut de fichiers XML et non XML peut être spécifié dans la section Options | Encodage.

Générer des DTD et des Schémas XML

Si vous souhaitez créer un schéma qui décrit la structure d'un document XML, utiliser la commande de menu **DTD/Schéma | Générer DTD/Schéma**. Dans le dialogue Générer DTD/Schéma qui apparaît, vous pouvez sélectionner si vous souhaitez générer un DTD ou un Schéma XML ainsi que certaines options de Schéma XML, comme par exemple si vous souhaitez générer des énumérations depuis les valeurs contenues dans le document XML.

Chercher et remplacer

Les fonctions Chercher et Remplacer (accédées par le biais du menu **Édition**) proposent des fonctions de recherche puissantes. Outre la casse, vous pouvez définir le terme de recherche et la correspondance de mots entiers, le terme peut aussi être exprimé en tant qu'expression régulière. La plage de recherche peut se limiter à une sélection dans le document et à des types de nœuds particuliers (*voir capture d'écran ci-dessous*).



La capture d'écran ci-dessus montre le dialogue Chercher dans le Mode Grille. Le dialogue et les fonctions varient selon le mode actif. Pour voir les fonctions Chercher et Remplacer dans le Mode Texte, voir ici.

Évaluer un XPath

Une expression XPath que vous saisissez dans la fenêtre XPath/XQuery peut être évaluée par rapport au document XML actif. Les résultats de l'évaluation sont affichés dans la fenêtre XPath/XQuery, et en cliquant sur un nœud dans le résultat, permet de marquer ce nœud dans l'affichage de document dans la Fenêtre principale. Veuillez noter que la fenêtre XPath/XQuery peut être rendue active en cliquant sur la commande **XML | Évaluer XPath**.

Importer et exporter du texte

Les données de texte peuvent être importées de et exportées vers tous les formats d'application. Les commandes pour ces fonctions se trouvent dans le menu **Convertir**.

© 2016-2022 Altova GmbH

6 DTD et Schémas XML

Site Internet Altova : Éditeur de Schéma XML
Cette section fournit un aperçu du travail avec les DTD et les Schémas XML. Elle décrit aussi SchemaAgent et la fonction puissante Chercher dans Schémas. Outre les fonctions d'édition, XMLSpy présente les fonctions de DTD/Schéma suivantes :

Mécanisme de catalogue

La prise en charge du mécanisme de catalogue OASIS permet de réorienter les URI sur des adresses locales, facilitant ainsi l'utilisation sur plusieurs postes de travail.

Règles Schéma

Un ensemble de contraintes supplémentaires définies par l'utilisateur peut être attribué au Schéma XML. XMLSpy contient un Éditeur de règles de schéma dans lequel un ensemble de règles de schéma peut être créé et édité pour un Schéma XML.

Sous-ensembles de schéma

Les composants d'un grand schéma peuvent, dans le Mode Schéma, être créés en tant que fichiers séparés. Ces petits sous-ensembles de schéma peuvent ensuite être inclus dans le schéma plus grand. L'opération inverse, connue sous la désignation Aplatir un schéma, place les composants des fichiers inclus directement dans le schéma plus important. Vous trouverez une description de la génération de sous-ensembles de schéma et d'aplatissement de schémas dans la section, Sous-ensembles de Schéma.

Convertir des DTD en Schémas XML et vice-versa

Une DTD peut être convertie en un Schéma XML et vice-versa, et les deux types de document peuvent être aplatis par le biais des commandes situées dans le menu **DTD/Schéma**. Lorsqu'une DTD est aplatie, les composants contenus dans les modules inclus/importés sont enregistrés directement dans le fichier parent et les composants non utilisés sont supprimés.

Générer un fichier XML échantillon

Vous pouvez générer, par le biais de la commande **DTD/Schéma | Générer un fichier XML échantillon**, un document XML squelette basé sur la DTD active ou sur le fichier de Schéma XML. Cela peut être très utile pour les créations rapides de fichiers XML basés sur le schéma actif.

Se rendre à la définition

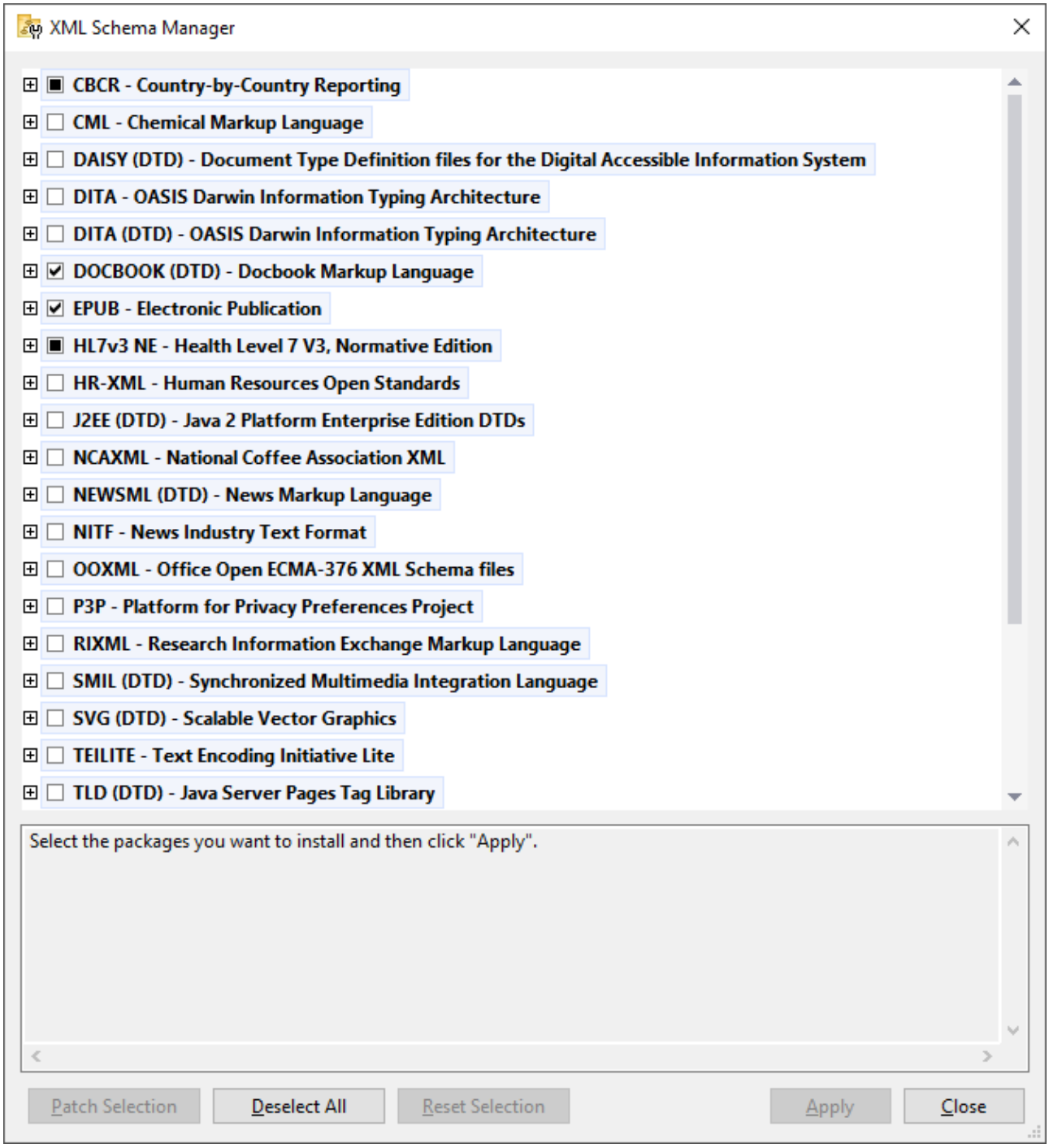
Lorsque le curseur est placé dans un nœud dans un document XML, cliquer sur la commande de menu **DTD/Schéma | Aller à la définition** pour ouvrir le fichier de schéma et marquer la définition du nœud XML sélectionné.

6.1 Gestionnaire de schéma

XML Gestionnaire de schéma est un outil qui propose un moyen centralisé d'installer et de gérer des schémas XML (DTD pour XML et Schémas XML) pour une utilisation sur toutes les applications XBRL Altova, y compris XMLSpy

- Sur Windows, Gestionnaire de schéma a une interface utilisateur graphique *(voir la capture d'écran ci-dessous*) et est aussi disponible dans la ligne de commande. (Les applications desktop d'Altova sont disponibles sur Windows uniquement ; *voir la liste ci-dessous*.)
- Sur Linux et Mac Gestionnaire de schéma, l'outil est disponible uniquement dans la ligne de commande. (Les applications serveur d'Altova sont disponibles sur Windows, Linux et macOS ; *voir la liste ci-dessous*.)

© 2016-2022 Altova GmbH



Application d'Altova qui fonctionnent avec Schema Manager

Applications desktop (Windows uniquement)	Applications de serveur (Windows, Linux, macOS)
XMLSpy (toutes éditions)	RaptorXML Server, RaptorXML+XBRL Server

MapForce (toutes éditions)	StyleVision Server
StyleVision (toutes éditions)	
Authentic Desktop Enterprise Edition	

Installation et désinstallation de Gestionnaire de schéma

Gestionnaire de schéma est installé automatiquement quand vous installez d'abord une nouvelle version de l'Altova Mission Kit Enterprise Edition ou toute application XML-schema-aware d'Altova (*voir la table ci-dessus*).

De même, il est supprimé automatiquement lorsque vous désinstallez la dernière application XML-schema-aware d'Altova depuis votre ordinateur.

Fonctions <% SCHEMA-MANAGER%>

Gestionnaire de schéma propose les fonctions suivantes :

- Affiche les schémas XML installés sur votre ordinateur et contrôle si de nouvelles versions sont disponibles pour le téléchargement.
- Télécharge des versions plus récentes des schémas XML indépendamment du cycle de release des produits Altova. (Altova stocke des schémas en ligne et vous pouvez les télécharger via Gestionnaire de schéma.)
- Installer ou désinstaller une des multiples versions d'un schéma donné (ou toutes les versions, si nécessaire).
- Un schéma XML peut avoir des dépendances sur d'autres schémas. Lorsque vous installez ou désinstallez un schéma particulier, Gestionnaire de schéma vous informe sur d'autres schémas dépendants et les installera ou désinstallera également automatiquement.
- Gestionnaire de schéma utilise le mécanisme du catalogue XML pour mapper les références de schéma aux fichiers locaux. Dans le cas de larges schémas XML, le traitement sera plus rapide que si les schémas étaient à un emplacement à distance.
- Tous les schémas majeurs sont disponibles via Gestionnaire de schéma et sont régulièrement mis à jour pour les dernières versions. Ceci vous fournit une ressource unique pour gérer tous vos schémas et les mettre à disposition de toutes les applications XML-schema-aware d'Altova.
- Les changements réalisés dans Gestionnaire de schéma prennent effet pour tous les produits d'Altova sur cet appareil.

Comment cela fonctionne ?

Altova stocke tous les schémas XML utilisés dans les produits Altova en ligne. Ce référentiel est mis à jour lorsque de nouvelles versions de schémas sont publiées. Gestionnaire de schéma affiche des informations sur les derniers schémas disponibles lorsqu'ils sont appelés dans son formulaire GUI de même que sur CLI. Vous pouvez ensuite installer, mettre à jour ou désinstaller les schémas via Gestionnaire de schéma.

Gestionnaire de schéma installe également les schémas d'une autre manière. Sur le site web d'Altova (https://www.altova.com/schema-manager), vous pouvez sélectionner un schéma et ses Schémas dépendants que vous souhaitez installer. Le site web préparera un fichier de type `.altova_xmlschemas` pour le téléchargement qui contient des informations sur la sélection de schéma. Lorsque vous double-cliquez sur ce fichier ou le passez à Gestionnaire de schéma via CLI comme argument de la commande `installer`, Gestionnaire de schéma installera les schémas que vous avez sélectionnés.

© 2016-2022 Altova GmbH

Cache local : suivre vos schémas
Toutes les informations sur les schémas installés sont suivies dans un répertoire cache centralisé sur votre ordinateur, situé ici :

Windows	`C:\ProgramData\Altova\pkgs\.cache`
Linux	`/var/opt/Altova/pkgs\.cache`
macOS	`/var/Altova/pkgs`

Ce répertoire cache est mis à jour régulièrement avec le dernier statut des schémas dans l'emplacement de stockage en ligne d'Altova. Ces mises à jour sont réalisées aux moments suivants :

- À chaque fois que vous lancez Gestionnaire de schéma.
- Lorsque vous exécutez XMLSpy pour la première fois dans un jour donné du calendrier.
- Si XMLSpy est ouvert plus de 24 heures, le cache est mis à jour toutes les 24 heures.
- Vous pouvez aussi mettre à jour le cache en exécutant la commande de `mise à jour` dans l'interface de ligne de commande.

Pour cette raison, le cache permet à Gestionnaire de schéma de suivre continuellement vos schémas installés par rapport aux schémas disponibles en ligne sur le site web d'Altova.

Ne modifiez pas le cache manuellement !

Le répertoire de cache local est entretenu automatiquement sur la base des schémas que vous installez ou désinstallez. Il ne devrait pas être altéré ou supprimé manuellement. Si vous êtes amené à réinitialiser Gestionnaire de schéma à son état original "intact", alors, sur l'interface de la ligne de commande (CLI) : (i) exécutez la commande `reset`, et (ii) exécutez la commande `initialize`. (En alternative, exécutez la commande **`reset`** avec l'option **`--i`**.)

6.1.1 Exécuter Schema Manager

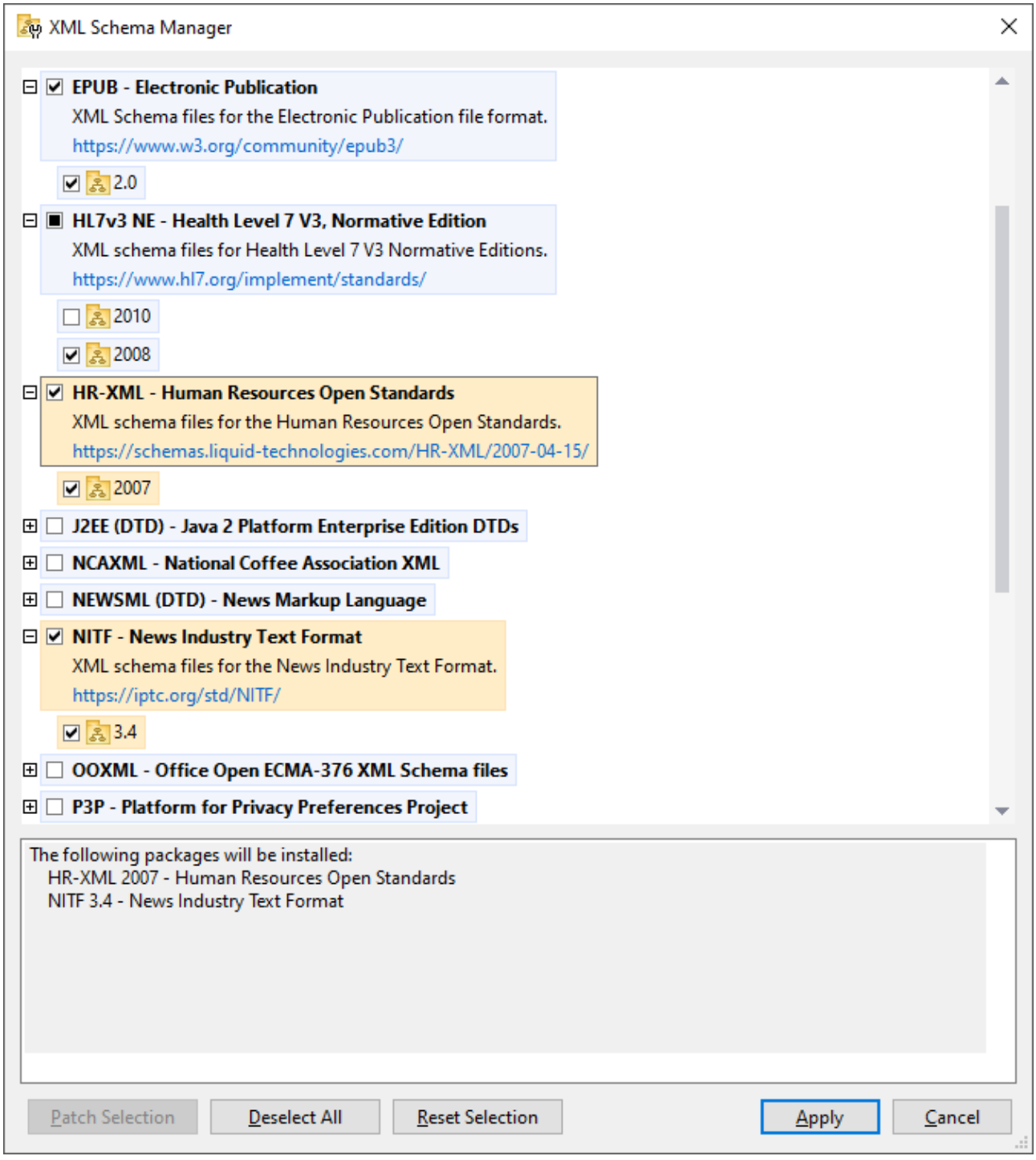
Interface utilisateur graphique

Vous pouvez accéder à la GUI de Gestionnaire de schéma des manières suivantes :

- *Durant l'installation de XMLSpy:* Vers la fin de la procédure d'installation, sélectionnez la case à cocher *Invoke Altova XML-Schema Manager* pour accéder directement à la GUI de Gestionnaire de schéma. Ceci vous permettra d'installer les schémas au cours de la procédure d'installation de votre application Altova.
- *Après l'installation de XMLSpy:* Une fois que votre installation a été installée, vous pouvez accéder à la GUI de Gestionnaire de schéma à tout moment, via la commande de menu **Tools | XML Schema Manager**.
- Via le fichier **`.altova_taxonomies`** téléchargé de Altova website: Double-cliquez sur le fichier téléchargé pour exécuter la GUI de Gestionnaire de schéma, qui sera configurée pour installer les schémas que vous avez sélectionnés (le site web) pour installation.

Une fois que la GUI de Gestionnaire de schéma (*capture d'écran ci-dessous*) a été ouverte, les schémas déjà installés seront affichés tels sélectionnés. Si vous voulez installer un schéma additionnel, sélectionnez-le. Si vous voulez désinstaller un schéma déjà installé, désélectionnez-le. Une fois que vous avez faits vos sélections et/ou désélections, vous êtes prêts pour appliquer vos changements. Les schémas qui seront installés ou désinstallés seront mis en surbrillance et un message sur les modifications à venir sera posté dans le volet Messages au niveau inférieur de la fenêtre Gestionnaire de schéma (*voir la capture d'écran*).

© 2016-2022 Altova GmbH



Interface de ligne de commande

Vous pouvez exécuter Gestionnaire de schéma depuis une interface de ligne de commande en sélectionnant son fichier exécutable, **`xmlschemamanager.exe`**.

Le fichier **xmlschemamanager.exe** est situé dans le dossier suivant :

- *Sur Windows :* `C:\ProgramData\Altova\SharedBetweenVersions`
- *Sur Linux ou macOS (application serveur uniquement) :* `%INSTALLDIR%/bin`, où `%INSTALLDIR%` est le répertoire d'installation du programme.

Vous pouvez alors utiliser toute commande dans la section de référence de la commande CLI.

Pour afficher l'aide pour la commande, exécutez l'étape suivante :

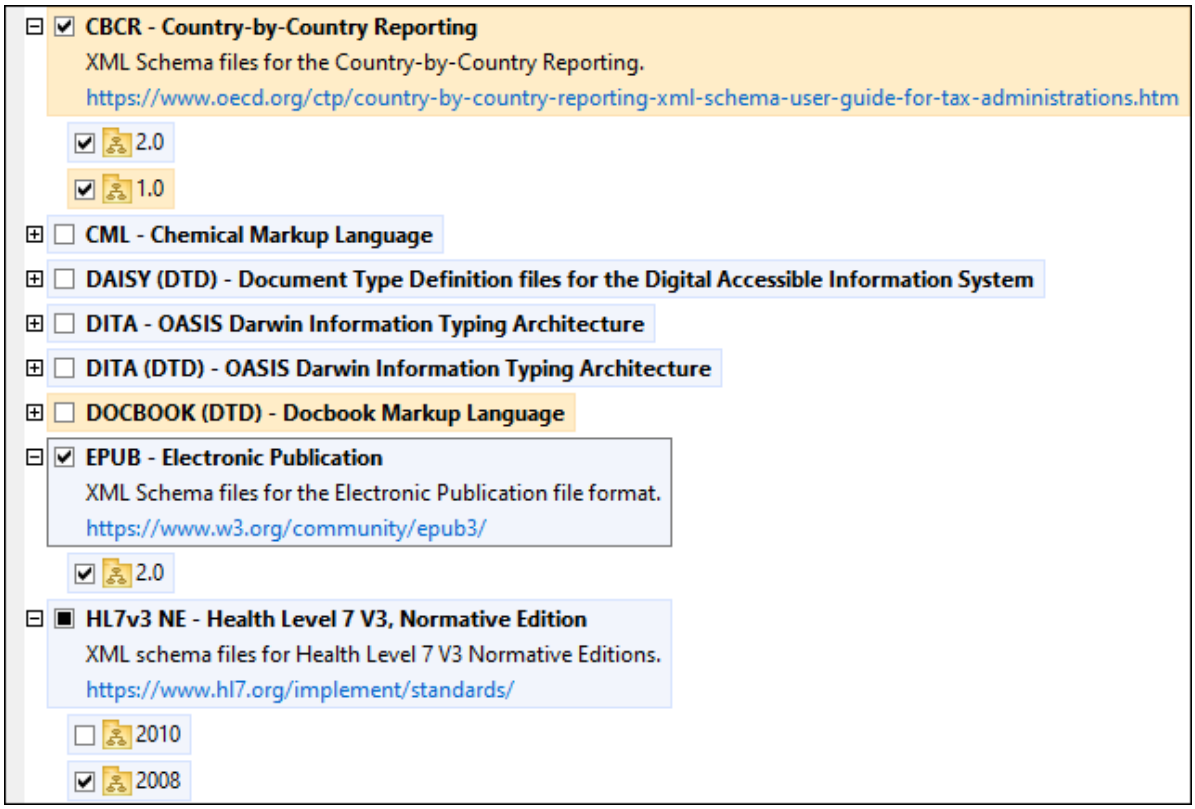
- *Sur Windows :* `xmlschemamanager.exe --help`
- *Sur Linux ou macOS (application serveur uniquement) :* `sudo ./xmlschemamanager --help`

6.1.2 Catégories de statut

Gestionnaire de schéma catégorise les schémas sous sa gestion comme suit :

- *Schémas installés.* Ceux-ci sont affichés dans la GUI avec leurs cases à cocher sélectionnées (*dans la capture d'écran ci-dessous, les versions cochées et bleues des schémas EPUB et HL7v3 NE sont des schémas installés*). Si toutes les versions de schéma sont sélectionnées, alors la marque de sélection est une coche. Si au moins une version de schéma est décochée, alors la coche de sélection est un carré coloré plein. Vous pouvez décocher un schéma installé pour le **désinstaller** ; (*dans la capture d'écran ci-dessous, le DocBook DTD est installé et a été désélectionné, le préparant ainsi pour la désinstallation*).
- *Schémas désinstallés disponibles.* Ils sont affichés dans la GUI avec leurs cases à cocher non sélectionnées. Vous pouvez sélectionner les schémas que vous souhaitez **installer**.

© 2016-2022 Altova GmbH



- Les *Schémas pouvant être mises à niveau* sont ceux qui ont été revus par leurs émetteurs depuis qu'ils ont été installés. Ils sont indiqués dans la GUI par une icône . Vous pouvez **retoucher** le schéma installé avec la révision disponible.

Points à noter

- Dans la capture d'écran ci-dessus, les deux schémas CBCR sont cochés. Celui avec un arrière-plan bleu est déjà installé. Celui avec un arrière-plan jaune est désinstallé et a été sélectionné pour l'installation. Notez que le schéma HL7v3 NE 2010 n'est pas installé et n'a pas été sélectionné pour l'installation.
- Un arrière-plan jaune signifie que le schéma sera modifié d'une manière ou d'une autre quand le bouton **Appliquer** est cliqué. Si un schéma est décoché et a un arrière-plan jaune, cela signifie qu'il sera désinstallé quand le bouton **Appliquer** est cliqué. Dans la capture d'écran ci-dessus, le DocBook DTD a un tel statut.
- Lorsque vous exécutez Gestionnaire de schéma depuis la ligne de commande, la commande `list` est utilisée avec différentes options pour pour recenser les différentes catégories de schémas :

`xmlschemamanager.exe list`	Recense tous les schémas installés et disponibles ; ceux pouvant être mis à niveau sont également indiqués.
`xmlschemamanager.exe list -i`	Recense les schémas installés uniquement ; ceux pouvant être mis à niveau sont également indiqués
`xmlschemamanager.exe list -u`	Recense les schémas pouvant être mis à niveau

Note : Sur Linux et macOS, **`use sudo ./ xmlschemamanager list`**

6.1.3 Retoucher ou Installer un schéma

Retoucher un schéma installé

Occasionnellement, des schémas XML peuvent recevoir des patchs (mises à niveau ou révisions) de leurs émetteurs. Lorsque Gestionnaire de schéma détecte que des patchs sont disponibles, ceux-ci sont indiqués dans les listes de schéma de Gestionnaire de schéma et vous pouvez installer les patchs rapidement.

Dans la GUI

Les patchs sont indiqués par l'icône . (*Voir aussi la rubrique précédente sur les catégories de statut.*) Si les patchs sont disponibles, le bouton **Patch Selection** sera activé. Cliquez dessus pour sélectionner et préparer tous les patchs pour installation. Dans la GUI, l'icône de chaque schéma sera patchée de  à , et le volet des Messages en bas du dialogue recense les patchs qui doivent être appliqués. Lorsque vous êtes prêt pour installer des patchs sélectionnés, cliquez sur **Appliquer**. Tous les correctifs seront appliqués ensemble. Notez que si vous décochez un schéma marqué pour une correction, vous désinstallerez de fait ce schéma.

Sur le CLI

Pour appliquer un patch dans l'interface de ligne de commande :

1. Exécuter la commande `list -u` . Cela liste tout schéma lorsque des mises à niveau sont disponibles.
2. Exécutez la commande `upgrade` pour installer les patchs.

Installer un schéma disponible

Vous pouvez installer des schémas en utilisant soit la GUI Gestionnaire de schéma ou en envoyant à Gestionnaire de schéma les instructions d'install via la ligne de commande.

Note : Si le schéma actuel référence d'autres schémas, les schémas référencées sont aussi installés.

Dans la GUI

Pour installer des schémas utilisant la GUI Gestionnaire de schéma GUI, sélectionnez les schémas que vous voulez installer et cliquez sur **Appliquer**.

Vous pouvez aussi sélectionner les schémas que vous voulez installer sur le site web d'Altova et générer un fichier téléchargeable **`.altova_xmlschemas`**. Lorsque vous double-cliquez sur ce fichier, il ouvrira Gestionnaire de schéma avec les schémas que vous vouliez présélectionner. La seule chose qui vous reste à faire, c'est cliquer sur **Appliquer**.

Sur le CLI

Pour installer des schémas via la ligne de commande, exécutez la commande `install` :

```
xmlschemamanager.exe install [options] Schema+
```


© 2016-2022 Altova GmbH

où **Schéma** est le schéma (ou les schémas) que vous voulez installer ou un fichier **.altova_xmlschemas**. Un schéma est référencé par un identifiant de format **<name>-<version>**. (Les identifiants de schémas sont affichés quand vous exécutez la commande list.) Vous pouvez saisir autant de schémas que vous le souhaitez. Pour plus de détails, voir la description de la commande install.

Note : sur Linux ou macOS, utilisez la commande **sudo ./xmlschemamanager**.

Installer un schéma requis
Lorsque vous exécutez une commande activée par XML dans XMLSpy, et que XMLSpy découvre qu'un schéma dont il a besoin pour exécuter la commande n'est pas présente ou est incomplète, Gestionnaire de schéma affichera l'information sur les schémas manquants. Vous pouvez ensuite installer directement tout schéma manquant via Gestionnaire de schéma.

Dans la GUI de Schema Manager, vous pouvez consulter tous les schémas précédemment installés à tout moment en exécutant Gestionnaire de schéma depuis **Outils | Gestionnaire de schéma**.

6.1.4 Désinstaller un schéma, Réinitialiser

Désinstaller un schéma

Vous pouvez désinstaller des schémas en utilisant soit la GUI Gestionnaire de schéma ou en envoyant à Gestionnaire de schéma les instructions d'installation via la ligne de commande.

Note : si la Police que vous voulez désinstaller référence d'autres Schémas, alors les Schémas référencées sont également désinstallées.

Dans la GUI
Pour désinstaller les schémas utilisant la GUI Gestionnaire de schéma, effacez leurs cases à cocher et cliquez sur **Appliquer**. Les schémas sélectionnés et leurs schémas référencées seront désinstallés.

Pour désinstaller tous les schémas, cliquez sur **Désélectionner tout** et cliquez sur **Appliquer**.

Sur le CLI
Pour désinstaller des schémas via la ligne de commande, exécutez la commande désinstaller :

```
xmlschemamanager.exe uninstall [options] Schema+
```

où chaque argument **Schéma** est le schéma que vous voulez désinstaller ou un fichier **.altova_xmlschemas**. Un schéma est spécifié par un identifiant qui a un format **<name>-<version>**. (Les identifiants de schémas sont affichés quand vous exécutez la commande list.) Vous pouvez saisir autant de schémas que vous le souhaitez. Pour plus de détails, voir la description de la commande désinstaller.

Note : sur Linux ou macOS, utilisez la commande **sudo ./xmlschemamanager**.

Réinitialiser Gestionnaire de schéma

Vous pouvez réinitialiser Gestionnaire de schéma. Ceci supprime toutes les schémas installés et le répertoire de mise sous cache.

- Dans la GUI, cliquez sur **Reset Selection**.
- Dans la CLI, exécutez la commande `reset`.

Une fois avoir exécuté cette commande, vous devrez exécuter la commande `initialize`, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécutez la commande `reset` avec l'option `-i`.

Notez que `reset -i` restaure l'installation originale du produit, il est recommandé d'exécuter la commande `update` après avoir réalisé la réinitialisation. En alternative, exécutez la commande `reset` avec les options `-i` and `-u`.

6.1.5 Interface de ligne de commande (CLI)

Pour appeler Gestionnaire de schéma dans la ligne de commande, vous devez connaître le chemin de l'exécutable. Par défaut, l'exécutable Gestionnaire de schéma est installé dans le chemin suivant :

```
C:\ProgramData\Altova\SharedBetweenVersions\XMLSchemaManager.exe
```

Note : sur les systèmes Linux et macOS, une fois que vous avez changé le répertoire à celui contenant l'exécutable, vous pouvez appeler l'exécutable avec **`sudo ./xmlschemamanager`**. Le préfixe **`./`** indique que l'exécutable est le répertoire actuel. Le préfixe **`sudo`** indique que la commande que la commande doit être exécutée avec des privilèges root.

Syntaxe de ligne de commande

La syntaxe générale pour utiliser la ligne de commande est la suivante :

```
<exec> -h | --help | --version | <command> [options] [arguments]
```

Dans l'extrait ci-dessus, la barre verticale **`|`** sépare un ensemble d'items mutuellement exclusifs. Les crochets **`[]`** indiquent des items optionnels. De manière générale, vous pouvez saisir le chemin d'exécutable suivi soit par les options `--h`, `--help`, ou `--version` ou par une commande. Chaque commande peut contenir des options et des arguments. La liste des commandes est décrite dans les sections suivantes.

6.1.5.1 help

Cette commande propose une aide contextuelle pour les commandes liées à l'exécutable Gestionnaire de schéma.

Syntaxe

```
<exec> help [command]
```


© 2016-2022 Altova GmbH

Où `[command]` est un argument optionnel qui spécifie un nom de commande valide.

Veuillez noter les points suivants :

- Vous pouvez invoquer de l'aide en saisissant la commande suivie par **`-h`** ou **`--help`**, par exemple : **`<exec> list-h`**
- Si vous tapez **`-h`** or **`--help`** directement après la commande exécutable et avant une commande, vous recevrez une aide générale (pas d'aide pour la commande), par exemple : **`<exec> -h list`**

Exemple

La commande suivante affiche une aide concernant la commande `list` :

```
xmlschemamanager help list
```

6.1.5.2 info

Cette commande affiche des informations détaillées pour chacun des schémas fournis en tant qu'argument de `Schéma`. Cette information inclut le titre, la version, description, l'éditeur et tout schéma soumis et tout schéma référencé, et informe si le schéma a été installé ou non.

Syntaxe

```
<exec> info [options] Schema+
```

- L'argument **`Schéma`** est le nom d'un schéma ou une partie du nom de schéma. (Pour afficher une ID de pack de schéma et des informations détaillées sur son statut d'installation, vous devriez utiliser la commande `list`.)
- Utiliser **`<exec> info -h`** pour afficher l'aide de la commande.

Exemple

La commande suivante affiche l'information sur les derniers schémas **`DocBook-DTD`** et **`NITF`** :

```
xmlschemamanager info doc nitf
```

6.1.5.3 initialize

Cette commande initialise l'environnement Gestionnaire de schéma. Elle crée un répertoire de cache où les informations concernant tous les schémas sont stockées localement. L'initialisation est réalisée automatiquement la première fois qu'une application schema-cognizant d'Altova est installée. Vous n'aurez pas besoin d'exécuter cette commande dans des circonstances normales, mais vous devrez l'exécuter généralement après la commande **`reset`**.

Syntaxe

```
<exec> initialize | init [options]
```

Options
La commande **initialize** accepte les options suivantes :

--silent, --s	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
--verbose, --v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
--help, --h	Afficher l'aide pour la commande.

Exemple
La commande suivante initialise Gestionnaire de schéma:

```
xmlschemamanager initialize
```

6.1.5.4 install

Cette commande installe un ou plusieurs schémas.

Syntaxe
```
<exec> install [options] Schema+
```

Pour installer de multiples schémas, ajoutez l'argument **Schéma** de nombreuses fois.

L'argument **Schéma** est l'un des suivants :

- Un identifiant de schéma (avoir un format de **<name>-<version>**, par exemple : **cbcr-2.0**). Pour trouver les identifiants de schémas que vous voulez, exécutez la commande list. Vous pouvez aussi utiliser des identifiants abrégés s'ils sont uniques, par exemple **docbook**. Si vous utilisez un identifiant abrégé, alors la dernière version de ce schéma sera installée.
- Le chemin vers un fichier **.altova_xmlschemas** téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir *Introduction à SchemaManager : Comment cela fonctionne-t-il ?*.

Options
La commande **install** accepte les options suivantes :

--silent, --s	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
--verbose, --v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
--help, --h	Afficher l'aide pour la commande.

© 2016-2022 Altova GmbH

Exemple

La commande suivante installe le schéma CBCR 2.0 (Country-By-Country Reporting) et le dernier DocBook DTD:

```
xmlschemamanager install cbcr-2.0 docbook
```

6.1.5.5 list

Cette commande recense les schémas sous la gestion de Gestionnaire de schéma. La liste affiche comme suit

- Tous les schémas disponibles
- Les schémas contenant le string dans leur nom soumis comme argument de **Schéma**
- Seuls les schémas installés
- Seuls les schémas qui peuvent être mis à niveau

Syntaxe

```
<exec> list | ls [options] Schema?
```

Si aucun argument de **Schéma** n'est soumis, alors toutes les schémas disponibles sont recensés. Autrement, les schémas sont recensés par des options soumises (*voir l'exemple ci-dessous*). Notez que vous pouvez soumettre l'argument de **Schéma** de nombreuses fois.

Options
La commande **list** accepte les options suivantes :

`--installed, --i`	Recenser uniquement la liste des schémas installés. Le réglage par défaut est **faux**.
`--upgradeable, --u`	Recenser uniquement les schémas lorsque des mises à niveau (patchs) sont disponibles. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemples

- Pour exécuter tous les schémas disponibles, exécutez : **xmlschemamanager list**
- Pour recenser les schémas installés, exécutez : **xmlschemamanager list -i**
- Pour recenser tous les schémas qui contiennent soit "doc", soit "nitf" dans leur nom, exécutez **xmlschemamanager list doc** :

6.1.5.6 reset

Cette commande supprime tous les schémas installés et le répertoire de mise sous cache. Vous réinitialiserez complètement votre environnement de schéma. Une fois avoir exécuté cette commande, vous devrez exécuter la commande initialize, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécuter la commande **reset** avec l'option **-i**. Pusique **reset -i** restaure l'installation originale du produit, nous vous recommandons que vous exécutiez la commande update après avoir réalisé la réinitialisation et l'initialisation. En alternative, exécutez la commande **reset** avec les options **-i** and **-u**

Syntaxe

```
<exec> reset [options]
```

Options
La commande **reset** accepte les options suivantes :

`--init, --i`	Initialiser Gestionnaire de schéma après le reset. Le réglage par défaut est **faux**.
`--update, --u`	Mettre à jour la liste de schémas disponibles dans le cache. Le réglage par défaut est **faux**.
`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemples

- Pour réinitialiser Gestionnaire de schéma, exécuter : **xmlschemamanager reset**
- Pour réinitialiser Gestionnaire de schéma et l'initialiser, exécutez : **xmlschemamanager reset -i**
- Pour réinitialiser Gestionnaire de schéma, initialiser-le et mettez à jour sa liste de schéma, exécutez : **xmlschemamanager reset -i-u**

6.1.5.7 uninstall

Cette commande désinstalle un ou plusieurs schémas. Par défaut, tout schéma référencé par la taxonomie actuelle sera également désinstallé. Pour désinstaller uniquement le schéma actuel et garder les schémas référencés, définir l'option `--k`.

Syntaxe

```
<exec> uninstall [options] Schema+
```

Pour désinstaller de multiples schémas, ajoutez l'argument **Schéma** de nombreuses fois.

© 2016-2022 Altova GmbH

L'argument **Schéma** est l'un des suivants :

- Un identifiant de schéma (avoir un format de **<name>-<version>**, par exemple : **cbcr-2.0**). Pour trouver les identifiants de schéma qui sont installés, exécutez la commande `list -i` . Vous pouvez aussi utiliser un nom de schéma abrégé s'il est unique, par exemple **docbook**. Si vous utilisez un nom abrégé, alors tous les schémas qui contiennent une abréviation dans leur nom seront désinstallés.
- Le chemin vers un fichier **.altova_xmlschemas** téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir *Introduction à SchemaManager : Comment cela fonctionne-t-il ?*.

Options
La commande **désinstaller** accepte les options suivantes :

`--keep-references, --k`	Définir cette option pour garder les schémas référencés. Le réglage par défaut est **faux**.
`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemple

La commande suivante désinstalle les schémas CBCR 2.0 et EPUB 2.0 et leurs dépendances :

```
xmlschemamanager uninstall cbcr-2.0 epub-2.0
```

La commande suivante désinstalle la taxonomie **eba-2.10** mais pas les schémas qu'elle référence :

```
xmlschemamanager uninstall --k cbcr-2.0
```

6.1.5.8 update

Cette commande requête la liste des schémas disponibles depuis l'emplacement de stockage en ligne et met à jour le répertoire de mise sous cache local. Vous devriez exécuter cette commande sauf si vous avez réalisé un `reset` et `initialize`.

Syntaxe

```
<exec> update [options]
```

Options
La commande **mise à jour** accepte les options suivantes :

`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemple

La commande suivante met à jour le cache local avec la liste des derniers schémas :

```
xmlschemamanager update
```

6.1.5.9 upgrade

Cette commande met à niveau toutes les schémas installés qui peuvent être mis à niveau à la dernière version *patchée* disponible. Vous pouvez identifier des schémas à mettre à niveau en exécutant la commande `list -u`.

Note : La commande **`mettre à niveau`** supprime une Police dépréciée si aucune version plus récente n'est disponible.

Syntaxe

```
<exec> upgrade [options]
```

Options

La commande **`mise à niveau`** accepte les options suivantes :

`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **`faux`**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **`faux`**.
`--help, --h`	Afficher l'aide pour la commande.

© 2016-2022 Altova GmbH

6.2 DTD

Un document DTD peut être édité dans le Mode Texte et le Mode Grille. Le mode par défaut peut être défini dans la section Types de fichier du dialogue Options.

Mode Texte

Dans le Mode Texte, le document est affiché avec une coloration syntaxique et doit être saisi. Ci-dessous, vous trouverez un échantillon d'un fragment DTD :

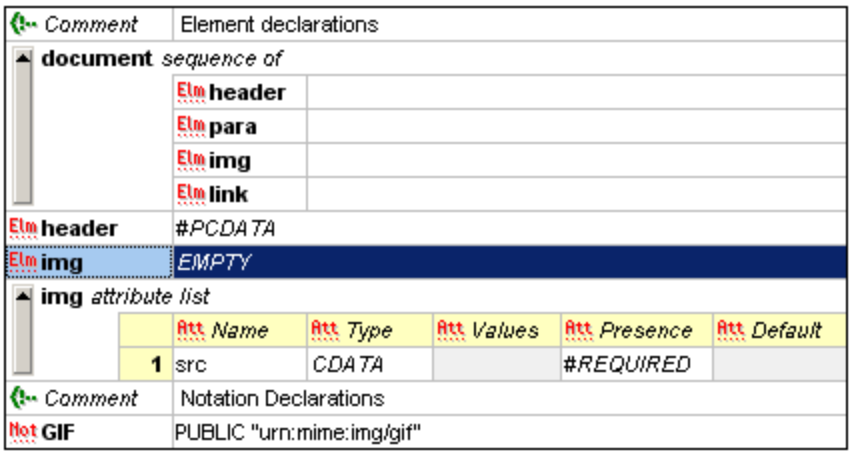
```
<!-- Element declarations -->
<!ELEMENT document (header, para, img, link)>
<!ELEMENT header (#PCDATA)>
<!ELEMENT img EMPTY>
  <!ATTLIST img
    src CDATA #REQUIRED
  >

<!-- Notation Declarations -->
<!NOTATION GIF PUBLIC "urn:mime:img/gif">
```

Le retrait est indiqué par des guides de retrait. Il est recommandé d'utiliser la touche du tabulateur. La taille du retrait peut être définie dans le dialogue Paramètres Mode Texte.

Mode Grille

Dans le Mode Grille, le document DTD est affiché en tant que table. La capture d'écran ci-dessous montre l'affichage du Mode Grille de la DTD ci-dessus.



Lorsque le curseur se trouve dans une ligne de table, ou si une ligne est sélectionnée, les commandes d'édition DTD dans le menu **XML** s'activent. Vous pouvez insérer, apposer et ajouter des nœuds enfants à la représentation graphique de la DTD. Les items DTD disponibles à un point particulier de la sélection sont activés dans le sous-menu respectif du menu **XML** (**Insérer**, **Apposer, Ajouter enfant**). Vous pouvez aussi convertir un item de DTD sélectionné en un autre item et déplacer l'item à gauche ou à droite afin de changer sa position dans la hiérarchie du document. Lorsqu'un nœud est sélectionné, les items de DTD disponibles sont aussi affichés en tant qu'items dans les assistants de saisie.

Fonctions DTD dans XMLSpy

XMLSpy propose les fonctions utiles suivantes :

- *Convertir une DTD en Schéma XML :* Grâce à la commande **DTD/Schéma | Convertir DTD/Schéma**, les DTD peuvent être converties en Schémas XML.
- *Générer un fichier XML échantillon depuis une DTD:* Grâce à la commande **DTD/Schéma | Générer un fichier XML/JSON échantillon**, un document XML basé sur la DTD active peut être généré.

© 2016-2022 Altova GmbH

6.3 Schémas XML

Les documents de Schéma XML peuvent être édités dans le Mode Texte, le Mode Grille et le Mode Schéma. Le mode par défaut dans lequel les documents de Schéma XML s'ouvrent peut être défini dans la section Types de fichier du dialogue Options. Vous pouvez sauter entre les modes pendant l'édition, en utilisant le mode qui vous paraît le plus utile à vos fins actuelles. Les documents de Schéma X sont généralement enregistrés avec l'extension `.xsd` ou `.xs`.

Éditer dans le Mode Texte

Dans le Mode Texte, un Schéma XML est édité en tant qu'un document XML ; les fonctions d'édition disponibles pour les documents XML sont aussi disponibles pour les Schémas XML. Comme pour tous les documents XML dont le schéma est identifiable et accessible, les assistants à la saisie du Mode Texte affichent les items disponibles pour un ajout à l'emplacement du curseur.

Éditer dans le Mode Grille

Dans le Mode Grille, un Schéma XML est édité en tant qu'un document XML ; les fonctions d'édition disponibles pour les documents XML sont aussi disponibles pour les Schémas XML. Lorsqu'un item est sélectionné dans le Mode Grille, les assistants à la saisie du Mode Grille affichent les items disponibles pour un ajout à l'emplacement du curseur.

Éditer dans le Mode Schéma

Le Mode Schéma est une interface graphique pour la conception de schémas. Lorsque vous créez/éditez le schéma dans le Mode Schéma, XMLSpy génère un document de texte correspondant derrière l'interface. Vous trouverez une description détaillée de la création et de l'édition des documents de Schéma XML pour le Mode Schéma dans la section Modes d'édition | Mode Schéma.

Site web Altova : Éditeur de schéma XML

Fonctions de Schéma XML dans XMLSpy

De plus, XMLSpy propose les fonctions utiles suivantes :

- *Convertir un Schéma XML en DTD :* Grâce à la commande **DTD/Schéma | Convertir Schéma en DTD**, les Schémas XML peuvent être convertis en DTD.
- *Générer un fichier XML échantillon depuis un Schéma XML :* Grâce à la commande **DTD/Schéma | Générer un fichier XML échantillon**, un document XML basé sur le Schéma XML actif peut être généré. Les valeurs d'échantillon peuvent aussi être spécifiées pour les éléments et les attributs dans le XML échantillon.
- Signatures XML pour les fichiers de Schéma XML (`.xsd`) dans le Mode Schéma, peuvent être créées en tant que fichiers de signature externe. Vous trouverez une description du maniement des signatures dans la section Signatures XML.

6.4 Sous-ensembles de Schéma

Il est possible de créer un ou plusieurs composants d'un Schéma XML en tant que fichier de schéma séparé, appelé sous-ensemble de schéma. L'avantage d'une utilisation de petits sous-ensembles de schéma pour composer des schémas importants (en utilisant Includes) est que les petits fichiers sont plus facilement gérables qu'un seul schéma complet.

Dans le Mode Schéma, un scénario de travail plausible, décrivant les différents aspects de la fonction des Sous-ensembles de Schéma pourrait être le suivant :

1. Créer un sous-ensemble de schéma contenant un ou plusieurs composants du schéma actif. Voir la description ci-dessous,
2. Créer des sous-ensembles de schéma supplémentaires le cas échéant.
3. Inclure le/s sous-ensemble/s de schéma récemment créés pour composer un schéma plus grand. Procéder à cette opération pour chaque sous-ensemble de schéma en ajoutant ou en insérant un composant Include dans la Fenêtre d'Aperçu de Schéma, et en sélectionnant le fichier de sous-ensemble de schéma récemment créé.
4. Supprimer tout composant qui était présent dans le schéma complet d'origine, mais qui est maintenant dupliqué en raison du sous-ensemble inclus.

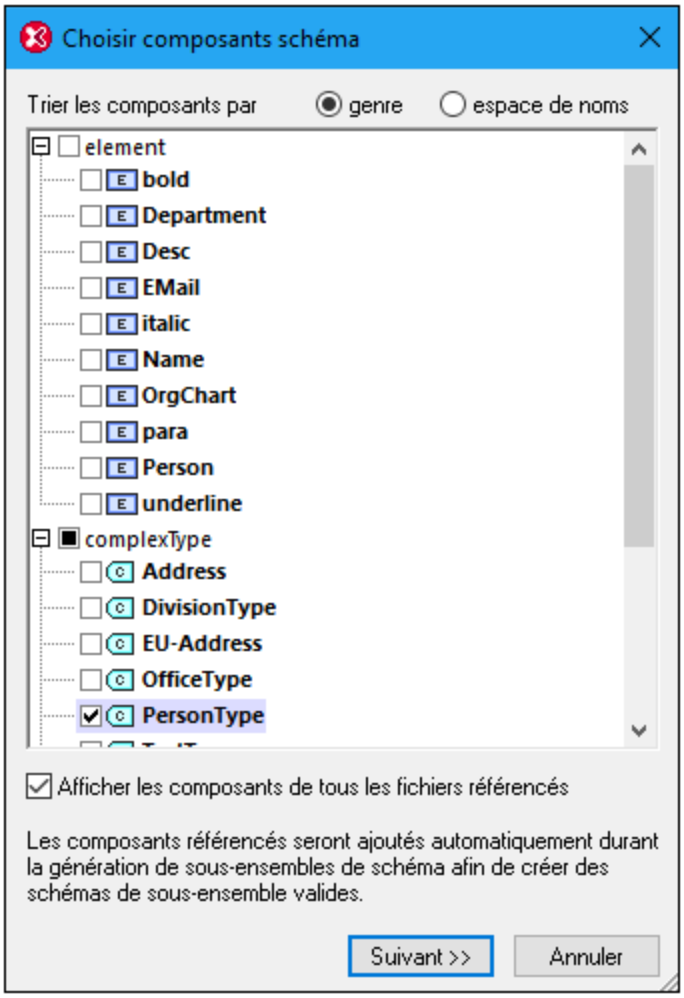
Vous pouvez donc procéder à l'opération inverse dans le Mode Schéma, c'est à dire, aplatir les sous-ensembles de schéma inclus de manière que : (i) les composants contenus dans les sous-ensembles de schéma soient ajoutés directement dans le schéma principal, et (ii) les sous-ensembles de schéma inclus soient supprimés du schéma principal. L'aplatissement d'un schéma est décrit ci-dessous.

Créer des sous-ensembles de schéma

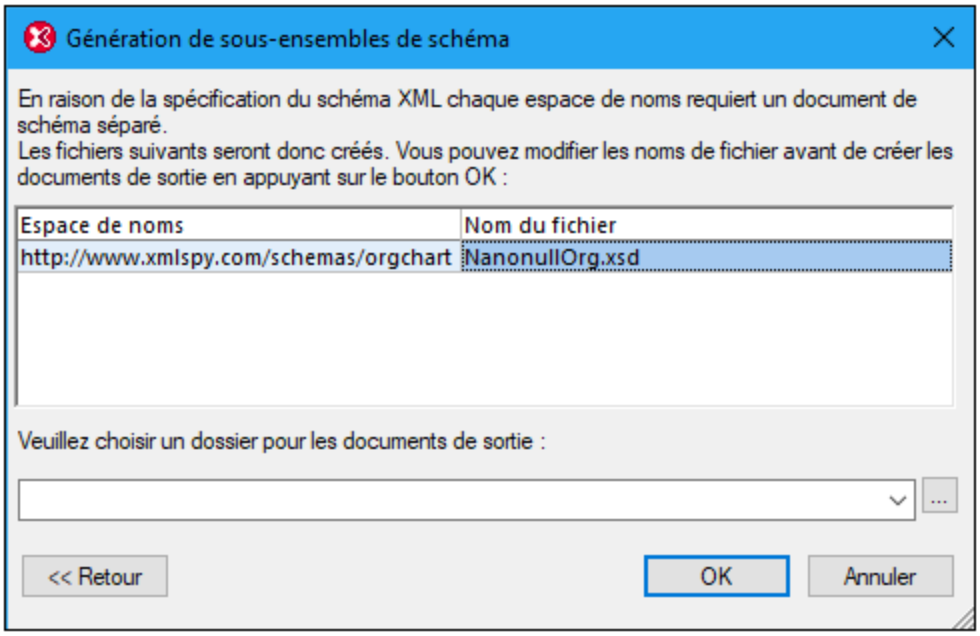
Pour créer un sous-ensemble de schéma, procéder comme suit :

1. Le Schéma XML requis étant actif dans le Mode Schéma, sélectionner la commande **Conception de Schéma | Créer sous-ensemble de schéma**. Le dialogue Choisir les composants de schéma s'ouvre (*capture d'écran ci-dessous*).
2. Dans le dialogue, cocher le composant ou les composants que vous souhaitez créer en tant que sous-ensemble unique de schéma, puis cliquer sur **Suivant**. (Veuillez noter qu'une case à cocher sous le volet permet aux composants de tous les fichiers référencés d'être recensés pour la sélection.)

© 2016-2022 Altova GmbH



3. Dans le dialogue Génération de sous-ensembles de schéma qui apparaît à présent (*capture d'écran ci-dessous*), saisir le nom que vous souhaiter donner au fichier du pack de sous-ensemble de schéma. Vous devrez aussi spécifier le dossier dans lequel vous souhaitez enregistrer les nouveaux fichiers de sous-ensemble de schéma. Un pack de sous-ensemble de schéma peut avoir plusieurs fichiers si un ou plusieurs composants créés sont un composant importé dans le schéma original. Un fichier de schéma séparé est créé pour chaque espace de noms dans le sous-ensemble de schéma. Le nom de fichier affiché dans le dialogue est, par défaut, le nom du fichier original. Mais puisque vous n'êtes pas autorisé à écraser les fichiers originaux, utiliser les nouveaux noms de fichier si vous souhaitez enregistrer les fichiers dans le même dossier que les les fichiers originaux.



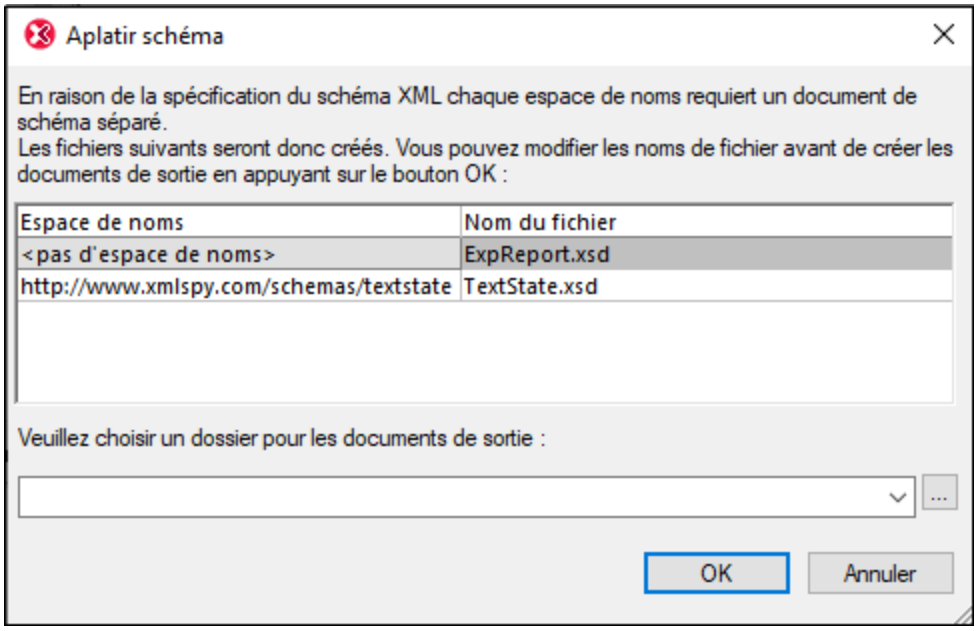
4. Cliquer sur **OK**, pour ouvrir le fichier de sous-ensemble de schéma avec l'espace de noms correspondant à celui du fichier actif dans le Mode Schéma. Tout autre fichier contenu dans le paquet est créé mais n'est pas ouvert dans le Mode Schéma.

Aplatir un schéma

L'aplatissement du schéma actif dans le Mode Schéma est le processus permettant : (i) d'ajouter les composants de tous les schémas inclus en tant que composants globaux du schéma actif, et (ii) de supprimer les schémas inclus.

Pour aplatir le schéma actif, sélectionner la commande **Conception de Schéma | Aplatir schéma**. Le dialogue Aplatir schéma s'ouvre (*capture d'écran ci-dessous*) qui contient les noms des fichiers séparés, un pour chaque espace de noms qui se trouvera dans le schéma aplati. Ces noms par défaut sont les mêmes que les noms de fichier d'origine. Mais puisque vous n'êtes pas autorisé à écraser les fichiers d'origine, les noms de fichier doivent être modifiés si vous souhaitez les enregistrer dans le même dossier que le fichier actif. Vous pouvez chercher un dossier dans lequel le schéma aplati et ses fichiers associés seront enregistrés.

© 2016-2022 Altova GmbH



Cliquer sur **OK**, pour ouvrir le fichier de schéma aplati dans le Mode Schéma.

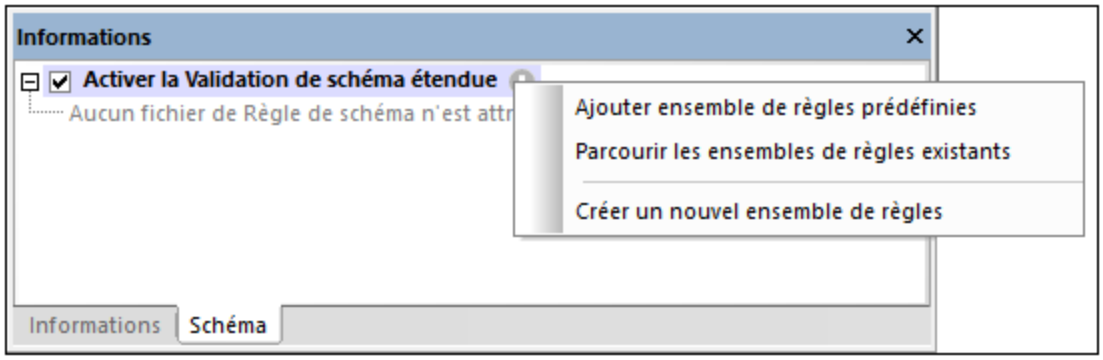
6.5 Règles de schéma

Un ensemble de Règles de schéma est un ensemble de règles à utiliser pour valider un Schéma XML. Par exemple, une règle peut spécifier que des noms d'attribut, tels que définis dans le Schéma XML, commencent avec une lettre minuscule, ou qu'un type complexe donné puisse uniquement être étendu depuis un type spécifique.

Les ensembles de règles de schéma sont enregistrés dans un fichier d'Ensembles de règles, qui est un fichier XML (`.xml`). XMLSpy contient un Éditeur de Règles de schéma dans lequel vous pouvez éditer des règles de schéma de manière graphique. Dans XMLSpy, un ou plusieurs fichiers d'Ensembles de règles peut être attribué à un Schéma XML. Le Schéma XML est validé dans le Mode Schéma par rapport à l'Ensemble de règles attribué en sélectionnant la commande **XML | Valider (F8)**.

6.5.1 Gérer les Ensembles de règles

Un ou plusieurs fichiers d'Ensemble de règles de schéma (fichiers `.xml`) peut être attribué au Schéma XML actif (fichier `.xsd`). Cela s'effectue par le biais de l'onglet Schéma de la fenêtre Info (*capture d'écran ci-dessous*).

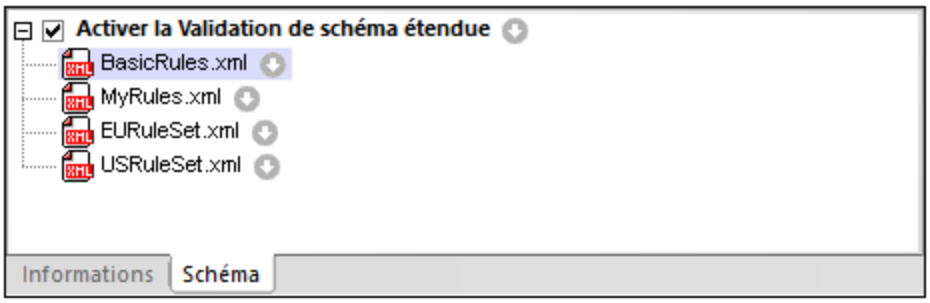


Ajouter des Ensembles de règles pour une validation étendue

Pour ajouter un fichier d'Ensemble de règles de schéma, cliquer sur le bouton du menu contextuel . Un menu apparaît (*voir capture d'écran ci-dessus*) dans lequel vous pouvez sélectionner comment vous souhaitez ajouter les fichiers d'Ensembles de règles de schéma dans le schéma XML. Les options suivantes sont disponibles :

- *Ajouter un Ensemble de règles prédéfinies :* vous pouvez effectuer votre choix depuis une liste d'Ensembles de règles prédéfinies qui ont été fournies avec XMLSpy. Ces fichiers d'Ensembles de règles ont été enregistrés dans le dossier `Validation étendue de schéma` situé dans le dossier d'application XMLSpy. Tout fichier d'Ensembles de règles ajouté à ce dossier sera affiché dans le dialogue Ensemble de règles prédéfinies et sera disponible pour tout ajout.
- *Parcourir les ensembles de règles existants :* vous pouvez chercher un fichier d'Ensembles de règles non-prédéfini.
- *Créer un nouvel ensemble de règles :* affiche l'Éditeur de règles de schéma dans lequel vous pouvez éditer les Règles de schéma dans un fichier d'Ensemble de règles. Vous trouverez une description du travail dans l'Éditeur de règles de schéma dans la section, Définir un Ensemble de règles. Après avoir

enregistré un fichier d'Ensembles de règles créé par le biais de cette commande, celui-ci sera ajouté à la liste du Schéma XML actif (*voir capture d'écran ci-dessous*).



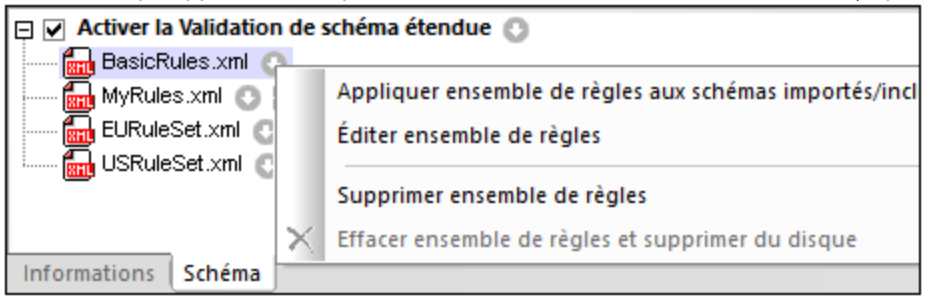
Vous pouvez ajouter autant d'Ensembles de règles que vous le souhaitez (*voir capture d'écran ci-dessus*). Si plus d'une règle de schéma a été attribuée à un schéma XML, les règles de tous les Ensembles de règles de schéma sont utilisées lorsque le Schéma XML est validé dans le Mode Schéma (**XML | Valider**).

Activer et désactiver la validation de schéma étendue

La validation de schéma étendue peut être activée ou désactivée en cliquant sur la case Activer la validation de schéma étendue.

Éditer et supprimer des Ensembles de règles

Les Ensembles de règles individuels attribués à un Schéma XML peuvent être gérés par le biais du menu contextuel qui apparaît en cliquant sur le bouton du menu contextuel  (*capture d'écran ci-dessous*).



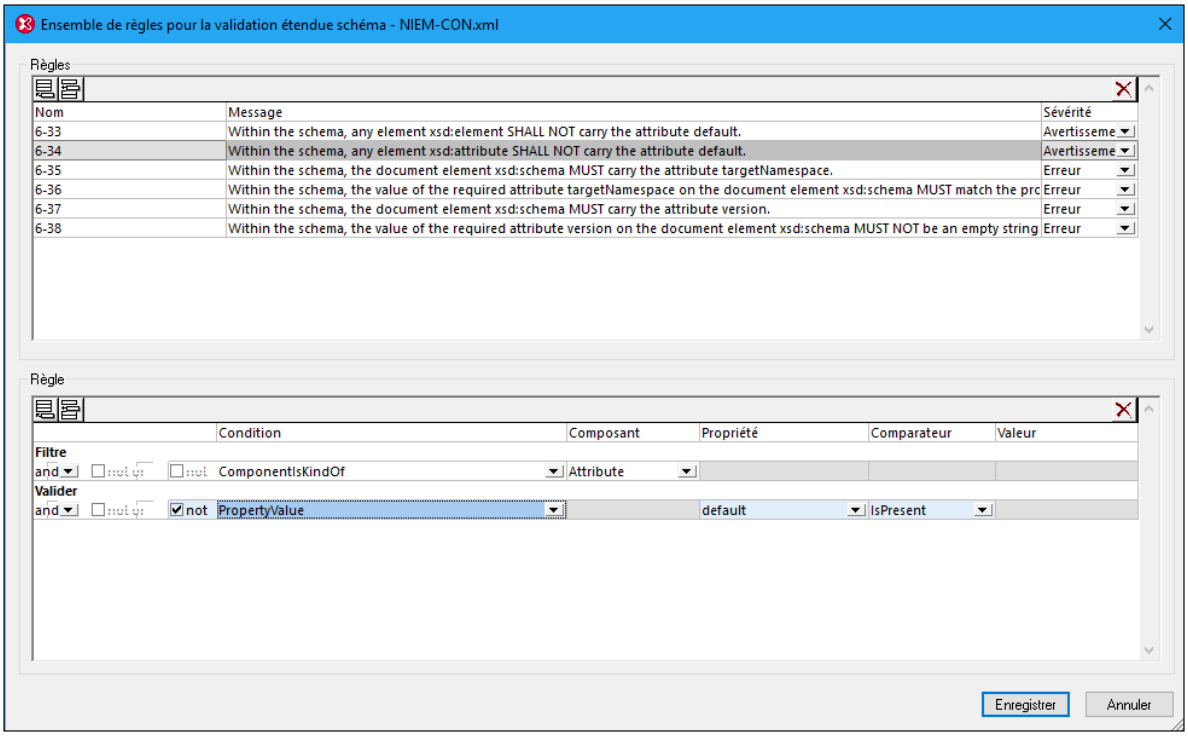
Les options suivantes sont disponibles :

- *Appliquer l'Ensemble de règles aux schémas importés et inclus :* si un Ensemble de règles est appliqué, les règles contenues seront utilisées pour tous les schémas que le schéma principal importe ou inclut.
- *Éditer règle :* ouvre l'Ensemble de règles de schéma dans l'éditeur de règles de Schéma.
- *Supprimer Ensemble de règles :* supprime l'Ensemble de règles de la liste des Ensembles de règles ajoutés.
- *Supprimer Ensemble de règles et supprimer du disque :* cette commande est activée pour tous les Ensembles de règles non-prédéfinis. Non seulement cette commande retire l'Ensemble de règles de la liste des Ensembles de règles ajoutés, mais elle supprime aussi l'Ensemble de règles.

6.5.2 Définir un Ensemble de règles

Un Ensemble de règles de schéma peut être ouvert pour être édité dans l'Éditeur de règles de schéma (*capture d'écran ci-dessous*). Vous pouvez ensuite créer, éditer et supprimer des ensembles de règles dans ce fichier d'Ensemble de règles de schéma. Pour ouvrir un Ensemble de règles dans l'Éditeur de règles de schéma, procéder comme suit :

1. Sélectionner l'Ensemble de règles dans la liste de l'Ensemble de règles dans la fenêtre Info.
2. Cliquer sur le bouton de menu contextuel  de cet Ensemble de règles.
3. Dans le menu contextuel qui apparaît, sélectionner Éditer règles.



Le dialogue Éditeur de règles de schéma comporte deux volets :

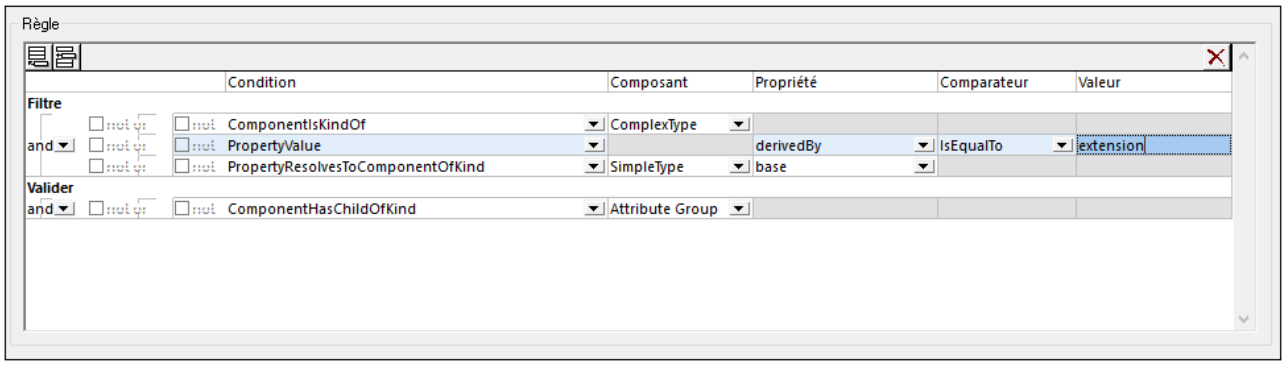
- Un volet Règles (dans la partie supérieure de l'Éditeur), dans lequel vous pouvez ajouter et supprimer des règles. Une ligne vide pour une règle peut être ajoutée ou insérée en cliquant sur le bouton respectif (**Ajouter** ou **Insérer**) situé dans la partie supérieure gauche du volet. Une règle peut être supprimée en la sélectionnant et en cliquant sur le bouton **Supprimer** dans la partie supérieure droite du volet. Chaque règle dans ce volet a un nom, un texte descriptif et un niveau de sévérité (si la règle est contredite, la validation sera définie pour retourner une erreur ou un avertissement).
- Un volet Règles (dans la partie inférieure de l'Éditeur). Ce volet affiche les détails de la règle qui a été sélectionnée dans le volet Règles situé au-dessus, et permet l'édition des détails de la règle. Pour plus de détails concernant la définition des règles, voir la section Définir une règle ci-dessous.

Une fois que les règles dans un fichier d'Ensemble de règles ont été éditées, cliquer sur **Enregistrer** pour enregistrer les règles dans le Fichier d'Ensemble de règles.

© 2016-2022 Altova GmbH

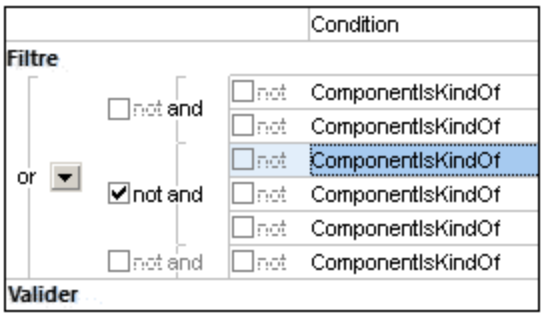
Définir une règle

Pour définir ou éditer une règle, sélectionner la règle depuis la liste dans le volet Règles supérieur. La définition de la règle sera affichées dans le volet Règles et peut être éditée. La capture d'écran ci-dessous affiche une règle qui peut être définie comme suit : *Si un type complexe est une extension d'un type simple, il doit avoir un type enfant AttributeGroup.*



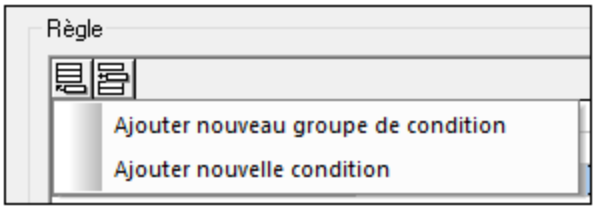
L'ensemble de condition Valider et l'ensemble de condition Filtre :

- Chaque règle dispose d'un ensemble de conditions Valider et d'un ensemble de conditions Filtre (*voir première colonne dans la capture d'écran ci-dessus*).
- L'ensemble des conditions Filtre doit finalement évaluer à Vrai pour que la condition Valider soit évaluée.
- Chaque ensemble de conditions (Valider ou Filtre) consiste en un ou plusieurs Groupes de Condition. Chaque Groupe de Condition contient une ou plusieurs conditions. Dans la capture d'écran ci-dessus, l'ensemble Valider contient un Groupe de Condition d'une condition, et l'ensemble Filtre contient trois Groupes de Condition, contenant chacun une condition. Dans la capture d'écran ci-dessous, l'ensemble Filtre contient trois Groupes de Condition : le premier contient deux conditions, le second contient trois conditions et le troisième contient une condition.



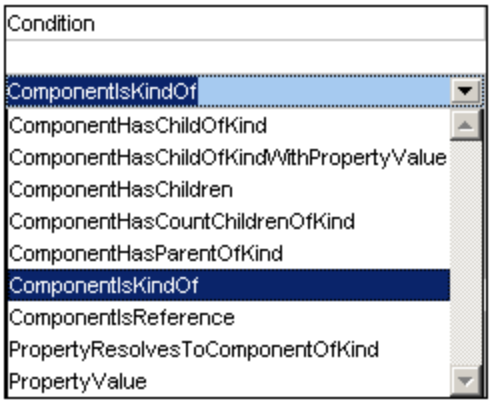
- Chaque condition individuelle peut être niée en cochant sa sa case *Not* (située à gauche de la condition).
- Dans un Groupe de Condition, les connecteurs logiques et ou ou indiquent, respectivement, si toutes les conditions dans le groupe ou si une condition dans le Groupe de Condition doivent évaluer à true pour que tout le Groupe de condition évalue à true. Dans la GUI, ces opérateurs logiques se trouvent à l'intérieur des deux colonnes des opérateurs logiques.
- Chaque Groupe de Condition peut être nié en cochant la case *Not* (située à gauche de l'opérateur logique du groupe de condition).

- Le connecteur logique externe "et" ou "ou" indique, respectivement, si tous les Groupes de condition dans l'ensemble (Valider ou Filtre) ou si un Groupe de condition doit évaluer à true pour que tout l'ensemble (Valider ou Filtre) évalue à true.
- Les connecteurs logiques peuvent être changés en choisissant l'option appropriée dans la liste de choix pour le connecteur logique externe (le connecteur de Groupe de Condition). La valeur des connecteurs logiques externes (les connecteurs pour les conditions dans un Groupe de condition) sont tous placés sur la valeur opposite de celle du connecteur logique externe.
- Un groupe de condition ou une Condition peut être ajoutée ou insérée relativement à la condition sélectionnée. Pour ce faire, sélectionner une condition, puis cliquez sur le bouton **Ajouter** ou **Insérer** (situé en haut et à gauche du volet) et puis en sélectionnant l'item requis (Groupe de condition ou Condition) à partir du menu (*voir capture d'écran ci-dessous*).



Types de condition
Une condition peut appartenir à un des trois groupes (*aussi voir capture d'écran ci-dessous*):

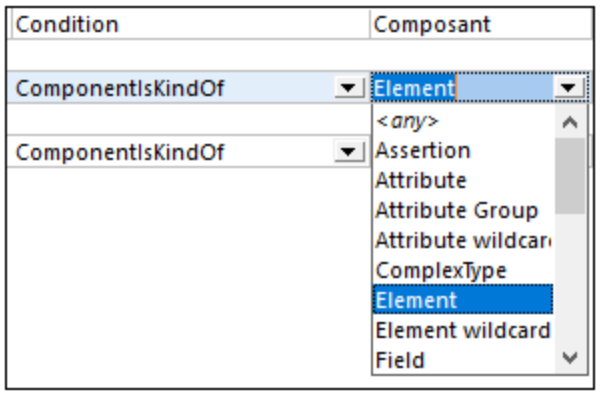
- Type de composant (dans la liste déroulante les types commençant avec *Composant ; voir capture d'écran ci-dessous*)
- Type de propriété (*Valeur de propriété*)
- Une combinaison de types de composant et de propriété (types comportant *Propriété* et *Composant* dans leurs noms)



Le type d'une condition est choisi depuis la liste déroulante dans la colonne *Condition* de la condition (*capture d'écran ci-dessus*). Chacun de ces trois groupes de conditions est décrit ci-dessous.

Conditions du type de composant
En ce qui concerne les conditions de type Composant (types commençant avec *Composant*), le composant doit être spécifié par la suite dans la colonne *Composant* (*voir capture d'écran ci-dessous*). Le composant est sélectionné depuis la liste déroulante dans le champ Composant d'une sous-condition. Puisqu'aucun autre champ (Propriété, Valeur de comparateur) ne doit être défini pour des conditions du type Composant, tous les autres champs sont grisés.

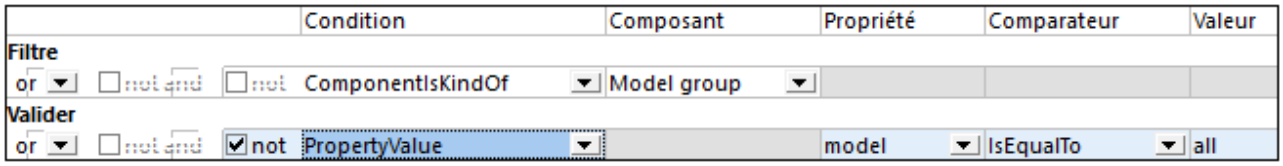
© 2016-2022 Altova GmbH



Dans la capture d'écran ci-dessus, la condition Filtre spécifie que la règle concerne les composants de type *Élément*. Si la condition Valider spécifie que le composant doit avoir un enfant de type *Notation*, alors la règle complète peut être déclarée comme : un composant *Élément* doit avoir un enfant de type *Notation*. Si l'option `NOT` de la condition Valider était cochée, la règle serait déclarée comme : un composant *Élément* ne doit pas avoir un enfant de type *Notation*.

Conditions de type Propriété

La condition du type Propriété est *Valeur de propriété* (*voir capture d'écran ci-dessous*). Ce type de condition spécifie la nature d'une propriété. Il nécessite dont des entrées dans les colonnes *Propriété* et *Comparateur*, et, en option, une entrée dans la colonne *Valeur*. Aucune entrée n'est nécessaire dans la colonne *Composant*, qui est donc grisée. Les propriétés recensées dans les listes déroulantes de la colonne *Propriété* incluent non seulement les attributs XML (comme `default` et `maxOccurs`) mais aussi les propriété logiques des composants (comme `derivedBy`).



La capture d'écran ci-dessus montre une règle dans laquelle la propriété *Modèle* a une valeur égale à *Tout* et est niée (avec la case à cocher *Not*). En conjonction avec le filtre dans le composants *Groupe de modèle*, cette règle déclare simplement qu'un schéma ne doit pas contenir d'élément `xsd:all`.

Note : Veuillez noter les points suivants :

- Lorsque vous utilisez le comparateur `IsQNameEqualTo`, la valeur correspondante doit être écrite sous la forme : `{URI}localName`. Par exemple, une valeur pourrait être : `{http://www.w3.org/2001/XMLSchema}NOTATION`.
- La propriété `default` peut être présente et vide (`<element name default=""/>`) ou elle peut être absente (`<element name/>`).

Conditions qui combinent les types Composant et Propriété

Les conditions qui sont une combinaison des types Composant et Propriété sont :

- `ComponentHasChildOfKindWithPropertyValue`: spécifie le type de composant d'un élément enfant et la valeur de propriété.

- `PropertyResolvesToComponentOfKind`: une propriété est spécifiée qui résout à un type de composant. Les colonnes *Comparateur* et *Valeur* sont vides.

Nier une condition
Une condition est niée en cochant la case *Not* située juste à sa gauche (les cases à cocher *Not* internes). Un groupe de condition est nié en cliquant sur la case *Not* située à gauche du connecteur logique pour les conditions dans ce groupe de condition.

© 2016-2022 Altova GmbH

6.6 Catalogues dans XMLSpy

<% XMLSPY%> prend en charge un sous-ensemble des catalogues XML OASIS. Le mécanisme du catalogue permet à XMLSpy d'extraire des schémas communément utilisés (y compris des feuilles de style ou autres fichiers) de répertoires utilisateurs locaux. Cela augmente la vitesse de traitement générale, permet aux utilisateurs de travailler hors ligne (c'est-à-dire sans connexion à un réseau) et améliore la portabilité des documents (parce que les URI ne devraient être modifiés uniquement dans les fichiers catalogue.)

Le mécanisme du catalogue dans XMLSpy fonctionne comme suit dans la présente section :

- Comment fonctionnent les catalogues
- Structure du catalogue dans XMLSpy
- Personnaliser vos catalogues
- Variables d'Environnement

Pour plus d'informations sur les catalogues, voir la spécification de catalogues XML.

6.6.1 Comment fonctionnent les catalogues

Les catalogues peuvent être utilisés pour rediriger les Schémas DTD et XML. Alors que le concept derrière les mécanismes dans les deux cas est le même, les détails sont différents et expliqués ci-dessous.

DTD

Les catalogues sont communément utilisés pour rediriger un appel vers un DTD ou un URI local. Pour ce faire, des identifiants publics ou système sont mappés dans le fichier catalogue vers l'URI local requis. Donc, si la déclaration **DOCTYPE** dans un fichier XML est lue, son identifiant public ou système localise la ressource locale requise par le biais du mappage du fichier catalogue.

Pour les schémas populaires, l'identifiant **PUBLIC** est normalement prédéfini, requérant uniquement que l'URI dans le fichier catalogue mappe l'identifiant **PUBLIC** à la copie locale correcte. Lorsque le document XML est parsé, l'identifiant **PUBLIC** qui le compose est lu. Si cet identifiant est trouvé dans un fichier catalogue, l'URL correspondant dans le catalogue fichier sera consulté et le schéma sera lu depuis cet emplacement. Donc, par exemple, si le fichier SVG est ouvert dans XMLSpy :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="20" height="20" xml:space="preserve">
  <g style="fill:red; stroke:#000000">
     <rect x="0" y="0" width="15" height="15"/>
     <rect x="5" y="5" width="15" height="15"/>
  </g>
</svg>
```

L'identifiant **PUBLIC** de ce fichier SVG file est recherché dans le catalogue. Disons que le fichier catalogue contient l'entrée suivante :

```
catalog>
  ...
    <public publicId="-//W3C//DTD SVG 1.1//EN" uri="schemas/svg/svg11.dtd"/>
  ...
</catalog>
```

Dans ce cas, il y a une correspondance pour l'identifiant **PUBLIC**. En conséquence, le lookup pour SVG DTD est redirigé vers l'URL **schemas/svg/svg11.dtd** (qui est associé au fichier catalogue). Il s'agit d'un fichier local qui sera utilisé en tant que DTD pour le fichier SVG. S'il n'y a pas de mappage pour l'ID **Public** dans le catalogue, l'URL dans le document XML sera utilisé (dans l'exemple du fichier SVG ci-dessus, l'URL Internet est la suivante : **http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd**).

Schémas XML

Dans XMLSpy, vous pouvez utiliser des catalogues avec des **Schémas XML**. Dans le fichier d'instance XML, la référence au schéma apparaîtra dans l'attribut **xsi:schemaLocation** de l'élément de premier niveau du document XML. Par exemple,

```
xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"
```

La valeur de l'attribut **xsi:schemaLocation** a deux parties : une partie d'espace de noms (vert ci-dessus) et une partie URI (en surbrillance). La partie d'espace de noms est utilisée dans le catalogue pour effectuer le mappage vers la ressource alternative. Par exemple, la saisie catalogue suivante redirige la référence du schéma ci-dessus vers un schéma à un emplacement alternatif.

```
<uri name="http://www.xmlspy.com/schemas/orgchart" uri="C:\MySchemas\OrgChart.xsd"/>
```

Normalement, la partie URI de la valeur de l'attribut **xsi:schemaLocation** est le chemin vers l'emplacement actuel du schéma. Toutefois, si le schéma est référencé par le biais du catalogue, la partie URI doit pointer vers le schéma XML actuel mais doit exister pour que la validité lexicale de l'attribut **xsi:schemaLocation** soit maintenu. Une valeur `foo`, par exemple, suffirait à la partie URI de la valeur de l'attribut pour qu'elle soit valide.

6.6.2 Structure du catalogue dans XMLSpy

Lorsque XMLSpy est lancé, il charge un fichier désigné **RootCatalog.xml** (*la structure est affichée dans la liste ci-dessous*), qui contient une liste des fichiers catalogue qui seront consultés. Vous pouvez modifier ce fichier et saisir autant de fichiers catalogue que vous souhaitez consulter, chacun est référencé dans un élément **nextCatalog**. Ces fichiers catalogue sont consultés et les URI contenus sont résolus conformément à leurs mappages.

Liste du RootCatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
   xmlns:spy="http://www.altova.com/catalog_ext"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:entity:xmlns:xml:catalog Catalog.xsd">
  <nextCatalog catalog="%PersonalFolder%/Altova/%AppAndVersionName%/CustomCatalog.xml"/>
  <!-- Inclut tous les catalogues sous les schémas communs du premier répertoire dans
l'arborescence -->
```


© 2016-2022 Altova GmbH

```
  <nextCatalog spy:recurseFrom="%CommonSchemasFolder%" catalog="catalog.xml"
spy:depth="1"/>
  <nextCatalog spy:recurseFrom="%ApplicationWritableDataFolder%/pkgs/.cache"
catalog="remapping.xml" spy:depth="0"/>
  <nextCatalog catalog="CoreCatalog.xml"/>
</catalog>
```

La liste des références ci-dessus renvoie à un catalogue personnalisé (désigné **CustomCatalog.xml**) et à un ensemble de catalogues qui localisent des schémas communément utilisés (tels que les Schémas W3C XML et le schéma SVG).

- **CustomCatalog.xml** est situé dans le sous-dossier De votre dossier personnel (situé par le biais de la variable **%PersonalFolder%**). CustomCatalog.xml est fichier squelette dans lequel vous pouvez créer vos propres mappages. Vous pouvez ajouter des mappages au **CustomCatalog.xml** pour chaque schéma dont vous avez besoin qui n'est pas adressé par les fichiers catalogue dans le dossier des Schémas communs. Pour ce faire, utilisez les éléments pris en charge par le mécanisme de catalogue OASIS (*voir prochaine section*).
- Le dossier des Schémas communs (localisé par la variable **%CommonSchemasFolder%**) contient un ensemble de schémas communément utilisés. À l'intérieur de chaque dossier de schéma, il y a un fichier **catalog.xml** qui mappe des identifiants publics et/ou de système aux URI qui dirigent vers des copies enregistrées localement des schémas respectifs.
- Les schémas liés à des XBRL et différentes taxonomies XBRL sont grands et installés localement sur demande avec l'aide de Altova Gestionnaire de taxonomie. Des schémas et taxonomies individuels sont mappés dans le catalogue **remapping.xml**, qui est situé dans le sous-dossier **pkgs/.cache** du dossier des données du Programme (localisé par le biais de la variable **%ApplicationWritableDatFolder%**). **Veuillez ne pas éditer ce fichier ;** la plus petite erreur pourrait sensiblement compromettre un large ensemble de références.
- **CoreCatalog.xml** est situé dans le dossier d'application in the XMLSpy et est utilisé pour localiser les schémas et les feuilles de style utilisés par des processus spécifiques de XMLSpy, tels que StyleVision Power Stylesheets qui sont des feuilles de style utilisées pour générer le mode Authentique des documents XML de Altova.

Variables d'emplacement
Les variables utilisées dans **RootCatalog.xml** (*liste ci-dessus*) ont les valeurs suivantes :

%PersonalFolder%	Dossier personnel de l'utilisateur actuel, par exemple C:\Users\<name>\Documents
%CommonSchemasFolder%	C:\ProgramData\Altova\Common2023\Schemas
%ApplicationWritableDataFolder%	C:\ProgramData\Altova

Emplacement des fichiers catalogue et des schémas
Veuillez noter l'emplacement des différents fichiers catalogue.

- **RootCatalog.xml** et **CoreCatalog.xml** sont dans le dossier d'application XMLSpy.
- **CustomCatalog.xml** est situé dans votre dossier **MyDocuments\Altova\XMLSpy**.
- Les fichiers **catalog.xml** sont chacun dans un dossier de schéma spécifique, ces dossiers de schéma étant dans le dossier des Schémas communs.

6.6.3 Personnaliser vos catalogues

Lorsque vous créez des entrées dans **CustomCatalog.xml** (ou tout autre fichier catalogue qui doit être lu par XMLSpy), utilisez uniquement les éléments suivants de la spécification de catalogue OASIS. Chacun des éléments ci-dessous est répertorié avec une explication de leurs valeurs attribut. Pour plus d'informations sur les catalogues, voir la spécification des catalogues XML. Notez que chaque élément peut prendre l'attribut **xml:base**, qui est utilisé pour spécifier la base URI de cet élément.

- `<public publicId="PublicID of Resource" uri="URL of local file"/>`
- `<system systemId="SystemID of Resource" uri="URL of local file"/>`
- `<uri name="filename" uri="URL of file identified by filename"/>`
- `<rewriteURI uriStartString="StartString of URI to rewrite" rewritePrefix="String to replace StartString"/>`
- `<rewriteSystem systemIdStartString="StartString of SystemID" rewritePrefix="Replacement string to locate resource locally"/>`

Veuillez noter les points suivants :

- Dans le cas où il n'y a pas d'identifiant public, comme pour toutes les feuilles de style, l'identifiant système peut être directement mappé avec un URL par le biais de l'élément **système**.
- Un URI peut être mappé avec un autre URI en utilisant l'élément `uri`.
- Les éléments `rewriteURI` et `rewriteSystem` permettent la réécriture respectivement de la partie initiale d'un URI ou d'un identifiant système. Ceci permet de lancer un chemin de fichier à remplacer et, par conséquent, permet de cibler un autre répertoire. Pour plus d'informations sur les éléments, voir la spécification des catalogues XML.

À partir de la version 2014, XMLSpy adhère étroitement à la spécification spécification des catalogues XML (OASIS Standard V1.1, 7 octobre 2005). Cette spécification sépare strictement les look-up d'identifiants externes (ceux avec une ID Publique ou une ID Système) des look-up URI (les URI qui sont pas des ID Publiques ou des ID Système). Les URI d'espace de noms doivent donc être considérés comme étant des URI simples —et pas des ID Publiques ou des ID Système—et doivent être utilisés en tant que look-up URI plutôt que des look-up d'identifiants externes. Dans les versions XMLSpy antérieures à la version 2014, les URI d'espace de noms ont été traduits par les mappages **<public>**. À partir de la version 2014, les mappages **<uri>** doivent être utilisés.

Avant v2014 : `<public publicID="http://www.MyMapping.com/ref" uri="file:///C:/MyDocs/Catalog/test.xsd"/>`

à partir de V-2014 : `<uri name="http://www.MyMapping.com/ref" uri="file:///C:/MyDocs/Catalog/test.xsd"/>`

Comment XMLSpy trouve un schéma référencé

Un schéma est référencé dans un document XML par le biais de l'attribut **xsi:scemaLocation** (*voir ci-dessous*). La valeur de l'attribut **xsi:schemaLocation** a deux parties : une partie d'espace de noms (vert) et une partie URI (en surbrillance).

```
xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"
```

Ci-dessous, vous trouverez les étapes à suivre pour trouver un schéma référencé, suivies de manière

© 2016-2022 Altova GmbH

séquentielle par XMLSpy. Le schéma est chargé lors de la première étape réussie.

1. Consultez le catalogue pour la partie URI de la valeur **xsi:schemaLocation**. Si un mappage est trouvé, y compris dans les mappages **rewriteURI**, utilisez l'URI qui en résulte pour charger le schéma.
2. Consultez le catalogue pour la partie espace de noms de la valeur **xsi:schemaLocation**. Si un mappage est trouvé, y compris dans les mappages **rewriteURI**, utilisez l'URI qui en résulte pour charger le schéma.
3. Utilisez la partie de l'URI de la valeur **xsi:schemaLocation** pour charger le schéma.

Extensions de fichier et édition intelligente conformément au schéma

Par le biais des fichiers catalogue, vous pouvez également spécifier si les documents dotés d'une extension de fichier particulière devraient disposer de fonctions d'édition intelligente de XMLSpyappliquées conformément aux règles du schéma que vous spécifiez. Par exemple, si vous créez une extension de fichier personnalisé `.myhtml` pour des fichiers (HTML) qui doivent être valides selon HTML DTD, vous pouvez activer l'édition intelligente pour les fichiers ayant ces extensions en ajoutant l'élément de texte suivant au **CustomCatalog.xml** en tant qu'enfant de l'élément **<catalog>** .

```
catalog>
  ...
   <spy:fileExtHelper ext="myhtml" uri="schemas/xhtml/xhtml1-transitional.dtd"/>
  ...
</catalog>
```

Ceci permettrait une édition intelligente (remplissage automatique, assistants de saisie, etc.) de fichiers **.myhtml** dans XMLSpy conformément à XHTML 1.0 Transitional DTD. Reportez-vous au fichier **catalog.xml** dans le dossier **%AltovaCommonSchemasFolder%\Schemas\xhtml**, qui contient des entrées similaires.

Spécifications de schéma XML

L'information de spécification de schéma XML est prédéfinie dans XMLSpy et la validité des documents de schéma XML (`.xsd`) est comparée à l'information interne. Pour cela, dans un document de schéma XML, il ne devrait pas y avoir de références faites à n'importe quel schéma qui définit la spécification de schéma XML.

Le fichier **catalog.xml** dans le dossier `%AltovaCommonSchemasFolder%\Schemas\schema` contient des références aux DTD qui implémentent des spécifications de schéma XML antérieures. Vous ne devriez pas valider vos documents de schéma XML par rapport à ces schémas. Les fichiers référencés sont inclus uniquement pour donner à XMLSpy des informations sur les assistants de saisie à des fins d'édition si vous vouliez créer des documents conformément à ces recommandations antérieures.

6.6.4 Variables d'Environnement

Les variables d'environnement shell peuvent être utilisées dans l'élément `nextCatalog` pour spécifier le chemin menant aux différents emplacements de système (*voir RootCatalog.xml liste ci-dessus*). Les variables d'environnement shell suivantes sont prises en charge :

`%PersonalFolder%`	Chemin complet vers le dossier Personnel de l'utilisateur actuel, par exemple `C:\Users\<name>\Documents`

%CommonSchemasFolder%	C:\ProgramData\Altova\Common2023\Schemas
%ApplicationWritableDataFolder%	C:\ProgramData\Altova
%AltovaCommonFolder%	C:\Program Files\Altova\Common2023
%DesktopFolder%	Chemin complet vers le dossier Bureau de l'utilisateur actuel.
%ProgramMenuFolder%	Chemin complet vers le dossier Menu Programme pour l'utilisateur actuel.
%StartMenuFolder%	Chemin complet vers le dossier Démarrage pour l'utilisateur actuel.
%StartUpFolder%	Chemin complet vers le dossier Démarrage pour l'utilisateur actuel.
%TemplateFolder%	Chemin complet vers le dossier Modèle pour l'utilisateur actuel.
%AdminToolsFolder%	Chemin complet vers le répertoire de système de fichier qui stocke des outils administratifs pour l'utilisateur actuel.
%AppDataFolder%	Chemin complet vers le dossier Données d'Application pour l'utilisateur actuel.
%CommonAppDataFolder%	Chemin complet vers le répertoire de fichier contenant les données d'application pour tous les utilisateurs.
%FavoritesFolder%	Chemin complet du dossier Favoris pour l'utilisateur actuel.
%PersonalFolder%	Chemin complet vers le dossier Personnel pour l'utilisateur actuel.
%SendToFolder%	Chemin complet vers le dossier EnvoyerÀ pour l'utilisateur actuel.
%FontsFolder%	Chemin complet vers le dossier Polices Système.
%ProgramFilesFolder%	Chemin complet vers le dossier Fichiers Programme pour l'utilisateur actuel.
%CommonFilesFolder%	Chemin complet vers le dossier Fichiers Communs pour l'utilisateur actuel.
%WindowsFolder%	Chemin complet vers le dossier Windows pour l'utilisateur actuel.
%SystemFolder%	Chemin complet vers le dossier Système pour l'utilisateur actuel.
%LocalAppDataFolder%	Chemin complet vers le répertoire de fichier système qui sert en tant qu'archivage de données pour les applications locales (nonroaming).
%MyPicturesFolder%	Chemin complet vers le dossier MesPhotos.

© 2016-2022 Altova GmbH

6.7 Travailler avec SchemaAgent

XMLSpy peut être configuré pour fonctionner avec la technologie SchemaAgent d'Altova.

Technologie SchemaAgent
La technologie SchemaAgent permet aux utilisateurs d'éditer des relations entre les schémas divers. Elle est composée de :

- un SchemaAgent Server, qui détient et sert des informations concernant les relations entre les schémas dans un ou plusieurs chemins de recherche (dossiers dans le réseau) spécifiés par vos soins.
- un client SchemaAgent, le produit SchemaAgent d'Altova, qui utilise les informations de schéma depuis le serveur SchemaAgent auquel il est connecté (i) pour construire des relations entre ces schémas ; et (ii) pour gérer ces schémas (renommer, déplacer, supprimer des schémas, etc).

Deux types de serveurs SchemaAgent sont disponibles :

- Altova SchemaAgent Server, qui peut être installé sur et accédé depuis un réseau et
- Altova SchemaAgent, qui est le produit client SchemaAgent. Il comprend une version de serveur plus légère, appelée LocalServer, qui peut uniquement être utilisée sur la même machine que sur laquelle SchemaAgent est installé.

XMLSpy utilise la technologie SchemaAgent pour éditer directement les schémas dans le Mode Schéma en utilisant les informations à propos d'autres schémas, obtenues à partir d'un serveur SchemaAgent. Dans cette configuration, XMLSpy est connecté à un serveur SchemaAgent, et, en interaction avec SchemaAgent Client, il envoie des requêtes à SchemaAgent Server. Lorsque XMLSpy a été configuré pour fonctionner avec SchemaAgent, les Assistants à la saisie dans le Mode Schema non seulement recensent les composants provenant du schéma actif actuellement dans le Mode Schéma, mais recensent également des composants provenant d'autres schémas dans les chemins de recherche du serveur SchemaAgent auquel il est connecté. Cela vous permet un accès direct à ces composants. Vous pouvez visualiser le modèle de contenu d'un composant appartenant à un autre schéma dans le Mode Schéma, et réutiliser ce composant avec ou sans modifications. Vous pouvez aussi créer des relations entre les schémas, vous permettant ainsi de modulariser et de gérer des schémas complexes directement depuis XMLSpy.

Installer SchemaAgent et SchemaAgent Server

Pour plus de détails concernant l'installation de SchemaAgent et de SchemaAgent Server et la configuration des chemins de recherche sur les serveurs, voir le manuel d'utilisateur SchemaAgent.

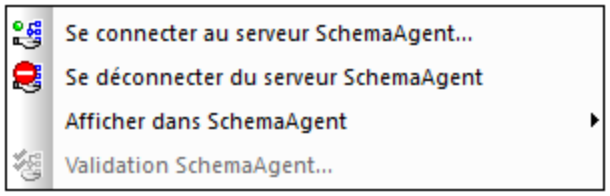
Configurer XMLSpy en tant que client SchemaAgent
Pour que XMLSpy puisse fonctionner en tant que client SchemaAgent, vous devez procéder comme suit :

- Télécharger SchemaAgent depuis le site web Altova. Vous pouvez maintenant utiliser le LocalServer de SchemaAgent pour servir les schémas. Pour plus d'informations concernant la configuration des chemins de recherche sur LocalServer, voir le manuel d'utilisateur SchemaAgent.
  Veuillez noter : SchemaAgent nécessite une licence valide, qui doit être achetée une fois que la période d'essai gratuite expire. Veuillez aussi noter que le paquet de produits Altova MissionKit, Enterprise Edition, contient le produit SchemaAgent et une licence pour ce produit. (Néanmoins, l'application SchemaAgent Server n'est pas incluse dans les paquets Altova MissionKit.)

- De plus, vous pouvez éventuellement télécharger et installer le SchemaAgent Server à base réseau depuis le site web Altova.
- Définir les chemins de recherche pour le serveur SchemaAgent (aussi appelé SchemaAgent Server de configuration). Une description détaillée expliquant comment faire est incluse dans le manuel d'utilisateur SchemaAgent. (Un chemin de recherche est un chemin menant vers le dossier contenant les schémas XML qui seront mappés pour leurs relations entre elles.)
- Lancer une connexion depuis XMLSpy vers un serveur SchemaAgent.

Important : tous les produits liés à SchemaAgent et SchemaAgent depuis Altova (y compris XMLSpy) commençant par la Version 2005 release 3 ne sont **pas compatibles** avec les versions précédentes de SchemaAgent ou de produits liés à SchemaAgent.

Commandes SchemaAgent dans XMLSpy
La fonction SchemaAgent dans XMLSpy est uniquement disponible dans le Mode Schéma et est accédée depuis des commandes de menu dans le menu de Conception de Schéma (*voir capture d'écran*) et en utilisant les Assistants à la saisie dans le Mode Schéma.



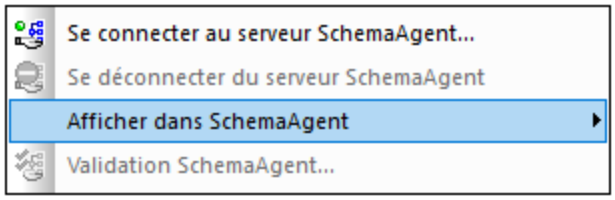
Les commandes de menu proposent des fonctions administratives générales. Les Assistants à la saisie (et les mécanismes GUI standard, comme les glisser/déposer) sont utilisés pour éditer des schémas.

Cette section décrit comment utiliser la fonction SchemaAgent disponible dans le Mode Schema.

6.7.1 Se connecter à SchemaAgent Server

Noter : SchemaAgent Client doit être installé afin que vous puissiez établir une connexion.

Avant de vous connecter à SchemaAgent Server, seule la commande **Se connecter au serveur SchemaAgent** est activée dans le menu **Conception de schéma** ; d'autres commandes SchemaAgent dans le menu **Conception de schéma** sont désactivées (*voir capture d'écran*). Les autres items de menu sont activés une fois qu'une connexion à un SchemaAgent Server a été effectuée avec succès.

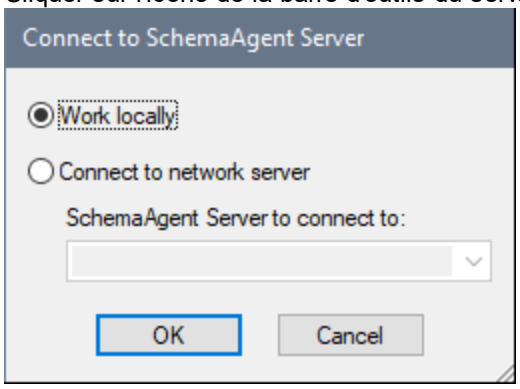


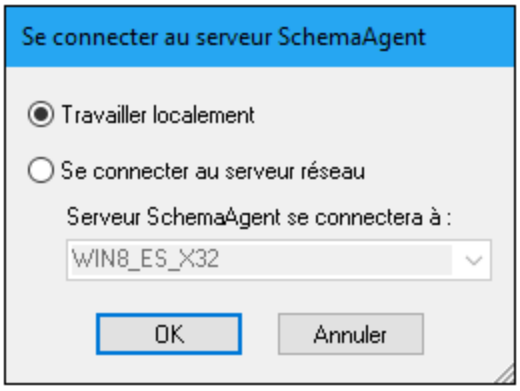
© 2016-2022 Altova GmbH

Étapes de connexion

Pour se connecter à un serveur SchemaAgent :

1. Cliquer sur l'icône de la barre d'outils du serveur Se connecter à SchemaAgent (**Conception de schéma | Se connecter au serveur SchemaAgent**). Le dialogue Se connecter au serveur SchemaAgent s'ouvre (*capture d'écran ci-dessous*) :





2. Vous pouvez utiliser soit le serveur local (le serveur SchemaAgent qui est contenu avec l'Altova SchemaAgent) ou un serveur de réseau (le produit SchemaAgent Server, qui est gratuit). Si vous choisissez Travailler localement, le serveur local de SchemaAgent sera lancé lorsque vous cliquez sur **OK** et qu'une connexion sera établie. Si vous choisissez Se connecter au serveur de réseau, le SchemaAgent Server sélectionné devra être exécuté pour qu'une connexion puisse être établie.

Note sur les serveur exécutés avec Windows XP SP2
Si le nom SchemaAgent Server est contenu dans le dialogue Se connecter à SchemaAgent Server mais que vous ne pouvez pas vous y connecter, il est possible que votre serveur ne prenne pas part au processus de résolution du nom de votre réseau. La résolution de nom est bloquées par le réglage par défaut du pare-feu Windows XP SP2.

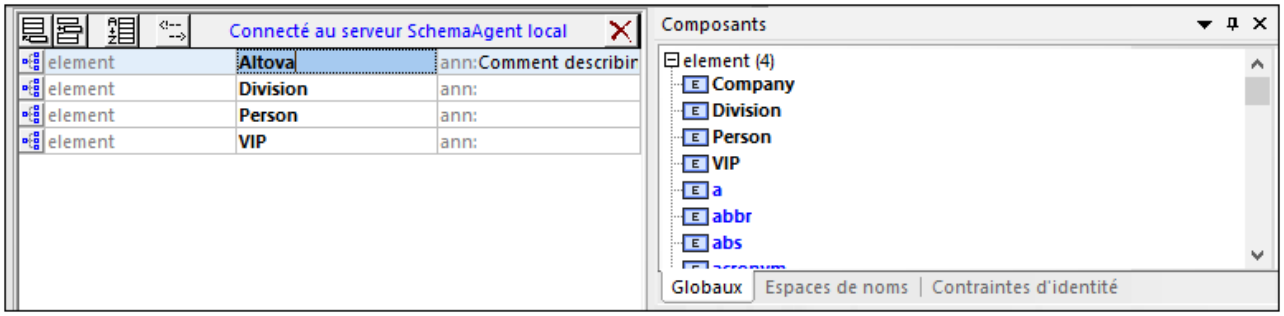
Pour vous connecter à ce type de serveur, veuillez procéder comme suit :

- Changer les paramètres du serveur pour permettre le processus de résolution de nom, ou

- Saisir l'adresse IP du serveur dans le champ Éditer du dialogue Se connecter.

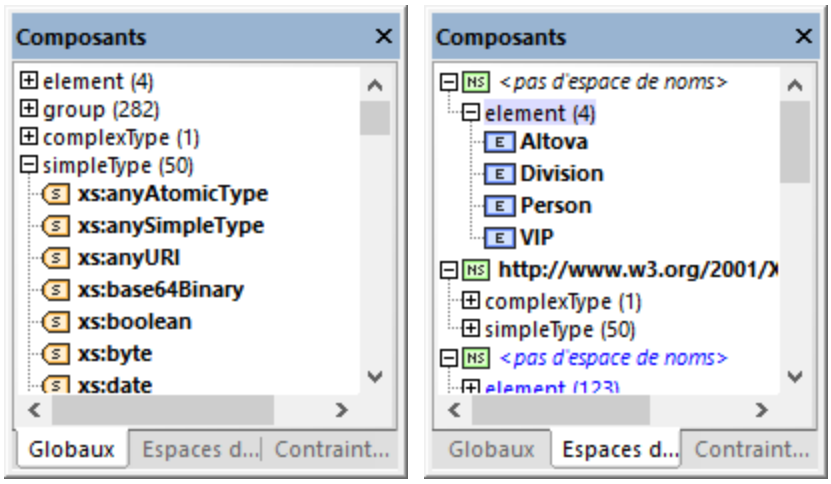
Cela devra être fait une seule fois étant donné que SchemaAgent Client stocke le string de connexion de cette dernière connexion réussie.

Mode Schéma après la connexion au serveur SchemaAgent
Une fois qu'une connexion à un serveur SchemaAgent a été établie, le Mode Schéma ressemblera à cela :



Noter :

- En haut du Mode Globaux, le texte "Connecté à SchemaAgent Server" apparaît, spécifiant le serveur auquel la connexion a été faite.
- Vous avez à présent le plein accès à tous les schémas et aux structures de schéma disponibles dans le chemin de recherche du serveur. Les structures de schéma SchemaAgent comme les éléments globaux, les types complexes et les types simples sont visibles en **texte bleu en gras**, et se trouvent en dessous des structures du schéma actif (**texte noir en gras**).



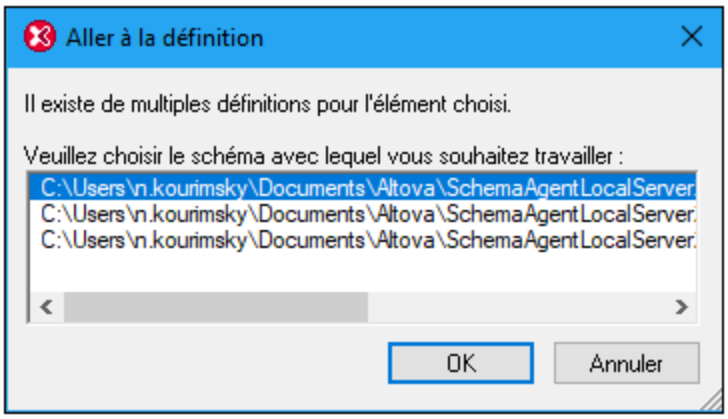
Les structures de schéma peuvent être consultées par Type (Globaux), par Espace de noms, ou par Contraintes d'identité dans l'onglet respectif de l'assistant de saisie Composants.

© 2016-2022 Altova GmbH

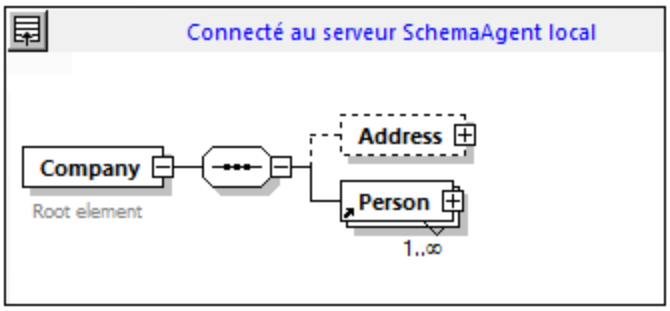
6.7.2 Ouvrir des Schémas trouvés dans le Chemin de recherche

Cet exemple montre comment ouvrir un schéma trouvé dans un chemin de recherche défini dans SchemaAgent Server. Il utilise le fichier `DB2schema.xsd` disponible dans le dossier `..\Tutorial` en tant que le schéma actif. L'onglet `Globaux` de l'Assistant à la saisie Composants est actif.

1. Faire défiler la liste jusqu'à l'entrée bleue `Company` dans l'Assistant à la saisie Composant et la double-cliquer. Le dialogue Aller à la définition s'ouvre.



2. Cliquer sur l'entrée `Addresslast.xsd` et cliquer sur **OK** pour confirmer. Cela ouvre le schéma `addresslast.xsd` et affiche le modèle de contenu de l'élément `Company`.



Noter : Double-cliquer sur une structure de schéma SchemaAgent, comme Element, complexType, ou simpleType, pour ouvrir le schéma associé (ainsi que tous les autres schémas inclus) dans XMLSpy.

6.7.3 Utiliser des IIRs

Le schéma XML fournit des instructions Import, Include et Redefine (IIR) pour vous assister dans la modularisation des schémas. Chaque méthode dispose d'exigences d'espaces de noms différentes. Ces exigences sont vérifiées automatiquement par SchemaAgent Client et XMLSpy lorsque vous tentez de créer des IIR.

Imports, Includes et Redefines (IIRs)

Les structures de schéma peuvent être "insérées" à l'aide de méthodes différentes :

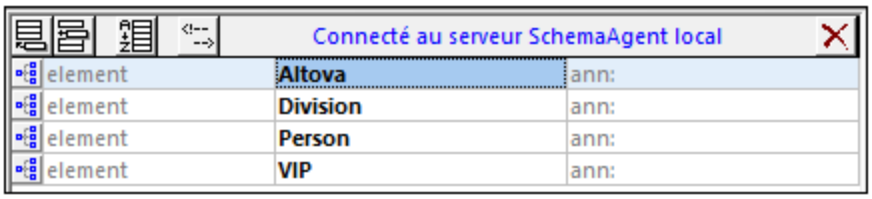
- les éléments globaux peuvent être glissés directement depuis l'Assistant de saisie Composants dans le modèle de contenu d'un composant de schéma (dans le Mode Schéma).
- les Composants, comme les Types complexes et les Types simples peuvent être sélectionnés depuis la fenêtre de liste qui s'ouvre automatiquement lorsque vous définissez de nouveaux éléments (attributs, etc.)
- les Composants, comme les Types complexes, peuvent être sélectionnés depuis l'Assistant de saisie Détails lors de la création de ce type de structure.

Incorporer des composants de schéma

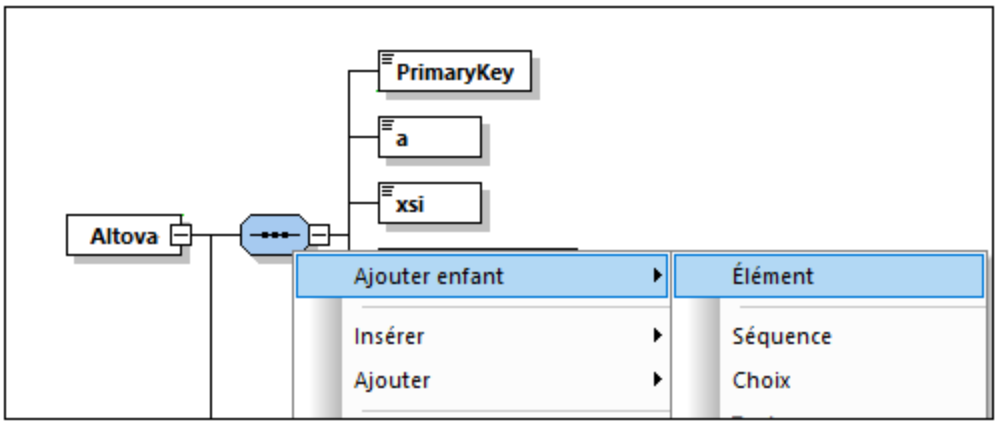
Cet exemple utilise le fichier `DB2schema.xsd` disponible dans le dossier `..\Tutorial` en tant que le schéma actif ; l'onglet `Global` de l'Assistant de saisie Composants est actif.

Pour utiliser les structures de schéma des schémas SchemaAgent Server :

1. Assurez-vous que vous êtes connecté à un serveur SchemaAgent (voir Se connecter à un serveur SchemaAgent).
2. Ouvrir et renommer le fichier `DB2Schema.xsd` pour cet exemple, par ex. en `Altova-office`.



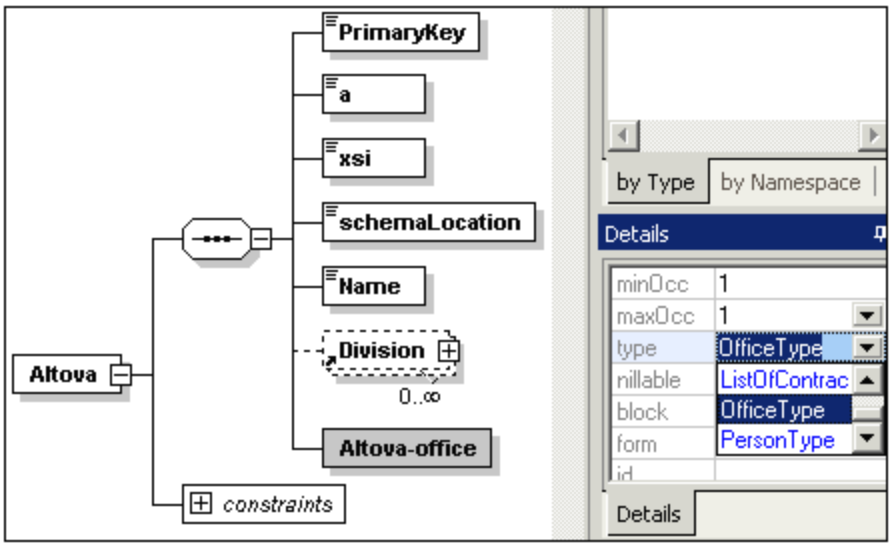
3. Cliquer sur l'icône de l'élément `Altova` dans l'Aperçu de Schéma pour voir son modèle de contenu.
4. Cliquer avec la touche de droite sur le compositeur de séquence `Altova` et sélectionner l'option de menu **Ajouter enfant | Élément**. Veuillez noter qu'une liste contenant tous les éléments globaux dans le chemin de serveur s'ouvre automatiquement à cet endroit. En sélectionner un pour incorporer cet élément.



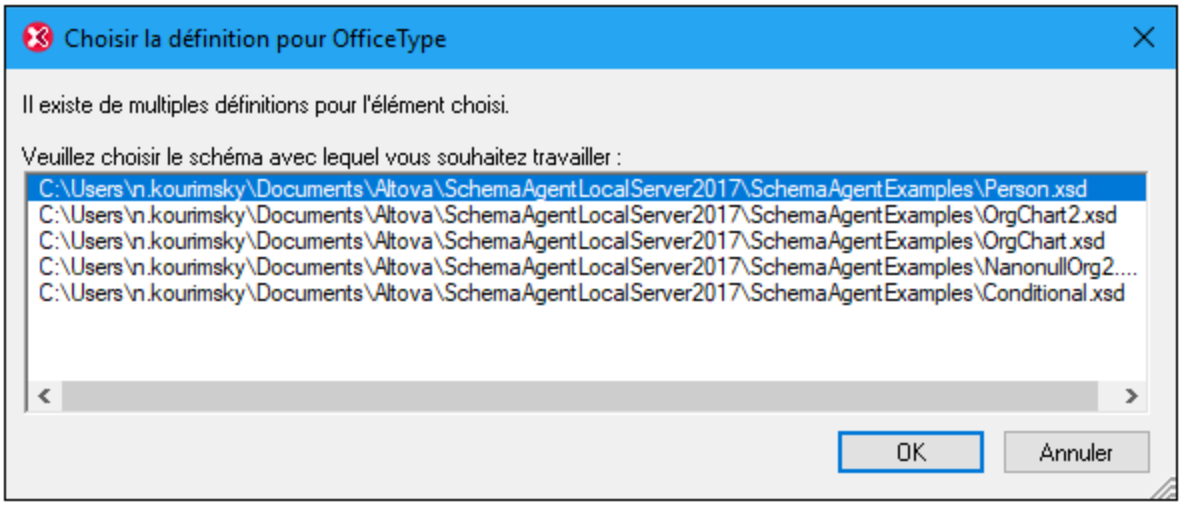
5. Saisir `Altova-office` en guise du nom pour ce nouvel élément et appuyer sur **Entrée**.

© 2016-2022 Altova GmbH

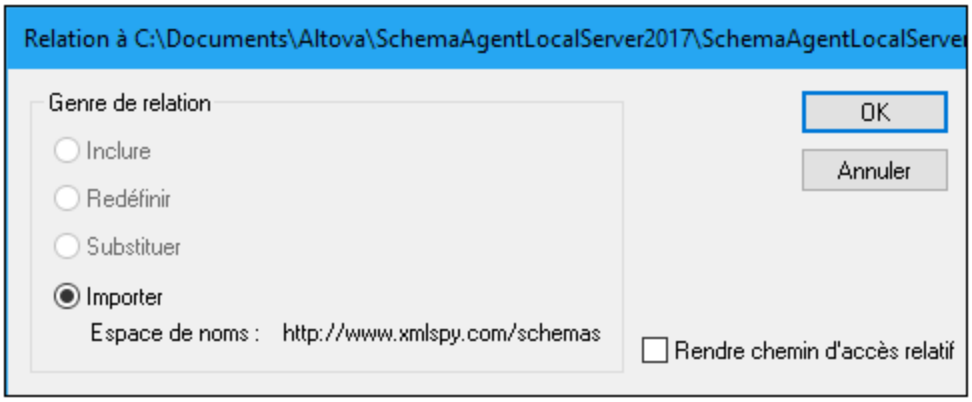
6. À l'aide de l'Assistant de saisie Détails, cliquer sur la liste de choix `type` et sélectionner l'entrée `OfficeType`.



Cela permet d'ouvrir le dialogue Sélectionner une définition pour l'OfficeType.



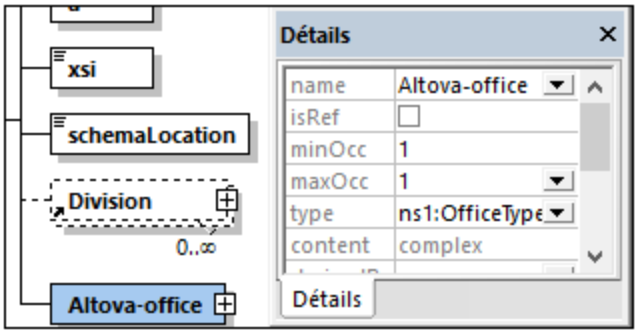
7. Choisir `Orgchart.xsd` et cliquer sur **OK** pour confirmer.



8. Cliquer sur **OK**. La commande Import a été sélectionnée automatiquement pour vous. Une icône d'agrandissement apparaît dans l'élément `Altova-office`.

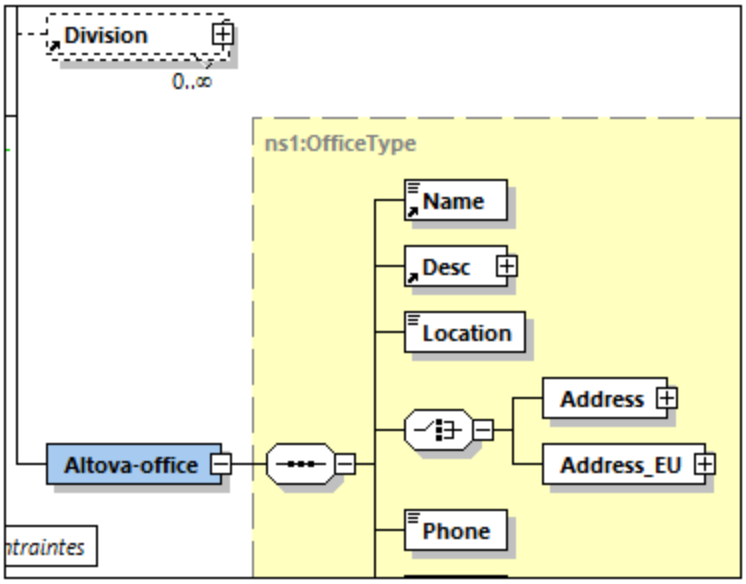


Noter : L'entrée `type` dans l'Assistant de saisie Détails a changé ; elle est maintenant affichée en tant que `ns1:OfficeType` en raison du fait que le fichier de schéma `Orgchart.xsd` a été importé et que les espaces de noms cibles doivent être différents dans les deux schémas. Une commande Import a aussi été ajoutée au schéma.



9. Cliquer sur la touche Agrandir pour voir le modèle de contenu `OfficeType`.

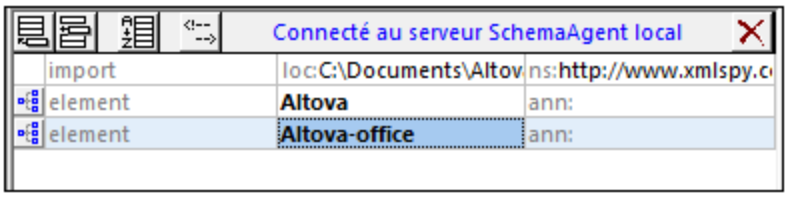
© 2016-2022 Altova GmbH



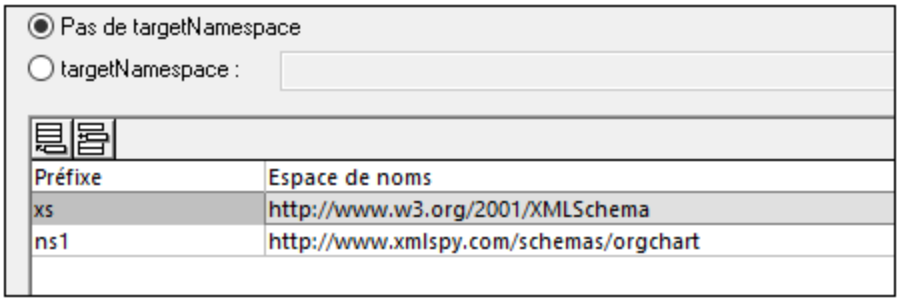
10. Appuyer sur **F8** pour valider le schéma. Le message "Schéma est valide" devrait apparaître à ce niveau.

Nettoyer le schéma :

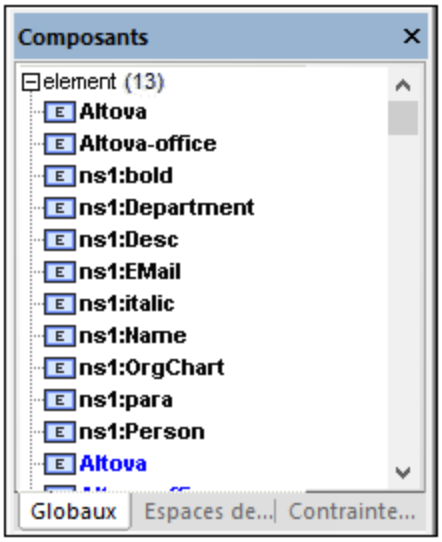
1. Supprimer l'élément `Division` dans le modèle de contenu.
2. Cliquer sur l'icône Retourner aux globaux pour passer à l'Aperçu de schéma.
3. Supprimer les éléments globaux suivants : `Division`, `Person` et `VIP`.



4. Sélectionner l'option de menu **Conception de schéma | Paramètres de schéma** pour voir comment les paramètres d'espace de noms ont été modifiés.



Le préfixe `ns1` a été ajouté automatiquement à l'espace de nom `www.xmlspy.com/schemas/orgchart`. Les composants (*voir capture d'écran*) et les Assistants de saisie Détails affichent toutes les structures importées avec le préfixe d'espace de noms `ns1:`.



Noter :

- Les changements effectués aux schémas sous le contrôle de SchemaAgent Server à l'aide de XMLSpy met à jour automatiquement d'autres schémas dans le chemin SchemaAgent Server qui a référencé le schéma modifié.
- Il est possible de voir les doubles des éléments de structures, de types simples, etc. dans les Assistant de saisie (en noir et en bleu), si le schéma sur lequel vous travaillez se trouve aussi dans le chemin SchemaAgent Server.

6.7.4 Consulter des schémas dans SchemaAgent

Pour travailler avec le schéma actif et ses schémas liés dans SchemaAgent, sélectionner l'option de menu **Conception de schéma | Afficher dans SchemaAgent |** ***schéma ou schémas liés*** *(voir capture d'écran).* Vous avez la possibilité d'ouvrir uniquement le schéma actif dans SchemaAgent (commande **Fichier seulement**), ou le schéma actif avec soit (i) tous les schémas référencés directement, soit (ii) tous les schémas directement référencés, ou encore (iii) tous les schémas liés directement.

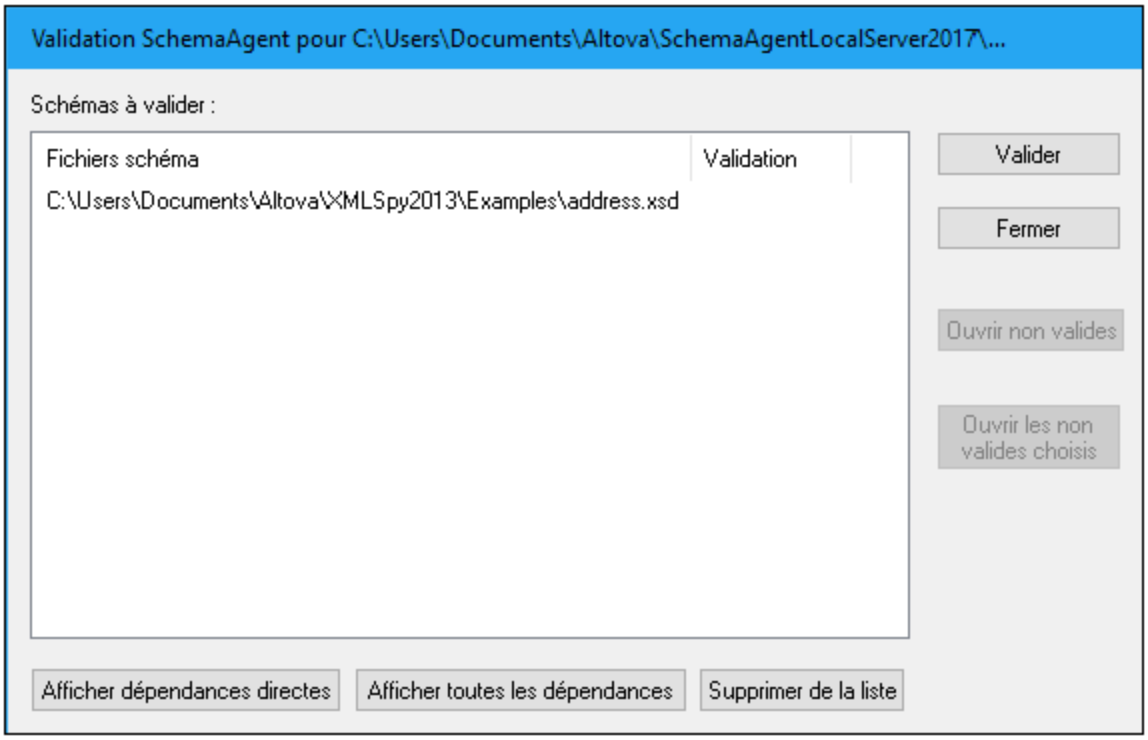
6.7.5 Validation de SchemaAgent

XMLSpy, utilisé de concert avec SchemaAgent, vous permet de valider non seulement le schéma actif actuel mais également les schémas liés au schéma actif actuellement. Nous appelons cela une validation de SchemaAgent. SchemaAgent fait la distinction entre deux types de schéma pour une validation étendue : (i) des schémas directement dépendants (des schémas référencés directement et des schémas de référencement directs), et (ii) tous les schémas dépendants (outre les dépendances directes, celles-ci comprennent des dépendances indirectes, il s'agit là du jeu de schémas liés l'un à l'autre par le biais d'un schéma intermédiaire.

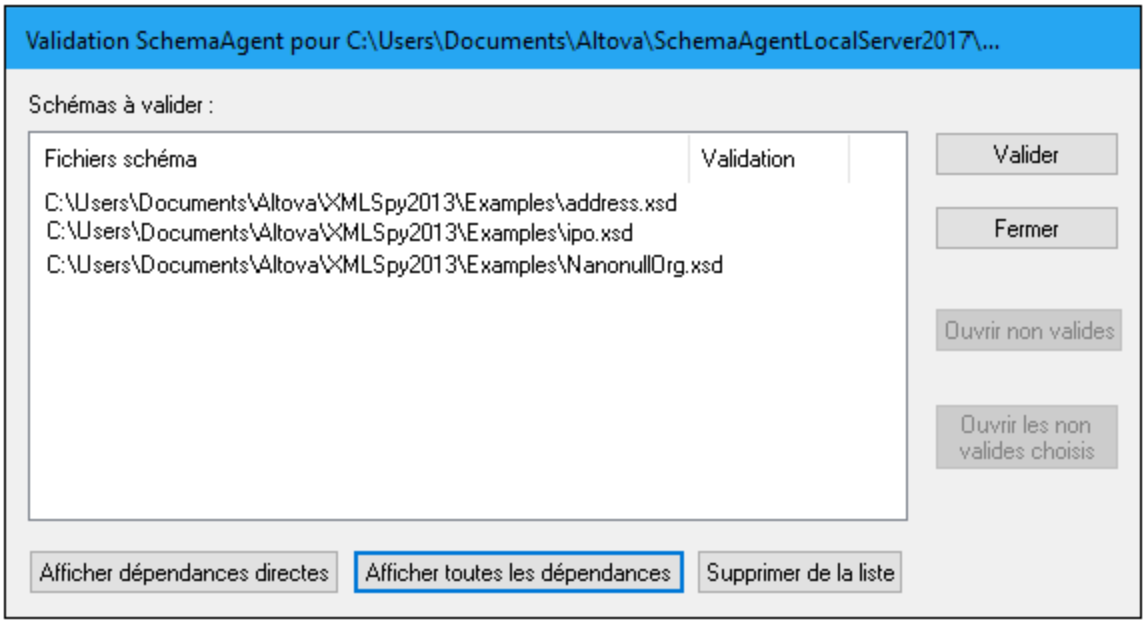
© 2016-2022 Altova GmbH

Vous trouverez ci-dessous un exemple vous indiquant comment effectuer une validation SchemaAgent. Cet exemple part du principe que le fichier de schéma `address.xsd` est un schéma actif dans le Mode Schéma de XMLSpy. Pour activer la commande **Validation de SchemaAgent**, veuillez vous assurer que les chemins de recherche sur SchemaAgent Server contiennent le fichier actif et certains fichiers dépendants. Ensuite, procéder comme suit :

1. Cliquer sur l'icône **Validation de SchemaAgent** dans la barre outils ou sur l'élément de menu **Conception de schéma | Validation de SchemaAgent**. Le dialogue Validation de SchemaAgent s'ouvre (*capture d'écran ci-dessous*), et vous pourrez choisir si vous souhaitez valider le schéma actif uniquement ou si vous souhaitez un ou plusieurs schémas liés également.

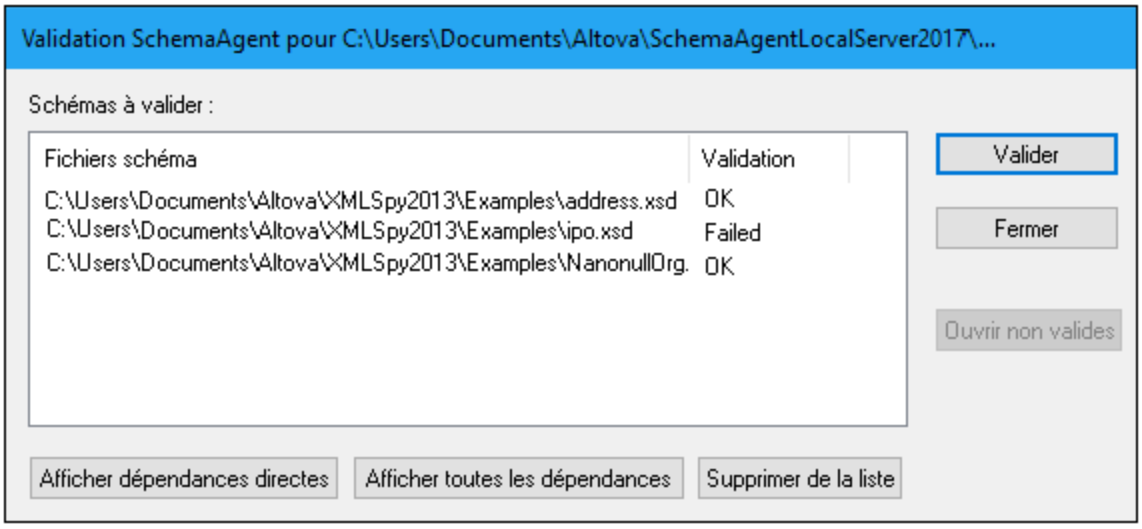


2. Pour insérer des schémas dans la liste, cliquer sur la touche **Afficher les dépendances directes** ou **Afficher toutes les dépendances** selon vos besoins. Dans cet exemple, nous avons cliqué sur la touche **Afficher toutes les dépendances**, cela insère tous les fichiers qui sont référencés directement ou indirectement dans la liste.



À cet endroit, vous pouvez supprimer un schéma de la liste (**Supprimer de la liste**) si vous souhaitez le faire.

3. Cliquer sur la touche **Valider** pour valider tous les schémas dans la fenêtre de liste.



La colonne Valider affiche si la validation a été réussie ou si elle a échoué.

Vous pouvez maintenant ouvrir tous les schémas non valides ou un ensemble de schémas non valides sélectionnés dans XMLSpy.

© 2016-2022 Altova GmbH

6.8 Chercher dans les schémas

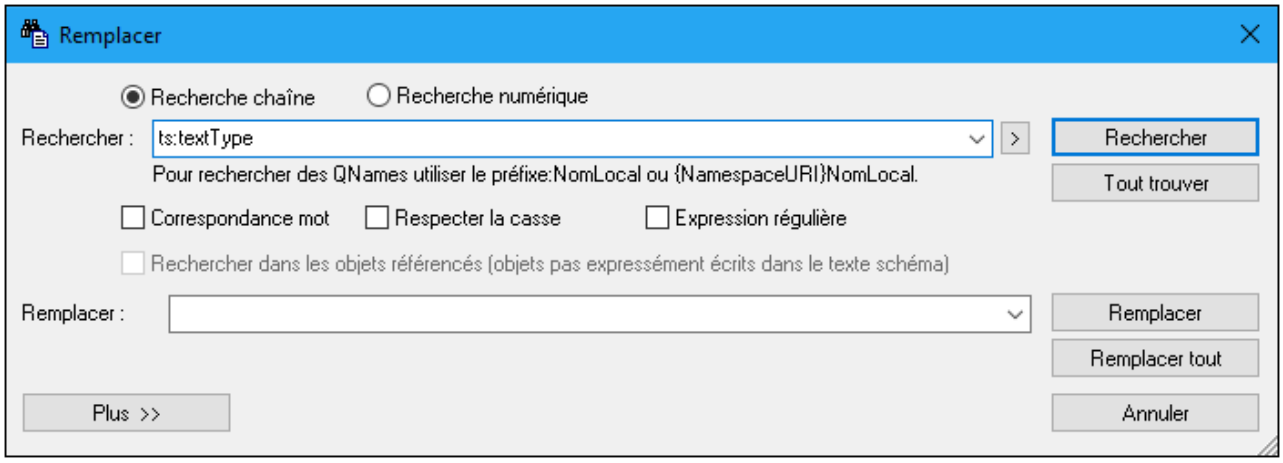
Dans le Mode Schéma, les schémas XML peuvent être cherchés de manière intelligente en utilisant la fonction Chercher et remplacer de XMLSpy dans le Mode Schéma.

La fonction Chercher et remplacer dans le Mode Schéma est activée lorsqu'un schéma est actif dans le Mode Schéma. L'accès peut se faire de deux manières :

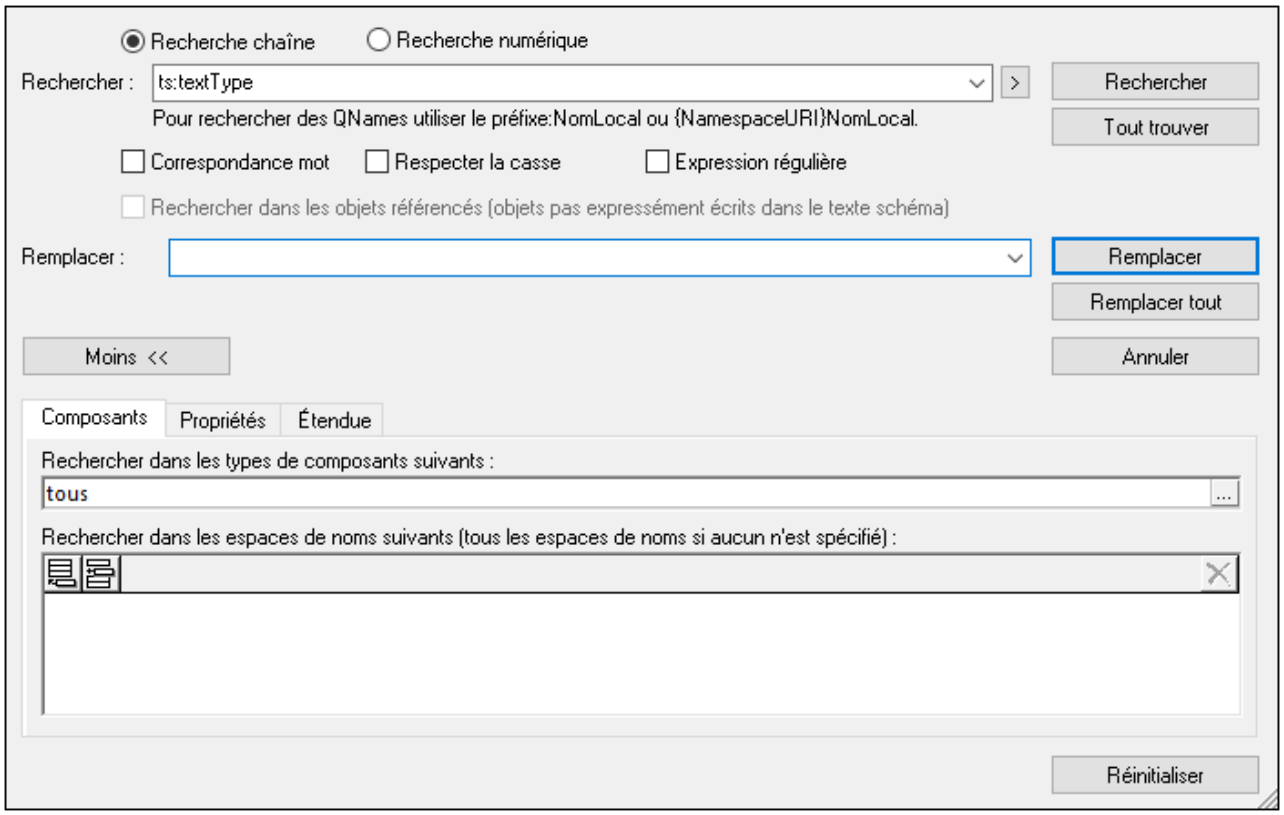
- Par le biais des commandes de menu **Édition | Recherche** et **Édition | Remplacer**.
- Par le biais des touches **Recherche** et **Remplacer** dans la fenêtre Rechercher dans les schémas.

Cliquer sur une commande ou sur une touche pour ouvrir le dialogue Recherche ou Remplacer, selon la touche/commande activée. Le dialogue Remplacer (*captures d'écran ci-dessous*) est différent du dialogue Rechercher car il a un champ de saisie pour le terme à Remplacer.

Le dialogue Remplacer standard ressemble à l'exemple ci-dessous :



Cliquer sur la touche **Plus** pour agrandir le dialogue et afficher les fonctions supplémentaires de recherche (*capture d'écran ci-dessous*).



Procédure à suivre :

- Saisir les termes de recherche et de remplacement dans les champs de saisie de texte Rechercher et remplacer
- Spécifier les composants de schéma à rechercher dans l'onglet Composants
- Spécifier les propriétés des composants à chercher ; cela aide à réduire la recherche
- Définir l'étendue de la recherche au document ou projet actuel, ou spécifier un dossier à chercher
- Exécuter la commande
- Utiliser la fenêtre Rechercher dans les schémas pour trouver rapidement un composant

La touche **Réinitialiser** située en bas du dialogue permet de réinitialiser les paramètres d'origine, qui sont les suivants :

- Aucun terme de recherche, aucun terme de remplacement
- Composants : `tous`
- Espaces de noms : aucun spécifié
- Restrictions de propriété : `partout`
- Restrictions de propriété supplémentaires : aucune
- Étendue : fichier actuel

Noter : Les expressions régulières ne sont pas prises en charge dans le champ Remplacer.

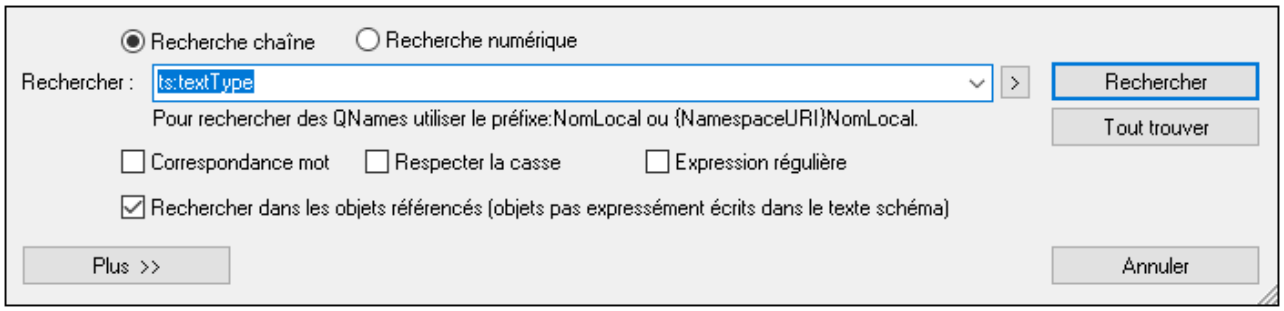
© 2016-2022 Altova GmbH

6.8.1 Terme de recherche

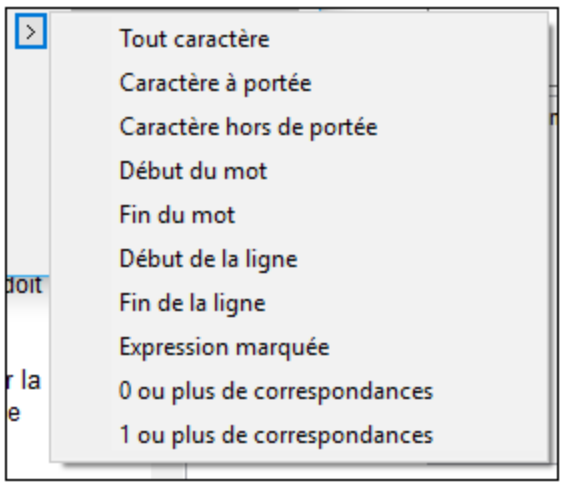
Le terme de recherche peut être saisi en tant que chaîne (sélectionner le bouton radio **Chaîne**) ou en tant que numéro (bouton radio **Numérique**).

Recherche Chaîne

Dans une recherche chaîne (*capture d'écran ci-dessous*), l'entrée peut être : (i) du texte ; (ii) un QName ; ou (iii) une expression régulière. Pour les recherches de QName, l'espace de noms est déterminé sur la base soit du préfixe utilisé dans le document, soit par l'URI de l'espace de noms. L'un des deux doit être saisi. Dans la capture d'écran ci-dessous, le préfixe `ts:` est le préfixe utilisé dans le document pour identifier un certain espace de noms.



Pour effectuer une recherche à l'aide d'une expression régulière, cocher la case Expression régulière puis saisir l'expression régulière. Les assistants à la saisie pour les expressions régulières sont disponibles dans un menu qui peut être activé en cliquant sur la touche fléchée orientée vers la droite située à droite du champ de saisie Recherche (*capture d'écran ci-dessous*).

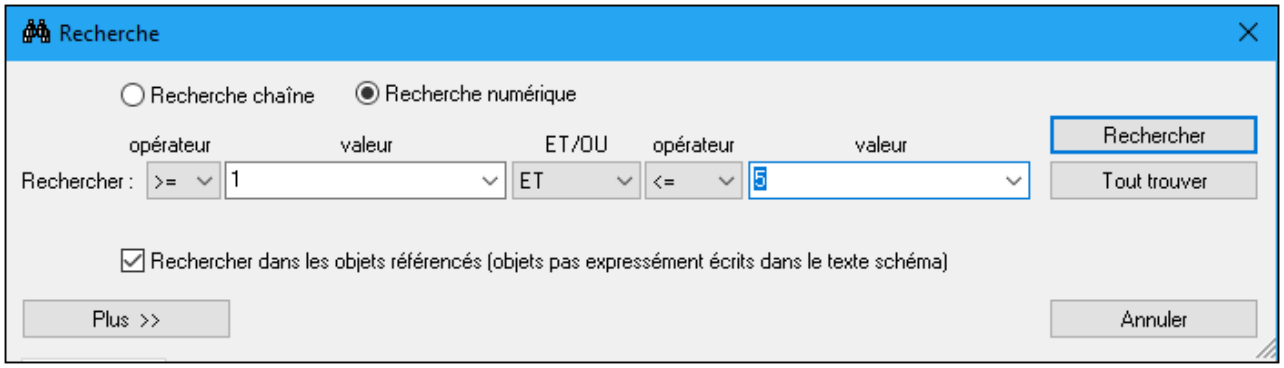


Vous pouvez aussi sélectionner si un terme de recherche doit correspondre à un mot entier dans le document ou/et si la casse du document doit correspondre. Utiliser les cases à cocher situées en dessous du champ de saisie pour spécifier ces options.

Si vous souhaitez procéder à une recherche dans les objets référencés (comme une définition de type complexe ou un élément global), cocher la case Chercher dans les objets référencés. Cette option est uniquement disponible dans le dialogue Rechercher ; elle est désactivée dans le dialogue Remplacer.

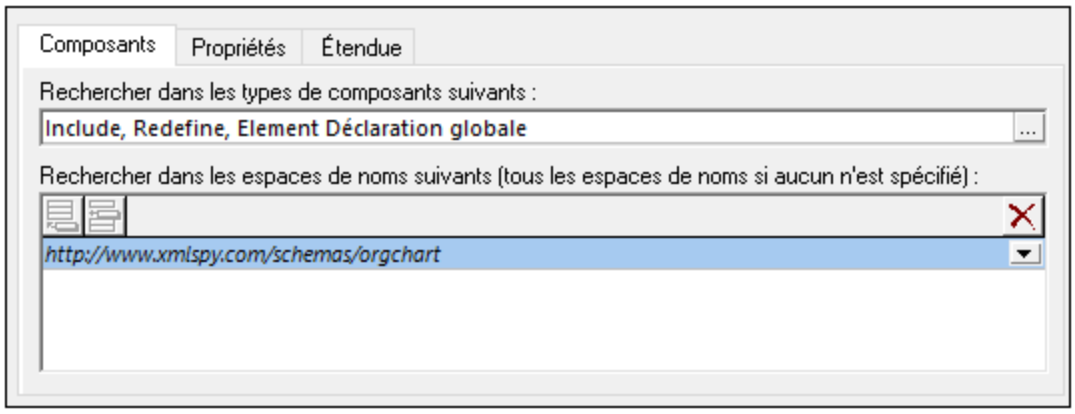
Recherche Numérique

Si le bouton radio "Recherche numérique" a été sélectionné, le terme de recherche peut être un seul paramètre de recherche "opérateur et nombre", ou en ensemble de deux paramètres de recherche "opérateur et nombre" joints par le connecteur logique `ET` ou `OU`. Dans la capture d'écran ci-dessous, vous trouverez deux paramètres de recherche qui créent un terme de recherche pour tous les entiers situés entre 1 et 5, inclus.



6.8.2 Composants

La recherche peut être limitée à un ou plusieurs types de composant et à un ou plusieurs espaces de noms cibles. Ces options sont disponibles dans l'onglet Composants. Agrandir le dialogue Chercher ou Remplacer en cliquant sur la touche **Plus**. Cela fera apparaître les onglets pour affiner la recherche. Parmi ces onglets, apparaîtra l'onglet Composants (*capture d'écran ci-dessous*).

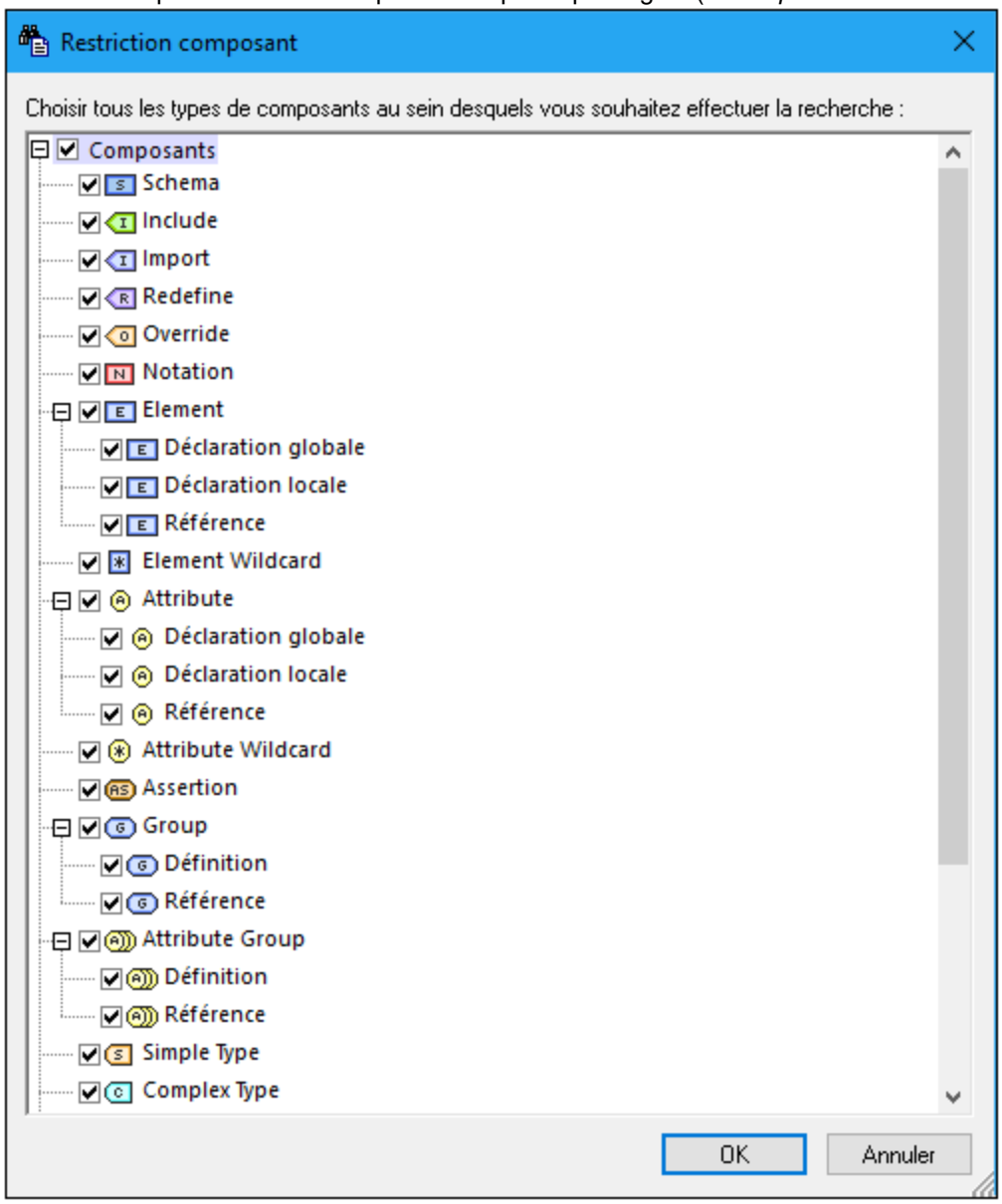


L'onglet Composant se compose de deux parties : (i) la sélection des types de composant à chercher, et (ii) la sélection des espaces de noms cibles à chercher.

© 2016-2022 Altova GmbH

Sélection Composant

Vous pouvez saisir les types de composants à chercher en cliquant sur l'icône **Ajouter** ... situé à la droite du champ de texte (*voir capture d'écran ci-dessus*). Le dialogue Restriction de composant apparaît (*capture d'écran ci-dessous*), dans lequel vous pouvez sélectionner les composants à chercher en les cochant. Le fait de cocher l'item `Components` situé en haut de la liste permet de sélectionner tous les composants (saisie de texte : `all`). Le fait de le décocher permet de désélectionner tous les composants (saisie de texte : `none`), y compris des composants sélectionnés individuellement. Les composants individuels peuvent donc être sélectionnés uniquement lorsque l'item `Components` est décoché. Les composants sélectionnés sont saisis dans le champ de texte en tant que liste séparée par virgule (*voir capture d'écran ci-dessus*).



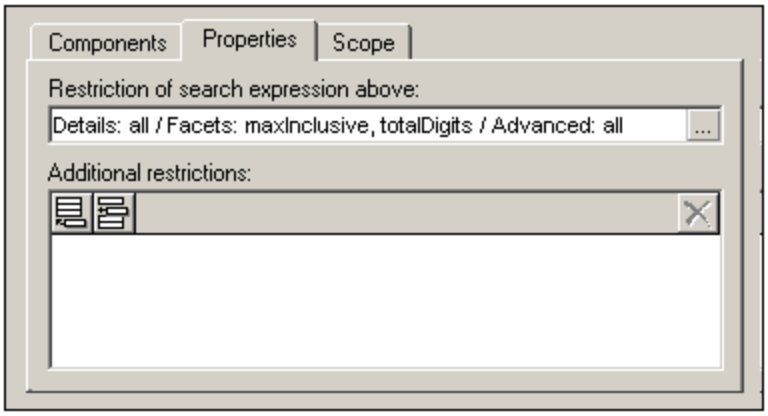
Noter : À chaque fois que l'onglet Composants ou le dialogue Chercher/Remplacer est ouvert, la sélection de composant précédente est retenue.

Sélection espace de noms

Afin de sélectionner un ou plusieurs espaces de noms cible à chercher, cliquer sur les icônes **Ajouter** ou **Insérer** et saisir l'espace de noms requis. Si aucun espace de noms cible n'est spécifié, tous les espaces de noms cibles seront cherchés. Pour supprimer un espace de noms cible qui a été saisi dans ce volet, sélectionner l'espace de noms cible et cliquet sur l'icône **Supprimer**.

6.8.3 Propriétés

La recherche peut être limitée à une ou plusieurs propriétés de composant (détails et facettes) en utilisant les options dans l'onglet Propriétés, ainsi que pour correspondre aux contenus des propriétés. Agrandir le dialogue Chercher ou Remplacer en cliquant sur la touche **Plus**, puis en sélectionnant l'onglet Propriétés (*capture d'écran ci-dessous*).

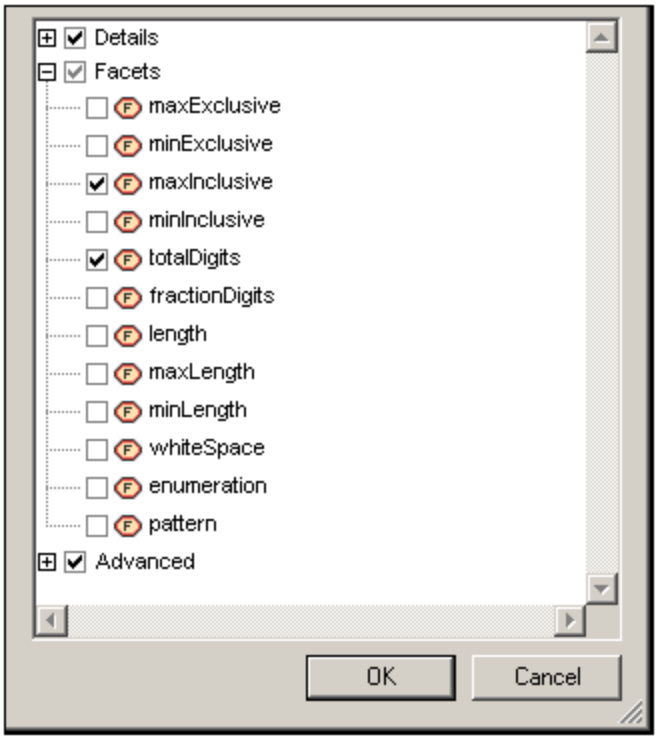


L'onglet Propriétés consiste en deux parties : (i) pour limiter le terme de recherche principal (saisi dans le champ de texte Chercher) ; et (ii) pour ajouter des restrictions de contenu supplémentaires (qui ont leur propre terme de correspondance) ; *voir la section Restrictions supplémentaires ci-dessous*.

Sélection de propriétés

Vous pouvez saisir les types de propriété que vous souhaitez chercher en cliquant sur l'icône **Ajouter** ..., qui se trouve à droite du champ de texte (*voir capture d'écran ci-dessus*). Le dialogue Limitations de propriété s'ouvre (*capture d'écran ci-dessous*), dans lequel vous pouvez choisir les propriétés que vous souhaitez chercher en les cochant. Les propriétés sont organisées en trois groupes : (i) Détails; (ii) Facettes; (iii) Avancé (comme la propriété DerivedFrom). Cocher Détails, Facettes, ou Avancé pour sélectionner toutes les propriétés dans ce groupe. Décocher un groupe pour désélectionner toutes les propriétés de ce groupe, y compris les propriétés qui ont été sélectionnées individuellement. C'est pourquoi les propriétés individuelles peuvent être sélectionnées uniquement lorsque l'item de groupe est décoché. Les propriétés sélectionnées sont saisies dans le champ de texte (*voir capture d'écran ci-dessus*).

© 2016-2022 Altova GmbH

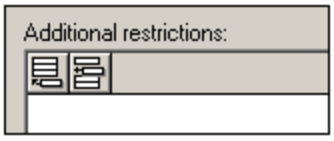


Noter : À chaque fois que l'onglet Propriétés ou le dialogue Chercher/Remplacer est ouvert, la sélection des propriétés précédentes est retenue.

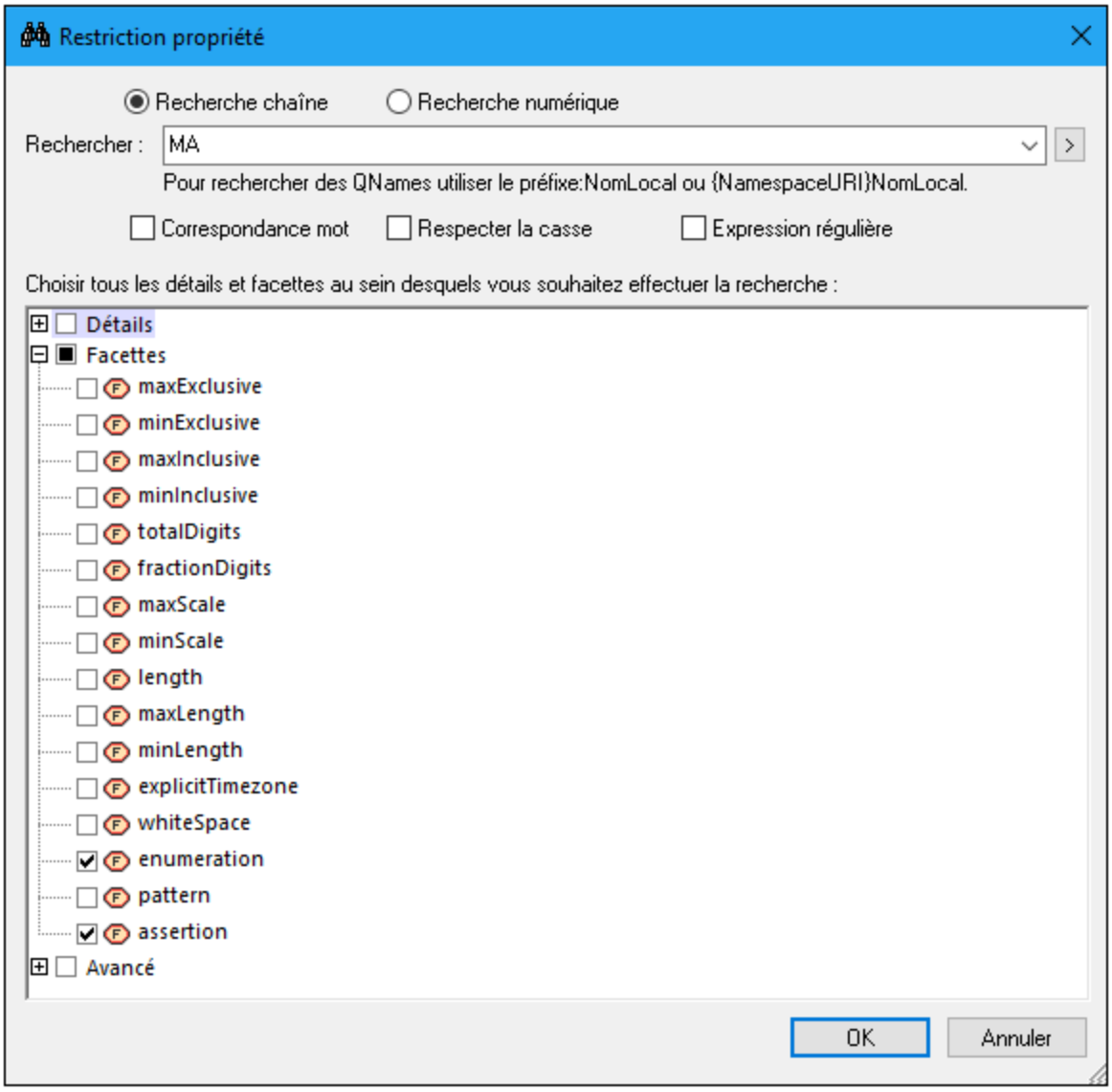
Restrictions supplémentaires

Une restriction supplémentaire vous permet de spécifier la valeur de la propriété que vous souhaitez rechercher. Par exemple, si vous recherchez un élément appelé `state` avec une énumération `MA` (pour l'État US de Massachusetts), vous pourriez spécifier la value `MA` de la propriété `enumeration` avec l'option Restrictions supplémentaires. Procéder comme suit :

1. Dans le volet Restrictions supplémentaires, cliquer sur l'icône **Ajouter** ou **Insérer** (*capture d'écran ci-dessous*).

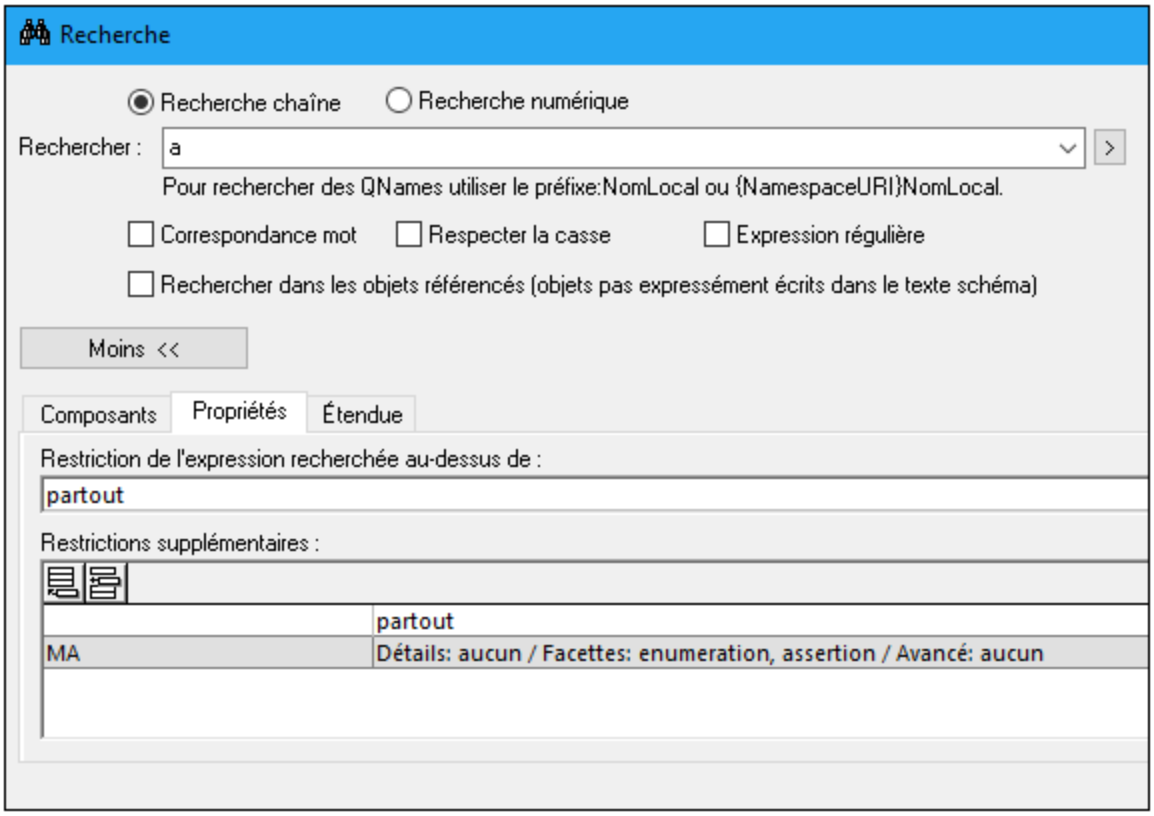


2. Cela permet d'ajouter une colonne au volet et le dialogue Restriction de propriété s'ouvre. Désélectionner toutes les propriétés et sélectionner uniquement la propriété `enumeration` (*capture d'écran ci-dessous*).



3. Dans le champ de texte situé en haut du dialogue, saisir la valeur d'énumération à chercher, dans ce cas, `MA` (*voir capture d'écran ci-dessus*).
4. Cliquer sur **OK**. La restriction supplémentaire est saisie dans la ligne récemment créée dans le volet Restrictions supplémentaires (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



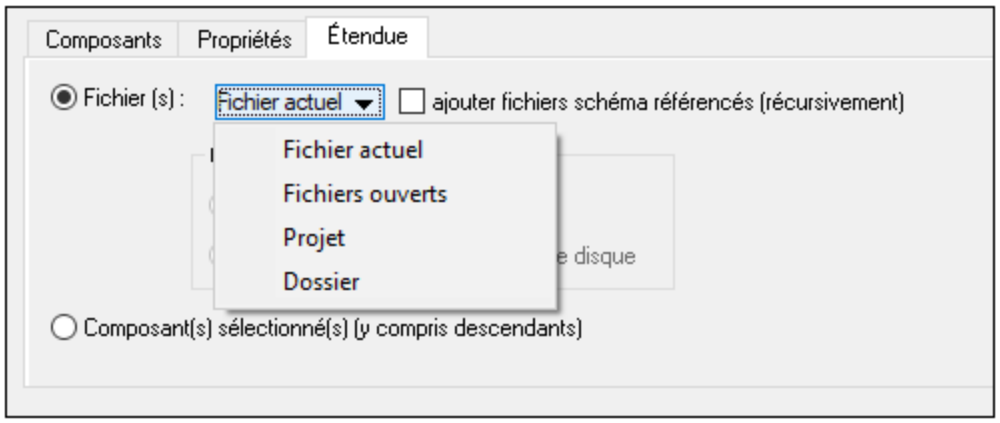
Dans la capture d'écran ci-dessus, veuillez noter que le terme de la recherche est `ipo:state`. Dans l'onglet Propriétés, `partout` spécifie que toutes les propriétés seront recherchées mais que la restriction supplémentaire spécifie que la recherche se limitera à des énumérations ayant une valeur de `MA`.

Plusieurs restrictions supplémentaires peuvent être rajoutées pour limiter encore la recherche. Afin de supprimer une restriction supplémentaire, sélectionner la restriction supplémentaire et cliquer sur l'icône **Supprimer**.

Noter : À chaque fois que l'onglet Propriétés ou que le dialogue Chercher/Replace est ouvert, les restrictions supplémentaires précédentes sont conservées.

6.8.4 Étendue

L'étendue de la recherche peut être définie dans l'onglet Étendue (*capture d'écran ci-dessous*). Vous pouvez sélectionner soit des fichiers doit le composant de schéma actuellement sélectionné dans le Mode Schéma.



Si l'option Fichiers est sélectionnée, vous pouvez également spécifier une option parmi la liste suivante :

- *Fichier actuel :* une option supplémentaire pour chercher les fichiers inclus, importés et redéfinis est aussi disponible.
- *Fichiers ouverts :* tous les fichiers de Schéma XML (XSD) qui sont ouverts dans XMLSpy. Seules les commandes Chercher tout et Remplacer tout sont activées ; la recherche à étape unique n'est pas disponible.
- *Projet :* le projet actif actuellement est sélectionné, avec l'option de sauter des dossiers externes. Seules les commandes Chercher tout et Remplacer tout sont activées ; la recherche à étape unique n'est pas disponible. Si le mode par défaut pour l'extension de fichier `.xsd` (**Outils | Options | Types de fichier | Mode par défaut**) n'est pas le Mode Schéma, alors les fichiers `.xsd` ne sont pas cherchés.
- *Dossier :* vous pouvez rechercher le dossier souhaité ; une option permettant de chercher les sous-dossiers est également disponible. Seules les commande Chercher tout et Remplacer tout sont activées ; la recherche à étape unique n'est pas disponible. Si le mode par défaut pour l'extension de fichier `.xsd` (**Outils | Options | Types de fichier | Mode par défaut**) n'est pas le Mode Schéma, alors les fichiers `.xsd` ne sont pas cherchés.
- *Fichiers Included, imported et redefined* peuvent être inclus dans l'étendue en cochant l'option pour les ajouter à l'étendue.

Dans le dialogue Remplacer, vous pouvez choisir si vous souhaitez copier le remplacement du fichier sur le disque dur ou si vous souhaitez ouvrir le fichier dans XMLSpy. Pour ce faire, sélectionner le bouton approprié dans le dialogue.

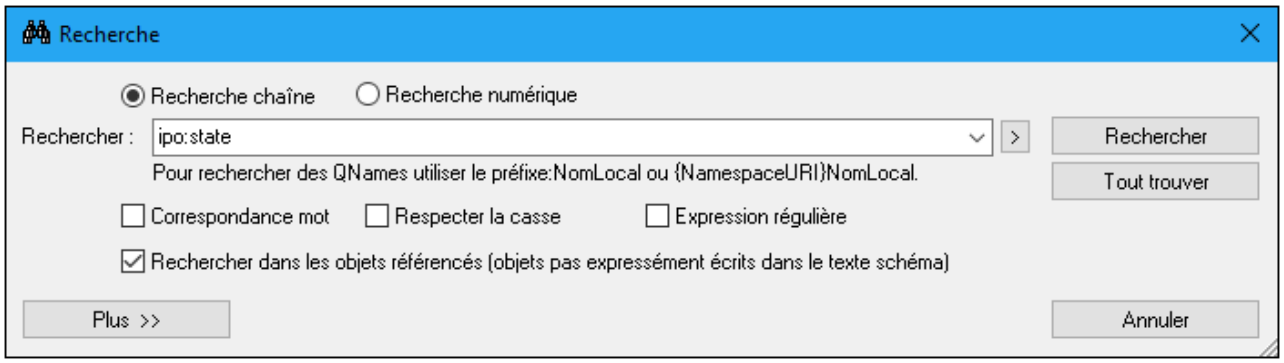
6.8.5 Commandes Recherche et Remplacer

La commande **Recherche** se comporte différemment dans le dialogue Recherche et dans le dialogue Remplacer. Le comportement de la commande **Recherche** dans les deux dialogues et de la commande **Remplacer** est décrit ci-dessous.

Dialogue **Recherche**

Une fois avoir saisi le terme recherché et, en option, avoir sélectionné d'autres critères de recherche, vous pouvez cliquer soit sur la commande **Recherche (F3)** ou **Tout trouver** (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



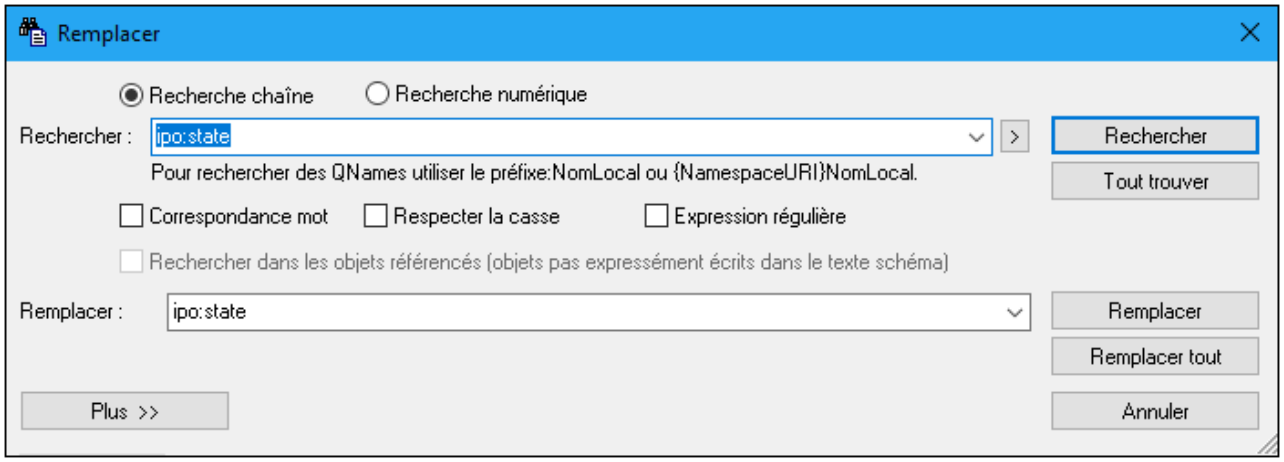
Cliquer sur la commande **Recherche (Ctrl+F)** dans le dialogue pour fermer le dialogue Recherche et trouver l'occurrence suivante du terme cherché dans le cadre de l'étendue spécifiée et des critères de recherche étendue. L'occurrence suivante sera trouvée relativement au composant sélectionné actuellement dans le Mode Schéma. Lorsque la recherche arrive à la fin de l'étendue, elle ne reprendra pas automatiquement du début de l'étendue. C'est pourquoi, avant de lancer la recherche, vous devriez vous assurer que le composant actuellement sélectionné dans le Mode Schéma se trouve avant la partie du document que vous souhaitez parcourir.

Le résultat de la **Recherche** est surligné dans le Mode Schéma et le résultat est également rapporté dans la fenêtre Rechercher dans schémas. Dans cette fenêtre, vous pouvez cliquer sur un résultat pour marquer cet item dans le Mode Schéma.

Cliquer sur la commande **Tout trouver** pour fermer le dialogue Recherche et regrouper tous les résultats de la recherche dans la fenêtre Rechercher dans schémas.

Dialogue Remplacer

Dans le dialogue Remplacer (*capture d'écran ci-dessous*), cliquer sur la commande **Recherche** pour trouver l'occurrence suivante du terme de recherche par rapport à la sélection actuelle dans le Mode Schéma. Vous pouvez ensuite cliquer sur **Remplacer** pour remplacer cette occurrence.



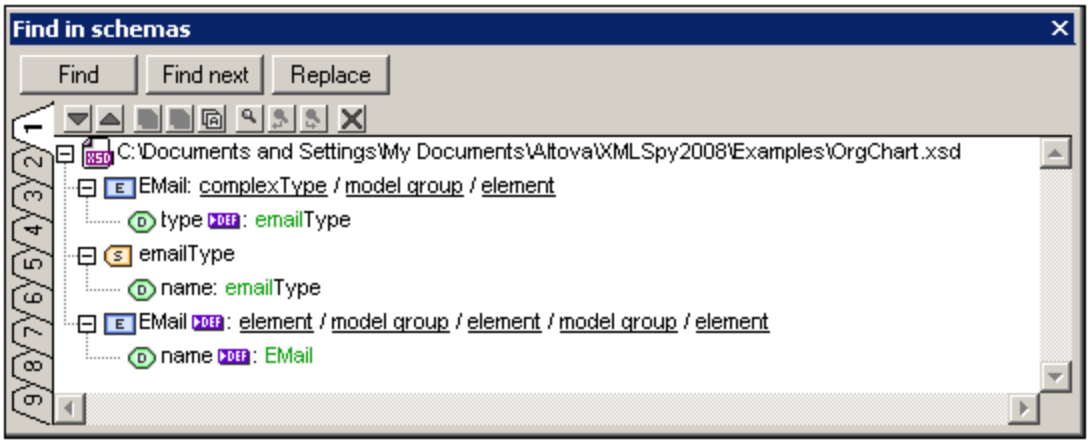
La commande **Tout trouver** ferme le dialogue Remplacer et regroupe tous les résultats de la recherche dans la fenêtre Rechercher dans schémas.

La commande **Tout remplacer** remplace toutes les occurrences du terme trouvé, ferme le dialogue Remplacer et regroupe tous les résultats de la recherche dans la fenêtre Rechercher dans schémas.

Noter : les expressions régulières ne sont pas prises en charge dans le champ Remplacer.

6.8.6 Résultats et informations

À chaque fois qu'une commande **Rechercher, Tout trouver, Remplacer** ou **Remplacer tout** est exécutée, les résultats de l'exécution de la commande sont affichés dans la fenêtre Rechercher dans Schéma (*capture d'écran ci-dessous*). Le terme recherché est affiché en vert ; (dans la capture d'écran ci-dessous, `email` était le terme recherché, sans restriction de casse spécifiée). Veuillez noter que l'emplacement du fichier de schéma est également donné.



Les commandes **Tout trouver** et **Remplacer tout** recensent toutes les occurrences trouvées dans le document.

Note : les touches **Chercher** et **Remplacer** situées en haut de cette fenêtre font apparaître le dialogue Rechercher et le dialogue Remplacer, respectivement. La touche **Chercher suivant** peut être utilisée pour trouver l'occurrence suivante du terme recherché.

Fonctions de la fenêtre Rechercher dans schémas

Les résultats sont affichés dans neuf onglets séparés (numérotés de 1 à 9). Pour garder les résultats d'une recherche dans un onglet, effectuer une nouvelle recherche, et comparer les résultats. Cliquer sur un résultat dans la fenêtre Rechercher dans Schéma pour faire apparaître et pour marquer le composant pertinent dans la fenêtre principale du Mode Schéma. Ainsi, vous pourrez chercher et parcourir rapidement pour trouver le composant recherché.

Les commandes de la barre outils de Rechercher dans schémas suivantes sont disponibles :

- Les icônes **Suivant** et **Précédent** sélectionnent, respectivement, les messages suivants et précédents en partant du message sélectionné actuellement.
- Les commandes **Copier Messages** copient, respectivement, le message sélectionné, le message sélectionné et ses messages enfants et tous les messages sur le presse-papiers.

© 2016-2022 Altova GmbH

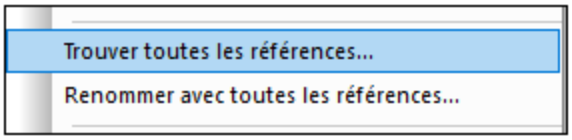
- Les commandes **Chercher** trouvent un texte dans la fenêtre Rechercher dans Schémas.
- La commande **Supprimer** supprime tous les messages dans l'onglet actif actuellement.

6.8.7 Chercher et renommer les Globaux

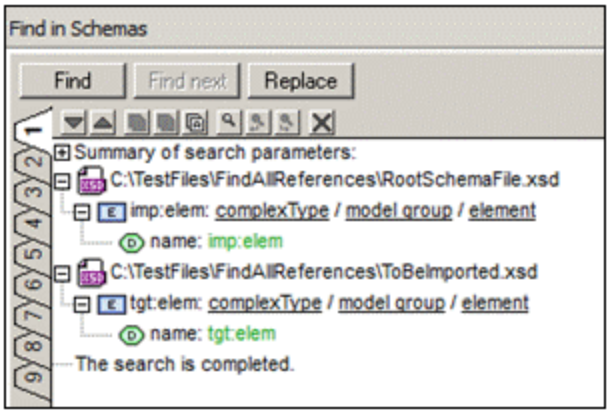
Les composants globaux nommés des Schémas XML peuvent être trouvés et renommés dans un fichier sélectionné et dans tous les fichiers de schéma liés au fichier sélectionné. Les composants globaux nommés sont tous des composants globaux sauf : Include, Import, Redefine, Annotation, Comment, et les composants PI.

Le processus fonctionne comme suit :

1. Dans l'aperçu de Schéma, sélectionner le composant global à trouver ou à renommer.
2. Dans le menu contextuel qui s'ouvre en cliquant avec la touche de droite sur le composant sélectionné, choisir la commande requise (**Trouver toutes les références** ou **Renommer avec toutes les références**).



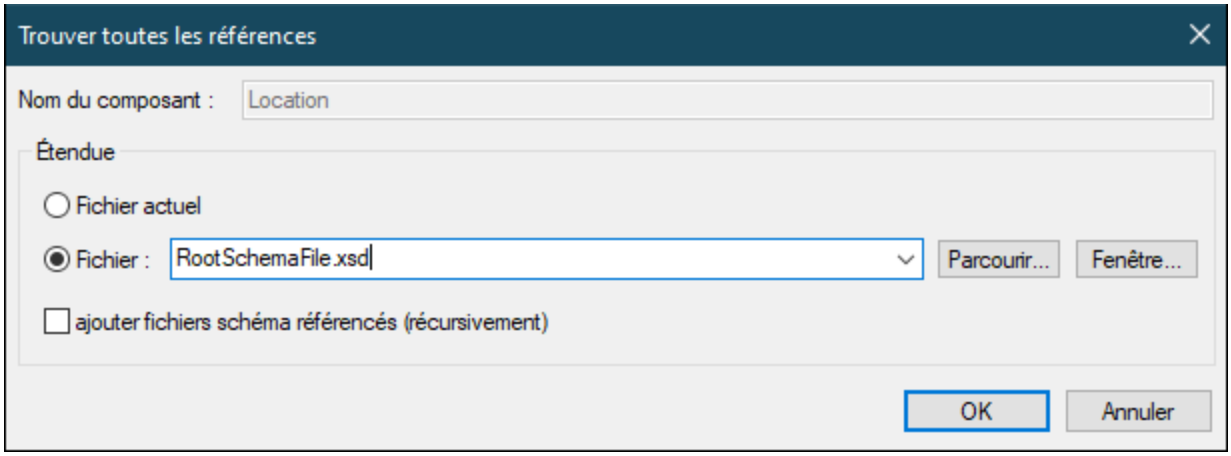
3. Dans le dialogue qui s'ouvre, sélectionner l'étendue de la recherche (ou renommer l'opération). Dans le cas d'une opération Renommer, saisir le nouveau nom du composant global.
4. Cliquer sur **OK** pour afficher les résultats de la recherche dans la fenêtre Rechercher dans schémas (*capture d'écran ci-dessous*).



Les emplacements de tous les fichiers dans lesquels des références au composant global sont trouvés, sont regroupés dans une liste (*voir capture d'écran ci-dessus*). Tous les composants renommés qui ont été trouvés sont aussi regroupés dans une liste.

Trouver toutes les références

Pour ouvrir le dialogue Trouver toutes les références (*capture d'écran ci-dessous*), procéder comme suit : (i) cliquer avec la touche de droite sur le composant global dans l'Aperçu de schéma, (ii) Dans le menu contextuel qui s'ouvre, sélectionner la commande **Trouver toutes les références**.



Le nom de composant global est affiché dans le champ *Nom de composant*, qui est grisé et ne peut pas être édité. Vous pouvez choisir si la recherche doit être effectuée dans le fichier actuel ou dans un autre fichier que vous pouvez parcourir (ou choisir dans une liste de fichiers ouverts). Vous pouvez aussi spécifier si vous souhaitez effectuer une recherche dans les fichiers liés (inclus, importés, redéfinis), en cochant la case *Ajouter fichiers Schéma référencé* située en bas du dialogue.

Renommer avec toutes les références

Afin de renommer un composant global, le cliquer avec la touche de droite et choisir **Renommer avec toutes les références** depuis le menu contextuel qui s'ouvre. Le dialogue Renommer avec toutes les références s'ouvre (*capture d'écran ci-dessous*).



Le nouveau nom que vous souhaitez donner au composant global sélectionné doit être saisi dans le champ de saisie *Nom de composant*. Vous pouvez choisir si vous souhaitez effectuer la recherche et le renommage dans le fichier actuel ou dans un autre fichier que vous pouvez parcourir (ou sélectionner depuis une liste de fichiers ouverts). Vous pouvez aussi spécifier si vous souhaitez parcourir les fichiers liés (inclus, importés, redéfinis), en cochant la case *Ajouter fichiers Schéma référencé* située en bas du dialogue.

© 2016-2022 Altova GmbH

7 XSLT

Site Internet Altova : Éditeur XSLT
Cette section concernant XSLT est organisée dans les sections suivantes :

- Éditer des documents XSLT : décrit la prise en charge d'édition pour les documents XSLT dans XMLSpy
- Traitement XSLT : montre les différents moyens d'effectuer des transformations XSLT dans la GUI XMLSpy avec des moteurs de votre choix. Cette section explique aussi les paramètres XSLT importants dans XMLSpy.
- XSL Outline : décrit les fenêtres XSL Outline et XSL Info, qui, ensemble, proposent un moyen très efficace de consulter, parcourir et gérer une collection de fichiers XSLT.

Évaluation XPath

Lorsqu'un document XML est actif, vous pouvez utiliser la fenêtre XPath/XQuery pour évaluer les expressions XPath. Il s'agit d'une fonction très utile qui permet de contrôler rapidement comment une expression XPath sera évaluée. Saisir une expression XPath et spécifier si elle doit être évaluée par rapport à la racine du document ou par rapport à un nœud contextuel sélectionné dans le document XML. Le résultat de l'évaluation sera affiché immédiatement dans la fenêtre XPath/XQuery. Vous trouverez une description de l'utilisation de la fenêtre XPath/XQuery dans la section GUI et Environnement | Fenêtre XPath/XQuery.

Profileur et Débogueur XSLT

XMLSpy contient également un Profileur XSLT et Débogueur XSLT pour vous aider à créer rapidement des feuilles de style XSLT de manière correcte et efficace. Ces deux fonctions sont décrites dans la section Débogueur XSLT et XQuery.

Fonctions XSLT supplémentaires

Vous trouverez des informations supplémentaires et détaillées concernant les différentes fonctions décrites dans cette section en vous rendant dans les descriptions des commandes de menu pertinentes (dans la section Références de l'utilisateur).

Moteurs Altova XSLT

Pour plus de détails concernant la manière dont sont implémentés les Moteurs Altova XSLT 1.0, 2.0 et 3.0, voir Informations Moteur XSLT et XQuery dans les Annexes.

RaptorXML pour la ligne de commande et le traitement en lot

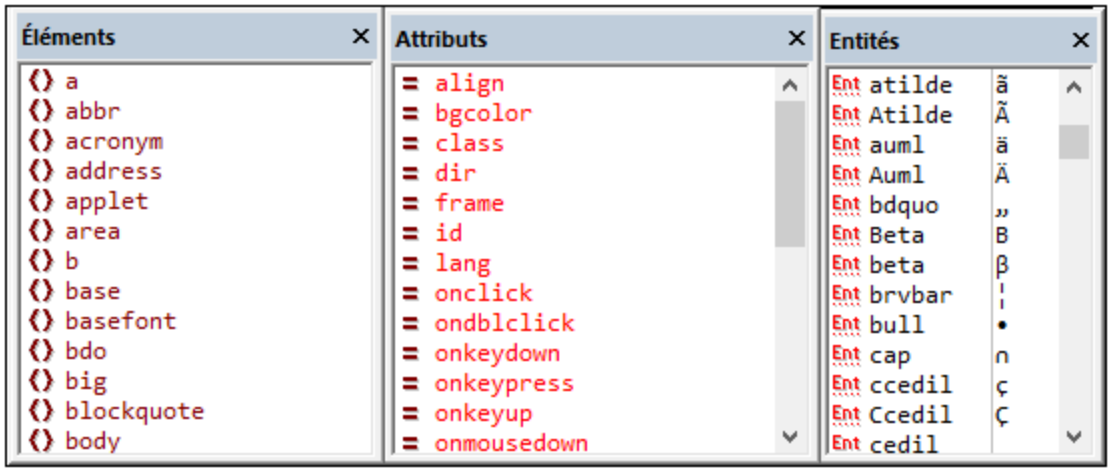
La GUI XMLSpy permet un traitement par lot par le biais des fonctions des projets. En revanche, si vous êtes à la recherche d'une solution plus flexible, nous vous recommandons d'essayer le produit RaptorXML d'Altova, qui permet une validation XML rapide, une transformation XSLT et une fonction d'exécution XQuery. RaptorXML est idéal si vous souhaitez effectuer des transformations XSLT depuis la ligne de commande ou à partir de traitement de lot.

7.1 Documents XSLT

Les documents XSLT 1.0, 2.0 et 3.0 peuvent être édités dans le Mode Texte et le Mode Grille, et ils sont édités comme tout autre document XML dans le Mode Texte et le Mode Grille. Le mode par défaut dans lequel un document XSLT est ouvert peut être défini dans la section Types de fichier du dialogue Options.

Assistants à la saisie

Les assistants à la saisie sont disponibles pour les éléments, les attributs et les entités. L'information concernant les items affichés dans les assistants à la saisie est intégrée dans XMLSpy, et n'est pas dépendante des références contenues dans le document XSLT.



Veuillez noter les points suivants :

1. Si un nouveau document XSLT est créé par le biais du dialogue Créer un nouveau document (**Fichier | Nouveau**), alors les éléments et attributs XSLT appropriés (XSLT 1.0, XSLT 2.0 ou XSLT 3.0, selon le type de document créé) sont chargés dans les assistants à la saisie. De plus, les éléments et attributs HTML sont chargés ainsi que les ensembles d'entité HTML 4.0, Latin-1, caractères spéciaux et les symboles.
2. Si un document XML est créé par le biais du dialogue Créer un nouveau document (**Fichier | Nouveau**) et est doté d'un contenu XSLT, aucun item d'assistant à la saisie n'est disponible sauf pour les entités de caractère XML.
3. Si un document XSLT créé en tant qu'un document XSLT est ouvert par le biais du dialogue Créer un nouveau document (**Fichier | Nouveau**), alors les assistants à la saisie seront comme dans le point 1 ci-dessus.
4. Si un document XSLT qui **n'a pas** été créé en tant que document XSLT est ouvert par le biais du dialogue Créer un nouveau document (**Fichier | Nouveau**), alors les assistants à la saisie seront comme dans le point 1 ci-dessus. De plus, les éléments et les attributs XSL-FO seront recensés dans les assistants à la saisie du Mode Texte.
5. Les préfixes des éléments dans l'assistant à la saisie Éléments sont comme suit et sont invariables : préfixe `xsl:` pour les éléments XSLT ; aucun préfixe pour les éléments HTML ; préfixe `fo:` pour les éléments XSL-FO. Par conséquence, pour utiliser les assistants à la saisie, les déclarations de l'espace de noms dans le document XSLT doivent définir des préfixes qui correspondent aux préfixes intégrés affichés dans les assistants à la saisie.

© 2016-2022 Altova GmbH

Saisie semi-automatique

Dans le Mode Texte, la saisie semi-automatique est disponible dans une fenêtre popup pendant que vous saisissez le texte. Le premier item correspondant le mieux au texte saisi est marqué dans la liste du popup. Lorsqu'un **élément** est saisi, une liste d'éléments s'affiche proposant une correspondance la plus proche possible, les correspondances étant classées par ordre alphabétique. De même, lorsqu'un **attribut** est saisi, une liste d'attributs applicables s'affiche. Les items contenus dans la liste sont déterminés selon les règles décrites dans la section précédente concernant les assistants à la saisie.

Édition intelligente XPath

Dans les emplacements dans le document XSLT où les expressions XPath peuvent être saisies (par exemple, dans la valeur d'un attribut `select`, dans les modèles de valeur d'attribut et les modèles de valeur XSLT 3.0), les fonctions suivantes sont disponibles.

- Coloration syntaxique pour les structures XPath, y compris des crochets correspondants pendant la saisie.
- Une infobulle d'aide est placée sur une fonction XPath. Elle contient des informations concernant la fonction.
- Les fonctions et les axes XPath sont suggérées dans des popups pendant votre saisie Vous pouvez vous déplacer dans la liste des suggestions avec les curseurs **Haut/Bas**. Si l'item qui est marqué dans le popup est une fonction, alors les informations concernant la fonction (sa signature) sont affichées dans un popup supplémentaire.
- Si un fichier XML a été attribué dans la fenêtre info, alors les éléments et les attributs du fichier XML seront aussi disponibles dans le popup.

Valider des documents XSLT

Le document XSLT peut être validé par rapport au schéma XSLT intégré dans XMLSpy (cliquer sur **XML | Valider (F8)**). Le schéma intégré correct est automatiquement sélectionné selon que le document soit un XSLT 1.0, XSLT 2.0 ou XSLT 3.0 (spécifié dans l'attribut `version` de l'élément `xsl:stylesheet`).

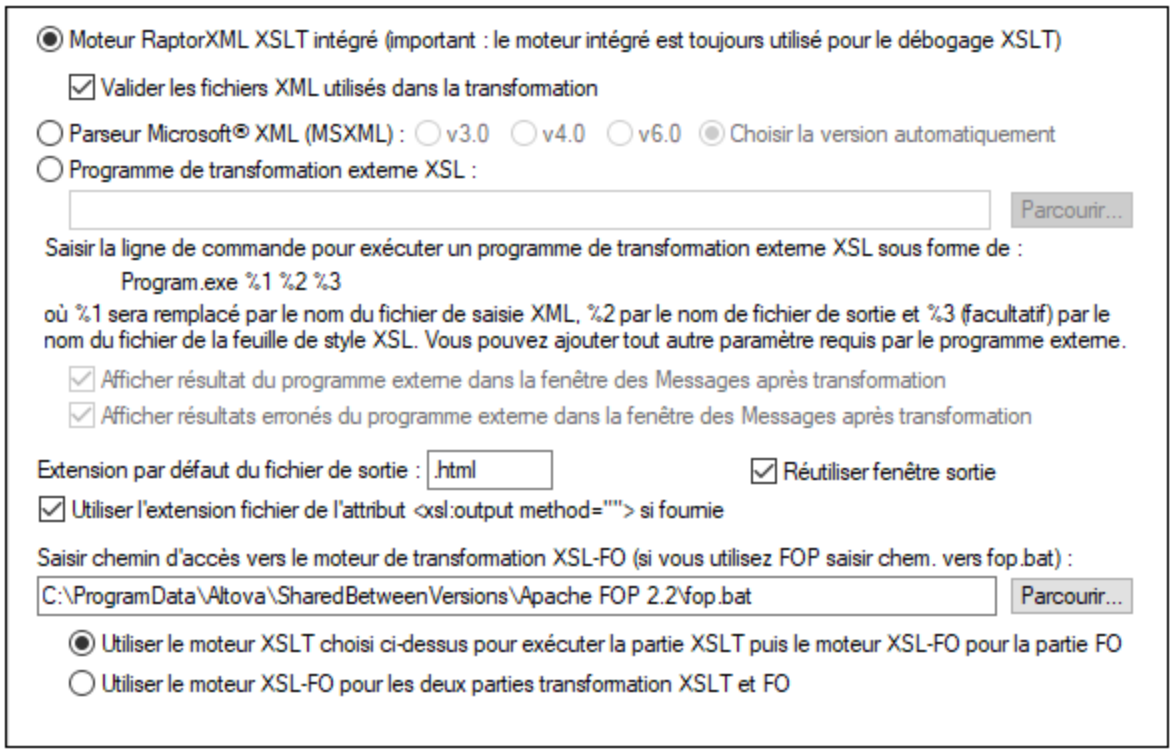
7.2 Traitement XSLT

Dans la GUI XMLSpy, deux types de transformations XSLT sont disponibles :

- La commande **XSL/XQuery | Transformation XSL (F10)** est utilisée pour les transformations XML directes avec une feuille de style XSLT pour obtenir les résultats spécifiés et décrits dans les feuilles de style.
- La commande **XSL/XQuery | Transformation XSL-FO** est utilisée : (i) pour les transformations de XML en FO en PDF en deux étapes et (ii) pour les transformations en une étape de FO en PDF.

Spécifier le processeur XSLT pour la transformation

Le moteur XSLT qui sera utilisé pour les transformations est spécifié dans la section XSL du dialogue Options (*capture d'écran ci-dessous*).



Les options disponibles sont expliquées dans la section Référence de l'utilisateur. Le moteur spécifié dans la section XSL sera utilisé pour toutes les transformations XSLT. Veuillez noter que pour la transformation XSL-FO, une option de moteur XSLT supplémentaire est disponible : le moteur XSLT qui est contenu dans un paquet avec certains processeurs FO. Pour sélectionner cette option, sélectionnez le bouton radio correspondant dans le bas de la section XSL (*voir la capture d'écran ci-dessus*).

Spécifier le processeur FO

Le processeur FO qui sera utilisé pour les transformations de FO en PDF est spécifié dans la fenêtre de texte située en bas de la section XSL du dialogue Options (*capture d'écran ci-dessus*).

© 2016-2022 Altova GmbH

XSLT 1.0, 2.0, 3.0 et les moteurs XSLT d'Altova

La version XSLT d'une feuille de style est spécifiée dans l'attribut `version` de l'élément `xsl:stylesheet` (ou `xsl:transform`). XMLSpy contient les moteurs built-in Altova XSLT 1.0, Altova XSLT 2.0 et Altova XSLT 3.0, et le moteur approprié est sélectionné conformément à la valeur de l'attribut de version (`1.0` ou `2.0` ou `3.0`).

Transformation XSLT

La commande de **Transformation XSLT (F8)** peut être utilisée dans les scénarios suivants :

- Pour transformer un document XML qui est activé dans la GUI et auquel un document XSLT a été attribué. Si aucun document XSLT n'est attribué, vous serez invité à procéder à une attribution lorsque vous cliquerez sur la commande **Transformation XSLT (F8)**.
- Pour transformer un document XSLT qui est activé dans la GUI. En cliquant sur la commande **Transformation XSLT (F8)**, vous serez invité à indiquer le fichier XML que vous souhaitez traiter avec la feuille de style XSLT active.
- Pour transformer des dossiers et des fichiers de projet. Cliquer avec la touche de droite sur le dossier ou le fichier de projet et choisir la commande.

Back-mapping

En activant la fonction Back-mapping, les transformations XSLT seront effectuées de manière à ce que le document de résultat puisse être remappé sur les documents XSLT+XML d'origine. Si vous cliquez sur un nœud dans le document de résultat, alors l'**instruction XSLT** ***et*** les **données source XML** qui généreront ce nœud de résultat particulier seront en surbrillance. De plus, si vous cliquez sur une instruction XSLT ou un nœud de données XML, alors les nœuds correspondants des deux autres documents sont en surbrillance. Voir la commande **XSL/XQuery | Enable Back-Mapping** pour les détails.

Transformation XSL:FO

La commande de **Transformation XSL:FO** peut être utilisée dans les scénarios suivants :

- Pour transformer un document XML qui est activé dans la GUI et auquel un document XSLT a été attribué. Le document XML sera d'abord transformé en FO avec le moteur XSLT spécifié. Pour transformer un document FO en PDF avec le processeur FO spécifié pour produire la sortie PDF. Si aucun document XSLT n'est attribué, vous serez invité à procéder à une attribution lorsque vous cliquerez sur la commande **Transformation XS:FO**.
- Pour transformer un document FO en PDF utilisant le processeur FO spécifié.
- Pour transformer un document XSLT qui est activé dans la GUI. En cliquant sur la commande **Transformation XSL:FO**, vous serez invité à indiquer le fichier XML que vous souhaitez traiter avec la feuille de style XSLT active.
- Pour transformer des dossiers et des fichiers de projet. Cliquer avec la touche de droite sur le dossier ou le fichier de projet et choisir la commande.

Pour une description des options contenus dans le dialogue de sortie XSL:FO, voir la section Référence de l'utilisateur.

Paramètres pour XSLT

Si vous utilisez les moteurs XSLT d'Altova, les paramètres XSLT peuvent être stockés dans un dialogue GUI pratique. Tous les paramètres stockés sont passés dans le document XSLT à chaque fois que vous effectuez

une transformation. Pour plus d'informations, voir la description de la commande Paramètres XSLT / Variable XQuery.

Traitement par lot avec RaptorXML

RaptorXML est une application autonome qui contient le validateur XML, les moteurs XSLT et les moteurs XQuery d'Altova les plus récents. Il peut être utilisé à partir de la ligne de commande, par le biais d'une interface COM, dans des programmes Java et dans les applications .NET pour valider les documents XML, transformer les documents XML avec des feuilles de style XSLT et exécuter des documents XQuery.

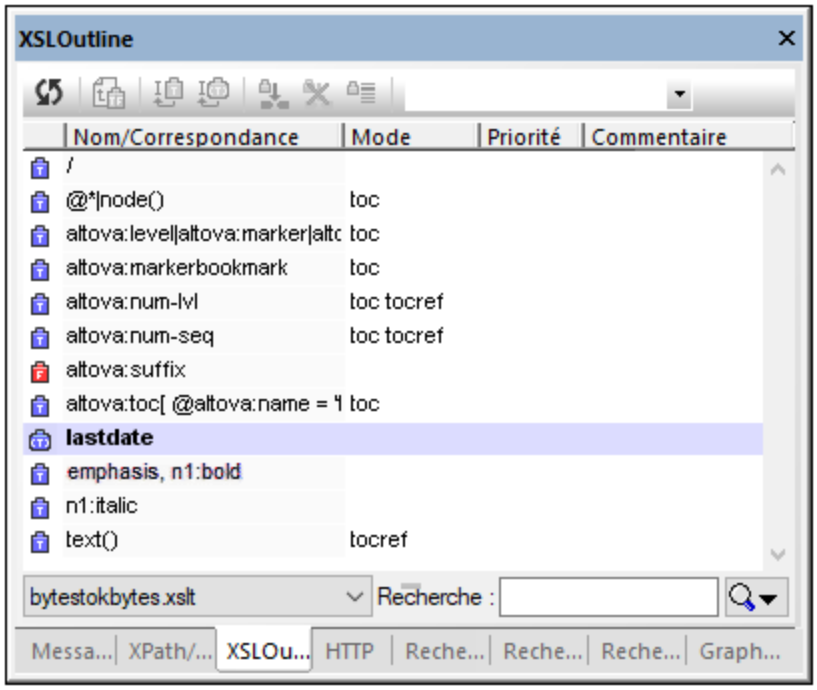
Les tâches de transformation XSLT peuvent donc être automatisées avec l'utilisation de RaptorXML. Par exemple, vous pouvez créer un fichier batch qui appelle RaptorXML pour transformer un ensemble de documents. Voir la documentation RaptorXML pour tout détail.

© 2016-2022 Altova GmbH

7.3 XSL Outline

Lorsqu'un document XSLT est le document actif dans XMLSpy, l'information concernant la structure du document est affichée dans la fenêtre XSL Outline et l'information concernant les fichiers liés au document XSLT actif est affichée dans l'onglet XSLT de la fenêtre Info (qui est affichée uniquement lorsqu'un document XSLT est le document actif dans XMLSpy). De plus, ces deux fenêtres vous permettent de recourir à un certain nombre de commandes vous permettant de faciliter l'édition du document XSLT et de gérer les fichiers qui y sont liés.

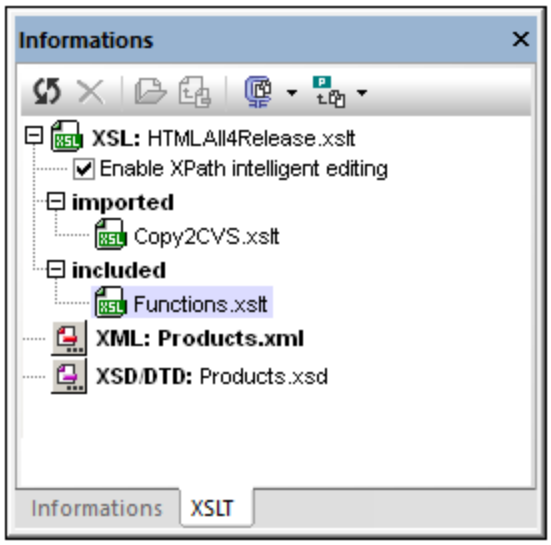
Dans la **fenêtre XSL Outline** (*capture d'écran ci-dessous*), vous pouvez :



- Consultez les modèles et les fonctions dans le document XSLT actif et dans tous les documents XSLT importés et inclus.
- Trier les modèles et les fonctions sur la base de leur nom ou faire correspondre les expressions, le mode, la priorité ou les commentaires.
- Cherchez des modèles spécifiques sur la base de leurs noms/expressions.
- Utilisez XSL Outline pour parcourir vers le modèle correspondant dans le document XSLT.
- Insérer rapidement des appels vers des modèles nommés.
- Définissez un modèle de nom sélectionné comme point d'entrée pour les transformations.

Voir la section fenêtre XSL Outline pour les détails.

Dans l'**onglet XSLT de la fenêtre d'info** (*voir la capture d'écran ci-dessous*), vous pouvez faire les choses suivantes :



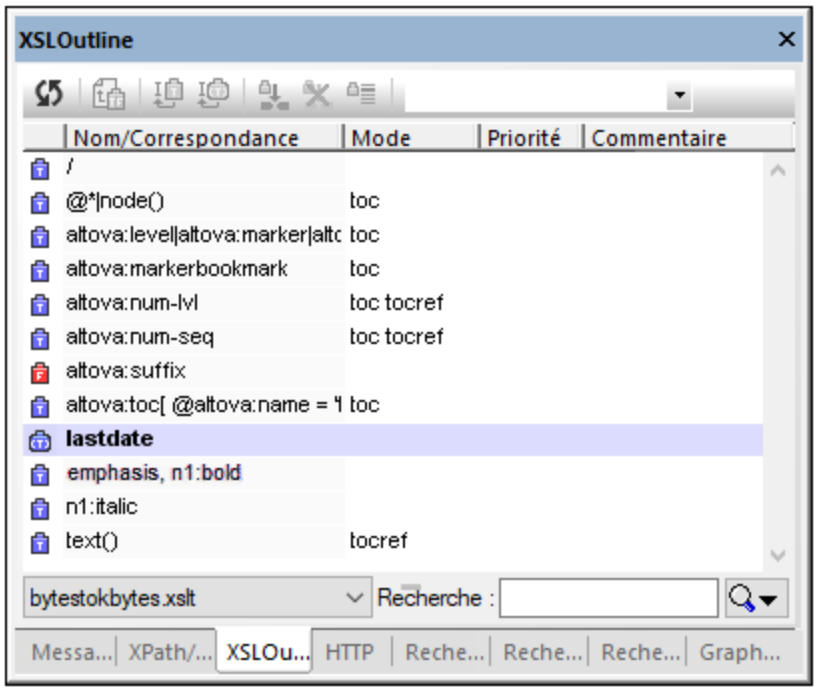
- Consulter les informations concernant tous les fichiers liés au document XSLT actif, comme les emplacements des fichiers importés et inclus.
- Définir un fichier XML pour une transformation avec le document XSLT actif. De même, le fichier de schéma (XSD/DTD) peut être configuré pour la validation du fichier XML sélectionné.
- Ouvrir un fichier lié depuis la fenêtre Info.
- Organiser rapidement tous les fichiers liés dans des projets XMLSpy.
- Comprimer tous les fichiers liés dans un emplacement défini par l'utilisateur.

La fenêtre XSL Outline et l'onglet XSLT de la fenêtre Info sont décrits en détail dans les sous-sections de cette section.

7.3.1 Fenêtre XSL Outline

Dans la fenêtre XSL Outline (*voir la capture d'écran*), tous les modèles et fonctions dans le document XSLT actif sont recensés. Les modèles sont indiqués par des icônes bleues (modèles sans un paramètre ; et modèles contenant des paramètres). Les fonctions sont indiquées par une icône rouge . Dans la liste de choix contenue dans la partie inférieure gauche de la fenêtre, vous pouvez sélectionner si tous les modèles et les fonctions recensées proviennent : (i) uniquement du document actif XSLT (comme dans la capture d'écran ci-dessous), ou (ii) du document XSLT actif et de toutes les feuilles de style incluses et importées.

© 2016-2022 Altova GmbH



Il y existe deux types de modèles : (i) modèles nommés et les (ii) modèles qui correspondent à une expression XPath. Chaque modèle est recensé avec :

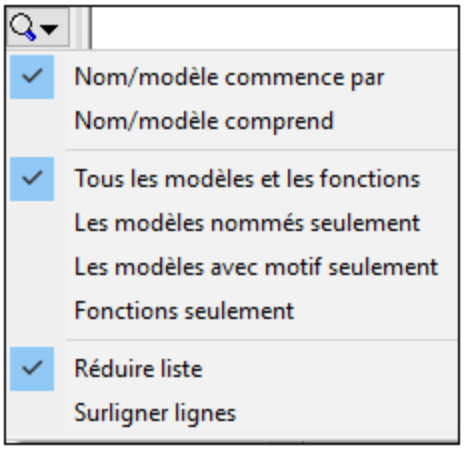
- Son nom (si le modèle a un attribut `name`) et/ou une expression XPath (si le modèle a un attribut `match`). Si le modèle possède les deux, un attribut `name` et un attribut `match`, ils seront recensés tous les deux, avec la valeur de l'attribut `name` en premier : `namevalue, matchvalue` (*voir le modèle nommé* `bold` *dans la* capture d'écran ci-dessus).
- Son mode, le cas échéant. Veuillez noter qu'un modèle peut avoir plus d'un mode (*voir capture d'écran ci-dessus*).
- Sa priorité, le cas échéant ;
- Le commentaire qui précède directement le modèle ou la fonction, le cas échéant.

Les fonctions dans la feuille de style sont recensées par leur nom. Les fonctions n'ont ni mode ni priorité.

Opérations

Les opérations suivantes peuvent être effectuées dans la fenêtre XSL Outline :

- *Filtrer :* La liste affichée dans la fenêtre peut être filtrée pour afficher l'une des choses suivantes : (i) tous les modèles et les fonctions (le paramètre par défaut à chaque fois que XMLSpy est lancé) ; (ii) les modèles nommés uniquement ; (iii) les modèles expression-XPath uniquement ; (iv) les fonctions uniquement. Pour sélectionner le filtre requis, cliquer sur la flèche descendante à droite de la fenêtre de Recherche (*capture d'écran ci-dessous*), et choisir le filtre requis (le deuxième groupe de commandes dans le menu). Le filtre sélectionné s'applique immédiatement et reste en vigueur jusqu'à ce qu'il soit modifié ou jusqu'à ce que XMLSpy soit fermé.



- *Trier et localiser :* Chaque colonne peut être triée de alphabétiquement en cliquant sur l'en-tête de colonne. Chaque clic suivant inverse l'ordre de tri précédent. Une fois qu'une colonne a été triée de cette manière, si vous sélectionnez un item dans la liste puis que vous saisissez rapidement un terme depuis la colonne triée, le premier item dans cette liste qui contient le terme saisi sera marqué. De cette manière, vous pouvez rapidement aller sur les modèles d'un nom/expression, mode ou priorité particulier.
- *Recherche :* Saisir dans la fenêtre de Recherche (en bas à droite) le nom ou l'expression XPath que vous souhaitez chercher. Les résultats de recherche s'affichent pendant la saisie. Les options de recherche suivantes sont disponibles dans la liste déroulante de recherche (*capture d'écran ci-dessous*) : (i) si le nom ou l'expression commence soit avec ou contienne le terme recherché (le premier groupe de commande dans le menu) ; l'option commencer-avec est l'option par défaut à chaque fois que XMLSpy est démarrée ; (ii) si les résultats de recherche doivent être affichés en tant que liste réduite ou marqués (le troisième groupe de commandes dans le menu) ; l'option liste réduite est le défaut à chaque fois que XMLSpy est ouvert Ces sélections sont appliquées immédiatement et restent en vigueur jusqu'à ce qu'elles soient changées ou que XMLSpy soit fermé.
- *Recharger :* Une fois que la feuille de style a été modifiée, cliquez sur l'icône **Synchroniser** dans la barre d'outils de la fenêtre pour mettre à jour l'XSL outline.
- *Aller à l'item :* lorsqu'un modèle ou une fonction a été sélectionnée dans la fenêtre XSL Outline, cliquer sur l'icône **Aller à la définition** icon dans la barre d'outils de la fenêtre pour marquer le modèle ou la fonction dans le document dans le Mode Design. En alternative, double-cliquez sur une saisie pour y aller.
- *Actions modèles nommés :* deux groupes d'actions peuvent être effectués concernant des modèles nommés : (i) les appels au modèle nommé (avec `xsl:call-template`) peuvent être insérés dans la feuille de style au niveau du point d'insertion du curseur ; et (ii) un modèle nommé peut être défini en tant que le point d'entrée pour une transformation. Les commandes pour ces actions sont effectuées par le biais d'icônes dans la barre d'outils et sont décrites ci-dessous.

Mode de modèle pour la transformation

La liste de choix dans la barre à outils, appelé *Configurer mode pour transformation*, recense (i) tous les modes dans la feuille de style, plus (ii) une entrée vide (qui sélectionne le mode par défaut) et, dans le cas des feuilles de style XSLT 3.0, (iii) le mode `#unnamed`. En sélectionnant un mode depuis la liste déroulante, vous configurez le mode sélectionné en tant que le mode pour la transformation. Le mode `#unnamed` (pour toutes les versions XSLT) s'applique à tous les modèles qui n'ont pas d'attribut `mode`.

© 2016-2022 Altova GmbH

Dans le cas des feuilles de styles XSLT 1.0 et XSLT2.0, le mode par défaut est le mode `#unnamed`. Donc en sélectionnant l'entrée vide, vous sélectionnez le mode par défaut (qui est le mode `#unnamed` et qui s'applique donc à tous les modèles sans attribut `mode`).

Dans les feuilles de style XSLT 3.0, l'élément de niveau supérieur `xslt` peut avoir un attribut `default-mode`, qui contient le mode par défaut pour la transformation. Si, dans la liste de choix *Configurer mode pour transformation*, l'entrée vide (mode par défaut) est sélectionné, le mode spécifié dans l'attribut `default-mode` sera utilisé en tant que le mode de transformation. Si un mode `#unnamed` est sélectionné dans la liste de choix, la transformation sera appliquée à tous les modèles avec un mode non nommé, c.-à.-d. à des modèles sans attribut de mode.

Note : un modèle peut recevoir une valeur de mode de `#all` pour le rendre applicable à tous les modes.

Modèles nommés

Lorsqu'un modèle nommé est sélectionné, une ou plusieurs commandes dans la barre à outils de la fenêtre liés aux modèles nommés sont activées (*capture d'écran ci-dessous*).

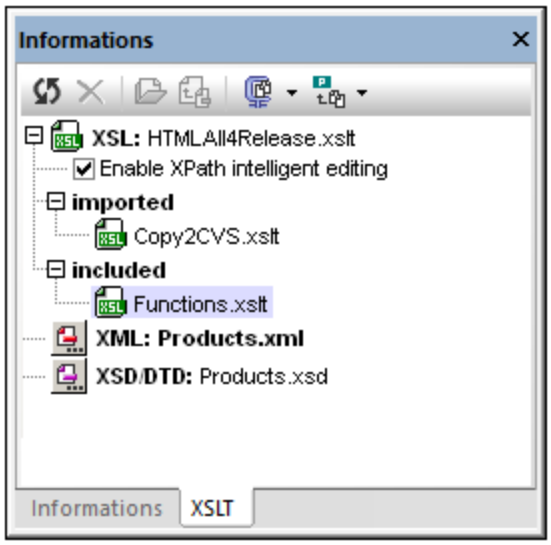


Les commandes dans la barre outils (*capture d'écran ci-dessous*) sont les suivantes, de gauche à droite :

- *Insérer xsl:call-template :* Cette commande devient active lorsque le modèle nommé est sélectionné dans la fenêtre Outline XSL. La commande insère un élément `xsl:call-template` au niveau du point d'insertion du curseur dans la feuille de style. L'attribut `name` de l'élément `xsl:call-template` qui est inséré dans la feuille de style reçoit une valeur qui sera la valeur de l'attribut `name` du modèle nommé sélectionné. Cela fait de `xsl:call-template` un appel au modèle nommé sélectionné.
- *Insérer xsl:call-template avec le paramètre :* Cette commande devient active lorsque le modèle nommé ayant un ou plusieurs éléments enfant `xsl:param` sélectionnés dans la fenêtre Outline XSL. Comme c'est le cas avec la commande **Insert xsl:call-template**, la commande insère un élément `xsl:call-template`, mais dans ce cas avec un élément enfant `xsl:with-param` correspondant pour chaque élément enfant `xsl:param` du modèle nommé sélectionné. Les noms de l'élément inséré `xsl:call-template` et de ses éléments enfants `xsl:with-param` correspondent aux noms du modèle nommé sélectionné et ses enfants `xsl:param`.
- *Définir le modèle nommé sélectionné comme point d'entrée pour la transformation :* Lorsqu'un modèle nommé est défini en tant que point d'entrée pour une transformation, les transformations exécutées dans XMLSpy commencent avec ce modèle nommé. Dans la fenêtre XSL Outline, les modèles nommés sont indiqués en police grasse (voir capture d'écran au début de cette section).
- *Supprimer le modèle nommé sélectionné comme point d'entrée pour la transformation :* Devient actif une fois que le modèle nommé a été défini comme point d'entrée pour les transformations.
- *Sautez vers le modèle nommé sélectionné comme point d'entrée pour les transformations :* Devient actif une fois que le modèle nommé a été défini comme point d'entrée pour les transformations. Lorsque l'attention dans la fenêtre XSL Outline se trouve dans un autre point que le modèle nommé défini en tant que le point d'entrée des transformations, cliquer sur cette icône pour marquer le modèle nommé dans la fenêtre XSL Outline, afin d'en accélérer l'accès.

7.3.2 Fenêtre Info

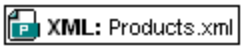
L'onglet XSLT de la fenêtre Info est affiché uniquement quand un document XSLT document est le document actif dans XMLSpy. Il affiche tous les fichiers XSLT importés et inclus associés au document XSLT actif. Vous pouvez aussi choisir un fichier XML pour transformer avec le XSLT lorsque la transformation est lancée si le XSLT est le document actif.



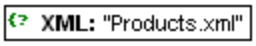
Les fichiers suivants sont affichés dans l'onglet XSLT de la fenêtre Info :

- *Fichiers XSLT :* Tous les fichiers XSLT importés et inclus sont recensés (*voir la capture d'écran ci-dessus*). L'emplacement du fichier XSD attribué est affiché dans une fenêtre pop up lorsque le curseur de la souris est placé sur le fichier. Double-cliquer sur un fichier importé ou inclus ou le sélectionner, puis cliquer sur l'icône **Ouvrir**dans la barre d'outils de la fenêtre Info, ouvre le fichier dans une nouvelle fenêtre. L'icône **Aller à l'emplacement Inclure/Importer** dans la barre d'outils met la déclaration inclure/importer en surbrillance dans le document XSLT actif.

- *Fichier XML :* Un fichier XML peut être attribué à la feuille de style XSLT active pour les transformations. L'emplacement du fichier XML attribué est affiché dans un pop-up quand le curseur de la souris est placé au-dessus du fichier. Si un fichier XML est spécifié et que la commande de menu **XSL/XQuery | Transformation XSL (F10)** est cliqué, une transformation est exécutée dans le fichier XML défini utilisant le document XSLT actif comme feuille de style. Le fichier XML peut être sélectionné en cliquant sur l'icône **XML** puis en parcourant les fichiers ; le fichier sélectionné est affiché en police grasse. En alternative, le fichier XML peut être assigné via le dialogue des propriétés de projet (*Entrée XML pour XSL/XQuery/Update transformation*) ou par le biais de l'instruction de traitement dans le document XSLT : `<?altova_samplexml "Products.xml"?>`. Dans tous les cas, le fichier XML sera affiché dans la fenêtre Info avec l'icône pertinente :

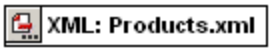
  

  attribué via le dialogue Propriétés de projet

  attribué via une instruction de traitement dans le document XSLT

© 2016-2022 Altova GmbH



attribué en cliquant sur l'icône XML et en parcourant le fichier requis ; l'entrée est en police grasse

Au cas où plus d'une des attributions mentionnées ci-dessus existe, la priorité de sélection est la suivante : (i) projet; (ii) instruction de traitement ; (iii) parcouru par l'utilisateur. Le fichier de schéma peut être ouvert en double-cliquant dessus ou en le sélectionnant puis en cliquant sur l'icône **Ouvrir** de la barre d'outils.

- *Fichier XSD/DTD :* Si le fichier XML sélectionné a une référence à un schéma (Schéma XML ou DTD), alors ce fichier de schéma est affiché dans l'entrée XSD/DTD. En alternative, comme pour le fichier XML, le fichier de schéma peut être sélectionné par le biais du dialogue Propriétés de projet (*Validation*) ou en cliquant sur l'icône **XSD/DTD** et en cherchant le fichier de schéma requis. Si le fichier de schéma est sélectionné par le biais du dialogue Propriétés de projet, une icône Projets est affichée à côté de l'entrée, sinon, l'icône cliquable **XSD/DTD** est affichée avec l'entrée de fichier soit dans une police normale (lorsque le schéma est référencé depuis le fichier XML) ou dans une police grasse (schéma parcouru par l'utilisateur par le biais de l'icône **XSD/DTD**). Si le fichier de schéma est attribué par plus d'une méthode, l'ordre de priorité sera le suivant : (i) projet; (ii) parcouru par l'utilisateur ; (iii) référence dans le document XML. L'emplacement du fichier XSD attribué est affiché dans un pop-up quand le curseur de la souris est placé au-dessus du fichier. Le fichier de schéma peut être ouvert en double-cliquant dessus ou en le sélectionnant puis en cliquant sur l'icône **Ouvrir** de la barre d'outils.

Note : Si un fichier XML ou XSD/DTD est sélectionné par le biais du dialogue Propriétés de Projet et si vous souhaitez effacer cette sélection, vous devrez vous rendre dans le dialogue Propriétés de Projet et supprimer la configuration s'y trouvant. Si la sélection a été effectuée en parcourant les fichiers avec les icônes **XML** ou **XSD/DTD**, pour supprimer cette configuration, sélectionner le fichier et cliquer sur l'icône **Supprimer** dans la barre d'outils de la fenêtre Info.

Options

Édition intelligente XPath : Si un fichier XML a été assigné, la structure du document XML est connue et l'édition Xpath intelligente sera élargie aux éléments et attributs. Aux emplacements dans le document XSLT où une expression XPath peut être saisie, des éléments et attributs disponibles seront affichés dans un popup. Cette option est activée par défaut. Pour désactiver l'édition XPath intelligente, décocher la case. Les paramètres sont enregistrés séparément pour chaque fichier XSLT lorsque le fichier est fermé et sera utilisé à chaque fois que le fichier sera ouvert.

Icônes de barre d'outils

Les icônes de la barre d'outils de la fenêtre Info (*capture d'écran ci-dessous*) sont les suivantes, de gauche à droite :



- *Info de rechargement :* Met à jour la fenêtre d'info pour refléter les modifications faites dans le document XSLT.
- *Supprimer l'assignation XML/XSD :* Supprime une assignation XML ou XSD/DTD effectuée par l'utilisateur en parcourant avec les icônes XML ou XSD/DTD, respectivement. Sélectionner le fichier à supprimer et cliquer sur l'icône.
- *Ouvrir Document :* Cliquer dessus pour ouvrir le document sélectionné.
- *Aller à l'emplacement importer/inclure :* Lorsqu'un fichier importé ou inclus est sélectionné, cliquer sur cette icône met en surbrillance la déclaration importer ou inclure dans le document XSLT.

- *Zipper tous les documents locaux :* Zipper tous les documents listés dans la fenêtre Info vers un emplacement défini par l'utilisateur. En alternative, seuls les documents sélectionnés pourront être comprimés ; pour ce faire, sélectionner, dans le menu déroulant de cette icône, la commande **Comprimer les documents locaux sélectionnés**.
- *Ajouter fichier actif aux projets :* Ajoutez tous les fichiers au projet actuel. En alternative, seuls les documents sélectionnés pourront être ajoutés ; pour ce faire, sélectionner, dans le menu déroulant de cette icône, la commande **Ajouter les fichiers sélectionnés au projet**.

© 2016-2022 Altova GmbH

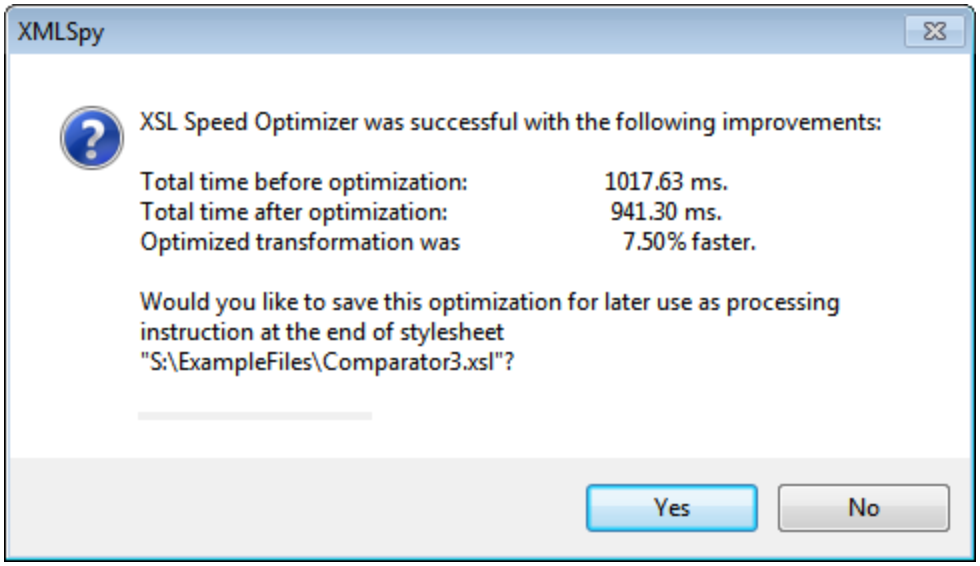
7.4 Optimiseur vitesse XSL

L'Optimiseur vitesse XSL (également appelé l'Optimiseur dans cette section) permet d'optimiser les feuilles de style XSLT de manière à ce que les transformations puissent être effectuées plus rapidement. L'Optimiseur fonctionne en exécutant la feuille de style XSLT pour être optimisé sur un document XML et en anlaysant la performance de la feuille de style. Une stratégie d'optimisation est dérivée de cette analyse et peut être enreigstrée avec la feuille de style XSLT (comme instruction de traitement à la fin de la feuille de style). La feuille de style optimisée peut être utilisée par la suite pour produire des transformations plus rapides.

Optimiser une feuille de style XSLT

Afin d'optimiser une feuille de style XSLT, il vous faudra, outre une feuille de style XSLT, un document XML qui servira d'échantillon. Cet ensemble de données doit être suffisamment volumineux pour que toutes les parties de la feuille de style XSLT soient utilisées et que la feuille de style puisse être analysée correctement. L'optimisation s'effectue comme suit :

1. Une fois que la feuille de style XSLT ou le document Word est activé, cliquer sur la commande de menu **XSL/XQuery | Optimiseur de vitesse XSL** ou cliquez sur l'icône de l'Optimiseur dans la barre d'outils principale.
2. Vous serez invité à choisir, selon qu'un document XSLT ou XML est actif, respectivement, un document XML ou une feuille de style XSLT. L'analyse démarre en cliquant sur **OK**. (si le document XSLT ou XML a déjà été attribué, cette étape peut être sautée ; l'analyse est lancée dès que la commande est invoquée.)
3. Si l'analyse d'optimisation a échoué, un message à cet effet sera affiché. (Les différentes possibilités d'échec d'une analyse d'optimisation sont décrites ci-dessous.) Si l'analyse réussit, un dialogue affichant les résultats de l'analyse s'affichera (*capture d'écran ci-dessous*).



Le dialogue vous permet d'enregistrer l'optimisation (instructions) dans la feuille de style XSLT (en tant qu'instruction de traitement à la fin de la feuille de style). Cliquer sur **Oui** pour enregistrer l'optimisation, **Non** pour l'abandonner. Lorsqu'une optimisation est enregistrée, elle écrase toute optimisation enregistrée précédemment.

La feuille de style optimisée peut maintenant être utilisée pour effectuer des transformations plus rapides.

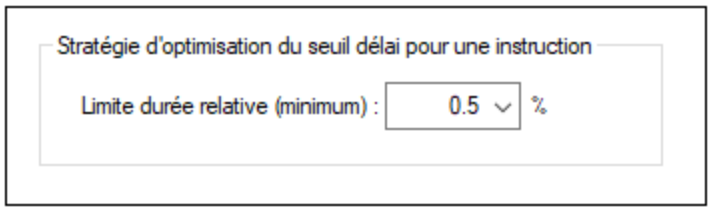
Possibilités d'échec d'une analyse d'optimisation

Si l'optimiseur de vitesse XSL n'est pas en mesure de dériver une optimisation, une (ou plusieurs) des raisons suivantes peuvent en être la cause :

- La feuille de style XSLT est déjà suffisamment efficace et n'a pas besoin d'être optimisée.
- L'ensemble de données XML soumis est trop petit pour être optimisé. Essayer à nouveau avec un document plus volumineux.
- Le seuil d'optimisation peut être trop élevé. Changer les seuils dans la section de 'Optimiseur de vitesse XSL du dialogue Options. *Voir ci-dessous*.
- Les optimisations pour cette structure XSLT spécifique ne sont pas disponibles dans l'Optimiseur. Veuillez contacter le support d'Altova.

Paramètres de l'optimiseur de vitesse XSL

Les paramètres de l'optimiseur sont réglés dans la section Optimiseur de vitesse XSL du dialogue Options (**Outils | Options,** capture d'écran ci-dessous).

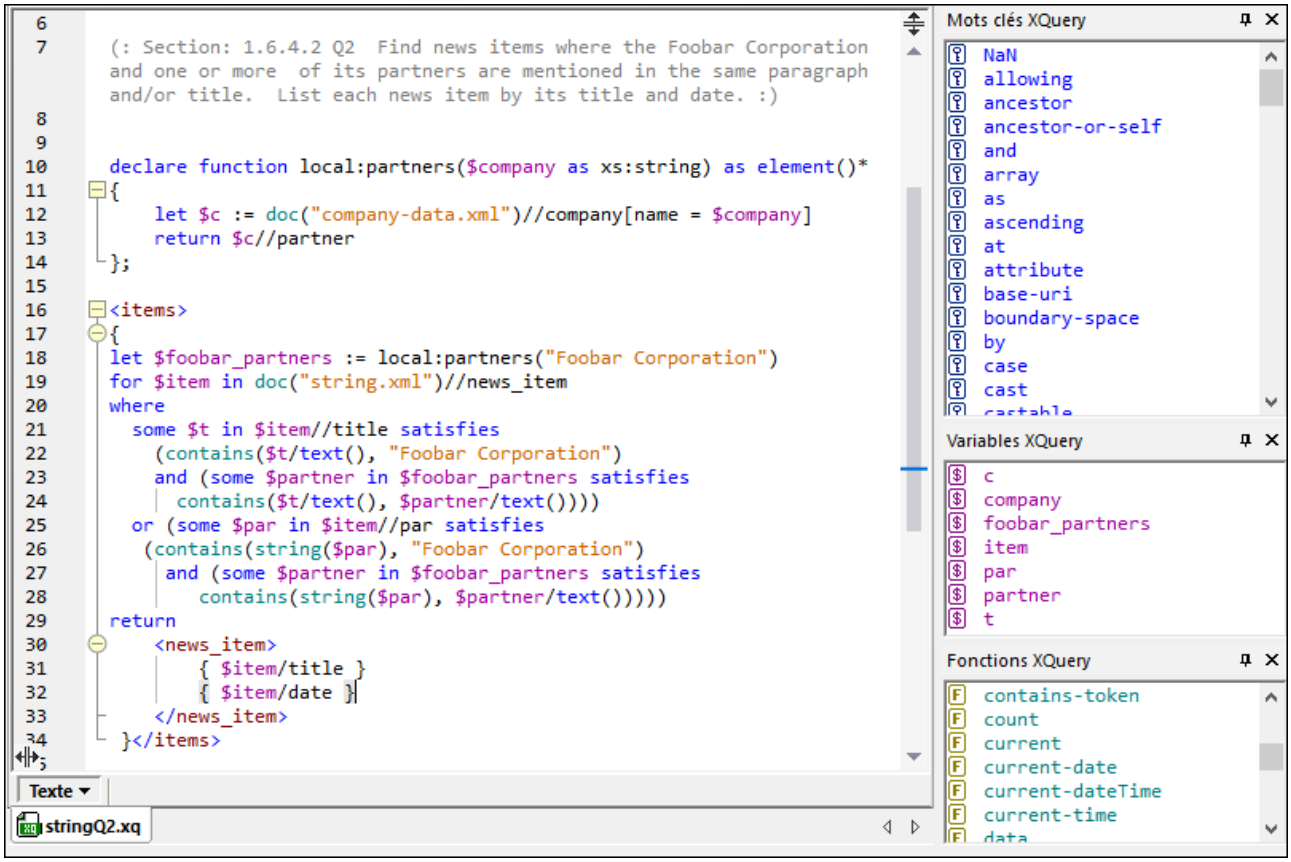


Un seuil de délai pour des instructions XSLT uniques dans une feuille de style XSLT peut être spécifié pour l'optimiseur. Les valeurs s'étendent de 0,1 % du temps de transformation total à 99 % du temps total. Si une instruction prend plus de temps à être exécutée que le temps spécifié en tant que le seuil, l'analyse d'optimisation est invoquée. Sinon, aucune analyse n'est exécutée. Si l'analyse d'optimisation échoue, il se peut que le seuil de délai spécifié dans les paramètres de l'optimiseur soit trop haut. Il faudra éventuellement le baisser.

© 2016-2022 Altova GmbH

8 XQuery

Site Internet Altova : Éditeur XQuery
Les documents XQuery et XQuery Update peuvent être édités dans le Mode Texte. Ce mode (*voir capture d'écran*) permet l'utilisation d'assistants à la saisie, de coloration de la syntaxe et d'édition intelligente pour faciliter l'édition. De plus, vous pouvez valider votre document XQuery et l'exécuter (avec un fichier XML optionnel le cas échéant) à l'aide du moteur Altova XQuery intégré.



Note : Les fichiers de mise à jour XQuery et XQuery peuvent être édités uniquement dans le Mode Texte. Aucune autre consultation des fichiers XQuery n'est disponible.

Association de fichiers de XQuery Update et XQuery

Dans XMLSpy, les documents XQuery Update et XQuery sont reconnus en tant que deux types de documents différents. Généralement, les documents XQuery comportent une extension `.xq` alors que les documents de mise à jour XQuery possèdent l'extension de fichier `.xqu`. Vous pouvez associer des extensions de fichier supplémentaires avec ces types de fichiers et aussi changer des associations de types de fichier à tout moment, dans la section *Type de fichier* du dialogue Options (**Outils | Options | Type de fichier**).

L'association de type de document d'une extension de fichier est importante parce que, selon cette association, soit une exécution XQuery, soit une XQuery Update sera effectuée lorsque la commande **XQuery/ Exécution de mise à jour** sera exécutée.

Dans cette section

Cette section est organisée comme suit :

- Éditer les documents XQuery
- Évaluation XQuery
- Validation XQuery
- XQuery Exécution/Mise à jour
- XQuery Update Facility
- XQuery et bases de données XML

Autres fonctions liées et informations :

- XSLT/XQuery Débogueur et profileur
- Mise en place Moteur XQuery
- Fenêtre de sortie : XPath/XQuery
- Outils | Options | Types de fichier
- Outils | Options | XQuery

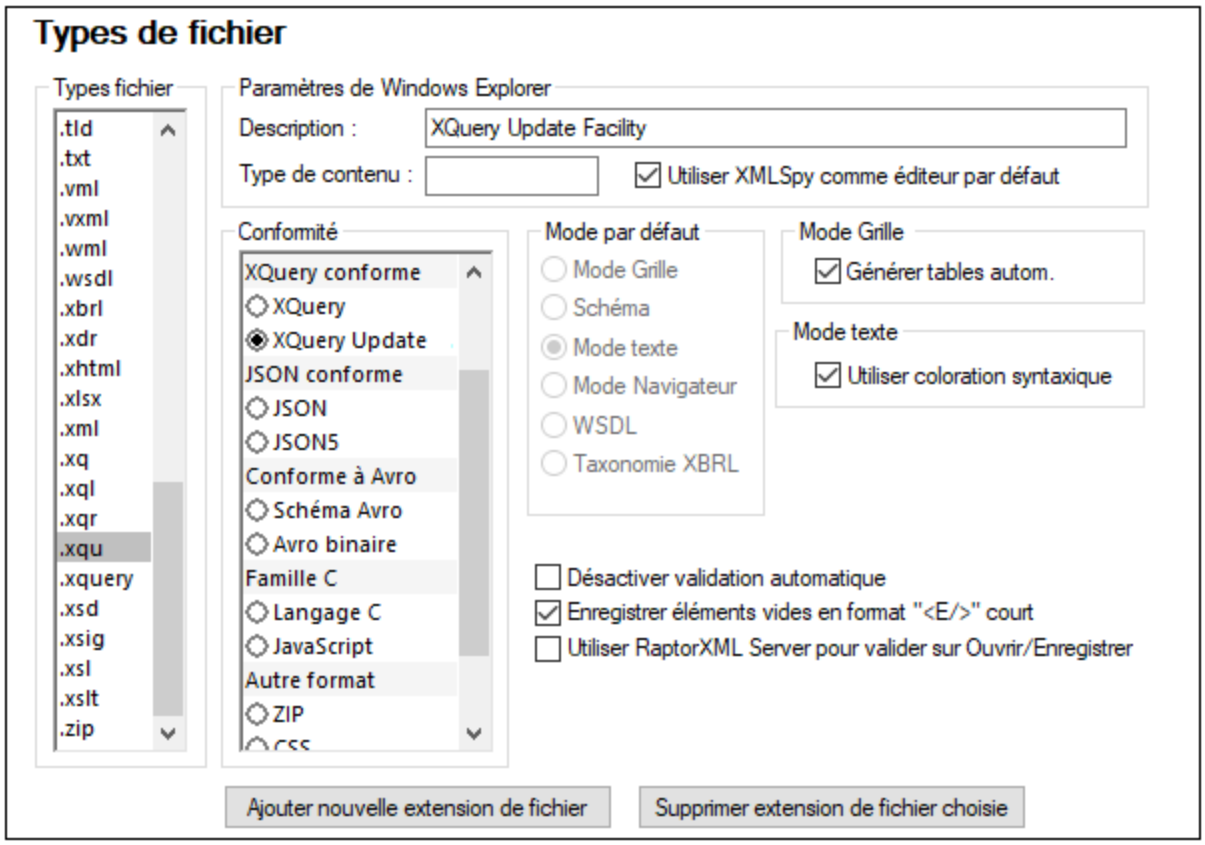
RaptorXML pour la ligne de commande et le traitement par lot

La GUI de XMLSpy permet un traitement par lot par le biais de la fonction des projets. Néanmoins, si vous souhaitez une solution plus flexible, vous devriez essayer le produit RaptorXML d'Altova, qui contient le Moteur XQuery le plus récent d'Altova. RaptorXML est la solution idéale si vous souhaitez effectuer des exécutions XQuery depuis la ligne de commande ou un traitement par lot.

© 2016-2022 Altova GmbH

8.1 Éditer les documents XQuery

Dans XMLSpy, les documents XQuery Update et XQuery sont reconnus en tant que deux types de document différents. Le type de document (XQuery ou XQuery Update) est attribué à une extension de fichier dans la section *Types de fichier* du dialogue Options (**Outils | Options | Type de fichier** *capture d'écran ci-dessous*). Lorsqu'un fichier de type XQuery ou XQuery Update est ouvert dans XMLSpy, les fonctions d'édition XQuery du Mode Texte sont disponibles pour ce fichier.



Extensions de fichier actuellement définies en tant que Mises à jour XQuery et XQuery dans XMLSpy

XQuery	`.xq` `.xql` `.xqr` `.xquery`
XQuery Update	`.xqu`

Note : Les fonctions d'édition décrites dans cette section sont identiques pour les documents XQuery et XQuery Update.

XQuery Exécution/Mise à jour

La commande de GUI **XSL/XQuery | XQuery Update/Exécution** exécute automatiquement soit une exécution XQuery soit une mise à jour XQuery selon le type de fichier du fichier XQuery qui a été sélectionné pour l'exécution. Voir la section XQuery Exécution/Mise à jour pour plus de détails.

8.1.1 Documents XQuery

Un document XQuery ou XQuery Update s'ouvre automatiquement dans le mode d'édition XQuery du Mode Texte s'il est conforme à XQuery ou XQuery Update. Un fichier est défini comme étant conformant à un certain type de document dans la section *Types de fichier* du dialogue Options (**Outils | Options | Type de fichier)**, (*capture d'écran ci-dessous*).

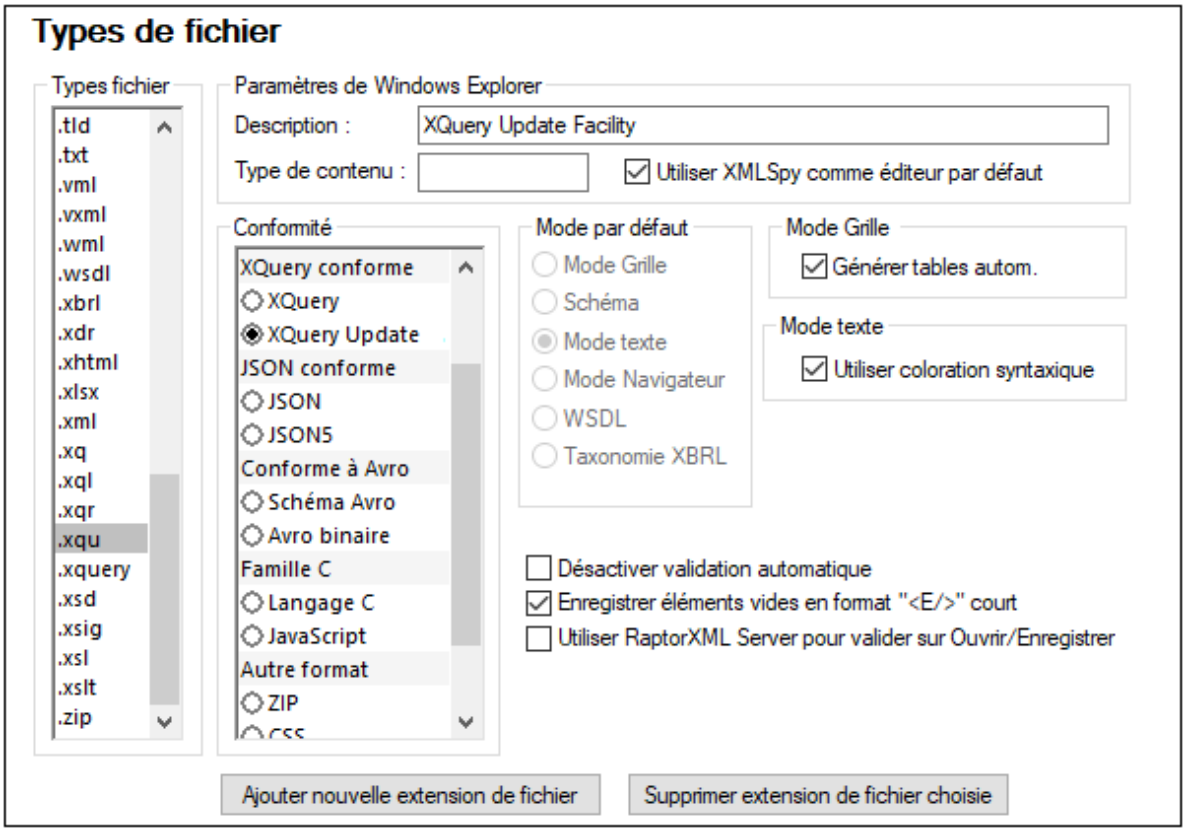
Extensions de fichier actuellement définies en tant que Mises à jour XQuery et XQuery dans XMLSpy

XQuery	`.xq` `.xql` `.xqr` `.xquery`
XQuery Update	`.xqu`

Configurer des extensions de fichier supplémentaires pour les rendre conforme à XQuery

Pour configurer des extensions de fichier supplémentaires pour être conformes à XQuery :

1. Sélectionner **Outils | Options**. Le dialogue Options apparaît (*capture d'écran ci-dessous*).



2. Choisir la section **Types de fichier**.
3. Cliquer sur **Ajouter nouvelle extension de fichier** pour ajouter la nouvelle extension de fichier à la liste des types de fichier.
4. Sous *Conformité*, sélectionner *Conforme à XQuery* et ensuite *XQuery* ou *XQuery Update*.

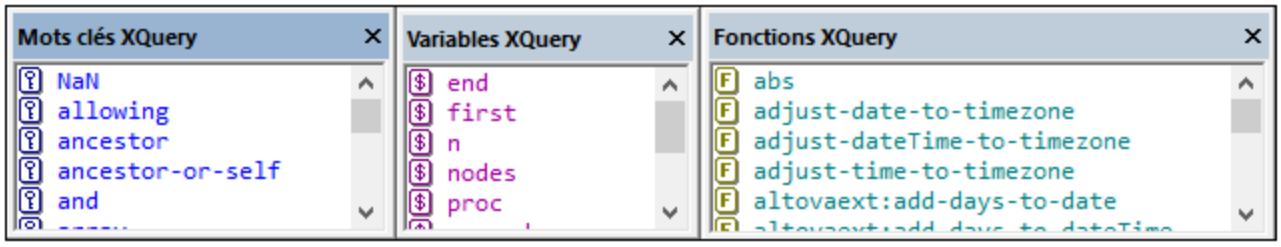
© 2016-2022 Altova GmbH

Vous devriez aussi procéder aux configurations suivantes :

- *Description :* `XML Query Language` ou `XQuery Update Facility`
- *Type de contenu :* `text/xml`
- Si vous souhaitez utiliser XMLSpy en tant que l'éditeur par défaut pour les fichiers XQuery, cocher la case *Utiliser XMLSpy en tant qu'éditeur par défaut*.

8.1.2 Assistants à la saisie XQuery

Il existe trois Assistants à la saisie dans le Mode XQuery du Mode Texte : Mots-clés XQuery (bleu), Variables XQuery (violet), et les fonctions XQuery (olive).



Veuillez noter les points suivants :

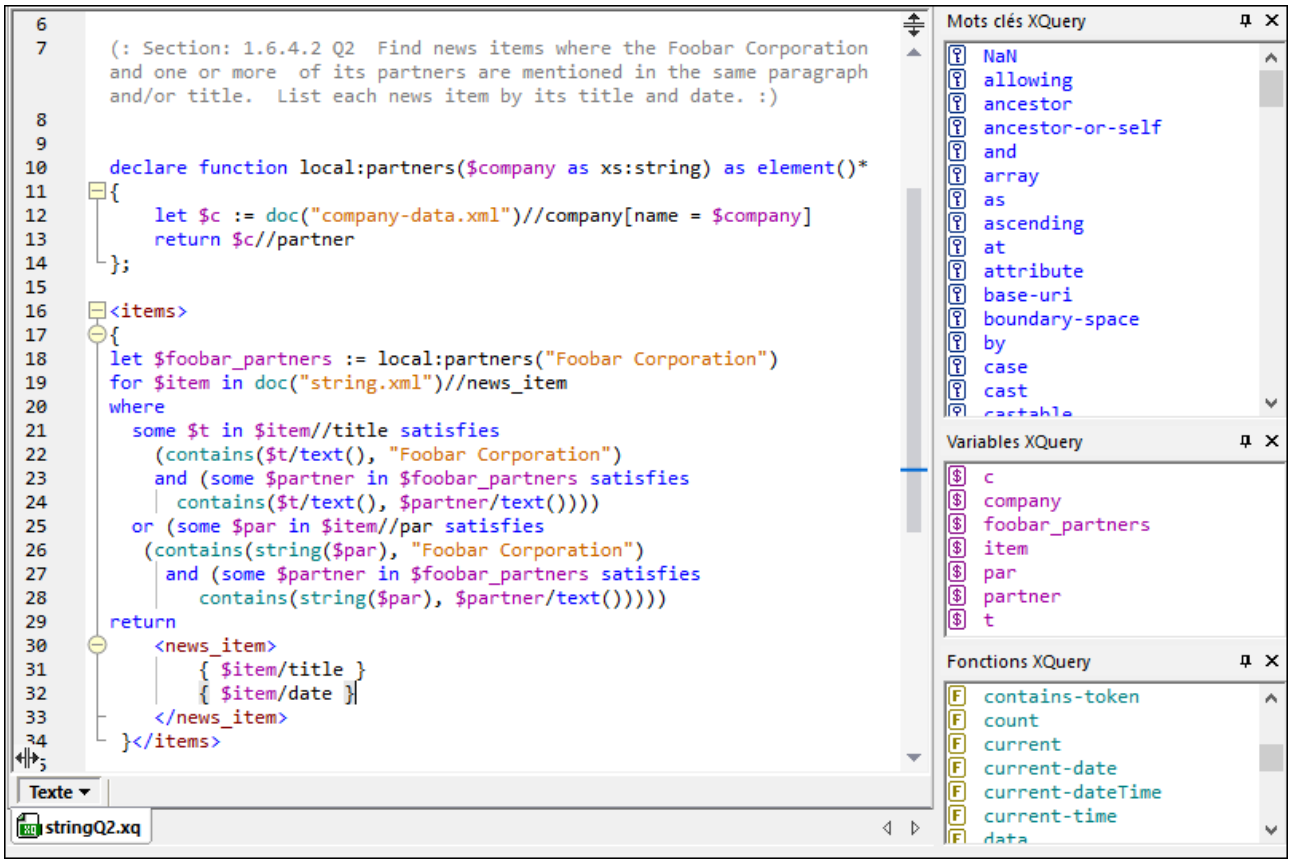
- La couleur des items dans les trois Assistants à la saisie est différente et correspond à la coloration de la syntaxe utilisée dans le texte. Ces couleurs ne peuvent pas être modifiées.
- Les mots clés recensés et les fonctions sont celles prises en charge par les Moteurs XQuery d'Altova.
- Les variables sont définies dans le document XQuery lui-même. Lorsqu'un caractère `$` suivi d'un caractère sont saisis dans le Mode Texte, le caractère est saisi dans l'Assistant à la saisie Variables (sauf si une variable consistant en ce caractère exactement existe). Dès qu'un nom de variable saisi correspond à un nom de variable existant déjà, le nom de variable qui vient d'être saisi disparaît de l'Assistant à la saisie.
- Pour parcourir un Assistant à la saisie, cliquer sur un item dans l'Assistant à la saisie puis utiliser soit la barre de défilement, la roue de la souris ou les touches vers le haut ou vers le bas pour faire défiler la liste.

Pour insérer un des items recensés dans les assistants à la saisie à l'intérieur du document, placer le curseur dans le point d'insertion requis et double-cliquer sur l'item. Veuillez noter que certains strings de caractères représentent aussi bien un mot-clé qu'une fonction (`empty`, `unordered`, et `except`). L'item approprié est inséré selon ce que vous avez double-cliqué.

8.1.3 Coloration de la syntaxe XQuery

Un document XQuery peut consister en un code XQuery ainsi qu'en un code XML. La coloration syntaxique par défaut pour le code XQuery est décrite dans cette section. La coloration syntaxique pour le code XML dans un document XQuery est le même que celui utilisé pour des documents XML réguliers. Toutes les colorations syntaxiques (aussi bien pour le code XQuery que le code XML) sont configurées dans la section Police et

couleurs du dialogue Options (**Outils | Options**). Veuillez noter que le code XQuery peut être contenu dans des éléments XML en plaçant le code XQuery dans des accolades { } (voir *capture d'écran pour un exemple*).



Dans le code XQuery (dans le Mode XQuery du Mode Texte), la coloration syntaxique par défaut suivante est utilisée :

- `(: Comments, including 'smiley' delimiters, are in gray :)`
- Les mots-clés XQuery sont en bleu : `keyword`
- Les Variables XQuery, y compris le signe dollar, sont en violet : `$start`
- Les Fonctions XQuery, mais **pas** leurs parenthèses, sont en vert : `function()`
- Les Strings sont en orange : `"Procedure"`
- Tous le texte restant, comme les expressions Xpath, sont en noir (*affiché ci-dessous en souligné*). Donc :

```
for $s in doc("report1.xml")//section[section.title = "Procedure"]
return ($s//incision)[2]/instrument)
```

Vous pouvez changer ces couleurs par défaut et d'autres propriétés de police dans la section Polices de texte du dialogue Options (**Outils | Options**).

Note : Dans la capture d'écran ci-dessus, une paire de parenthèses colorées pour un commentaire est affichée en noir et en gras. Cela est dû à la fonction de Correspondance des parenthèses (voir Édition intelligente XQuery).

© 2016-2022 Altova GmbH

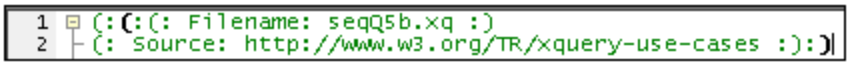
8.1.4 Édition intelligente XQuery

Le mode XQuery du Mode Texte propose les fonctions d'édition intelligences suivantes :

- Correspondance des parenthèses
- Mots-clés
- Variables
- Fonctions
- Guides visuels

Correspondance des parenthèses

La fonction de correspondance des parenthèses marque l'ouverture et la fermeture d'une paire de parenthèses, vous permettant d'apercevoir clairement le contenu d'une paire de parenthèses. Cela est particulièrement utile lorsque les parenthèses sont imbriquées, comme cela est le cas dans les commentaires XQuery (*voir capture d'écran ci-dessous*).

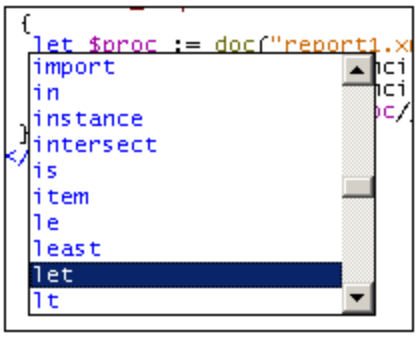


- La correspondance de parenthèses est activée lorsque le curseur est placé soit immédiatement avant ou immédiatement après une parenthèse (ouvrante ou fermante). La première parenthèse est marquée (en noir gras) avec sa parenthèse correspondante. Veuillez noter la position du curseur dans la capture d'écran ci-dessus.
- La correspondance de parenthèses est activée pour les parenthèses `()`, les crochets `[]` et les accolades `{}`. Seule exception, les chevrons **<>**, qui sont utilisés en tant que balises XML.

Note : Lorsque vous placez le curseur juste après une parenthèse de démarrage ou de fin, les deux parenthèses sont marquées. Appuyer sur **Ctrl+E** pour déplacer le curseur vers l'autre membre de la paire. Appuyer sur **Ctrl+E** plusieurs fois pour passer entre la parenthèse de démarrage et de fin. Cette fonction vous permet de naviguer plus facilement dans votre document.

Mots-clés

Les mots-clés XQuery sont des instructions utilisées dans les expressions query, et sont affichées en bleu. Vous sélectionnez un mot-clé en plaçant le curseur dans un mot-clé ou juste avant ou après le mot. Une fois le mot-clé sélectionné, appuyer sur **Ctrl+Espace** entraîne l'affichage d'une liste complète de mots-clés dans un menu popup. Vous pouvez parcourir la liste et double-cliquer un mot-clé que vous souhaitez remplacer par le mot-clé sélectionné.

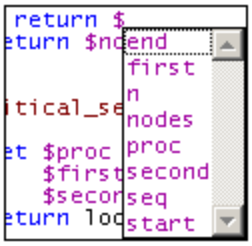


Dans la capture d'écran ci-dessus, le curseur était placé dans le mot-clé `let`. En double-cliquant un mot-clé de la liste, le mot-clé `let` sera remplacé par ce mot-clé.

Variables

Les noms des variables sont préfixés avec le signe **$**, elles sont affichées en violet. Ce mécanisme de fonction d'édition intelligente est semblable à celui utilisé pour les mots-clés. Il existe deux manières d'accéder à la liste pop-up de toutes les variables dans un document:

- Une fois avoir saisi un caractère **$**, appuyer sur **Ctrl+Espace**
- Sélectionner une variable et appuyer sur **Ctrl+Espace**. (Une variable est sélectionnée lorsque vous placez le curseur juste après le caractère **$**, ou dans le nom d'une variable, ou juste après le nom d'une variable.)

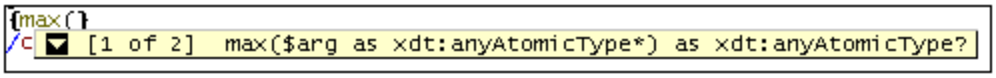


Pour insérer une variable après le caractère **$** (pendant la saisie), ou pour remplacer une variable sélectionnée, double-cliquer sur la variable que vous souhaitez dans le menu qui apparaît.

Fonctions

De même que pour les mots-clés et les variables, un menu popup contenant des fonctions intégrées s'affiche lorsque vous sélectionnez une fonction (affichée en olive) et appuyez sur **Ctrl+Espace**. (Une fonction est sélectionnée lorsque vous placez le curseur dans un nom de fonction ou juste après ou juste avant le nom d'une fonction. Le curseur ne doit pas être placé entre les parenthèses qui suivent le nom de la fonction.) Double-cliquer sur le nom d'une fonction dans le menu qui s'affiche pour remplacer le nom de la fonction sélectionnée avec la fonction provenant du menu popup.

Pour afficher une information contenant la signature d'une fonction (*capture d'écran ci-dessous*), placer le curseur immédiatement après la parenthèse d'ouverture et appuyer sur **Ctrl+Espace**. Veuillez noter que la signature ne s'affiche que pour les fonctions standard XQuery.



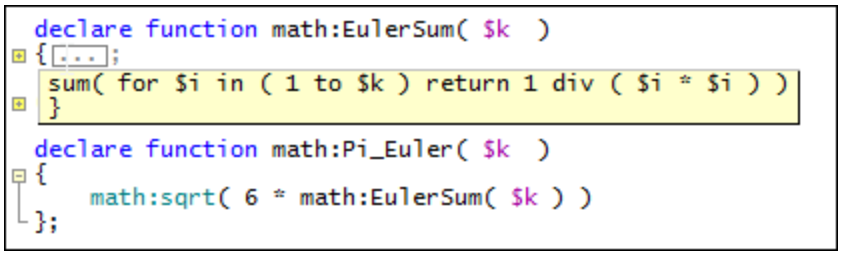
La flèche orientée vers le bas indique qu'il y a plus d'une seule fonction avec le même nom, mais comportant des arguments ou des types de return différents. Cliquer sur la flèche pour afficher la signature de la fonction de texte (si disponible) ; cliquer à nouveau pour passer à travers toutes les fonctions portant ce nom. En alternative, vous pouvez utiliser la combinaison de touches **Ctrl+Shift+Haut** ou **Ctrl+Shift+Bas** pour vous déplacer dans une séquence.

Guides visuels

Le pliage de texte (ou le pliage de source) est activé pour les accolades XQuery, les commentaires XQuery, les éléments XML et les commentaires XML et réfère à la capacité d'agrandir et de réduire ces nœuds. Ces nœuds sont indiqués dans la marge de pliage de la source par un signe +/- (*voir capture d'écran ci-dessous*). La

© 2016-2022 Altova GmbH

marge peut être ouverte ou fermée dans le dialogue Paramètres de Mode Texte. Lorsqu'un nœud est réduit, cela est indiqué visuellement par une ellipse (*voir capture d'écran ci-dessous*). Si le curseur de la souris est placé au-dessus d'une ellipse, le contenu du nœud réduit est affiché dans une fenêtre popup (*capture d'écran ci-dessous*). Si le contenu est trop grand pour une fenêtre popup, cela est indiqué par une ellipse en bas de la fenêtre.



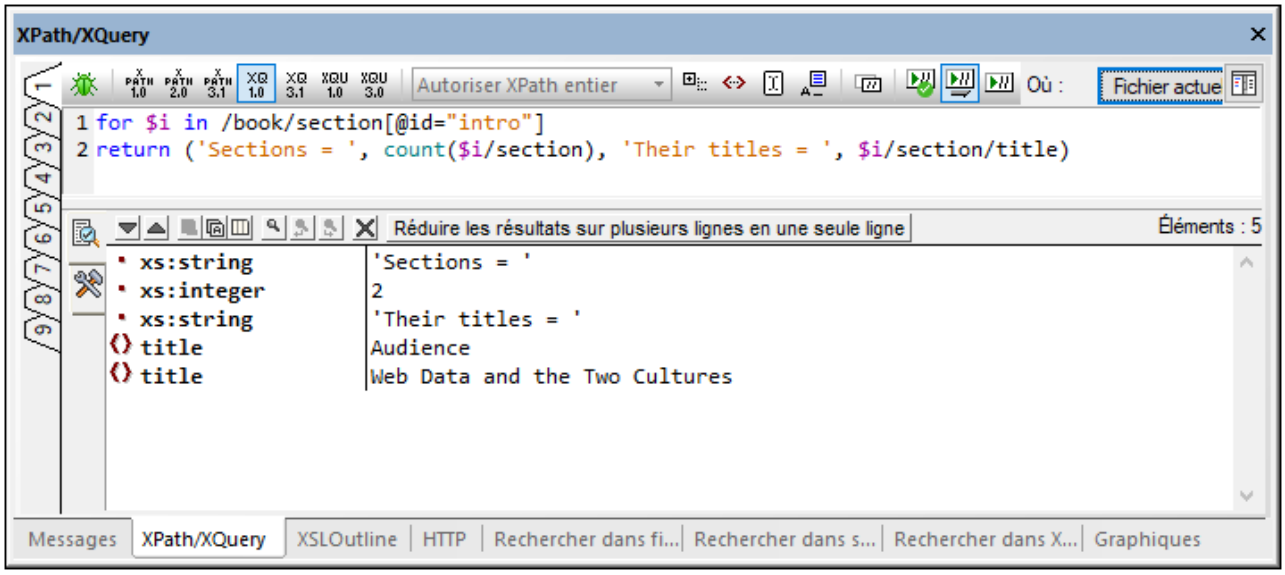
L'icône **Basculer tous les plis** dans la barre d'outils Texte bascule **tous** les nœuds dans leur forme agrandie ou réduit tous les nœuds au niveau de l'élément du document de niveau supérieur.

Les options suivantes sont disponibles en cliquant sur l'icône +/- du nœud :

Clic [-]	Réduit le nœud.
Clic [+]	Agrandit le nœud de manière à ce que les nœuds descendants s'affichent sous leur forme agrandie ou réduite selon l'état dans lequel ils étaient avant la réduction du nœud.
Shift+Clic [-]	Réduit tous les nœuds descendants, mais laisse le nœud qui a été cliqué dans sa forme agrandie.
Ctrl+Clic [+]	Agrandit le nœud cliqué ainsi que tous ses nœuds descendants.

8.2 Évaluation XQuery

Les expressions XQuery peuvent être évaluées par rapport à un ou plusieurs documents dans la Fenêtre de sortie XPath/XQuery (*capture d'écran ci-dessous*).



Procéder comme suit :

1. Saisir l'expression XQuery dans le volet supérieur de la fenêtre.
2. Dans la liste de choix *Où* (*voir capture d'écran ci-dessus*), sélectionner où le document requis est situé. Les options sont : (i) fichier actuel ; (ii) fichiers ouverts ; (iii) projet ; (iv) dossier.
3. Cliquer sur **Évaluer Expression XPath/XQuery (F5)**. L'expression est évaluée par rapport au/x fichier/s XML. Si l'emplacement spécifié (*Où*) contient un ou plusieurs fichiers XML, tous les fichiers XML sont parcourus pour trouver des structures de données ou des contenus correspondants à l'expression. Les résultats de toutes les correspondances disponibles sont affichées dans le volet inférieur.

Dans la capture d'écran ci-dessus, une requête est faite pour un élément de section qui a l'attribut `@id='intro'`. La requête retourne le nombre de sous-sections de cette section `intro` et leurs titres.

Pour plus d'informations, voir aussi Fenêtre Sortie : XPath/XQuery et Prévisualiser et appliquer XQuery Updates.

© 2016-2022 Altova GmbH

8.3 Validation XQuery

Pour valider un document XQuery ou XQuery Update, procéder comme suit :

1. Faire du document XQuery le document actif.
2. Choisir **XML | Valider**, ou appuyer sur la touche **F8**, ou cliquer sur l'icône de barre d'outils **Valider**.



Valider l'icône de la barre outils

Le document en ce qui concerne la syntaxe XQuery correcte.

8.4 Exécution XQuery/Update

Un document XQuery ou XQuery Update peut être exécuté des manières suivantes :

- Lorsque le document XQuery ou XQuery Update est actif.
- Lorsqu'un document XML est actif.

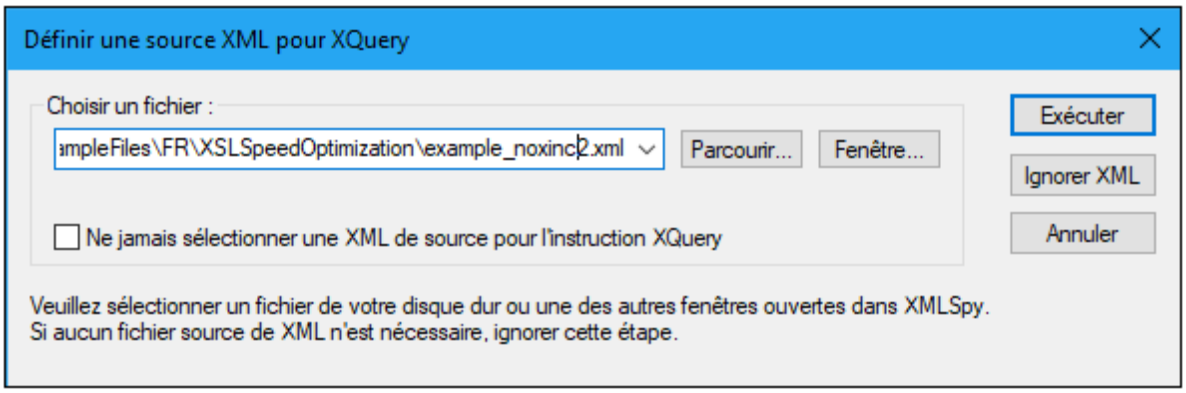
Note : L'extension de fichier d'un document permet de spécifier si le document est un document XQuery ou XQuery Update. XMLSpy reconnaît des associations de type de fichier conformément aux définitions faites dans la section Filetypes du dialogue Options. (**Outils | Options | Types de fichier**).

Note : pour une mise à jour de XQuery, vous pouvez aussi saisir Mettre à jour les expressions dans la fenêtre de sortie XPath/XQuery et les mises à jour de l'aperçu. Si les mises à jour sont acceptables, vous pouvez les appliquer et enregistrer le fichier mis à jour. Voir XQuery Update Facility et Previewing and Applying Updates pour plus de détails.

Exécution avec un document actif XQuery ou XQuery Update

Afin d'exécuter un document XQuery ou XQuery Update avec la fonction *document XQuery / XQuery Update actif*, procédez comme suit

1. Faire du document XQuery ou XQuery Update le document actif.
2. Choisir **XSL/XQuery | XQuery/ Update Execution** ou cliquer sur l'icône de barre d'outils de la commande. Le dialogue Définir la source XML pour XQuery s'ouvre (*capture d'écran ci-dessous*).



3. Parcourir la liste pour trouver un fichier XML et l'exécuter ou sauter la sélection d'une source XML.

Icône de barre d'outils XQuery/ Update Exécution

Généralement, un document XQuery **n'est pas** associé avec un seul document XML. Cela est dû au fait que les expressions XQuery peuvent sélectionner un nombre quelconque de documents XML avec la fonction `doc()`. Dans XMLSpy, néanmoins, avant d'exécuter des documents XQuery individuels, vous pouvez sélectionner un document XML source pour l'exécution. Dans ces cas, le nœud du document de la source XML sélectionnée est l'item de contexte de démarrage disponible au niveau racine du document XQuery. Les chemins qui commencent avec une barre oblique sont résolus avec ce nœud de document en tant que son item contextuel.

© 2016-2022 Altova GmbH

Note : la commande **XQuery/ Update Execution** est aussi disponible dans le menu contextuel des items de la fenêtre Projet.

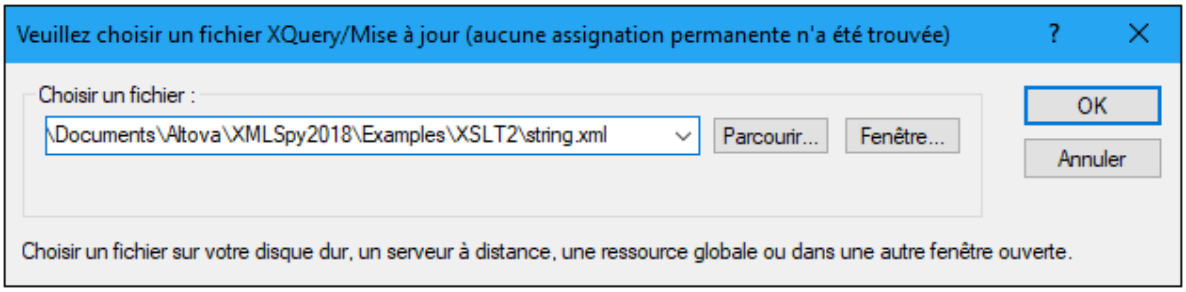
Résultat de l'exécution / mise à jour

- *Exécution XQuery :* Le document de résultat est généré comme fichier temporaire qui peut être enregistré à tout emplacement avec le format de fichier désiré et l'extension.
- *Mise à jour Xquery :* La mise à jour est enregistrée dans le fichier, ou le fichier mis à jour est ouvert, permettant de le consulter, puis soit de l'enregistrer soit de le fermer sans l'enregistrer. Vous pouvez spécifier laquelle de ces deux actions vous souhaitez effectuer. Pour ce faire, utilisez la section *XQuery* du dialogue Options (**Outils | Options | XQuery**).

Exécution avec un document XML actif

Afin d'exécuter un document XQuery ou XQuery Update sur un *document XML actif*, procédez comme suit

1. Faire du document XML le document actif.
2. Choisir **XSL/XQuery | XQuery/ Update Execution** ou cliquer sur l'icône de barre d'outils de la commande. Cela permet d'ouvrir le dialogue XQuery/Update File (*voir la capture d'écran ci-dessous*).



3. Parcourez pour trouver le fichier XQuery ou XQuery Update et cliquez sur **OK**.



Icône de barre d'outils XQuery/ Update Execution

Résultat de l'exécution / mise à jour

- *Exécution XQuery :* Le document de résultat est généré comme fichier temporaire qui peut être enregistré à tout emplacement avec le format de fichier désiré et l'extension.
- *Mise à jour XQuery :* La mise à jour est enregistrée dans le fichier, ou le fichier mis à jour est ouvert, permettant de le consulter, puis soit de l'enregistrer soit de le fermer sans l'enregistrer. Vous pouvez spécifier laquelle de ces deux actions vous souhaitez effectuer. Pour ce faire, utilisez la section *XQuery* du dialogue Options (**Outils | Options | XQuery**).

Back-mapping

En activant la fonction Back-mapping, l'exécution XQuery sera effectuée de manière à ce que le document de résultat puisse être remappé sur les documents XQuery+XML d'origine. Si vous cliquez sur un nœud dans le document de résultat, alors l'**instruction XQuery** *et* les **données source XML** qui généreront ce fragment de résultat particulier seront en surbrillance. De plus, si vous cliquez sur une instruction XQuery ou un nœud de données XML, alors les nœuds correspondants des deux autres documents sont en surbrillance. Voir la commande **XSL/XQuery | Enable Back-Mapping** pour les détails.

Variables XQuery

Si vous utilisez les moteurs Altova XQuery, les variables peuvent être stockées dans un dialogue GUI pratique. Toutes les variables stockées sont passées dans le document XQuery à chaque fois que vous exécutez un document XQuery via XMLSpy. Pour plus d'informations, voir la description de la commande Paramètres XSLT / Variable XQuery.

Moteurs Altova XQuery

Pour plus de détails concernant la mise en place des moteurs Altova XQuery et du traitement des fichiers XQuery, voir Implémentation Moteurs XQuery.

© 2016-2022 Altova GmbH

8.5 XQuery Update Facility

XQuery Update Facility est une extension du langage XQuery qui permet de modifier des parties de documents XML. Dans une exécution XQuery normale, l'ensemble du document est régénéré et doit être enregistré dans son emplacement d'origine. Cette opération peut s'avérer inefficace lorsqu'il s'agit de ne modifier que des petites parties du document. Avec Update Facility, seules les parties du document nécessitant une mise à jours le sont.

XQuery Update Facility est décrit en tant qu'extensions de XQuery 1.0 et XQuery 3.1, dans les spécifications suivantes, respectivement :

- XQuery Update Facility 1.0 (W3C Recommandation du 17 mars 2011)
- XQuery Update Facility 3.0 (W3C Ébauche de travail du 19 février 2015)

XQuery Update Facility dans XMLSpy

Les points suivants expliquent comment fonctionne XQuery Update dans XMLSpy :

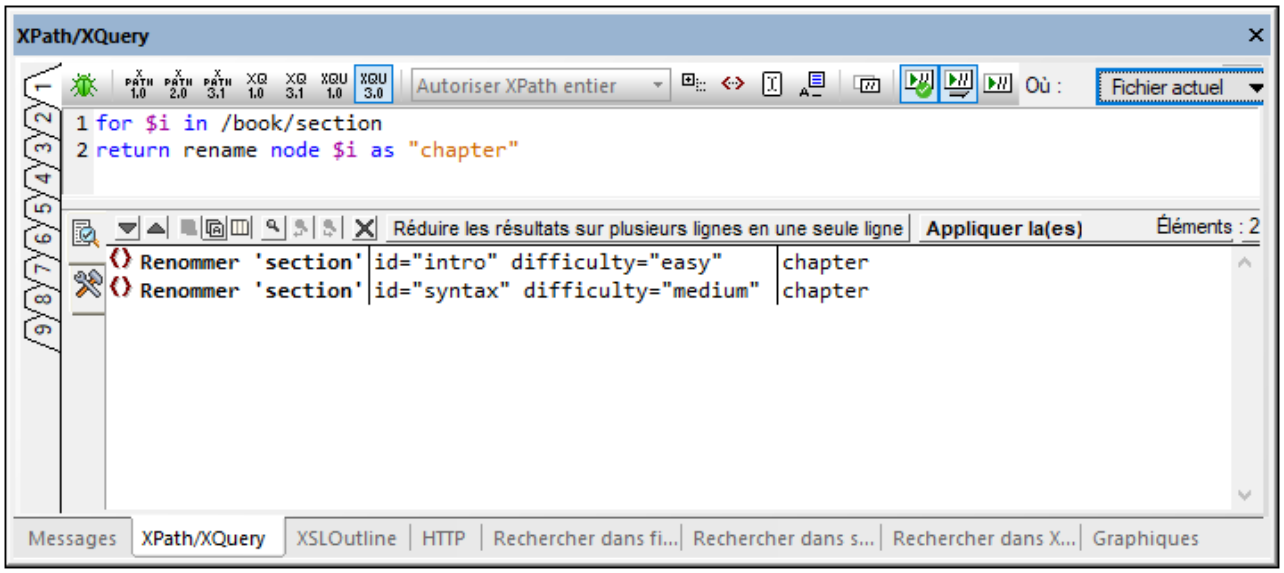
- Une mise à jour est effectuée par une expression de mise à jour. Par exemple, une expression de mise à jour peut spécifier qu'un nœud dans une document XML a été renommé :
  ```
  rename node /documents/doc-01 as "document-01"
  ```
- Dans la pratique, de nombreuses expressions de mise à jour multiples ont été saisies dans un seul document : le document XQuery Update.
- Alors que chaque expression de mise à jour exécute dans le document de mise à jour, le résultat n'est pas appliqué immédiatement, mais est ajouté à une liste de mises à jour en cours : *Pending Updates List (PUL)*. Par conséquence, la PUL contient les résultats de toutes les expressions de mise à jour. Toutes les mises à jours contenues dans la PUL sont ensuite appliquées ensemble, simultanément.
- Dans XMLSpy, les mises à jour PUL sont appliquées d'une de deux manières :
  (i) Après avoir été prévisualisées par l'utilisateur dans la GUI. L'avantage est que la mise à jour peut être abandonnée si la prévisualisation montre des résultats indésirables. La prévisualisation est disponible lors de l'exécution de la commande Exécution XQuery/Update, ou lors de l'évaluation des expressions XQuery Update dans la fenêtre de sortie XPath/XQuery. La configuration de l'option de prévisualisation est expliquée dans les descriptions respectives.
  (ii) Directement et sans aucune intervention de la part de l'utilisateur. L'avantage est que la mise à jour est effectuée de manière silencieuse et sans l'intervention de l'utilisateur. L'application directe des mises à jour (sans prévisualisation) est disponible lors de l'exécution de la commande Exécution XQuery/Update, ou lors de l'évaluation des expressions XQuery Update dans la fenêtre de sortie XPath/XQuery. La configuration de l'option de mise à jour directe est expliquée dans les descriptions respectives.

XMLSpy propose une fonction puissante de prévisualisation des mises à jour XQuery qui vous permet de prévisualiser les effets des expressions de mise à jour dans le document XML actif puis de les appliquer. Cette fonction est décrite dans la section Prévisualiser et appliquer les mises à jour.

8.5.1 Prévisualiser et appliquer XQuery Updates

Si vous souhaitez modifier un document XML avec XQuery Update, vous pouvez prévisualiser les mises à jour avant de les appliquer au document XML et enregistrer le document modifié.

Dans la fenêtre de sortie XPath/XQuery (*capture d'écran ci-dessous*), vous pouvez saisie une ou plusieurs expressions de mise à jour puis prévisualiser les mises à jour dans la liste des mises à jour en attente : **pending update list (PUL)** qui est affichée dans le volet inférieur (*voir capture d'écran ci-dessous*). Si la PUL vous convient, vous pouvez appliquer les mises à jour dans le document pour enregistrer le document modifié. Si vous ne souhaitez pas procéder aux modifications dans la PUL, vous pouvez décider de ne pas appliquer les modifications ou de ne pas enregistrer le fichier.



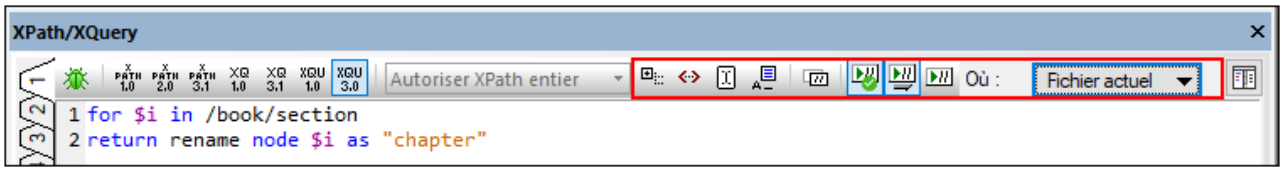
Pour créer une PUL pour un fichier XML actif, procéder comme suit :

1. Dans la barre d'outils de la fenêtre de sortie XPath/XQuery (*capture d'écran ci-dessus*), sélectionner l'icône **XQU 1.0** ou **XQU 3.0** (respectivement pour XQuery Update 1.0 ou XQuery Update 3.0).
2. Saisir une ou plusieurs expressions de mise à jour dans le panneau supérieur de la fenêtre. Pour une description des expressions de mise à jour et de leur syntaxe, voir la section Mettre à jour les opérations et la syntaxe.
3. Dans la liste de choix scanner emplacement de la barre d'outils, sélectionner l'emplacement à scanner pour les mises à jour :
   Fichiers ouverts : tous les fichiers actuellement ouverts dans XMLSpy seront scannés
   Fichier actuel : seul le fichier actuellement actif est scanné. Si l'emplacement sélectionné pour le scan est *Fichier actuel,* l'icône de la barre d'outils **Évaluer l'expression XPath/XQuery pendant la saisie** est activée
   Projet : le projet actif est scanné
   Dossier : Vous pouvez choisir un dossier à scanner
4. Pour exécuter l'/les expression/s et afficher la PUL, cliquer sur icône de la barre d'outils **Évaluer l'expression XPath/XQuery pendant la saisie**.

Barre d'outils de la fenêtre de sortie XPath/XQuery

Les commandes suivantes de la barre d'outils de la fenêtre de sortie XPath/XQuery suivantes (*encadrées en rouge dans la capture d'écran ci-dessous*) sont disponibles. De gauche à droite :

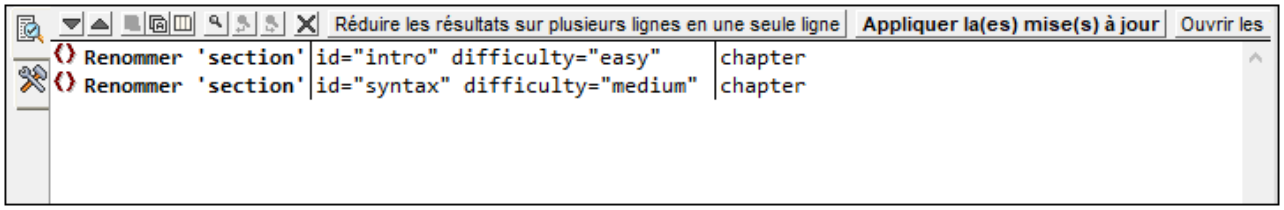
© 2016-2022 Altova GmbH



- La commande bascule *Afficher en-tête* spécifie si vous souhaitez afficher le nom de fichier des emplacements mis à jour ou pas.
- La commande bascule *Afficher résultat complet* spécifie si vous souhaitez afficher le contenu du nœud complet ou uniquement les attributs du nœud.
- Si sélectionné, *Configurer l'origine XPath/XQuery*, permet de définir l'emplacement du curseur dans le fichier actif en tant que l'origine des expressions XPath relatives. L'expression est évaluée par rapport à ce nœud.
- Au lieu de saisir manuellement l'expression de chemin de localisation d'un nœud, vous pouvez laisser XMLSpy la saisir pour vous. Pour ce faire : (i) Placer le curseur au niveau de l'expression XPath où vous souhaitez saisir le chemin de localisation; (ii) Placer le curseur dans la balise de démarrage du nœud que vous souhaitez cibler ; (iii) Cliquer sur l'icône *Copie le XPath de la sélection actuelle dans le champ d'édition* pour saisir le chemin de localisation dans l'expression. Le chemin de localisation sera saisi en tant que chemin absolu commençant au nœud de racine du document.
- *Valider fichier XML* active et désactive la validation de fichier XML.
- La commande *Évaluer Expression XPath/XQuery lors de la saisie* est activée lorsque l'option de parsage est *Fichier actuel*. Si la commande est activée, la PUL est générée pendant la saisie de l'expression de mise à jour.
- La commande *Évaluer XPath/XQuery Expression* évalue l'expression de mise à jour et génère la PUL.
- L'option de liste de choix *Scanner emplacement* est décrite ci-dessus.

Le volet Pending Update List (PUL)

Le volet PUL montre toutes les mises à jour qui seront effectuées. Si l'option Afficher en-tête a été activée dans la barre outils de la fenêtre, les emplacements des fichiers cibles seront affichés. L'affichage PUL est divisé en trois sections verticales (*voir capture d'écran ci-dessous*): (i) l'action de mise à jour à effectuer ; (ii) le contenu du nœud cible à mettre à jour ; (iii) le résultat de l'action de mise à jour.



Les commandes suivantes de la barre d'outils du volet PUL sont disponibles :

- Les icônes *Suivant* et *Précédent* sélectionnent, respectivement, les messages suivants et précédents du message actuellement sélectionné.
- Les commandes *Copy Line/s* copient, respectivement, la ligne sélectionnée et toutes les lignes sur le presse-papiers.
- Les commandes *Rechercher* trouvent du texte dans le volet PUL.
- La commande *Supprimer* supprime toutes les lignes dans le volet PUL.
- La commande *Réduire les résultats multilignes* réduit les items de liste de mise à jour à lignes multiples à des lignes simples.

- La commande *Appliquer les mises à jour* applique les mises à jour en attente aux emplacements cibles. Lors de la mise à jour, les mises à jour peuvent être enregistrées sur un fichier ou le fichier mis à jour peut être affiché (puis, ultérieurement, être enregistré manuellement ou pas). Voir l'option suivante.
- La liste de choix *Mettre à jour les fichiers directement* vous permet de sélectionner (i) si les fichiers sont mis à jour sur le disque de manière silencieuse ou (ii) si les fichiers mis à jour sont ouverts et rendus actifs. Si cette dernière option a été sélectionnée, les fichiers cibles non-actifs sont rendus actifs. Vous pouvez ensuite choisir si vous souhaitez enregistrer le document modifié ou pas.

Note : Si un ou plusieurs fichiers ont été mis à jour directement sur un disque, une liste des fichiers changés est affichée. Chaque item dans la liste montre l'emplacement du fichier et est un lien cliquable dans le fichier.

8.5.2 Mise à jour des opérations et de la syntaxe

XQuery Update Facility permet les opérations suivantes :

- Supprimer un ou plusieurs nœuds
- Insérer un ou plusieurs nœuds avant, après ou à l'intérieur d'un nœud spécifié
- Renommer un nœud
- Remplacer un nœud avec une séquence d'items
- Remplacer la valeur d'un nœud avec la valeur string d'une séquence d'items

Les mots-clés et la syntaxe de ces opérations sont décrites dans les sous-sections de cette section.

8.5.2.1 Supprimer les nœuds

Description et syntaxe
Supprime un ou plusieurs nœuds.

```
delete node nodeSequence
delete nodes nodeSequence
```

Détails

- L'expression *nodeSequence* retourne une séquence du/des nœud/s à supprimer. Tous les nœuds sélectionnés seront marqués pour la suppression.
- Il est possible d'utiliser le **node** singulier ou le **nodes** pluriel. Aucune correspondance n'est nécessaire avec le nombre d'items dans *nodeSequence*.

Exemples

```
for $i in /book/section return
delete nodes $i/@id
```


© 2016-2022 Altova GmbH

8.5.2.2 Insérer les nœuds

Description et syntaxe
Insère un ou plusieurs nœuds avant, après ou dans le nœud cible spécifié.

```
insert (node|nodes) items into targetNode
insert (node|nodes) items as first into targetNode
insert (node|nodes) items as last into targetNode
insert (node|nodes) items before targetNode
insert (node|nodes) items after targetNode
```

Détails

- L'expression *items* doit retourner une séquence d'items. Bien que le mot-clé **node|nodes** soit utilisé, *items* peut être une séquence d'items non-nœud.
- L'expression *targetNode* doit pointer vers un nœud cible unique.
- Si le mot-clé **into** est utilisé, *targetNode* doit être un nœud d'élément ou un nœud document-élément.
- Si la phrase-clé **as first** ou **as last** est utilisée, l'insertion a lieu en tant que premiers ou derniers enfants, respectivement.
- Si le mot-clé **into** est utilisé seul, alors les attributs sont apposés aux attributs existants, et les éléments sont insérés en tant que premiers enfants.
- Si le mot-clé **before** ou **after** est utilisé, *targetNode* peut être de n'importe quel type.
- Si un attribut est inséré, son nom ne doit pas dupliquer celui d'un attribut déjà existant.

Exemples

```
for $i in /book/section return
insert nodes (attribute id { 'somevalue' }, <newelement>some content including the numbers
"{ 1 to 3}"</newelement>)
into $i
```

8.5.2.3 Renommer le nœud

Description et syntaxe
Renames an element, attribute, or processing instruction nœud.

```
rename node targetNode as name
```

Détails

- L'expression *targetNode* doit pointer vers un nœud cible unique, qui peut être un élément, un attribut ou une instruction de traitement.
- L'expression *name* doit évaluer à un QName ou un string.
- Si un QName est construit, l'espace de noms obligatoire est déclaré localement.

Exemples

```
rename node /book/title as 'header-1'

rename node /book/title as QName("http://www.altova.com/xquf", "header-1")
```

8.5.2.4 Remplacer le nœud

Description et syntaxe
Remplace un nœud avec une séquence de tout type d'items.

```
replace node targetNode with items
```

Détails

- L'expression *targetNode* doit pointer vers un nœud cible unique.
- L'expression *items* doit retourner une séquence d'items. Cette séquence remplacera le nœud cible.
- Sauf pour les nœuds d'attribut, un nœud cible peut être remplacé par tout type de séquence.
- Un nœud d'attribut peut uniquement être remplacé par un nœud d'attribut. *Voir exemple ci-dessous*.

Exemples

```
replace node //hr with '<line/>'

for $i in //@height return
replace node $i with (attribute line-height{'12pt'})
```

8.5.2.5 Remplacer la valeur du nœud

Description et syntaxe
Remplace la valeur d'un nœud par la valeur string d'une séquence d'items.

```
replace value of node targetNode with items
```

Détails

- L'expression *targetNode* doit pointer vers un nœud cible unique.
- L'expression *items* doit retourner une séquence d'items.
- Les contenus d'un nœud cible sont remplacés par la valeur string de la séquence retournée par l'expression *items*. Cela signifie que le nœud cible contiendra un seul nœud de texte.

Exemples

© 2016-2022 Altova GmbH

```
for $i in //title return
replace value of node $i with ('Draft Title')
```

8.6 XQuery et bases de données XML

Un document XQuery peut être utilisé pour requêter une base de données XML (BD XML). Actuellement, cette fonction XQuery est uniquement prise en charge pour les bases de données IBM DB2. Le mécanisme de requête d'une BD XML avec XQuery implique essentiellement : (i) indiquer au moteur XQuery qu'un XML doit être requis dans une BD, contrairement à un XML dans un document XML ; et (ii) accéder aux données XML dans la BD.

Les étapes de mise en place de ce mécanisme sont les suivants et sont décrits en détail ci-dessous :

1. Configurer le document XQuery pour requêter la BD XML en insérant le mot-clé `XQUERY` au début du document.
2. En ce qui concerne le document XQuery actif, activer la prise en charge de BD (par le biais de la fenêtre Info) et se connecter à la BD (avec le dialogue Connexion rapide).
3. Dans le document XQuery, insérer les extensions XQuery spécifiques à la BD afin de pouvoir accéder aux données de BD et de les rendre disponibles pour les opérations XQuery.
4. Exécuter le document XQuery dans XMLSpy.

Configurer le document XQuery pour requêter la BD XML

Afin de configurer le document XQuery pour requêter une BD XML, ouvrir le document XQuery (ou créer un nouveau document XQuery) et saisir le mot-clé `XQUERY` (quelle que soit la casse) au début du document (avant le prologue) ; *voir exemples ci-dessous*.

```
XQUERY (: Retrieve details of all customers :)
declare default element namespace "http://www.altova.com/xquery/databases/db2";
<a> {db2-fn:xmlcolumn("CUSTOMER.INFO")} </a>
```

Si le document utilise l'expression optionnelle `xquery version`, le mot-clé `XQUERY` est toujours requis :

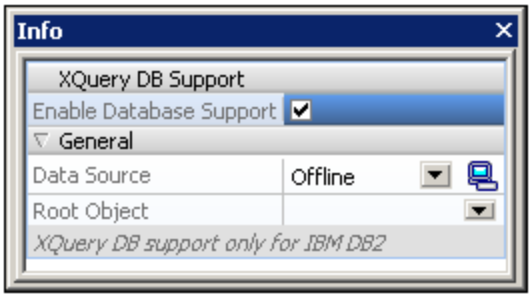
```
XQUERY xquery version "1.0"; (: Retrieve details of all customers :)
declare default element namespace "http://http://www.altova.com/xquery/databases/db2";
<a> {db2-fn:xmlcolumn("CUSTOMER.INFO")} </a>
```

Note : Les moteurs XQuery intégrés de XMLSpy lisent le mot-clé `XQUERY` d'une manière qui indique qu'une BD XML doit être accédée. En conséquence, toute tentative d'exécuter un document XQuery contenant le mot-clé `XQUERY` sur un document XML autre qu'un document contenu dans une BD XML, entraînera une erreur.

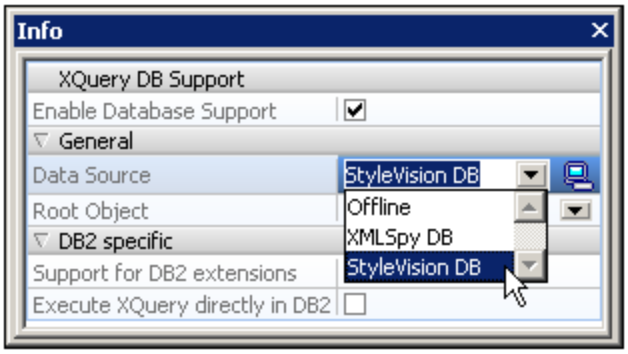
Activer la prise en charge de XQuery et se connecter à la BD

La prise en charge de la BD d'un document XQuery est activée en cochant la case Activer la prise en charge de base de données dans la fenêtre Info (*capture d'écran ci-dessous*). Veuillez noter que la prise en charge BD doit être activée pour chaque document XQuery séparément et qu'à chaque fois qu'un document XQuery est ouvert pour la première fois.

© 2016-2022 Altova GmbH



Lorsque vous activez la prise en charge BD dans la fenêtre Info, un dialogue Connexion rapide s'affiche qui vous permet de vous connecter à une base de données. Actuellement, seules les bases de données IBM DB2 sont prises en charge. La connexion à une BD est décrite dans la section Se connecter à une base de données. Si des connexions à des sources de données existent déjà, elles seront recensées dans la liste de choix Sources de données de la fenêtre Info (*capture d'écran ci-dessous*), et une de ces sources de données peut être sélectionnée en tant que la source de données pour le document XQuery actif. Dans la fenêtre Info, vous pouvez aussi sélectionner l'objet racine parmi les objets disponibles dans la liste de choix Objet Racine.



Le dialogue Connexion rapide (qui vous permet de vous connecter à une BD) peut être accédé à tout moment en cliquant sur l'icône dans la fenêtre Info.

Note : Lorsque vous fermez un document XQuery, la connexion à la BD est fermée également. Si, ultérieurement, vous rouvrez le document XQuery, vous devrez également vous reconnecter à la BD.

Extensions de langage XQuery spécifiques à IBM DB2

Deux fonctions IBM spécifiques à DB2 peuvent être utilisées dans les documents XQuery pour extraire des données depuis une base de données IBM DB2 :

- `db2-fn:xmlcolumn` extrait une colonne XML complète sans chercher ou filtrer la colonne.
- `db2-fn:sqlquery` extrait des valeurs basées sur une déclaration SQL `SELECT`

Les données XML extraites à l'aide de ces fonctions peuvent ensuite être manipulées à l'aide de structures XQuery standard. *Voir exemples ci-dessous*.

`db2-fn:xmlcolumn`: L'argument de la fonction est un littéral de chaîne sensible à la casse qui identifie une colonne XML dans une table. L'argument littéral de chaîne doit être un nom de colonne qualifié de type XML. La fonction retourne toutes les données XML dans la colonne en tant que séquence, sans y appliquer une

condition de recherche. Dans l'exemple suivant, toutes les données de la colonne INFO (XML) de la table CUSTOMER sont retournées dans un élément <newdocelement> de niveau supérieur :

```
XQUERY (: Retrieve details of all customers :)
declare default element namespace "http://www.altova.com/xquery/databases/db2";
<newdocelement> {db2-fn:xmlcolumn("CUSTOMER.INFO")} </newdocelement>
```

Les données extraites peuvent être requises avec des structures XQuery. Dans l'exemple ci-dessous, les données XML extraites depuis la colonne (XML) INFO de la table CUSTOMER est filtrée à l'aide d'une structure XQuery de manière à ce que seuls les profils des clients situés à Toronto sont extraits.

```
XQUERY (: Retrieve details of Toronto customers :)
declare default element namespace "http://www.altova.com/xquery/databases/db2";
<newdocelement> {db2-fn:xmlcolumn("CUSTOMER.INFO")/customerinfo[addr/city='Toronto']}
</newdocelement>
```

Note : dans l'exemple ci-dessus, l'élément de document des fichiers XML dans chaque cellule est customerinfo et le nœud de racine de la séquence XML retournée par db2-fn:xmlcolumn est considérée être un nœud abstrait placé au-dessus des nœuds customerinfo.

db2-fn:sqlquery : la fonction prend une déclaration SQL Select en tant que son argument et retourne une séquence de valeurs XML. La séquence extraite est ensuite requêtée avec des structures XQuery. Dans l'exemple suivant, la colonne INFO est filtrée pour des enregistrements dans la table CUSTOMER comportant un champ CID contenant une valeur située entre 1000 et 1004. Veuillez noter qu'alors que SQL n'est pas sensible à la casse, XQuery l'est.

```
XQUERY (: Retrieve details of customers by Cid:)
declare default element namespace "http://www.altova.com/xquery/databases/db2";

<persons>
   {db2-fn:sqlquery("SELECT info FROM customer WHERE CID>1000 AND CID<1004")/
   <person>
      <id>{data(@Cid)}</id>
      <name>{data(name)}</name>
   </person>}
</persons>
```

Le document XQuery ci-dessus retourne la sortie suivante :

```
<persons xmlns="http://www.altova.com/xquery/databases/db2">
   <person>
      <id>1001</id>
      <name>Kathy Smith</name>
   </person>
   <person>
      <id>1002</id>
      <name>Jim Jones</name>
   </person>
   <person>
      <id>1003</id>
      <name>Robert Shoemaker</name>
   </person>
</persons>
```


© 2016-2022 Altova GmbH

Veuillez noter les points suivants :

- La déclaration d'espace de noms de l'élément par défaut dans le prologue s'applique pour tout le document XQuery et est utilisé pour parcourir le document XML et pour la construction de nouveaux éléments. Cela signifie que le sélecteur XQuery `name` est élargi à `<default-element-namespace>:name`, et que les éléments construits, comme `persons`, se trouvent dans l'espace de noms élément par défaut.
- La déclaration SQL Select n'est pas sensible à la casse.
- La clause `WHERE` de la déclaration Select devrait référencer un autre item de base de données et non un nœud dans le fichier XML en cours d'accession.
- La barre oblique "`/`" située après la fonction `db2-fn:sqlquery` représente le premier item de la séquence retournée et cet item est le nœud contextuel pour une navigation ultérieure.

Exécuter le document XQuery

Afin d'exécuter le document XQuery, sélectionner la commande **Exécution XQuery** (menu **XSL/XQuery**). En alternative, appuyer sur **Alt+F10** ou cliquer sur l'icône d'exécution XQuery . Le résultat de l'exécution est affiché dans un nouveau document.

9 XSLT/XQuery débogueur et profileur

XMLSpy contient un Débogueur XSLT/XQuery et un Profileur XSLT/XQuery afin de vous aider à créer plus rapidement des documents XQuery corrects. Ces deux fonctions sont décrites dans les sous-sections de cette section.

© 2016-2022 Altova GmbH

9.1 Débogueur XSLT et XQuery

Le Débogueur XSLT et XQuery vous permet de tester et de déboguer des feuilles de style XSLT et des documents XQuery. L'interface Débogueur XSLT et XQuery présente des modes simultanés du document XSLT/XQuery, du document résultant et du document XML source. Vous pouvez ensuite procéder étape par étape dans le document XSLT/XQuery. La sortie correspondante est générée étape par étape et, si un fichier de source XML est affiché, la position correspondante dans le fichier XML est marquée par chaque étape. Dans le même temps, les fenêtres dans l'interface fournissent des informations de débogage.

Le Débogueur XSLT et XQuery s'ouvre toujours dans une **session de débogage**. Les sessions de débogage peuvent être de type suivant :

- XSLT 1.0, qui utilise le moteur intégré Altova XSLT 1.0
- XSLT 2.0, qui utilise le moteur intégré Altova XSLT 2.0
- XSLT 3.0, qui utilise le moteur intégré Altova XSLT 3.0
- XQuery 1.0, qui utilise le moteur intégré Altova Altova XQuery 1.0
- XQuery 3.1, qui utilise le moteur intégré Altova Altova XQuery 3.1

Note : XQuery Update Facility 1.0 et XQuery Update Facility 3.0 ne sont actuellement pas pris en charge dans Débogueur XSLT et XQuery.

Le type de session de débogage qui s'ouvre est déterminé automatiquement par le type de document à partir duquel la session de débogage a été ouverte (ci-après appelé le *document actif* ou le *fichier actif*). Les sessions de débogage XSLT sont ouvertes depuis les fichiers XSLT (le choix de la version dépend de la valeur de l'attribut `version` de l'élément `xsl:stylesheet` (ou `xsl:transform`) dans la feuille de style XSLT (`"1.0"` pour XSLT 1.0, `"2.0"` pour XSLT 2.0, et `"3.0"` pour XSLT 3.0)). Les sessions de débogage XQuery s'ouvrent depuis les fichiers XQuery de manière semblable. Si le fichier actif est un fichier XML, la sélection dépend de si vous choisissez d'exécuter un fichier XSLT ou XQuery sur le fichier XML ; dans le premier cas, la sélection dépend aussi de la feuille de style (XSLT 1.0, XSLT 2.0, ou XSLT 3.0).

Cette information est résumée dans la table ci-dessous.

Fichier actif	Fichier associé	Session de débogage
XSLT 1.0	XML; (exigé)	XSLT 1.0 (qui utilise le moteur intégré Altova XSLT 1.0)
XSLT 2.0	XML; (exigé)	XSLT 2.0 (qui utilise le moteur intégré Altova XSLT 2.0)
XSLT 3.0	XML; (exigé)	XSLT 3.0 (qui utilise le moteur intégré Altova XSLT 3.0)
XQuery 1.0	XML; (optionnel)	XQuery 1.0 (qui utilise le moteur intégré Altova XQuery 1.0)
XQuery 3.1	XML; (optionnel)	XQuery 3.1 (qui utilise le moteur intégré Altova Altova XQuery 3.0)
XML	XSLT 1.0, 2.0, ou 3.0, ou XQuery 1.0 ou 3.1; (exigé)	XSLT 1.0, 2.0 ou 3.0, ou XQuery 1.0 ou 3.1. La version XSLT dépend de la valeur de l'attribut `version` de l'élément `xsl:stylesheet` (ou `xsl:transform`) de la feuille de style XSLT.

Pour plus de détails concernant les trois moteurs Altova, veuillez voir la section Informations de moteur XSLT et XQuery dans l'Annexe.

Automatisation des tâches XSLT et XQuery avec l'Altova RaptorXML Server 2023

Altova RaptorXML Server contient le Validateur XML, les moteurs XSLT 1.0, 2.0 et 3.0, et les moteurs XQuery 1.0 et 3.1 d'Altova. Le programme peut être utilisé depuis la ligne de commande, par le biais d'une interface COM, dans des programmes Java et dans des applications .NET pour valider des documents XML, transformer des documents XML avec des feuilles de style XSLT et exécuter des documents XQuery.

Les tâches XSLT et XQuery peuvent ensuite être automatisées avec l'utilisation d'Altova RaptorXML Server. Par exemple, vous pouvez créer un fichier batch qui appelle Altova RaptorXML pour transformer un ensemble de documents XML ou pour exécuter un ensemble de documents XQuery. Voir la documentation RaptorXML Server pour plus de détails.

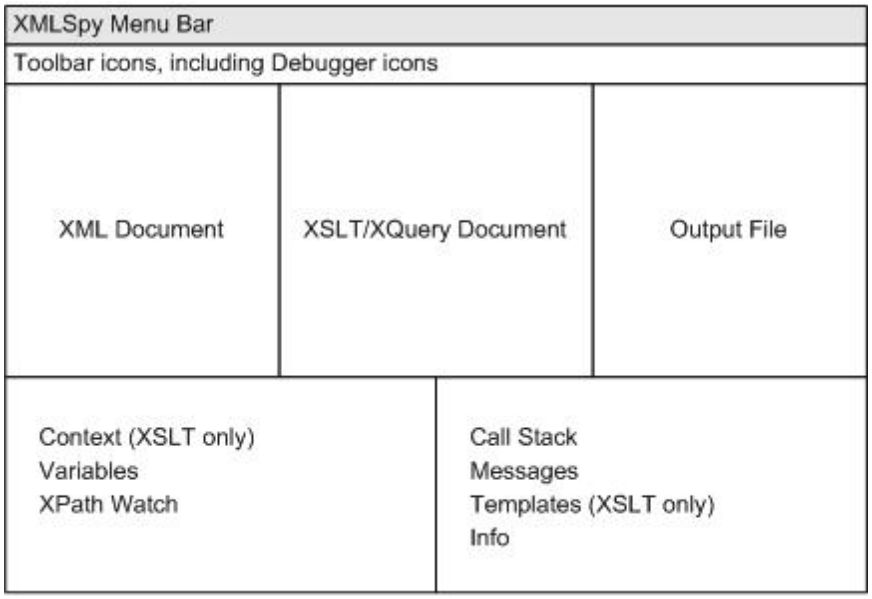
Site Altova : Débogueur XSLT, Débogueur XQuery

9.1.1 Mécanisme et interface

Le mécanisme de base utilisé pour déboguer les fichiers XSLT et XQuery à l'aide du Débogueur XSLT et XQuery est indiqué ci-dessous. Veuillez noter qu'il existe une différence entre une session de débogage et le débogueur, bien que tous deux sont lancés avec la commande **XSL/XQuery | Démarrer débogueur / OK**. Vous devez tout d'abord démarrer la session de débogage puis progresser dans le document XSLT ou XQuery à l'aide du débogueur.

- Ouvrir une session de débogage (avec la commande **XSL/XQuery | Démarrer débogueur / OK**). La session appropriée (XSLT 1.0, XSLT 2.0, XSLT 3.0, XQuery 1.0 ou XQuery 3.1) est sélectionnée sur la base du fichier actif (*voir* Débogueur XSLT et XQuery). Le Débogueur XSLT et XQuery fonctionne uniquement dans le Mode Texte et Grille. Si le mode du document actif n'est pas Texte ou Grille lors du démarrage de la session de débogage, vous serez invité à passer au mode Texte, qui est le mode par défaut du Débogueur XSLT et XQuery. Dans le Dialogue Paramètres de débogage, vous pouvez aussi décider de configurer cette option en permanence.
- Passer dans le document XSLT ou XQuery avec les commandes **Pas à pas détaillé, Pas à pas sortant**, et **Pas à pas principal** dans le menu **XSL/XQuery**. (Si, à ce moment, vous cliquez sur la commande **Démarrer débogueur / OK**, le débogueur passera sur toute la transformation ou exécution, et ne s'arrêtera qu'aux points d'arrêt. Si aucun point d'arrêt n'a été défini, il passera à travers toute la transformation en une étape, sans montrer de résultats de débogage.) Si un fichier XML est associé avec la session, les emplacements correspondants dans le fichier XML sont marqués. Simultanément, la sortie pour les étapes correspondantes est générée dans le fichier de résultat et le document de résultat est construit étape par étape. Ainsi, vous pouvez analyser ce que chaque déclaration du fichier XSLT ou XQuery fait.

© 2016-2022 Altova GmbH



En alternative au mode des trois documents (XML, XSLT/XQuery, Sortie) affichés ci-dessus, un mode de deux documents (XSLT/XQuery et Sortie), ou un mode d'un des documents peut être sélectionné.

- Pendant qu'une session de débogage est ouverte, les fenêtres d'information dans l'interface fournissent des informations concernant les divers aspects de la transformation/exécution (Variables, Suivi XPath, Pile d'appel, Messages, Info, etc).
- Pendant qu'une session de débogage est ouverte, vous pouvez interrompre le débogueur (ce qui n'est pas la même chose que d'arrêter la session de débogage) pour effectuer des changements à un des documents. Toutes les fonctions d'édition qui sont disponibles dans votre environnement XMLSpy sont aussi disponibles pour l'édition d'un fichier pendant une session de débogage. Lorsque le débogueur est arrêté, l'interface du Débogueur XSLT et XQuery reste ouverte et vous pouvez accéder à toutes les informations dans la fenêtre d'Information. Une fois avoir arrêté le débogueur dans une session de débogage, vous pouvez redémarrer le débogueur (depuis le début du document XSLT/XQuery) dans la même session de débogage.
- Les points d'arrêt peuvent être configurés dans le fichier XSLT de manière à interrompre le traitement au niveau des points sélectionnés. Cela accélère les sessions de débogage étant donné que vous n'avez pas à passer manuellement par chaque déclaration dans le document XSLT ou XQuery.
- Les points de trace peuvent être configurés dans le fichier XSLT. Pour obtenir des instructions lorsqu'un point de trace est défini, la valeur de cette instruction est sortie lorsque l'instruction est atteinte.
- Interrompre une session de débogage. Cela permet de fermer l'interface du Débogueur XSLT et XQuery et vous permet de retourner à votre environnement XMLSpy précédent. L'information contenue dans les fenêtres d'information n'est plus disponible. En revanche, l'information de point d'arrêt et de trace est retenue dans les fichiers respectifs jusqu'à ce que le fichier se ferme. (Donc si vous lancez une autre session de débogage impliquant un fichier contenant des points d'arrêt, les points d'arrêt s'appliqueront dans la nouvelle session de débogage récemment ouverte.)

Note : La barre d'outils du débogueur avec les icônes de débogueur apparaît automatiquement lorsqu'une session de débogage est lancée.

9.1.2 Commandes et icônes de barre d'outils

Les commandes de débogage sont disponibles dans le menu **XSL/XQuery** et en tant qu'icônes de barre d'outils. Les icônes du débogueur sont automatiquement rendues disponibles dans la barre outils lorsqu'une session de débogage est ouverte. Ces icônes sont recensés ci-dessous.



Démarrer débogueur / OK (Alt+F11)

Démarre ou reprend le traitement du document XSLT/XQuery jusqu'à la fin. Si des points d'arrêt ont été définis, le traitement s'interrompra à cet endroit. Si la session du débogueur n'a pas été démarrée, cette touche lancera la session et s'arrêtera au premier nœud à traiter. Si la session est en cours d'exécution, le document XSLT/XQuery sera traité jusqu'à la fin ou jusqu'à ce que le point d'arrêt suivant survienne. Si des points de trace ont été définis, la valeur de l'instruction pour laquelle le point de trace a été défini est affichée dans la fenêtre Trace lorsque cette instruction est atteinte.



Afficher uniquement le document actif

Agrandit la fenêtre du document actuellement actif dans l'interface du débogueur.



Afficher XSLT/XQuery et sortie

Affiche les documents XSLT et de Sortie dans leurs fenêtres et dissimule simultanément le document XML.



Afficher XML, XSLT/XQuery et sortie

Affiche les documents XML, XSLT/XQuery et de Sortie. Il s'agit du mode par défaut lorsqu'un document XML est associé pour la session de débogage.



© 2016-2022 Altova GmbH

Arrêter débogueur

Arrête le débogueur. Ce n'est pas la même chose que d'arrêter la session de débogage dans laquelle le débogueur est en cours d'exécution. Cette fonction est pratique si vous souhaitez éditer un document pendant une session de débogage ou utiliser des fichiers alternatifs dans le cadre de la même session de débogage. Après avoir arrêté le débogueur, vous devez redémarrer le débogueur pour commencer au début du document XSLT/XQuery.



Pas à pas détaillé (F11)
Procède par étapes individuelles à travers de tous les nœuds et les expressions XPath de la feuille de style. Cette commande est aussi utilisée pour redémarrer le débogueur après qu'il ait été stoppé.



Pas à pas sortant (Ctrl+F11)

Passe au-dessus du nœud actuel vers le nœud suivant au même niveau ou vers le nœud suivant situé au niveau supérieur suivant à celui du nœud actuel. Cette commande est aussi utilisée pour redémarrer le débogueur après qu'il ait été stoppé.



Pas à pas principal (Shift+F11)

Sort du nœud actuel pour aller vers l'enfant suivant du nœud parent, ou vers le nœud suivant situé au niveau supérieur suivant à celui du nœud parent.



Afficher le nœud d'exécution actuel

Affiche/sélectionne le nœud d'exécution actuel dans le document XSLT/XQuery et le nœud correspondant dans le document XML. Cette fonction est utile lorsque vous avez cliqué dans d'autres onglets qui montrent ou marque un code spécifique dans la feuille de style XSLT ou le fichier XML, et que vous souhaitez retourner à l'endroit où vous étiez avant de procéder à cette opération.



Redémarrer le débogueur

Vide la fenêtre de sortie et redémarre la session de débogage avec les fichiers sélectionnés actuellement.



Insérer/Supprimer le point d'arrêt (F9)

Insère ou supprime un point d'arrêt à l'emplacement actuel du curseur. Les points d'arrêt inline peuvent être définis pour des nœuds dans les documents XSLT/XQuery et XML, et déterminent où le traitement devrait s'interrompre. Lorsque vous définissez un point d'arrêt, une ligne en pointillés rouge apparaît au-dessus du nœud. Les points d'arrêt ne peuvent pas être définis sur les nœuds de fermeture, et les points d'arrêt des attributs dans les documents XSLT seront ignorés. Cette commande est également disponible en cliquant avec la touche de droite sur l'emplacement du point d'arrêt.



Insérer/Supprimer point de trace (Shift+F9)

Insère ou supprime un point de trace à l'emplacement actuel du curseur. Les points de trace peuvent être définis pour des nœuds dans les documents XSLT. Pendant le débogage, lorsqu'une déclaration à point de trace est atteinte, le résultat de cette déclaration est sortie dans la fenêtre Trace. Lorsque vous définissez un point de trace, une ligne en pointillés bleu apparaît au-dessus du nœud. Les points de trace ne peuvent pas être définis pour des nœuds de fermeture. Cette commande est également disponible en cliquant avec la touche de droite sur l'emplacement du point de trace.



Activer/Désactiver point d'arrêt (CTRL+F9)

Cette commande (il n'existe pas d'icône de barre d'outils) active ou désactive des points d'arrêt déjà définis. Le marquage de point d'arrêt rouge passe au gris lorsque le point d'arrêt est désactivé. Le débogueur ne s'arrête pas aux points d'arrêt désactivés. Pour désactiver/activer un point d'arrêt, placer le curseur dans le nom de ce nœud et cliquer sur la commande Activer/Désactiver point d'arrêt. Cette commande est également disponible en cliquant avec la touche de droite sur l'emplacement du point d'arrêt.



Activer/Désactiver point de trace (Shift+CTRL+F9)

Cette commande (il n'existe pas d'icône de barre d'outils) active ou désactive des points de trace déjà définis. Le marquage de point de trace bleu passe au gris lorsque le point de trace est désactivé. Aucune sortie ne sera effectuée dans la fenêtre Trace pour les points de trace désactivés. Pour désactiver/activer un point de

© 2016-2022 Altova GmbH

trace, placer le curseur dans le nom de ce nœud et cliquer sur la commande Activer/Désactiver point de trace. Cette commande est également disponible en cliquant avec la touche de droite sur l'emplacement du point de trace.



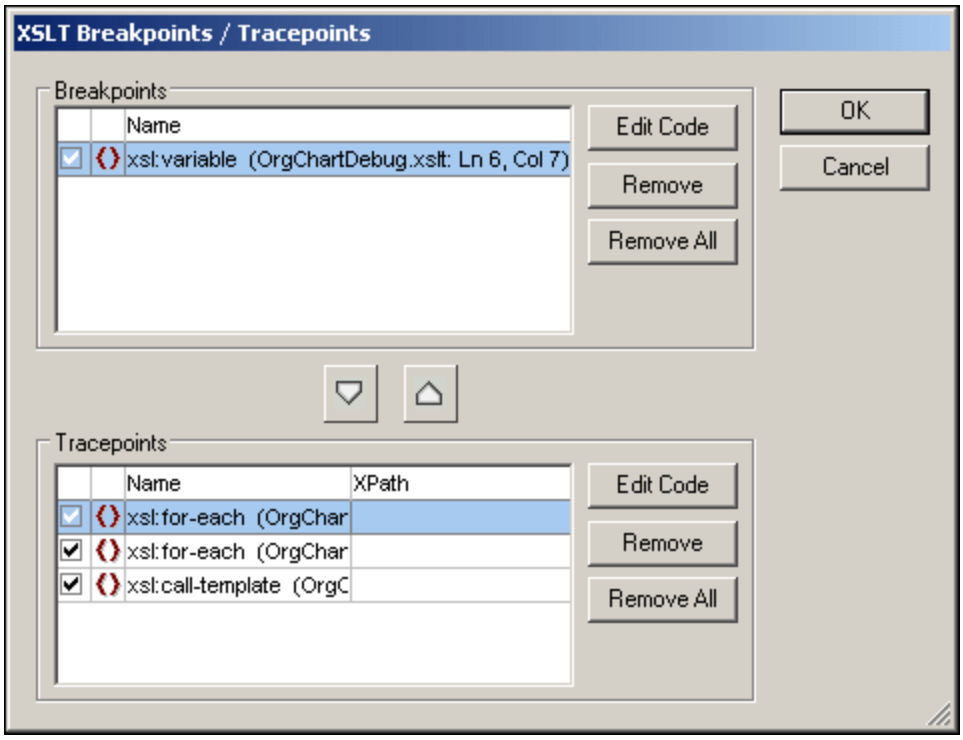
Terminer la session de débogage

Termine la session de débogage et vous ramène au mode XMLSpy normal qui était actif avant le démarrage de la session de débogage. Selon les configurations définies dans le dialogue Paramètres de débogage XSLT/XQuery, les documents de sortie qui ont été ouverts pour la session de débogage restent ouverts ou non.



Points d'arrêt XSLT / Dialogue points de trace

Cette commande ouvre le dialogue points d'arrêt / points de trace XSLT/XQuery, qui affiche une liste de tous les points d'arrêt/ points de trace actuellement définis (y compris les points désactivés) dans tous les fichiers dans la session de débogage actuelle.

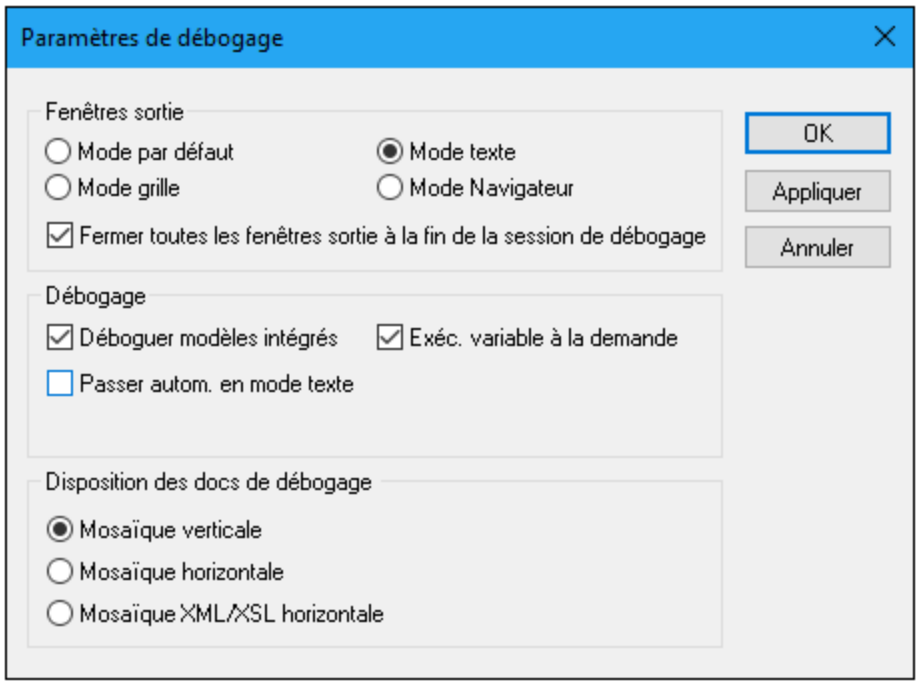


Les cases à cocher indiquent si un point d'arrêt / point de trace est activé (coché) ou désactivé. Vous pouvez supprimer le point d'arrêt/point de trace marqué en cliquant sur la touche correspondante **Supprimer** et en supprimant tous les points d'arrêt ou les points de trace en cliquant sur la touche correspondante **Supprimer**

tout. La touche correspondante **Éditer Code** vous emmène directement au point d'arrêt/point de trace sélectionné dans le fichier.

9.1.3 Paramètres du débogueur

Le dialogue Paramètres du Débogueur XSLT et XQuery vous permet de configurer les options de débogage et de sortie qui sont applicable pour toutes les sessions de débogage. Pour accéder au dialogue Paramètres, cliquer sur **XSL/XQuery | Paramètres de débogage** ou cliquer sur l'icône  dans la barre d'outils. Les paramètres sont décrits ci-dessous.



Fenêtre Sortie

Définit le mode de la fenêtre de document de la sortie (Défaut, Texte, Grille ou Navigateur). Le Mode par défaut est celui sélectionné pour un type de fichier (identifié par son extension de fichier, par exemple `.xslt` ou `.xq`) dans la section *Types de fichier* du dialogue d'Options (Outils | Options). En ce qui concerne les transformations XSLT, le type de fichier de sortie est défini dans le fichier XSLT. En ce qui concerne les exécutions XQuery, le type de fichier de sortie est déterminé par le format de sérialisation que vous choisissez dans les paramètres XQuery de ce dialogue (*voir ci-dessous*).

L'option Fermer toutes les fenêtres de sortie vous donne la possibilité de laisser les fenêtres de document de sortie ouvertes qui ont été ouvertes pendant la session de débogage lorsque la session de débogage se termine.

© 2016-2022 Altova GmbH

Débogage

Le paramètre Déboguer les modèles intégrés permet au débogueur de **passer dans** le code des modèles intégrés le cas échéant. Il n'est pas lié à l'**affichage** des modèles intégrés lorsque vous cliquez sur ce type d'entrée de modèle dans l'onglet Modèles, ou si la pile d'appels montre un nœud provenant du fichier de modèle intégré.

Le débogueur XSLT ne marche que dans le Mode Texte ou Grille. L'option Changement automatique au Mode Texte vous permet de passer automatiquement au Mode Texte d'un document pour le débogage lorsqu'un document ne se trouve pas dans le Mode Texte ou Grille. (Le débogueur XQuery fonctionne uniquement dans le Mode Texte.) Si la case *Exécution variable à la demande* est cochée, la définition d'une variable suivra le pas à pas détaille lorsque la variable est appelée. Dans le cas contraire, le débogueur n'effectuera pas un pas à pas détaillé dans la définition variable lorsqu'il rencontrera un appel pour une variable ; mais il passera à l'étape suivante.

Mise en page des documents du débogueur

Les Documents du débogueur sont les documents qui sont ouverts dans le débogueur. Vous pouvez sélectionner si ces documents doivent être placés en mosaïque verticale, horizontale ou en XML/XSLT horizontal avec le document de résultat placés en mosaïque verticale relativement à XML et XSLT.

9.1.4 Démarrer une session de débogage

La manière la plus simple de démarrer une session de débogage est d'en lancer une depuis un fichier XSLT, XQuery ou XML. Si le fichier associé nécessaire (voir Table des fichiers associés) a déjà été attribué au fichier actif, la session de débogage est lancée immédiatement. Sinon, vous êtes invité à choisir le fichier associé nécessaire. Puisque les fichiers XQuery ne nécessitent pas d'association de fichier XML et n'en contiennent pas non plus, vous pouvez choisir d'être invité (ou pas) à choisir une association de fichier XML en option à chaque fois que vous lancerez une session de débogage XQuery depuis un fichier XQuery.

Associations prédéfinies

Les associations prédéfinies sont uniquement pertinentes pour les sessions de débogage de XSLT, et réfèrent à des cas dans lesquels l'attribution de fichier associé est déjà présente dans le fichier actif. Afin d'effectuer une attribution dans un fichier XML ou XSLT, procéder comme suit :

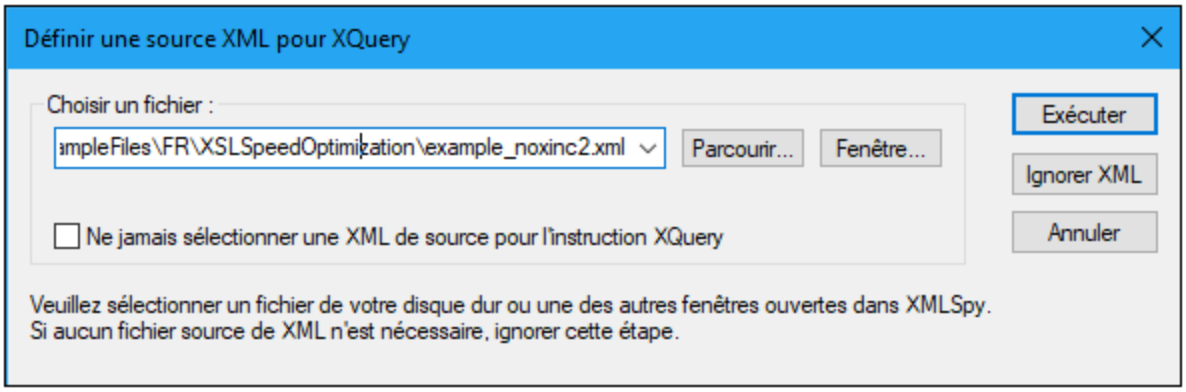
- Dans les fichiers XML : ouvrir le fichier, cliquer sur **XSL/XQuery | Assigner XSL**, et sélectionner le fichier XSLT.
- Dans les fichiers XSLT : ouvrir le fichier, cliquer sur **XSL/XQuery | Assigner fichier XML modèle...**, et sélectionner le fichier XML.

Lorsque vous cliquez sur **XSL/XQuery | Démarrer le débogueur / OK**, la session de débogage est lancée directement, donc sans que vous soyez invité à choisir un fichier à associer avec le fichier actif.

Assignation directe

Si aucune association prédéfinie n'est présente dans le fichier actif, vous êtes invité à choisir une association. Lorsque vous sélectionnez **XSL/XQuery | Démarrer débogueur / OK**, les situations suivantes se produisent :

- Pour les fichiers XML : vous êtes invité à choisir un fichier XSLT ou XQuery.
- Pour les fichiers JSON : vous êtes invité à choisir un fichier XSLT, XQuery ou XQuery Update
- Pour les fichiers XSLT : vous êtes invité à choisir un fichier XML.
- Pour les fichiers XQuery : vous avez la possibilité de choisir un fichier XML, option que vous pouvez sauter.



(Le dialogue affiché dans la capture d'écran apparaît lorsque vous commencez une session de débogage à partir d'un fichier XQuery.)

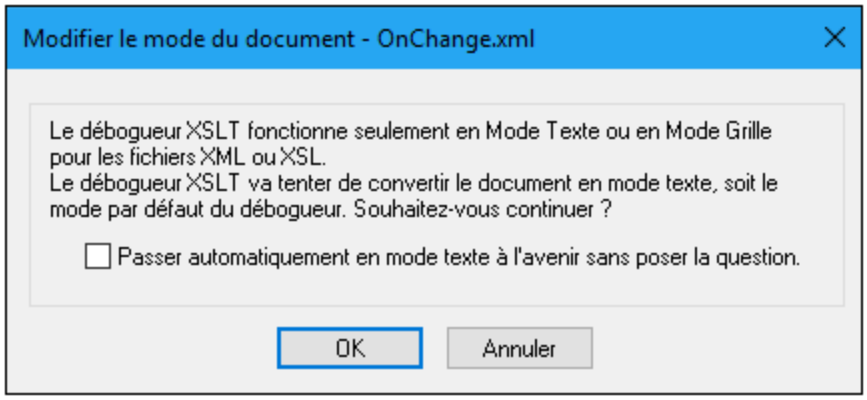
Une fois avoir sélectionné le fichier associé ou avoir sauté une association optionnelle, la session de débogage démarre.

Méthode alternative d'association de fichier

Dans le dialogue Propriétés de projet, vous pouvez effectuer des associations prédéfinies. Cliquer sur **Projet | Propriétés de projet**, et assigner les fichiers requis en cliquant sur la case **Utiliser ce XSL** / **Utiliser ce XML**.

Mode débogueur

Le Débogueur XSLT et XQuery fonctionne uniquement dans le Mode Texte et le Mode de Grille amélioré. Si votre fichier XML ou XSLT est ouvert dans un autre mode que Texte ou Grille, ou si un fichier SPS est associé avec un fichier XML, le dialogue suivant s'affiche lorsque vous démarrez une session de débogage impliquant un de ces fichiers.



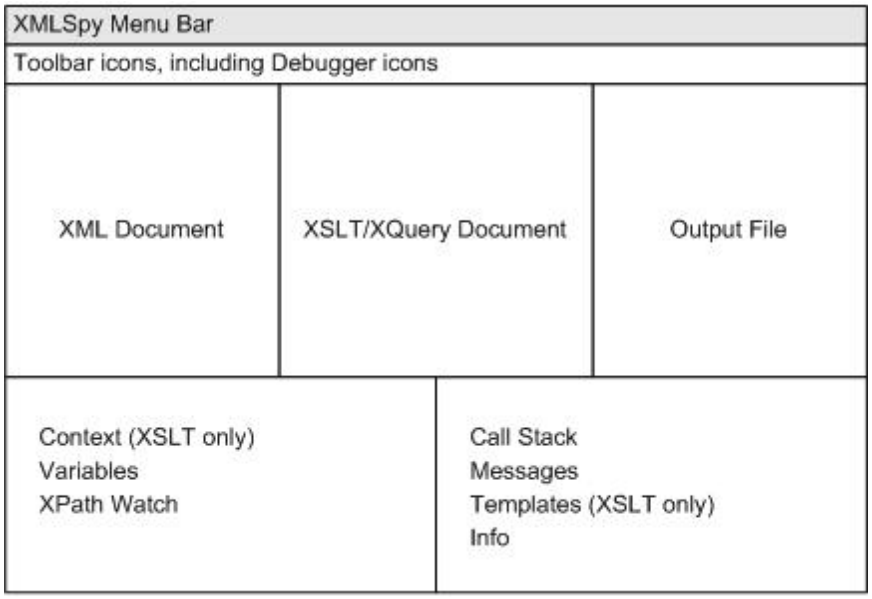
 © 2016-2022 Altova GmbH

Cliquer sur **OK** pour ouvrir le document en Mode Texte. Veuillez noter que les fichiers XQuery sont toujours affichés dans le Mode Texte.

9.1.5 Fenêtres d'Information

Les fenêtres d'information qui s'ouvrent dans l'interface du Débogueur XSLT et XQuery pendant une session de débogage contiennent des informations concernant des aspects variés de la transformation XSLT ou de l'exécution XQuery. Cette information est importante pour vous aider à déboguer vos fichiers XSLT et XQuery.

Il existe huit fenêtres d'information dans les sessions de débogage XSLT et cinq fenêtres d'information dans les sessions de débogage XQuery. Ces fenêtres sont organisées en deux groupes par défaut qui sont situés en bas de l'interface du Débogueur XSLT et XQuery (*voir illustration ci-dessous*). Ces fenêtres et l'information qu'elles affichent sont décrites en détail dans cette section.



La mise en page par défaut de l'interface du Débogueur XSLT et XQuery.

Le premier groupe des fenêtres d'information affiche les fenêtres suivantes dans des onglets regroupés dans une seule fenêtre :

- Contexte (uniquement pour les sessions de débogage XSLT)
- Variables
- Suivi XPath

Le second groupe des fenêtres d'information affiche les fenêtres suivantes dans des onglets regroupés dans une seule fenêtre :

- Pile d'appels
- Messages
- Modèles (uniquement pour les sessions de débogage XSLT)

- Info
- Trace

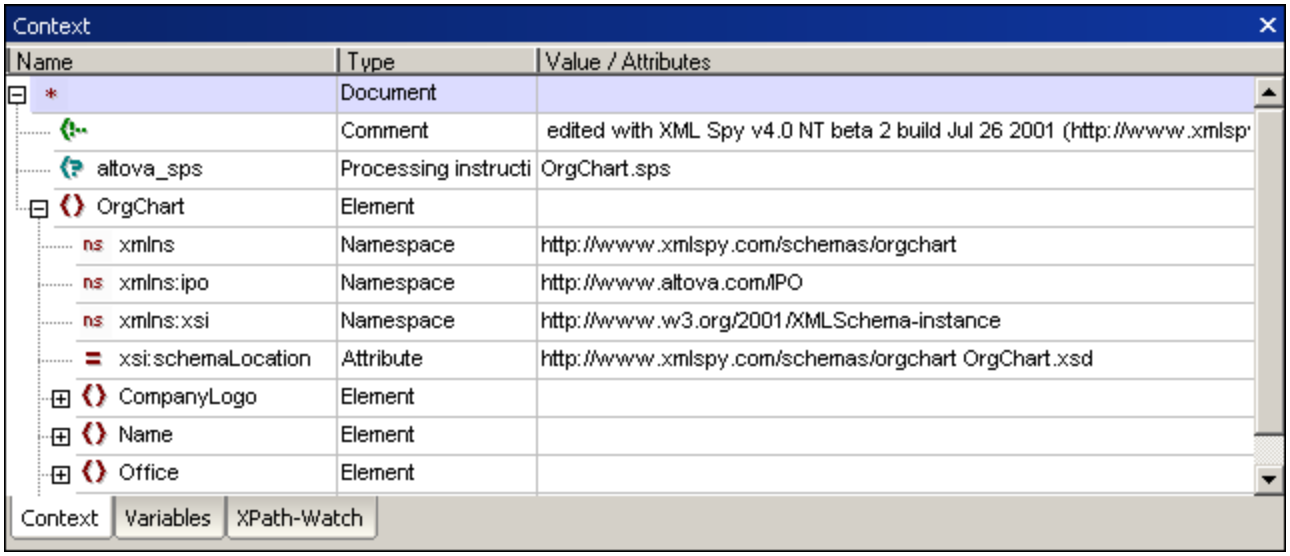
C'est pourquoi vous verrez, dans la mise en page par défaut, deux groupes de fenêtre, chacun présentant des onglets pour les différentes fenêtres qu'ils contiennent. Un onglet est actif à la fois. Ainsi, par exemple, pour afficher des informations concernant les Variables dans le premier groupe de fenêtre d'information, cliquer sur l'onglet **Variables**. La fenêtre d'information Variables s'ouvre et les fenêtres d'information Contexte et Suivi XPath sont dissimulées. Veuillez noter que dans certains onglets, vous pouvez utiliser l'affichage d'information en tant qu'outils de navigation : cliquer sur un item pour vous amener sur cet item dans le fichier XML, XSLT ou XQuery. Voir la documentation des fenêtres d'information respectives (Contexte, Pile d'appels, Modèles) pour plus de détails.

Les deux groupes d'information peuvent être redimensionnés en tirant leur bordure. Des fenêtres individuelles peuvent être glissées hors du groupe les contenant en cliquant sur le nom de l'onglet et en glissant la fenêtre hors du groupe. Une fenêtre peut être ajoutée à un groupe en glissant sa barre de titre dans la barre de titre du groupe. Veuillez noter qu'il n'y a pas de bouton de réinitialisation pour retourner la mise en page à la mise en page par défaut.

9.1.5.1 Fenêtre Contexte

La fenêtre Contexte est uniquement disponible dans les sessions de débogage XSLT ; elle n'est pas disponible dans les sessions de débogage XQuery.

Au cours du traitement de la feuille de style XSLT, le contexte du processeur se trouve toujours dans le cadre d'un modèle que correspond à certaines séquences (de nœuds ou de valeurs atomiques). La fenêtre Contexte affiche le contexte de traitement actuel, qui peut être une séquence de nœuds, un nœud simple ou une valeur atomique (comme une chaîne). Selon le type d'item contextuel, sa/ses valeur/s est/sont affichée/s. Par exemple, si l'item de contexte est un élément, les attributs de l'élément sont affichés. Si l'item de contexte est un attribut ou un nœud de texte, la valeur du nœud est affichée.

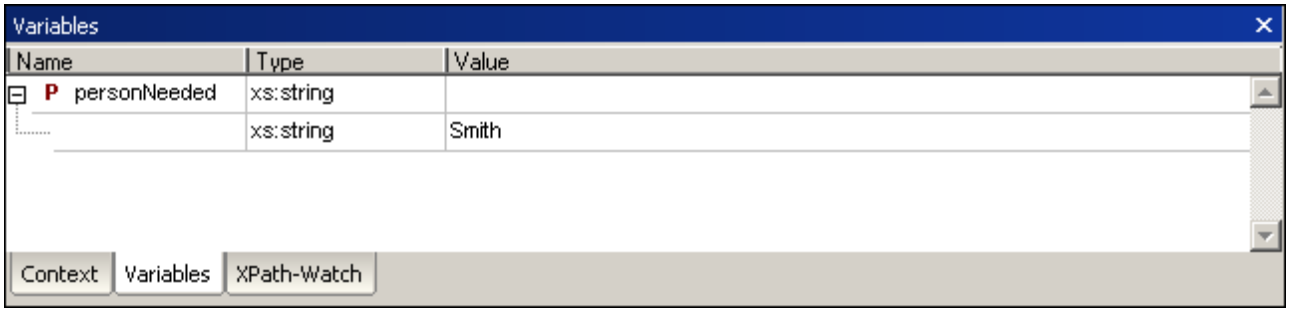


Cliquer sur une entrée dans la fenêtre Contexte pour afficher cet item dans le document XML. Si le document XML n'est pas affiché actuellement dans l'interface, une fenêtre pour le document XML sera ouverte.

© 2016-2022 Altova GmbH

9.1.5.2 Fenêtre Variables

La fenêtre Variables est disponible dans les sessions de débogage XSLT et XQuery. Elle indique les variables et les paramètres qui sont utilisés dans le document XSLT/XQuery lorsqu'ils sont dans leur étendu d'application, et leurs valeurs.

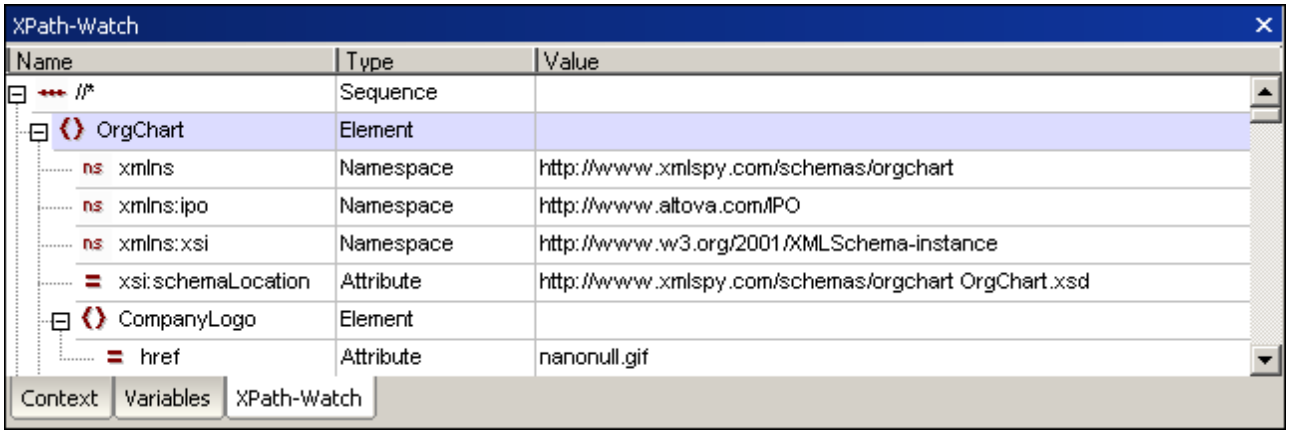


Les paramètres sont indiqués par `P`, les variables globales (déclarées au niveau supérieur d'une feuille de style) sont indiquées par `G`, et les variables locales (déclarées dans un modèle XSLT) sont indiquées par `L`. Le type des valeurs des variables et des paramètres est aussi indiqué par des icônes dans le champ Valeur. Les types suivants sont distingués : Ensemble de nœud, Fragment de nœud, Chaîne, Nombre, Booléenne.

9.1.5.3 Fenêtre Suivi XPath

La fenêtre Suivi XPath est disponibles dans les sessions de débogage XSLT et XQuery.

Elle vous permet de saisir des expressions XPath que vous souhaitez évaluer dans un ou plusieurs contextes. Alors que vous parcourez le document XSLT, l'expression XPath est évaluée dans le contexte actuel et le résultat est affiché dans la colonne Valeur.



Pour saisir une expression XPath, double-cliquer dans le champ de saisie sous la colonne Name et saisir le XPath. En alternative, glisser une expression XPath depuis un fichier et la déposer dans la fenêtre Suivi XPath.

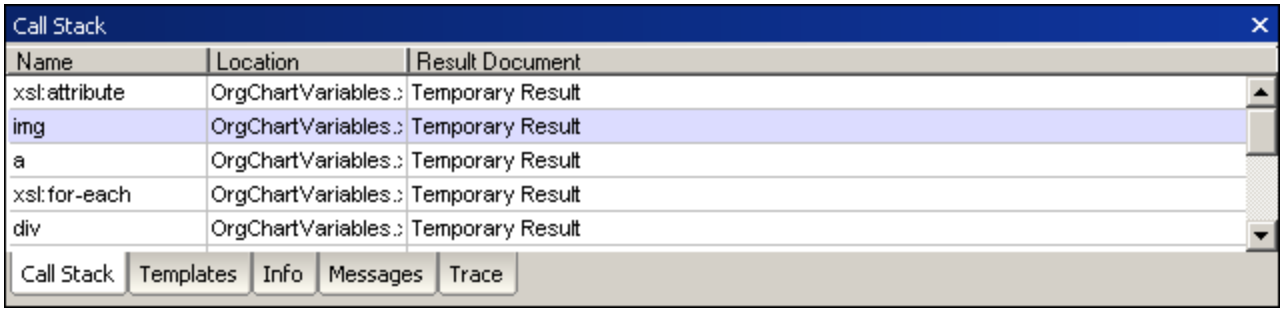
Utiliser les expressions qui sont correctes conformément à la version XPath qui correspond à la version XSLT de la feuille de style XSLT (XPath 1.0 pour XSLT 1.0, XPath 2.0 pour XSLT 2.0, et XPath 3.0 pour XSLT 3.0).

Note : Si les espaces de noms ont été utilisés dans le fichier XML ou XSLT, vous devez utiliser les préfixes d'espace de noms corrects dans vos expressions XPath.

9.1.5.4 Fenêtre Pile d'appel

La fenêtre Pile d'appels est affichée dans les sessions de débogage XSLT et XQuery.

La fenêtre Pile d'appels affiche une liste des modèles et instructions XSLT précédemment traités, le modèle/instruction actuel apparaissant en haut de la liste.



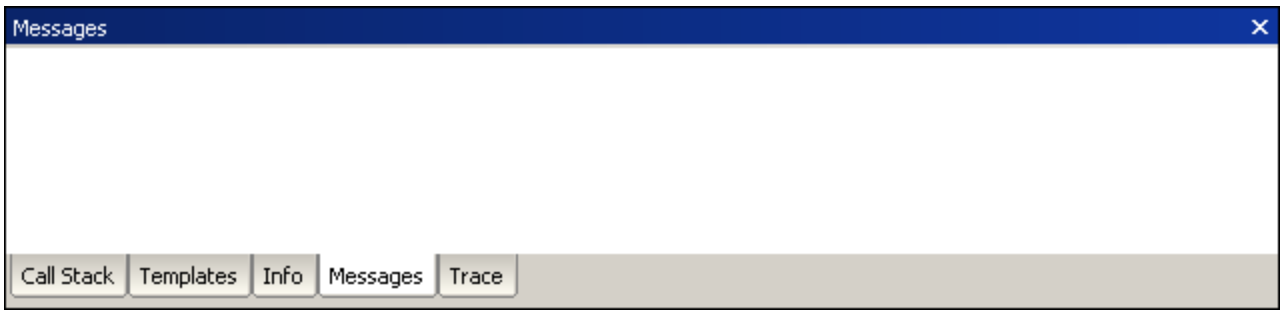
Cliquer sur une entrée dans cette fenêtre pour afficher le modèle/instruction XSLT sélectionné dans la fenêtre de document XSLT. Cliquer sur un modèle/instruction qui référence un modèle intégré permet de marquer le modèle intégré dans une fenêtre séparée qui affiche tous les modèles intégrés.

9.1.5.5 Fenêtre Messages

La fenêtre Messages est affichée dans les sessions de débogage XSLT et XQuery.

XSLT 1.0, XSLT 2.0 et XSLT 3.0

Dans les sessions de débogage XSLT, l'onglet Messages affiche des erreurs de message, l'instruction `xsl:message`, ou tout message d'erreur pouvant se produire durant le débogage.



© 2016-2022 Altova GmbH

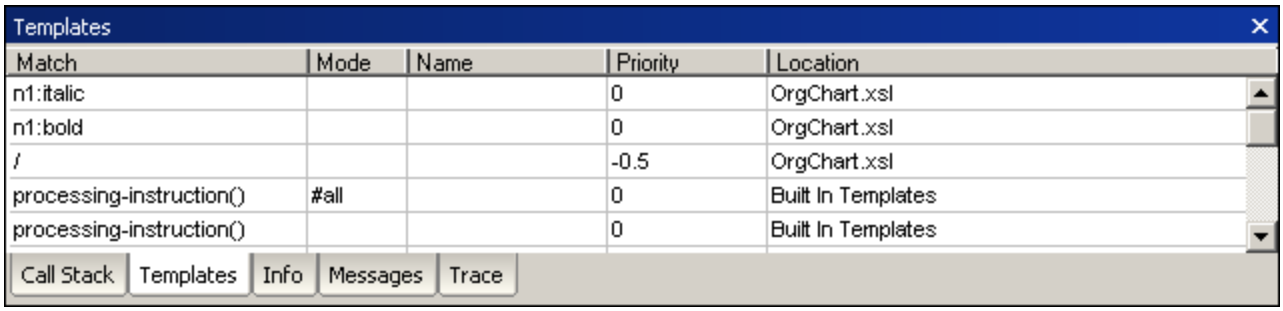
XQuery

Dans les sessions de débogage XQuery, la fenêtre Messages affiche des messages d'erreur.

9.1.5.6 Fenêtre Modèles

La fenêtre Modèles (*voir capture d'écran*) est uniquement disponible dans les sessions de débogage XSLT et XQuery ; elle n'est pas disponible dans les sessions de débogage XQuery.

La fenêtre Modèles affiche les différents modèles utilisés dans la feuille de style XSLT, y compris les modèles intégrés et les modèles nommés. Les modèles correspondants sont recensés par les nœuds auxquels ils correspondent. Les modèles nommés sont recensés par leur nom. En ce qui concerne les deux types de modèles, le mode, la priorité et l'emplacement du modèle sont affichés.

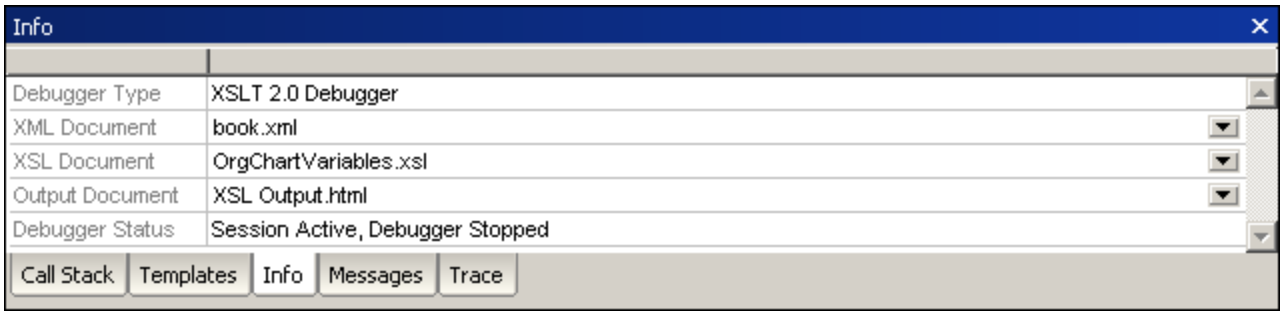


Dans la capture d'écran ci-dessus, la feuille de style XSLT contient trois modèles correspondants : un modèle qui correspond au nœud `/` du document et des modèles qui correspondent aux nœuds `n1:italic` et `n1:bold`. Tous les autres modèles sont des modèles intégrés (indiqués par une entrée à cet effet dans le champ *Location*).

Cliquer sur une entrée dans cette fenêtre pour marquer le modèle dans la fenêtre de document XSLT. Si vous cliquez sur un modèle intégré, le modèle est marqué dans une fenêtre séparée qui affiche tous les modèles intégrés.

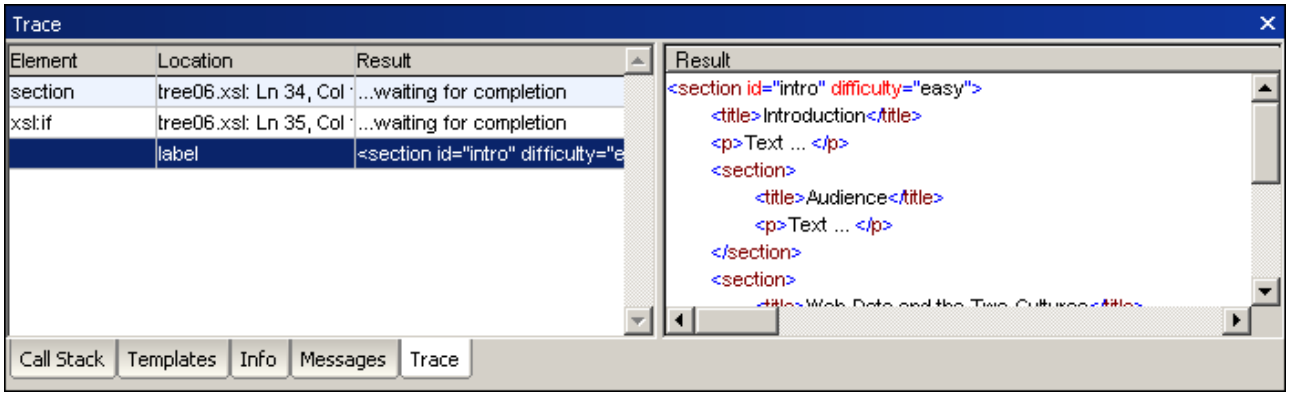
9.1.5.7 Fenêtre Info

La fenêtre Info est disponible dans les sessions de débogage XSLT et XQuery. Elle fournit des méta-informations concernant la session de débogage actuelle. Cette information comprend le type de débogueur utilisé, les noms des documents source et sortie et le statut du débogueur.



9.1.5.8 Fenêtre Trace

La fenêtre Trace est affichée dans les sessions de débogage XSLT et XQuery.



La fenêtre Trace contient l'élément pour lequel le point de trace a été défini, son emplacement dans la feuille de style XSLT et le résultat généré lorsque cet élément est exécuté. Cliquer sur une ligne dans le côté gauche de la fenêtre pour afficher le résultat complet à droite.

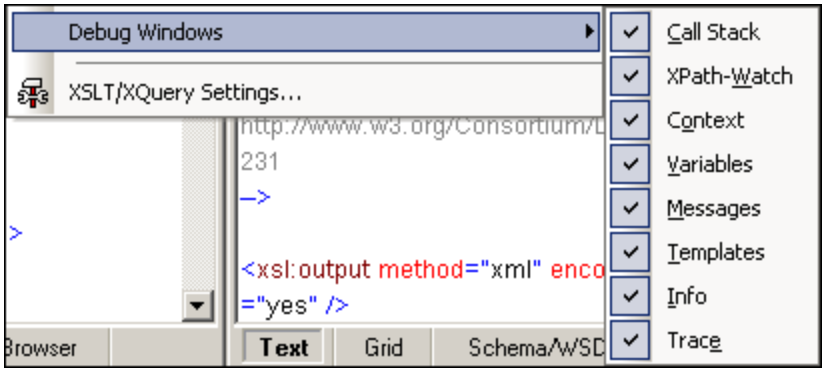
9.1.5.9 Disposer les fenêtres d'Information

Les fenêtres d'Information peuvent être disposées dans l'interface du Débogueur XSLT et XQuery. Dans l'interface, les fenêtres peuvent être ancrées, elles peuvent y flotter et peuvent être disposées comme une série de volets dans une fenêtre. Vous pouvez utiliser les mécanismes suivants pour disposer les fenêtres.

Menu

Dans le menu **XSL/XQuery**, placer le curseur au-dessus de l'item **Fenêtres de débogage** pour ouvrir la liste des fenêtres d'information. Vous pouvez dissimuler ou afficher des fenêtres individuelles en cliquant sur la fenêtre.

© 2016-2022 Altova GmbH



Cela permet d'activer et de désactiver l'affichage de la fenêtre.

Menu contextuel

Le menu contextuel peut être accédé en cliquant avec la touche de droite sur un onglet de fenêtre ou sur une barre de titre.



Cliquer sur l'option requise pour ancrer la fenêtre, la faire flotter ou la dissimuler.

Glisser/déposer

Vous pouvez faire glisser une fenêtre en la déplaçant par son onglet puis en la plaçant à l'endroit désiré.

En outre, vous pouvez ancrer la fenêtre dans une autre fenêtre ou sur l'interface à l'aide des commandes de placement qui apparaissent lorsque vous déplacez une fenêtre :

- Lorsque vous déplacez une fenêtre sur une autre fenêtre, une commande de placement apparaît (*voir capture d'écran ci-dessous*). Cette commande est divisée en cinq secteurs de placement. Lorsque vous relâchez le bouton de la souris sur un de ces secteurs, la fenêtre déplacée s'ancre dans le secteur respectif de la **fenêtre cible**. Les quatre secteurs fléchés ancrent la fenêtre déplacée dans les côtés respectifs de la fenêtre cible. Le bouton central ancre la fenêtre en tant qu'onglet de la fenêtre cible. Vous pouvez aussi ancrer une fenêtre en tant qu'onglet dans une autre fenêtre en la faisant glisser dans la barre d'onglets et en l'y déposant.



- Lorsque vous déplacez une fenêtre, une commande de placement divisée en quatre secteurs fléchés apparaît. Chaque flèche correspond à un côté de l'**interface du débogueur**. Lorsque vous relâchez une fenêtre déplacée sur une de ces flèches, la fenêtre déplacée est ancrée dans un des côtés de l'interface du débogueur.



Vous pouvez aussi double-cliquer sur la barre de titre d'une fenêtre pour la faire basculer entre la position ancrée et flottante.

9.1.6 Points d'arrêt

Le Débogueur XSLT et XQuery vous permet de définir des points d'arrêt dans des documents XSLT, XQuery et XML. Les points d'arrêt sont affichés sous la forme d'une ligne en pointillés rouge (*voir la capture d'écran ci-dessous*).

Note : il est possible de définir un point de trace et un point d'arrêt pour la même instruction. Dans ce cas la ligne apparaîtra en pointillés bleus et rouges (*voir capture d'écran*).

```
<!-- Local Variable reused -->
<xsl:variable name="OfficeName" select="n1:Name"/>
<!-- Display the company name if the varaible is true ********* -->
<xsl:if test="$Show_Company_Name">
    <h3>
        <xsl:value-of select="$OfficeName"/>
    </h3>
    <xsl:message>
        <xsl:text>Company Named Displayed</xsl:text>
    </xsl:message>
</xsl:if>
```

Lorsque vous lancez le débogueur pendant une session de débogage, le débogage s'interrompra à chaque point d'arrêt rencontré. Cela vous permet d'identifier des zones spécifiques nécessitant un débogage et de limiter l'attention à ces zones dans les documents XSLT, XQuery et/ou XML. Vous pouvez définir un nombre quelconque de points d'arrêt.

Note : Les points d'arrêt définis pour un document restent dans ce document jusqu'à ce qu'il soit fermé. Néanmoins, si vous souhaiter passer au Mode Schéma (par exemple, dans le cas de documents XSD), alors les points d'arrêt seront supprimés, lorsque vous retournez au Mode Texte ou au Mode Grille (depuis le Mode Schéma), il n'y aura pas de point d'arrêt.

© 2016-2022 Altova GmbH

Les points d'arrêts dans les documents XML

Vous pouvez définir des points d'arrêt sur n'importe quel nœud dans un document XML. L'interruption du traitement aura lieu au début de ce nœud.

Les points d'arrêts dans les documents XSLT

Vous pouvez définir des points d'arrêt dans les points suivants dans un document XSLT :

- Au début des modèles et des instructions de modèle (par ex., `xsl:for-each`).
- Dans une expression XPath (XPath 1.0 ou XPath 2.0).
- Dans n'importe quel nœud d'un fragment XML construit littéralement. L'interruption du traitement aura lieu au début de ce nœud.

Les points d'arrêts dans les documents XQuery

Vous pouvez définir des points d'arrêt dans les endroits suivants d'un document XQuery :

- Au début des déclarations XQuery.
- Dans une expression XQuery.
- Dans n'importe quel nœud d'un fragment XML construit littéralement. L'interruption du traitement aura lieu au début de ce nœud.

Insérer/supprimer les points d'arrêt

Pour insérer un point d'arrêt :

1. Placer le curseur à un endroit du document où vous souhaitez insérer le point d'arrêt (*voir paragraphes ci-dessus*). Dans les sessions de débogage XSLT, vous pouvez définir des points d'arrêt dans le Mode Texte et le Mode Grille. Les sessions de débogage XQuery sont uniquement disponibles dans le Mode Texte.
2. Choisir une des méthodes suivantes :

   - Sélectionner **XSL/XQuery | Insérer/Supprimer point d'arrêt**.
   - Appuyer sur **F9**.
   - Cliquer avec la touche de droite et sélectionner **Insérer/Supprimer point d'arrêt**.

Pour supprimer un point d'arrêt :

1. Placer le curseur à un endroit du document contenant le point d'arrêt.
2. Choisir une des méthodes suivantes :

   - Sélectionner **XSL/XQuery | Insérer/Supprimer point d'arrêt**.
   - Appuyer sur **F9**.
   - Cliquer avec la touche de droite et sélectionner **Insérer/Supprimer point d'arrêt**.

En alternative, vous pouvez utiliser le dialogue Points d'arrêt pour supprimer un point d'arrêt :

1. Sélectionner l'option de menu **XSL/XQuery | Point d'arrêt...**.
2. Cliquer sur le point d'arrêt dans la fenêtre de dialogue et cliquer sur **Supprimer**.

La touche **Supprimer tout** supprime tous les points d'arrêt de la fenêtre de dialogue (et toutes les feuilles de style XSLT).

Activer/désactiver les points d'arrêt :

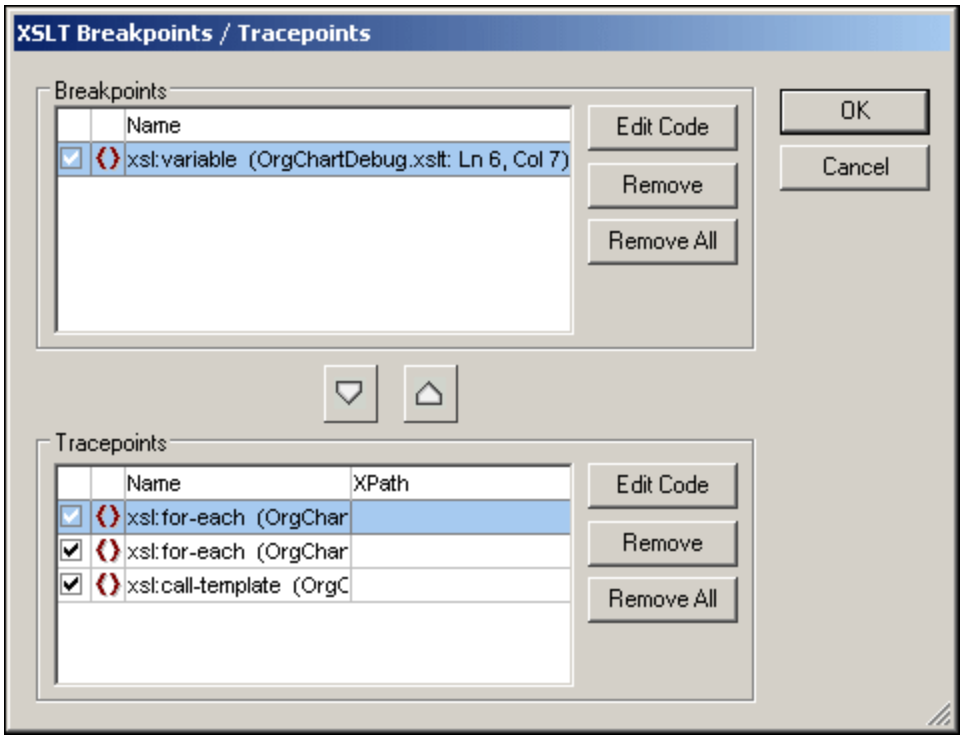
Une fois avoir inséré les points d'arrêt, vous pouvez les désactiver si vous souhaitez passer au-dessus des points d'arrêt sans devoir les supprimer. Vous pourrez les activer à nouveau le cas échéant.

Pour désactiver un point d'arrêt :

1. Placer le curseur à un endroit du nœud ou de l'expression contenant le point d'arrêt.
2. Sélectionner **XSL/XQuery | Activer/désactiver le point d'arrêt** (ou appuyer sur **Ctrl+F9**). Le point d'arrêt passe du rouge au gris, indiquant qu'il a été désactivé.

En alternative, vous pouvez utiliser le dialogue Points d'arrêt pour désactiver un point d'arrêt :

1. Sélectionner l'option de menu **XSL/XQuery | Points d'arrêt/Points de trace...**. Le dialogue Points d'arrêt / Points de trace XSLT s'ouvre et affiche les points d'arrêt actuellement définis dans tous les documents de source XML et les feuilles de style XSLT ouverts.



2. Décocher la case des points d'arrêt que vous souhaitez désactiver et cliquer sur **OK** pour confirmer. Le point d'arrêt passe du rouge au gris, indiquant qu'il a été désactivé.

Pour activer un point d'arrêt :

1. Placer le curseur à un endroit du nœud ou de l'expression contenant le point d'arrêt.
2. Sélectionner **XSL/XQuery | Activer/désactiver le point d'arrêt** (ou appuyer sur **Ctrl+F9**). Le point d'arrêt passe du rouge au gris, indiquant qu'il a été désactivé.

© 2016-2022 Altova GmbH

Trouver un point d'arrêt spécifique

Pour trouver un point d'arrêt spécifique :

1. Sélectionner l'option de menu **XSL/XQuery | Points d'arrêt/Points de trace...**. Le dialogue Points d'arrêt /Points de trace XSLT apparaît.
2. Cliquer sur le point d'arrêt dans la liste de points d'arrêt.
3. Cliquer sur la touche **Éditer code**. Le dialogue Points d'arrêt se ferme et le curseur de texte est placé directement devant le point d'arrêt dans le mode Texte. Dans le mode Grille amélioré, la cellule de table contenant le point d'arrêt est marquée en rouge.

Reprendre le débogage après un point d'arrêt

Pour continuer le débogage après un point d'arrêt :

- Sélectionner **XSL/XQuery | Pas à pas détaillé** ou la commande **XSL/XQuery | Démarrer débogueur/OK**.

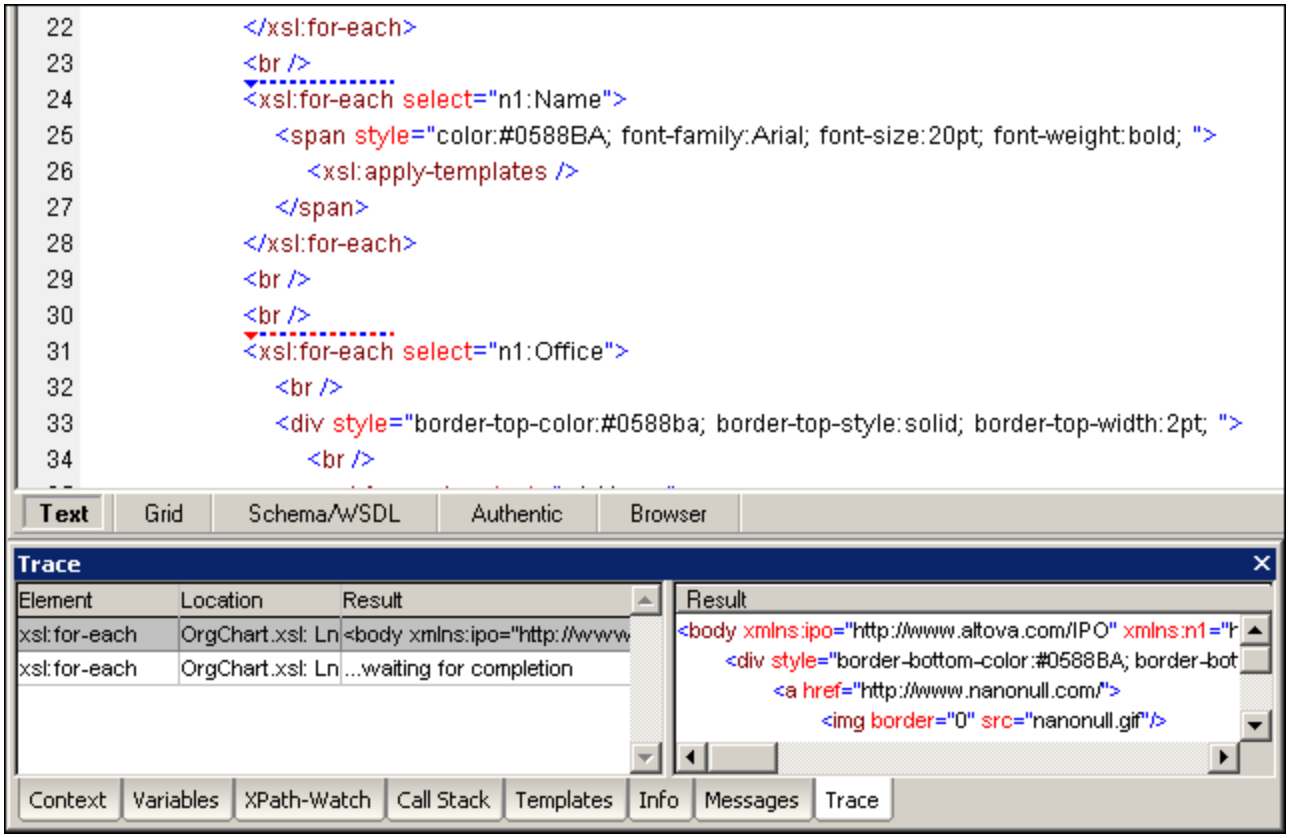
9.1.7 Points de trace

Le Débogueur XSLT et XQuery vous permet de définir des points de trace dans les documents XSLT.

Les points de trace vous permettent de tracer un contenu généré par une instruction ou de consulter le résultat d'une expression XPath à l'endroit où le point de trace a été défini sans devoir éditer la feuille de style XSLT, par exemple, en utilisant l'élément `xsl:message` pour sortir des messages de débogage.

Les points de trace sont affichés en tant que ligne en pointillés bleus dans les feuilles de style XSLT (*affichés dans la capture d'écran ci-dessous*).

Note : il est possible de définir un point de trace et un point d'arrêt pour la même instruction. Dans ce cas la ligne apparaîtra en pointillés bleus et rouges (*voir capture d'écran*).



Le débogueur sort le contenu généré par chaque instruction pour lequel un point de trace a été défini. Cette sortie est visible dans la fenêtre Trace. Vous pouvez définir n'importe quel nombre de points de trace dans une feuille de style XSLT.

Note : Les points de trace définis pour un document restent dans ce document jusqu'à ce qu'il soit fermé.

Les points de trace dans des documents XSLT

Vous pouvez définir des points de trace dans des instructions XSL et des résultats littéraux dans une feuille de style XSLT.

Les points de trace dans des documents XML et XQuery

Vous pouvez définir des points de trace dans des documents XML et XQuery.

Insérer/supprimer les points de trace

Pour insérer un point de trace :

1. Placer le curseur à un endroit du document XSLT où vous souhaitez insérer le point de trace. Pendant les sessions de débogage, vous pouvez définir des points de trace dans le Mode Texte et le Mode Grille.
2. Choisir une des méthodes suivantes :

© 2016-2022 Altova GmbH

- Sélectionner **XSL/XQuery | Insérer/supprimer le point de trace**.
- Appuyer sur **Shift+F9**.
- Cliquer avec la touche de droite et sélectionner **Insérer/supprimer le point de trace**.

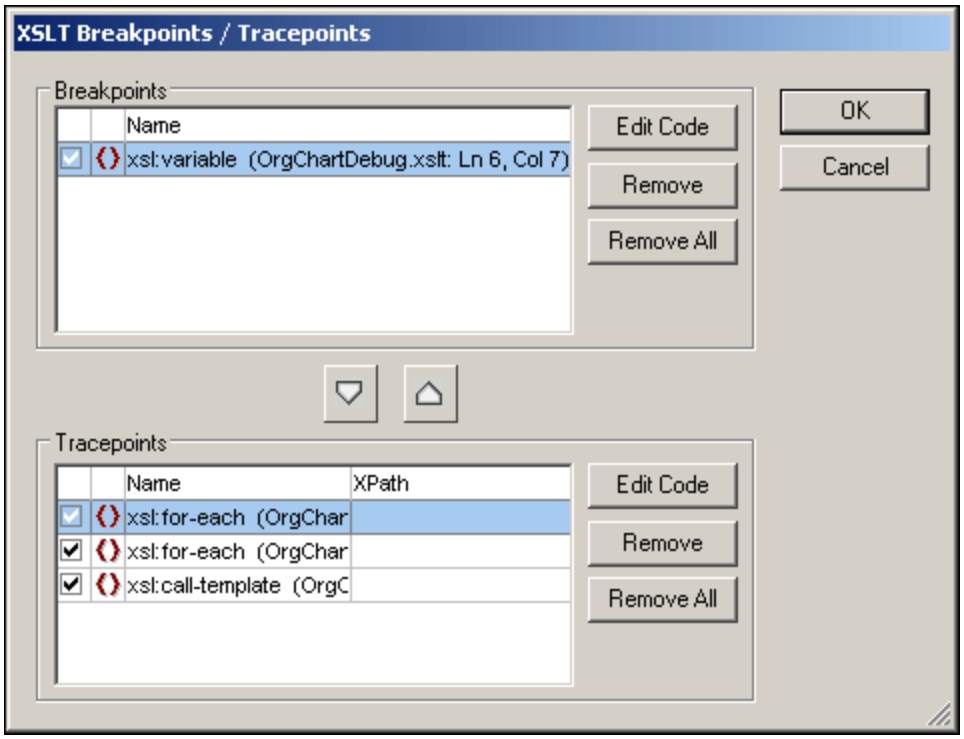
Pour supprimer un point de trace :

1. Placer le curseur à un endroit du document XSLT document contenant le point de trace.
2. Choisir une des méthodes suivantes :

   - Sélectionner **XSL/XQuery | Insérer/supprimer le point de trace**.
   - Appuyer sur **Shift+F9**.
   - Cliquer avec la touche de droite et sélectionner **Insérer/supprimer le point de trace**.

En alternative, vous pouvez utiliser le dialogue Points d'arrêt pour supprimer un point de trace :

1. Sélectionner l'option de menu **XSL/XQuery | Points d'arrêt/Points de trace...**.
2. Cliquer sur le point de trace dans le dialogue (*voir capture d'écran*) et cliquer sur **Supprimer**.



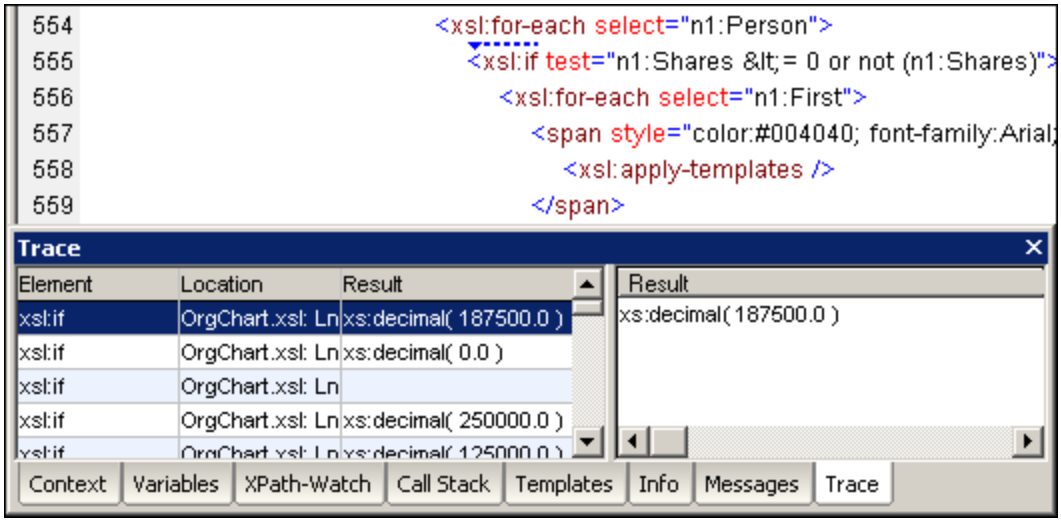
La touche **Supprimer tout** dans le volet Points de trace supprime tous les points de trace du dialogue (et de toutes les feuilles de style XSLT).

Définir un XPath pour un point de trace
Vous pouvez définir un XPath pour un point de trace. Lorsque vous définissez un XPath pour un point de trace, le résultat de l'évaluation du XPath est affichée dans la fenêtre Trace au lieu du contenu généré par la déclaration pour laquelle le point de trace a été défini. Le XPath est évalué relativement au nœud contextuel à l'endroit où le point de trace a été défini.

Pour définir un XPath pour un point de trace :

1. Sélectionner l'option de menu **XSL/XQuery | Points d'arrêt/Points de trace...**. Cela permet d'ouvrir le dialogue Points d'arrêt / Points de trace XSLT qui affiche les points de trace actuellement définis dans tous les documents de feuille de style XSLT ouverts.
2. Saisir le XPath dans la colonne **XPath** dans la ligne qui correspond au point de trace.

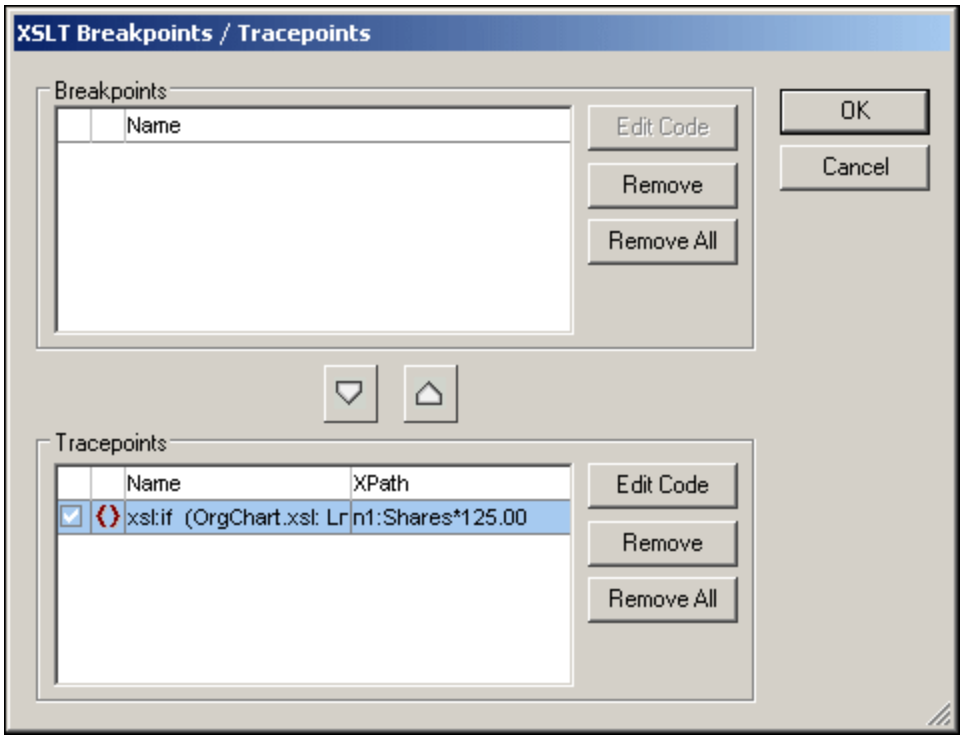
Exemple
Dans l'exemple ci-dessous, le point de trace est défini de manière à ce que le nœud contextuel est Person. L'élément Person contient un élément Shares. Nous souhaitons afficher le nombre de shares que chaque personne a, multiplié par 125 (la valeur de chaque share).



Procéder comme suit :

1. Définir un point de trace au niveau de la ligne juste après l'instruction `xsl:for-each` qui sélectionne l'élément `n1:Person` (ligne 555 dans la capture d'écran ci-dessus).
2. Ouvrir le dialogue Points d'arrêt / Points de trace XSLT et saisir le XPath `n1:Shares*125.00` pour le point de trace que vous venez de définir.

© 2016-2022 Altova GmbH

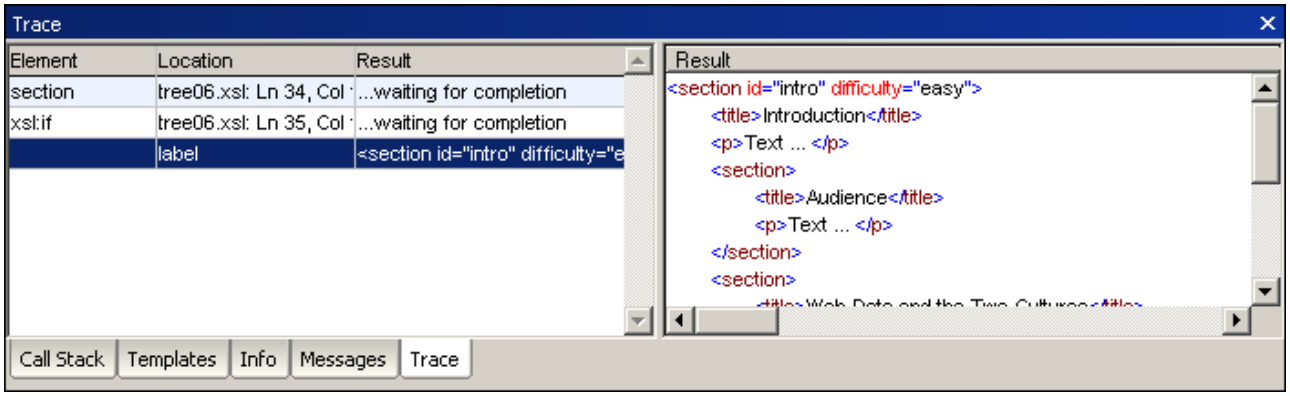


3. Lancer le débogueur. Les résultats du XPath que vous avez saisi pour le point de trace apparaissent dans la fenêtre Trace.

La fenêtre Trace

Sélectionner **XSL/XQuery** | **Démarrer débogueur/OK** pour commencer le débogage. La sortie des instructions pour lesquelles des points de trace ont été définis est affichée dans la fenêtre Trace (*voir capture d'écran*). Cliquer sur une ligne dans la fenêtre Trace pour afficher les résultats complets de cette déclaration dans la partie droite de la fenêtre (*voir capture d'écran*).

Note : les résultats sont affichés dans la fenêtre Trace uniquement après l'achèvement de l'instruction tracée.



Désactiver/activer les points de trace

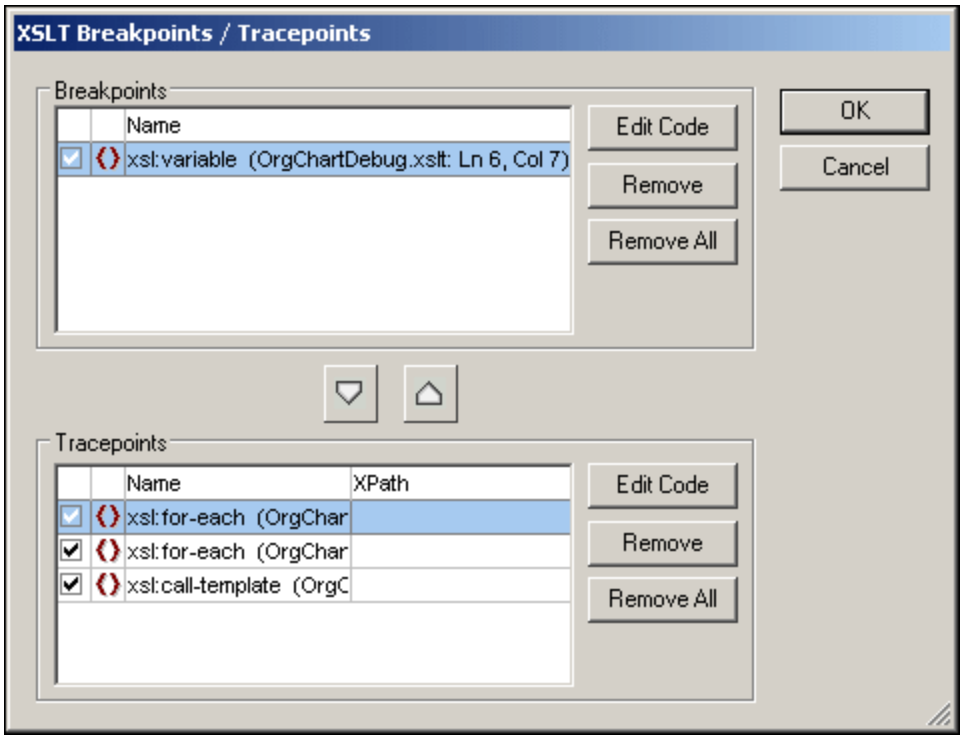
Après avoir insérer les points de trace, vous pouvez les désactiver si vous souhaitez les sauter sans devoir les supprimer. Vous pouvez les réactiver le cas échéant.

Pour désactiver un point de trace :

1. Placer le curseur à l'endroit de la feuille de style XSLT contenant le point de trace.
2. Sélectionner **XSL/XQuery | Activer/Désactiver le point de trace** (ou appuyer sur **Ctrl+Shift+F9**). Le point de trace passe du bleu au gris, indiquant qu'il a été désactivé.

En alternative, vous pouvez utiliser le dialogue Points d'arrêt / Points de trace XSLT pour désactiver un point de trace :

1. Sélectionner l'option de menu **XSL/XQuery | Points d'arrêt/Point de trace...**. Le dialogue Points d'arrêt / Points de trace XSLT s'ouvre, qui affiche les points de trace actuellement définis dans tous les documents de feuille de style XSLT ouverts.



2. Retirer la coche de chaque point de trace que vous souhaitez désactiver et cliquer sur **OK** pour confirmer. Les points de trace passent du bleu au gris, indiquant qu'ils ont été désactivés.

Pour activer un point de trace :

1. Placer le curseur à l'endroit du document XSLT contenant le point de trace.
2. Sélectionner **XSL/XQuery | Activer/Désactiver le point de trace** (ou appuyer sur **Ctrl+Shift+F9**). Le point de trace passe du gris au bleu, indiquant qu'il a été activé.

© 2016-2022 Altova GmbH

Trouver un point de trace spécifique
Pour trouver un point de trace spécifique :

1. Sélectionner l'option de menu **XSL/XQuery | Points d'arrêt/Point de trace...**. Le dialogue Points d'arrêt / Points de trace XSLT apparaît.
2. Cliquer sur le point de trace requis dans la liste de point de trace.
3. Cliquer sur la touche **Éditer Code**. Le dialogue Points d'arrêt / Points de trace XSLT se ferme et le curseur de texte se place directement devant le point de trace dans le Mode Texte du document XSLT. Dans le Mode Grille amélioré, la cellule de table contenant le point de trace est marquée en bleu.

9.1.8 Raccourcis débogueur

Les raccourcis de déboguer par défaut sont recensés ci-dessous.

Commandes de débogueur

F9	Insérer/Supprimer point d'arrêt
F9 + Shift	Insérer/Supprimer point de trace
F9 + CTRL	Activer/Désactiver point d'arrêt
F9 + Shift + CTRL	Activer/Désactiver point de trace
F11	Pas à pas détaillé
F11 + Shift	Pas à pas principal
F11 + CTRL	Pas à pas sortant
F11 + Alt	Démarrer débogueur/OK

9.2 Profileur XSLT et XQuery

Site Altova : Profileur XSLT, Profileur XQuery
Le profileur XSLT/XQuery est un outil qui est utilisé pour analyser les délais d'exécution des feuilles de style XSLT (1.0 et 2.0) et des documents XQuery depuis XMLSpy. Il vous révèle combien de temps nécessite chaque instruction de feuille de style XSLT ou XQuery document pour l'exécution, et vous pouvez utiliser cette information pour optimiser les délais d'exécution de ces fichiers.

Le profileur est utilisé pour trouver les instructions qui présentent le délai d'exécution total le plus élevé afin que cette information puisse être utilisée pour optimiser ces instructions. Les instructions peuvent avoir un délai total d'exécution élevé pour la ou les raisons suivantes :

- l'instruction exige beaucoup de temps
- l'instruction a été évaluée souvent (décompte d'appel élevé)

Profilage de décompte des appels et profilage du graphe d'appel

Le profileur vous permet de choisir entre le profilage *décompte des appels* et *graphe d'appels*. Les deux types de profilage montrent les statistiques des délais d'exécution pour chaque instruction.
En ce qui concerne les objectifs d'optimisation, normalement, vous utiliserez un profilage de décompte d'appel, qui affiche une ligne dans le profileur pour chaque instruction.
Le profilage d'appel de graphe montre l'histoire complète de l'exécution d'une transformation XSLT ou d'une exécution XQuery, c. à. d. quels modèles/quelles fonctions ont été appelées et dans quel ordre, pendant la transformation. Dans les résultats du profilage du graphe d'appel, vous trouverez une ligne pour chaque moment auquel une instruction est appelée et non pas une ligne pour chaque instruction.

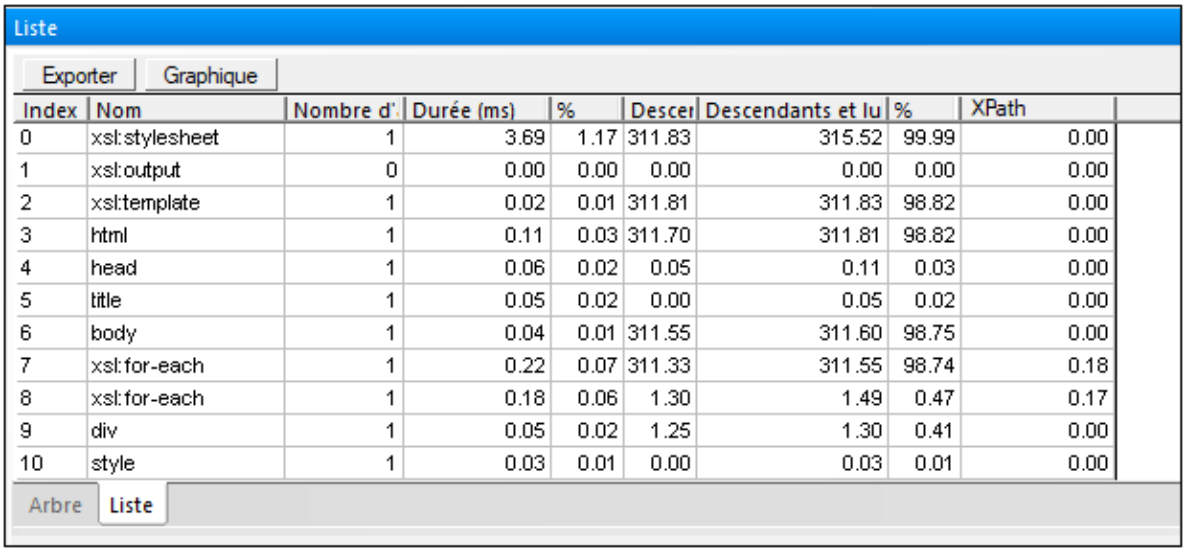
Pour utiliser le profileur XSLT/XQuery, voir Profilage XSLT ou Profilage XQuery.

Modes Profileur

Plusieurs modes vous permettent de consulter les résultats de l'analyse en cliquant sur l'onglet correspondant :

- **Mode Liste :** les statistiques de profilage sont affichées en tant qu'une liste qui peut être triée par exemple, par la durée de l'exécution d'instruction ou la durée de l'exécution de l'instruction et de ses descendants.

© 2016-2022 Altova GmbH

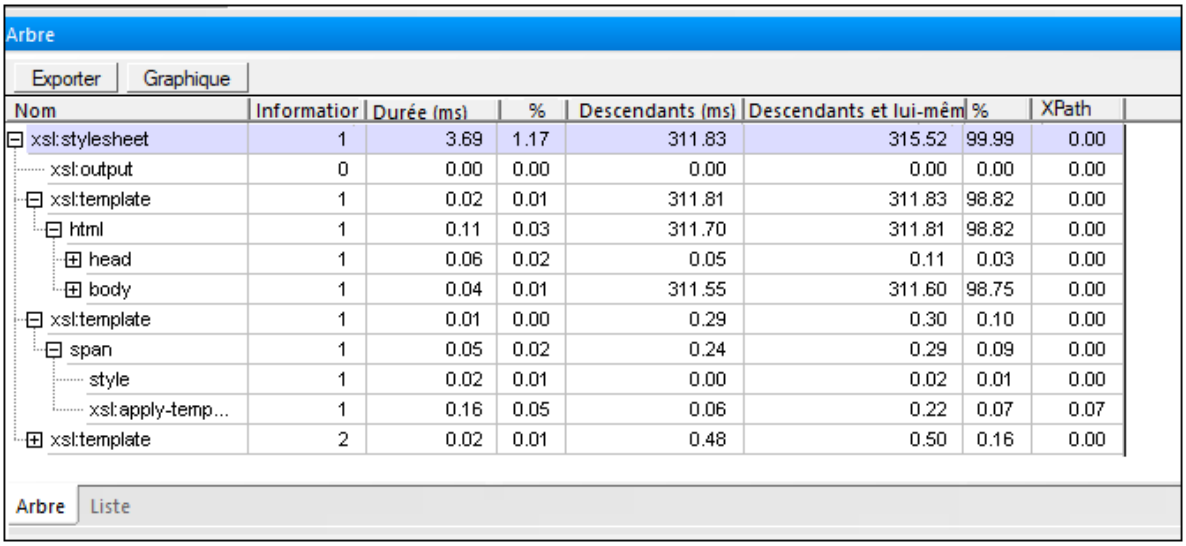

Liste

Exporter | Graphique

Index	Nom	Nombre d'	Durée (ms)	%	Descer	Descendants et lu	%	XPath
0	xsl:stylesheet	1	3.69	1.17	311.83	315.52	99.99	0.00
1	xsl:output	0	0.00	0.00	0.00	0.00	0.00	0.00
2	xsl:template	1	0.02	0.01	311.81	311.83	98.82	0.00
3	html	1	0.11	0.03	311.70	311.81	98.82	0.00
4	head	1	0.06	0.02	0.05	0.11	0.03	0.00
5	title	1	0.05	0.02	0.00	0.05	0.02	0.00
6	body	1	0.04	0.01	311.55	311.60	98.75	0.00
7	xsl:for-each	1	0.22	0.07	311.33	311.55	98.74	0.18
8	xsl:for-each	1	0.18	0.06	1.30	1.49	0.47	0.17
9	div	1	0.05	0.02	1.25	1.30	0.41	0.00
10	style	1	0.03	0.01	0.00	0.03	0.01	0.00

Arbre | Liste

- **Mode Arborescence :** les statistiques sont affichées dans une structure arborescente. Il est possible de voir, par exemple combien de temps une fonction a pris à exécuter, puis augmenter l'arborescence pour cette fonction et voir combien de temps chaque instruction contenue dans la fonction a pris pour s'exécuter et combien de fois elle a été exécutée.


Arbre

Exporter | Graphique

Nom	Informatior	Durée (ms)	%	Descendants (ms)	Descendants et lui-mêm	%	XPath
⊟ xsl:stylesheet	1	3.69	1.17	311.83	315.52	99.99	0.00
xsl:output	0	0.00	0.00	0.00	0.00	0.00	0.00
⊟ xsl:template	1	0.02	0.01	311.81	311.83	98.82	0.00
⊟ html	1	0.11	0.03	311.70	311.81	98.82	0.00
⊞ head	1	0.06	0.02	0.05	0.11	0.03	0.00
⊞ body	1	0.04	0.01	311.55	311.60	98.75	0.00
⊟ xsl:template	1	0.01	0.00	0.29	0.30	0.10	0.00
⊟ span	1	0.05	0.02	0.24	0.29	0.09	0.00
style	1	0.02	0.01	0.00	0.02	0.01	0.00
xsl:apply-temp...	1	0.16	0.05	0.06	0.22	0.07	0.07
⊞ xsl:template	2	0.02	0.01	0.48	0.50	0.16	0.00

Arbre | Liste

Résultats de tri

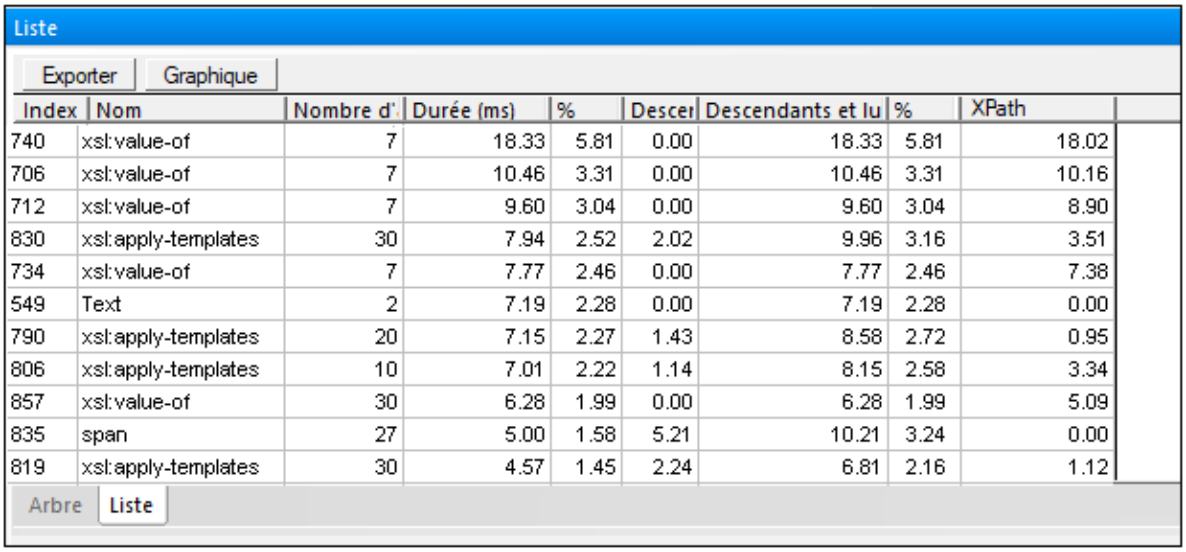
Une fois que vous avez exécuté le Profileur, vous pouvez trier par la quantité de temps qu'a pris une instruction à exécuter ou par le nombre de fois que cette instruction a été appelée.

Pour trier l'information dans le Profileur :

1. Cliquer sur l'onglet **Liste**.

2. Cliquer sur l'en-tête de colonne par laquelle vous souhaitez effectuer votre tri (par ex., **Décompte d'appel** pour trier par le nombre de fois qu'une instruction a été appelée ou **Durée** pour trier par le temps qu'a mis l'instruction à s'exécuter).

Cette capture d'écran montre les contenus du Profileur trié par la durée d'instruction dans l'ordre descendant.



Index	Nom	Nombre d'	Durée (ms)	%	Descer	Descendants et lu	%	XPath
740	xsl:value-of	7	18.33	5.81	0.00	18.33	5.81	18.02
706	xsl:value-of	7	10.46	3.31	0.00	10.46	3.31	10.16
712	xsl:value-of	7	9.60	3.04	0.00	9.60	3.04	8.90
830	xsl:apply-templates	30	7.94	2.52	2.02	9.96	3.16	3.51
734	xsl:value-of	7	7.77	2.46	0.00	7.77	2.46	7.38
549	Text	2	7.19	2.28	0.00	7.19	2.28	0.00
790	xsl:apply-templates	20	7.15	2.27	1.43	8.58	2.72	0.95
806	xsl:apply-templates	10	7.01	2.22	1.14	8.15	2.58	3.34
857	xsl:value-of	30	6.28	1.99	0.00	6.28	1.99	5.09
835	span	27	5.00	1.58	5.21	10.21	3.24	0.00
819	xsl:apply-templates	30	4.57	1.45	2.24	6.81	2.16	1.12

Optimiser vos feuilles de style XSLT et vos documents XQuery

Veuillez garder en tête les lignes directrices suivantes lors de l'optimisation des délais d'exécution des instructions dans les feuilles de style XSLT et les documents XQuery :

- Éviter d'utiliser des variables dans une instruction si la variable est utilisée une seule fois uniquement, parce que l'initialisation d'une variable peut exiger beaucoup de temps.

  Les fragments de code XSLT suivants montrent un exemple d'optimisation de code en supprimant des variables superflues. Les deux exemples font la même chose mais le deuxième exemple le fait sans utiliser les variables `name` et `containsResult` :

  Fragment de code 1 :

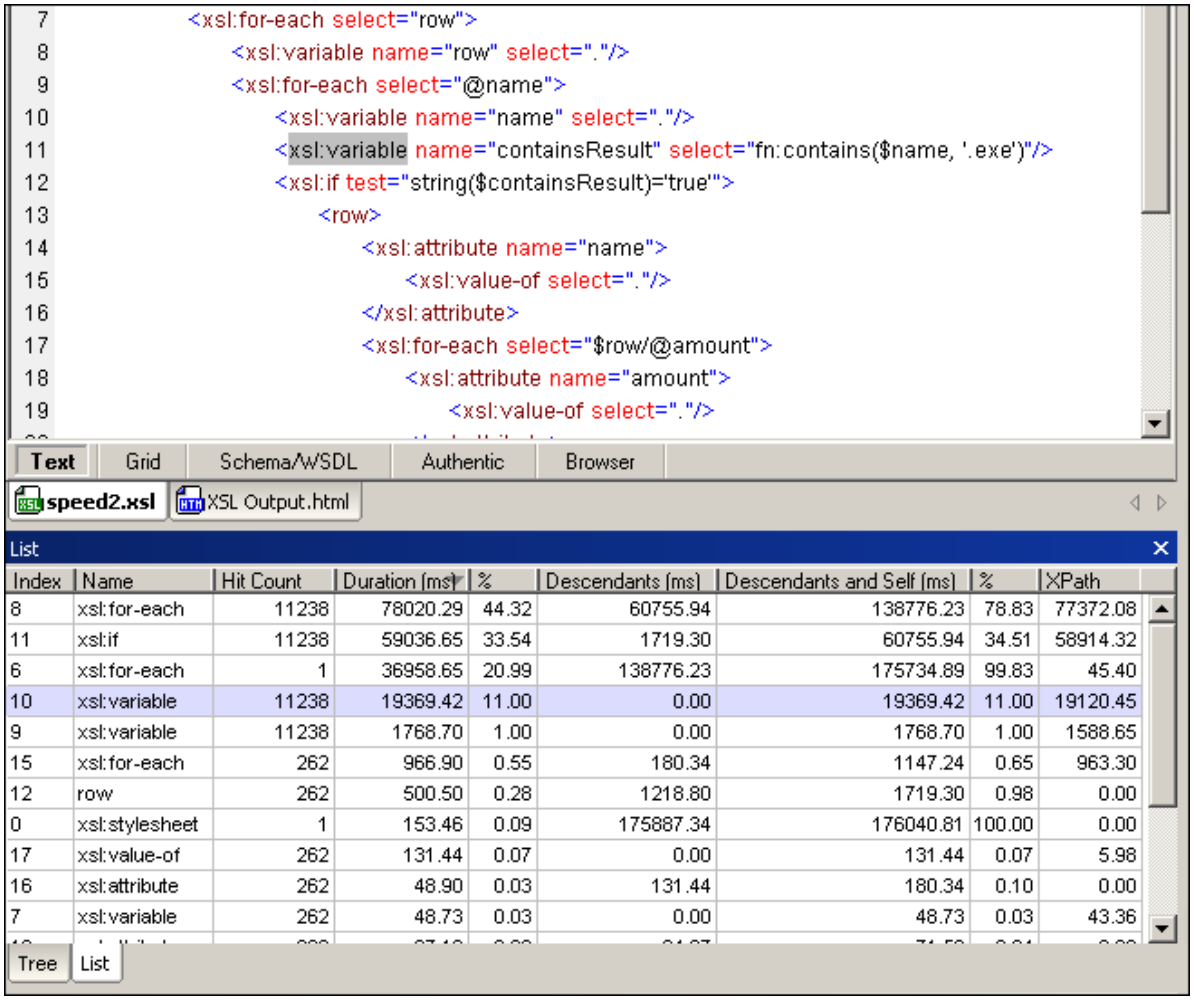
```
<xsl:for-each select="row">
	<xsl:variable name="row" select="."/>
	<xsl:for-each select="@name">
		<xsl:variable name="name" select="."/>
		<xsl:variable name="containsResult" select="fn:contains($name,
'.exe')"/>
		<xsl:if test="string($containsResult)='true'">

		...

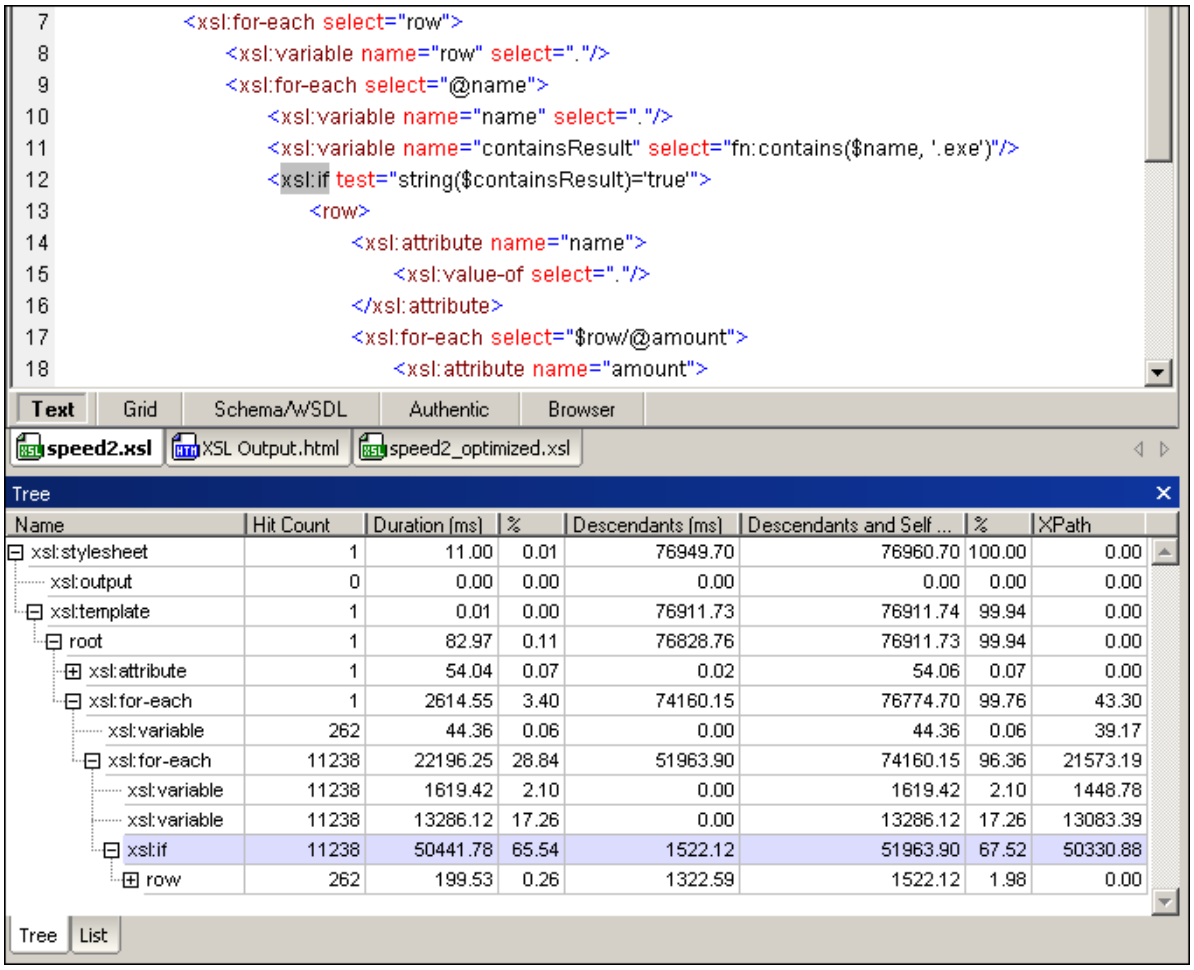
		</xsl:if>
	</xsl:for-each>
</xsl:for-each>
```


© 2016-2022 Altova GmbH

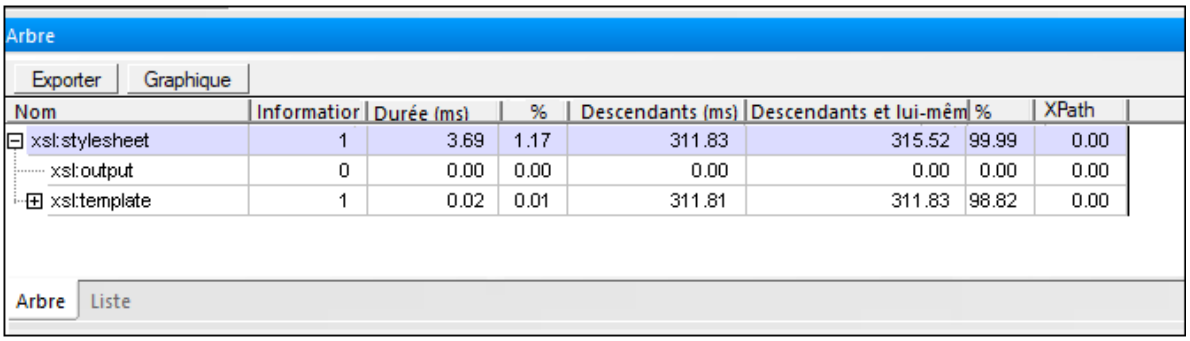
La capture d'écran ci-dessous montre les résultats de l'analyse du fichier qui contient ce fragment de code, trié par la durée des instructions. L'instruction dans laquelle la variable containsResult est initialisée nécessite au total environ 19 secondes de temps d'exécution.



La capture d'écran ci-dessous montre les résultats dans le mode arborescence. Ici, vous pouvez constater que la déclaration If qui utilise la variable containsResult nécessite un temps d'exécution d'environ 50 secondes au total :



La transformation XSLT prend un total d'environ 74 secondes :



Fragment de code 2 :

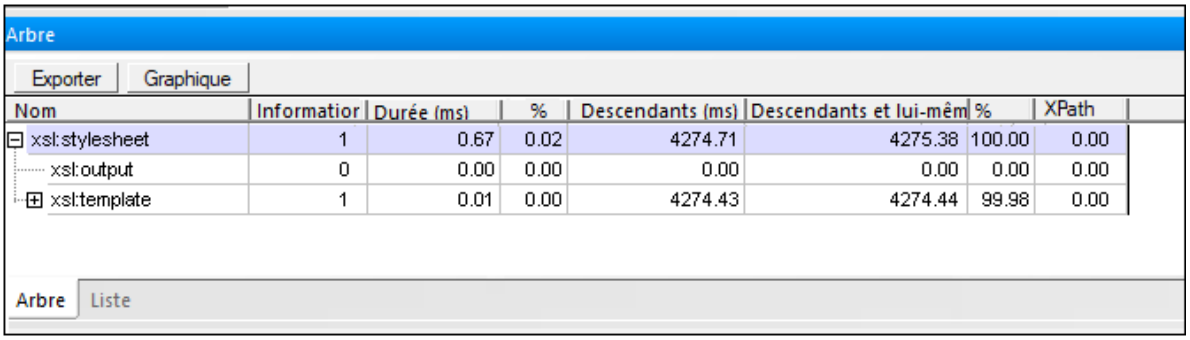
```
<xsl:for-each select="row">
	<xsl:variable name="row" select="."/>
	<xsl:for-each select="@name">
		<xsl:if test="fn:contains(., '.exe')">

		...
```

 © 2016-2022 Altova GmbH

```
            </xsl:if>
        </xsl:for-each>
</xsl:for-each>
```

Une fois que la feuille de style a été réécrite sans utiliser ces variables, son délai d'exécution total est de seulement 4,3 secondes :



- Utiliser des variables si une valeur ou une expression est utilisée de manière répétitive.
- Éviter de créer des variables de constante locales dans une fonction ; au lieu de cela, créer des variables globales.
- Éviter de créer des fragments d'arborescence constants dans une fonction ; au lieu de cela, les créer globalement.
- Limiter votre utilisation des prédicats, puisque le filtrage avec des prédicats est évalué séparément pour chaque nœud. Vous pouvez réduire le nombre d'appels auprès des prédicats, en préfiltrant les noms. Dans cet exemple, `*` est utilisé avec deux prédicats :

  ```
  //*[node-name()=Book][author="Steve"]
  ```

  Dans cette instruction équivalente, le nom `Book` et un seul prédicat sont utilisés :

  ```
  //Book[@Author="Steve"]
  ```

- Partager les instructions de manière à ce que des parties de l'instruction qui doivent uniquement être exécutées une fois sont uniquement utilisées une fois. Créer des variables globales à partir des parties qui sont uniquement dépendantes du contexte global.

9.2.1 Profilage XSLT

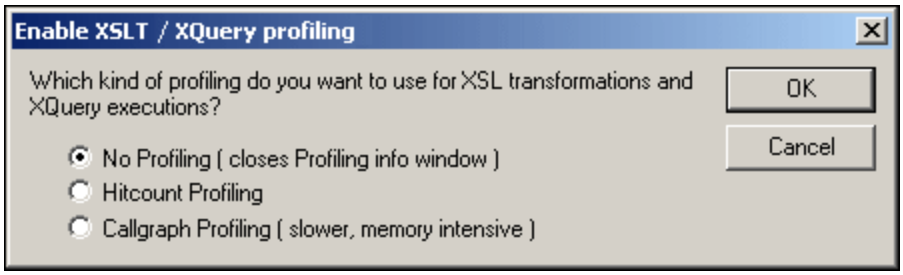
Lancer le profileur

Note : Les résultats de temps d'exécution affichés dans le profileur peuvent être influencés par d'autres applications qui sont exécutées sur votre ordinateur. Lors de l'analyse des fichiers à l'aide du profileur, il est préférable de ne pas exécuter d'autres applications que XMLSpy.

Pour analyser une feuille de style XSLT :

1. Dans XMLSpy, ouvrir le fichier XML qui sera utilisé en tant que données d'entrée pour la transformation XSLT.

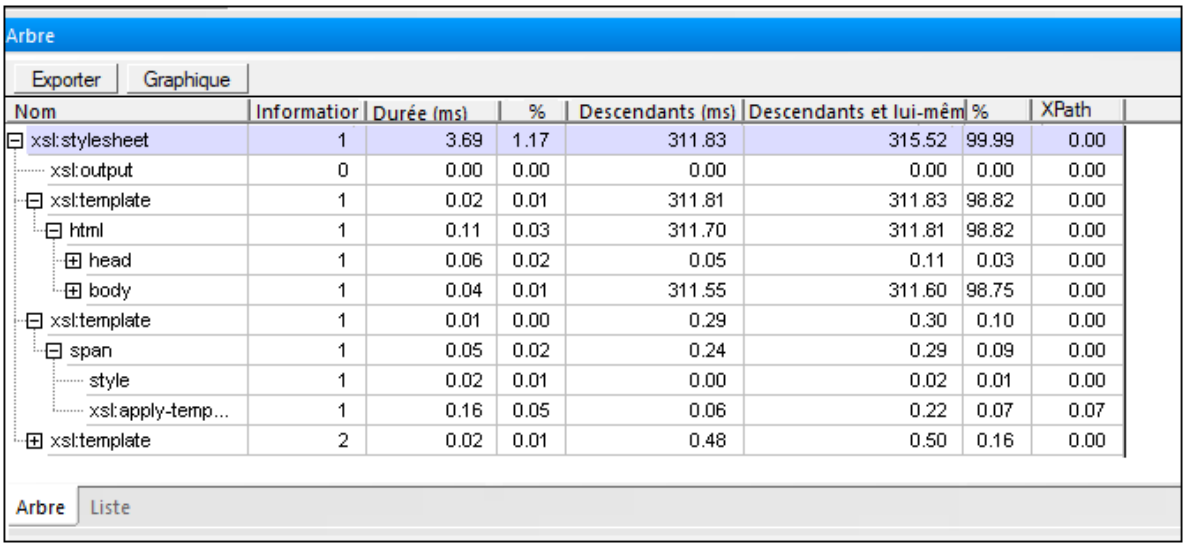
2. Activer le profileur en sélectionnant **XSL/XQuery** | **Activer profilage XSLT / XQuery**. Un dialogue s'ouvre.



3. Sélectionner **Profilage nombre d'appels** ou **Profilage graphe des appels**. Cliquer sur **OK** pour confirmer. Une fenêtre de profileur vide apparaît.
4. Exécuter la transformation XSL (**XSL/XQuery | Transformation XSL**). Un dialogue s'ouvre dans lequel vous pouvez sélectionner le chemin vers la feuille de style XSLT que vous souhaitez analyser. Une fois la transformation terminée, les statistiques de temps d'exécution apparaissent dans le profileur.
5. Cliquer sur les icônes "+" pour agrandir les lignes dans le profileur afin de consulter les statistiques de temps d'exécution pour la feuille de styles XSLT (*voir capture d'écran*). Veuillez noter qu'en ce qui concerne ces captures d'écran, l'option **Profilage nombre d'appels** avait été sélectionnée.

Cliquer sur une ligne dans le profileur pour marquer l'instruction correspondante dans le fichier qui a été analysé.

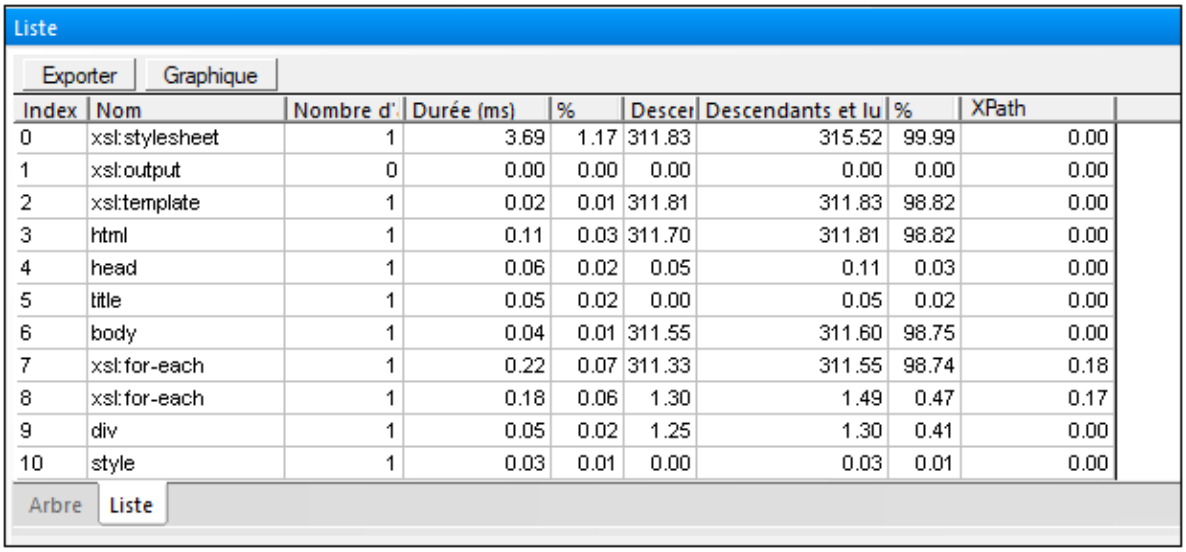
La capture d'écran suivante montre le mode Arborescence dans le profileur :



Nom	Informatior	Durée (ms)	%	Descendants (ms)	Descendants et lui-même	%	XPath
xsl:stylesheet	1	3.69	1.17	311.83	315.52	99.99	0.00
xsl:output	0	0.00	0.00	0.00	0.00	0.00	0.00
xsl:template	1	0.02	0.01	311.81	311.83	98.82	0.00
html	1	0.11	0.03	311.70	311.81	98.82	0.00
head	1	0.06	0.02	0.05	0.11	0.03	0.00
body	1	0.04	0.01	311.55	311.60	98.75	0.00
xsl:template	1	0.01	0.00	0.29	0.30	0.10	0.00
span	1	0.05	0.02	0.24	0.29	0.09	0.00
style	1	0.02	0.01	0.00	0.02	0.01	0.00
xsl:apply-temp...	1	0.16	0.05	0.06	0.22	0.07	0.07
xsl:template	2	0.02	0.01	0.48	0.50	0.16	0.00

La capture d'écran suivante montre le Mode Liste dans le profileur pour la même feuille de style XSLT :

© 2016-2022 Altova GmbH


Liste

Exporter Graphique

Index	Nom	Nombre d'	Durée (ms)	%	Descer	Descendants et lu	%	XPath
0	xsl:stylesheet	1	3.69	1.17	311.83	315.52	99.99	0.00
1	xsl:output	0	0.00	0.00	0.00	0.00	0.00	0.00
2	xsl:template	1	0.02	0.01	311.81	311.83	98.82	0.00
3	html	1	0.11	0.03	311.70	311.81	98.82	0.00
4	head	1	0.06	0.02	0.05	0.11	0.03	0.00
5	title	1	0.05	0.02	0.00	0.05	0.02	0.00
6	body	1	0.04	0.01	311.55	311.60	98.75	0.00
7	xsl:for-each	1	0.22	0.07	311.33	311.55	98.74	0.18
8	xsl:for-each	1	0.18	0.06	1.30	1.49	0.47	0.17
9	div	1	0.05	0.02	1.25	1.30	0.41	0.00
10	style	1	0.03	0.01	0.00	0.03	0.01	0.00

Arbre Liste

Utiliser les informations dans le profileur

Le profileur affiche les informations suivantes concernant chaque instruction dans la feuille de style XSLT :

- **Index :** un nombre attribué à chaque instruction dans l'ordre dans lequel l'instruction a été appelée.
- **Nom :** le nom de l'instruction XSLT.
- **Nombre d'appels :** le nombre total de fois que l'instruction a été appelée pendant la transformation.
- **Durée (ms)** et **% :** le nombre de millisecondes que l'instruction a prise pour s'exécuter sans prendre en compte le temps d'exécution de ses descendants et le pourcentage du temps d'exécution total.
- **Descendants (ms) :** le nombre de millisecondes que les descendants de l'instruction ont pris à s'exécuter.
- **Descendants et lui-même** et **% :** le nombre de millisecondes que l'instruction et ses descendants ont pris à s'exécuter, et le pourcentage du temps d'exécution total.
- **XPath :** si l'instruction contient une instruction XPath, cette colonne contient le temps qu'il a fallu à l'instruction pour s'exécuter.

Note : lors de l'utilisation du profilage de nombre d'appels, les temps contenus dans la fenêtre du Profileur sont la somme totale du temps d'exécution pour tous les appels au niveau de l'instruction. Lors de l'utilisateur d'un profilage de graphe d'appel, étant donné que chaque appel de l'instruction est recensé séparément, les temps montrés dans la fenêtre du profileur sont les durées d'une seule exécution de l'instruction.

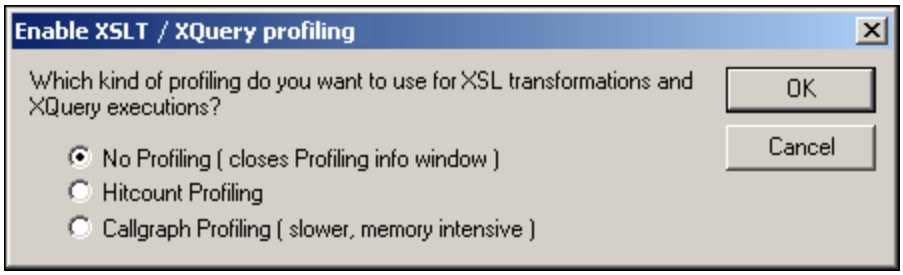
9.2.2 Profilage XQuery

Lancer le profileur

Note : Les résultats de temps d'exécution affichés dans le profileur peuvent être influencés par d'autres applications qui sont exécutées sur votre ordinateur. Lors de l'analyse des fichiers à l'aide du profileur, il est préférable de ne pas exécuter d'autres applications que XMLSpy.

Pour analyser un document XQuery :

1. Dans XMLSpy, ouvrir le document XQuery que vous souhaitez analyser.
2. Activer le profileur en sélectionnant **XSL/XQuery** | **Activer profilage XSLT 2 / XQuery**. Un dialogue s'ouvre.

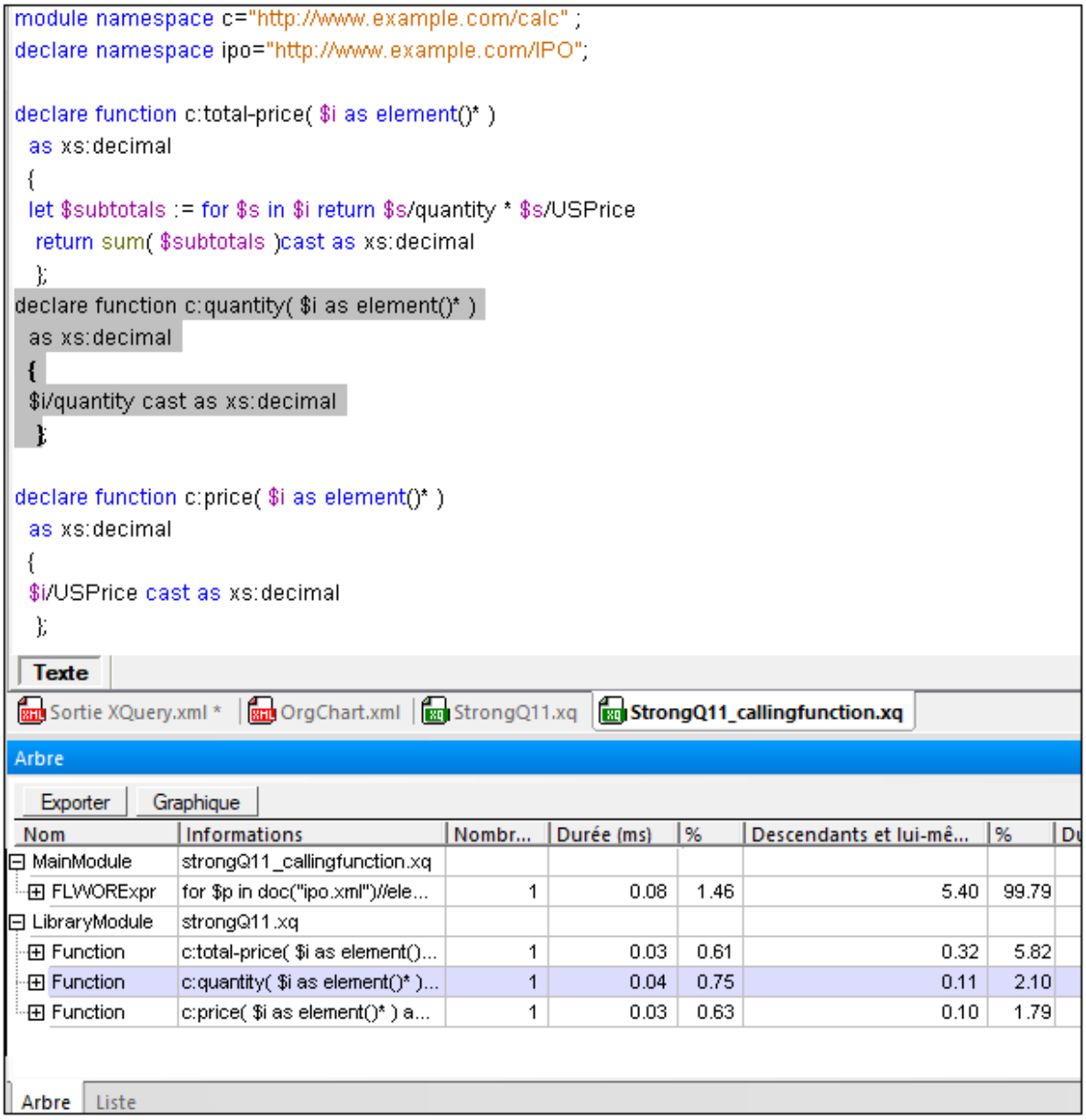


3. Sélectionner **Profilage nombre d'appels** ou **Profilage graphe des appels**. Cliquer sur **OK** pour confirmer. Une fenêtre de profileur vide apparaît.
4. Exécuter XQuery (**XSL/XQuery | Exécution XQuery**). Une fois l'exécution terminée, les statistiques de temps d'exécution apparaissent dans le profileur.
5. Cliquer sur les icônes "+" pour agrandir les lignes dans le profileur afin de consulter les statistiques des instructions dans le document XQuery (*voir capture d'écran*). Veuillez noter qu'en ce qui concerne ces captures d'écran, l'option **Profilage nombre d'appels** avait été sélectionnée.

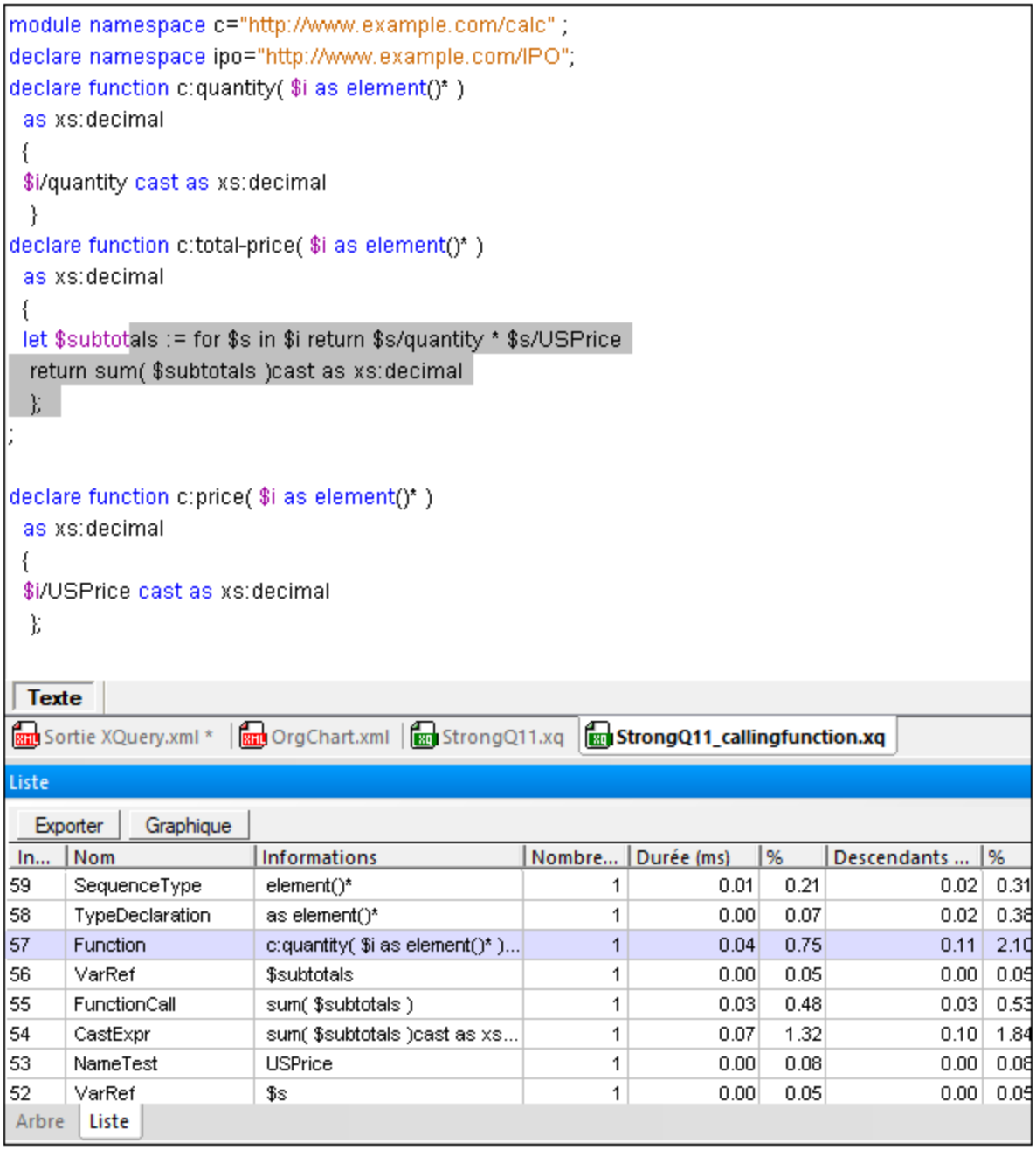
Cliquer sur une ligne dans le profileur pour marquer l'instruction correspondante dans le fichier qui a été analysé.

La capture d'écran suivante montre le mode Arborescence dans le profileur:

© 2016-2022 Altova GmbH



La capture d'écran suivante montre le Mode Liste dans le profileur pour le même document XQuery.



Utiliser les informations dans le profileur

Le profileur affiche les informations suivantes concernant chaque instruction dans le document XQuery :

- **Index :** un nombre attribué à chaque instruction dans l'ordre dans lequel l'instruction a été appelée.
- **Nom :** le nom de l'instruction XQuery.
- **Info :** Information about the instruction. For example, if the instruction is a variable declaration, this column contains the name of the variable and its value; if it is a function, then this contains the name and parameters of the function.

© 2016-2022 Altova GmbH

- **Hit Count: Nombre d'appels :** le nombre total de fois que l'instruction a été appelée pendant l'exécution.
- **Durée (ms)** et **% :** le nombre de millisecondes que l'instruction a prise pour s'exécuter sans prendre en compte le temps d'exécution de ses descendants et le pourcentage du temps d'exécution total.
- **Descendants et lui-même** et **% :** le temps passé à exécuter l'instruction et ses descendants et le pourcentage du temps d'exécution total.

Note : lors de l'utilisation du profilage de nombre d'appels, les temps contenus dans la fenêtre du Profileur sont la somme totale du temps d'exécution pour tous les appels au niveau de l'instruction. Lors de l'utilisateur d'un profilage de graphe d'appel, étant donné que chaque appel de l'instruction est recensé séparément, les temps montrés dans la fenêtre du profileur sont les durées d'une seule exécution de l'instruction.

9.2.3 Résultats du profileur : exportations et graphiques

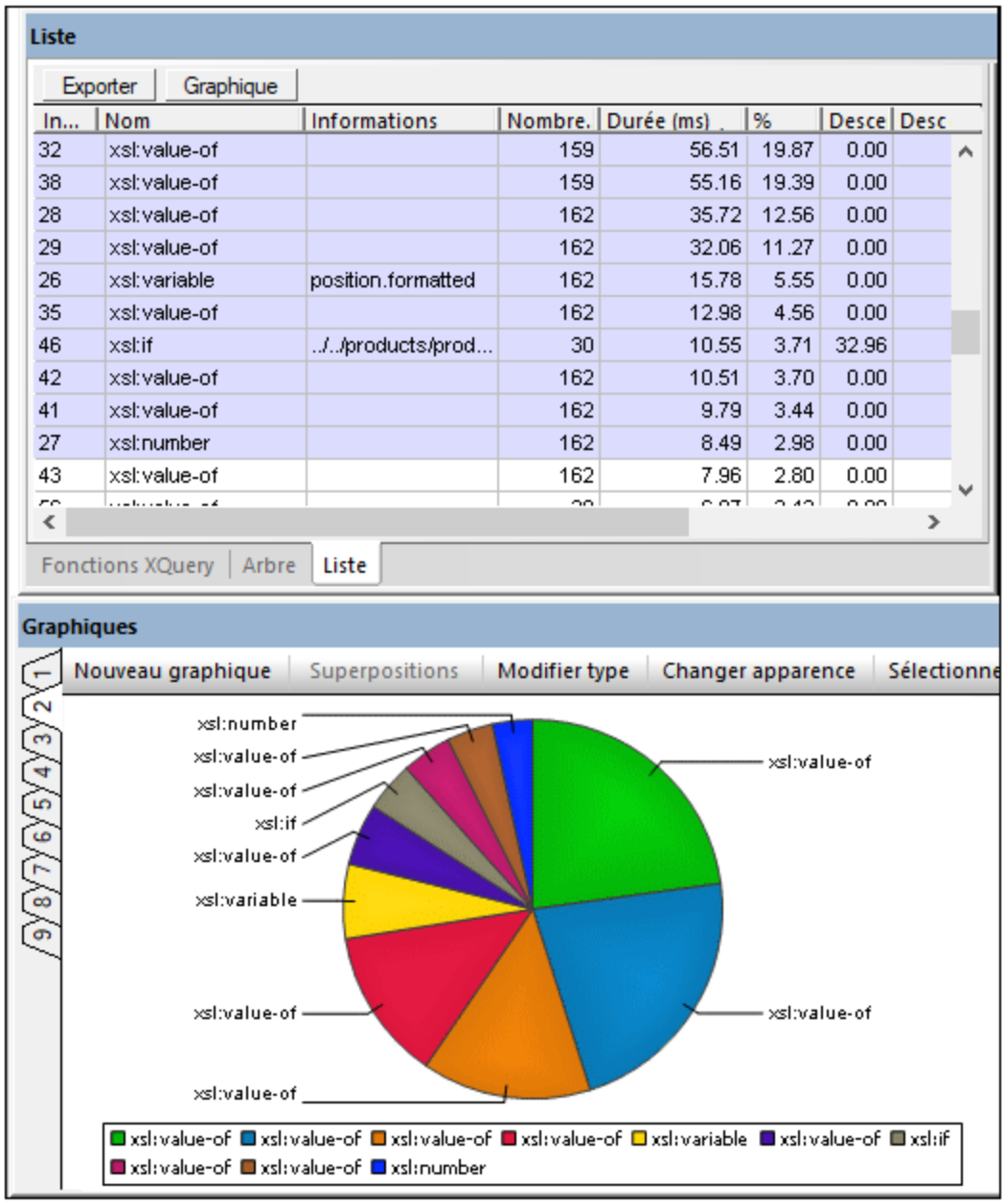
Après avoir exécuté le Profileur XSLT/XQuery, les résultats peuvent être exportés à un fichier XML et vers un graphique qui peut être sauvegardé en tant que fichier image.

Exportation

Cliquer sur la touche **Exporter** pour sélectionner un emplacement et un nom de fichier pour le fichier XML vers lequel les résultats du profileur peuvent être enregistrés. Pour obtenir un aperçu plus clair de la structure, il est conseillé de consulter le fichier XML dans le mode Grille. Par exemple, lorsque vous consultez un résultat de Profileur XSLT dans le mode Grille, vous verrez la structure du document XML sous la forme de trois niveaux hiérarchiques, chacun identifié par un élément `node`. Le premier élément `node` représente la racine de document, le deuxième élément `node` l'élément `xsl:stylesheet`, et le troisième élément `node` les éléments globaux (comme `xsl:output` et `xsl:template`). Les données de profilage sont stockées en tant qu'attributs de chacun des éléments `node`.

Graphique

Une fois le profileur XSLT/XQuery exécuté, un graphique des résultats ou un sous-ensemble des résultats peut être généré. Dans la fenêtre Profileur (*voir capture d'écran ci-dessous*), cliquer sur la touche **Graphique** pour générer le graphique dans la fenêtre de sortie Graphiques (*voir capture d'écran ci-dessous*).

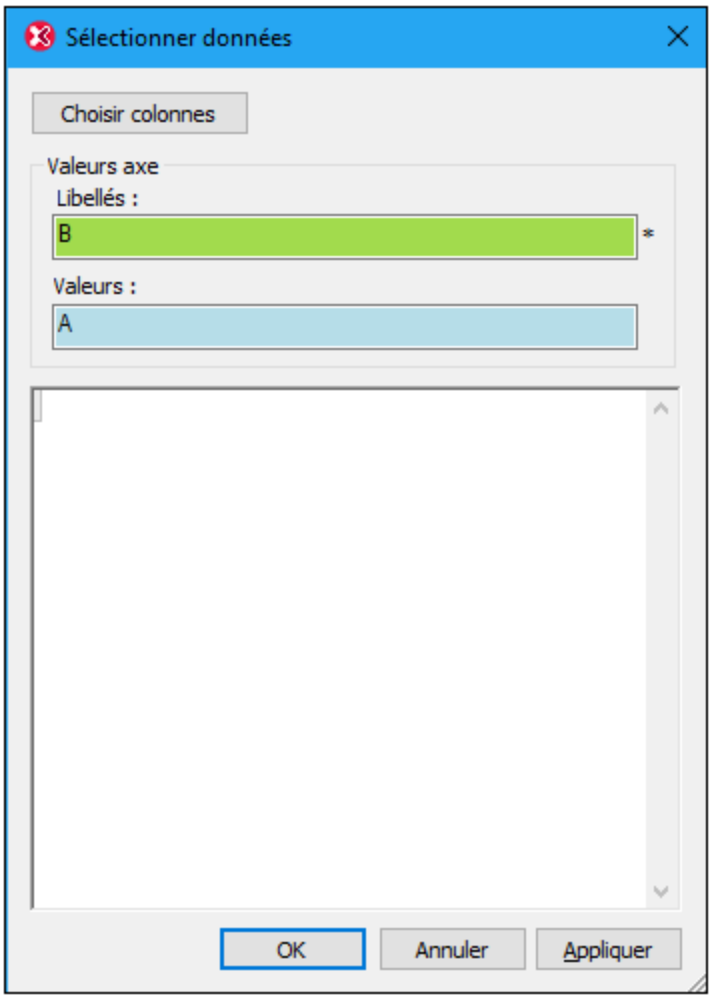


Veuillez noter les points suivants :

- Dans la fenêtre du Profileur (Mode Arborescence ou Mode Liste), un sous-ensemble des résultats peut être sélectionné en les marquant. Cliquer avec les touches **Ctrl** et/ou **Shift** pour marquer plusieurs items. Dans le Mode Liste, les résultats peuvent être triés sur la base des valeurs d'une colonne en cliquant sur l'en-tête de cette colonne. Cela peut être utile, par exemple, pour trier les items de résultat conformément aux items les plus chronophages puis pour sélectionner un sous-ensemble qui prend la plus grande partie du temps de transformation. En sélectionnant les sous-ensembles, il est possible de filtrer les items de résultats indésirables. Dans la capture d'écran ci-dessus, les items de résultat marqués ont été sélectionnés.

© 2016-2022 Altova GmbH

- Une fois qu'un graphique a été créé, son type (camembert, graphique à barres, à lignes, etc.) peut être changé en cliquant sur la touche **Changer type** de la fenêtre de sortie Graphiques. Les différents types de graphiques sont décrits en détail dans la section Graphiques de la documentation.
- Cliquer sur la touche **Sélectionner les données** de la fenêtre de sortie Graphiques pour faire apparaître le dialogue Sélectionner les données (*capture d'écran ci-dessous*). Dans ce dialogue, vous pouvez sélectionner les données pour l'axe X et pour l'axe Y à partir de la table des données qui est produite au cours du processus Sélectionner les colonnes. Afin de sélectionner les données pour l'axe X, cliquer dans la fenêtre Valeurs d'axe, puis, soit saisir la plage des valeurs de table (par exemple, `A1:A7`) ou glisser le curseur du début vers la fin de la plage. Procéder de même pour l'axe Y.



Cliquer sur la touche **Sélectionner les colonnes** pour changer la sélection des données pour la table des données. Voir XPath de source, Sélection de l'axe X, et Sélection de l'axe Y pour plus d'informations concernant le fonctionnement de la sélection de colonnes.

Pour plus d'informations concernant les graphiques, voir la section Graphiques de la documentation.

10 Expressions XPath/XQuery

Les expressions XPath et XQuery sont utilisées pour parcourir les arborescences XML. Avec l'ajout de la prise en charge des mappages et des arrays dans la version 3.1, les expressions XPath et XQuery de la version 3.1 peuvent aussi être utilisées pour parcourir des structures JSON. XPath est un sous-ensemble de XQuery, et toute expression qui est valide dans les deux langages retournera le même résultat dans les deux langages. Pour plus d'informations concernant ces deux langages, voir la Recommandation XPath 3.1 et la Recommandation XQuery 3.1.

Évaluer les expressions XPath/XQuery dans XMLSpy

XMLSpy fournit des fonctions puissantes pour générer des expressions XPath/XQuery, et pour évaluer et déboguer des expressions par rapport à des documents XML et JSON. Cela vous permet de créer et tester rapidement des expressions par rapport aux documents XML ou JSON sur lesquels vous prévoyez de les utiliser.

Ces fonctions d'analyse sont disponibles dans la **Fenêtre XPath/XQuery**, qui est une fenêtre de sortie située par défaut parmi les autres fenêtres de sortie en bas de l'interface de l'application. Les fonctions de la Fenêtre XPath/XQuery sont décrites dans les sous-sections de cette section.

Dans un scénario d'utilisateur typique, vous suivriez les étapes suivantes :

1. Ouvrir le document XML ou JSON pour lequel vous souhaitez créer ou évaluer une expression.
2. Saisir l'expression XPath/XQuery dans la Fenêtre XPath/XQuery.
3. Exécuter l'Évaluateur ou le débogueur pour voir les résultats. L'Évaluateur montre le résultat final, alors que le Débogueur vous permet d'avancer étape par étape dans le processus d'évaluation, vous montrant les aspects du résultat à chaque étape.

Dans cette section

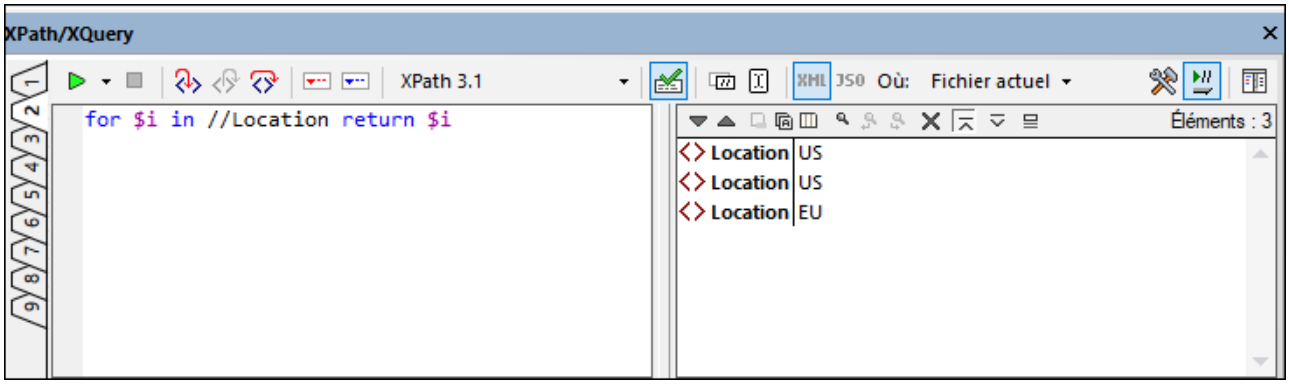
Cette section décrit les fonctions de la Fenêtre XPath/XQuery. Elle est organisée comme suit :

- À propos de la fenêtre XPath/XQuery
- Mode Évaluer : pour évaluer les expressions XPath/XQuery
- Mode Débogage : pour déboguer les expressions XPath/XQuery
- Générateur d'expression : pour créer des expressions XPath/XQuery
- Expressions XQuery pour JSON
- Points importants

© 2016-2022 Altova GmbH

10.1 À propos de la fenêtre XPath/XQuery

La Fenêtre XPath/XQuery (*capture d'écran ci-dessous*) vous permet de créer, d'évaluer et de déboguer des expressions XPath et XQuery par rapport aux documents XML ou JSON. (Les fonctions qui activent les requêtes JSON ont été introduites dans XPath/XQuery 3.1. *Voir également Transformations JSON avec XSLT/XQuery.)*

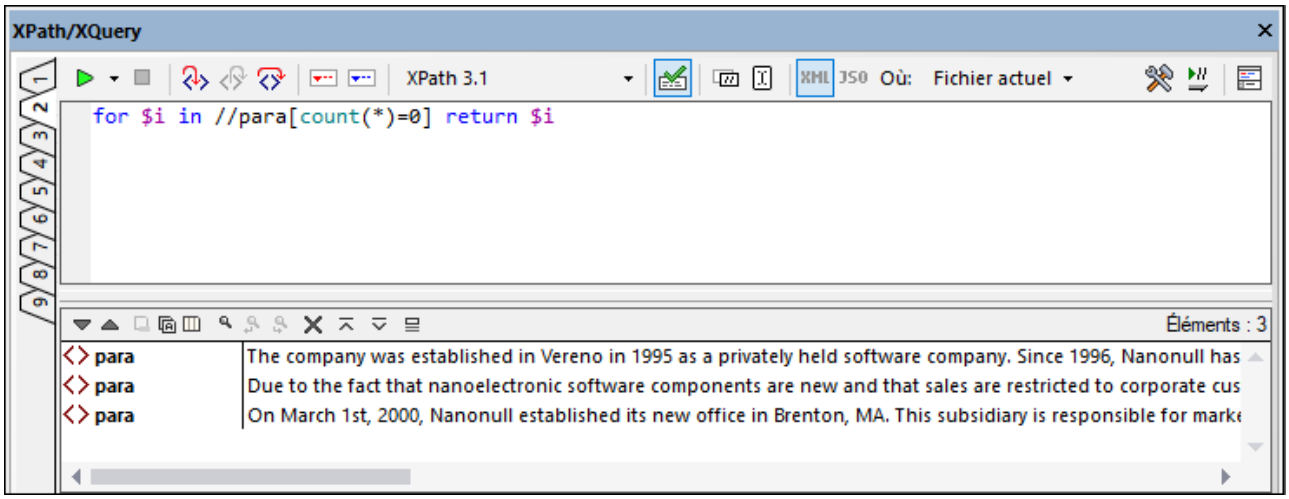


Pour une description complète de la fenêtre et sa barre d'outils, voir Fenêtre XPath/XQuery dans la description de l'interface.

Mises en page horizontales et verticales

Dans le coin droit de la barre d'outils, vous trouverez une touche (*voir les captures d'écran ci-dessus et ci-dessous*) qui vous permet de passer entre une mise en page horizontale et verticale. Vous pouvez changer de mise en page à tout moment et dans tous les modes (*voir Mode d'évaluation et Mode de débogage ci-dessous*). La capture d'écran ci-dessus montre la mise en page verticale, qui est utile lorsque l'expression XPath/XQuery (dans le volet de gauche, dans la capture d'écran ci-dessus) s'étend sur plusieurs lignes.

La mise en page horizontale (*capture d'écran ci-dessous*) est utile lorsque le résultat a de longues lignes qui s'étendent à l'horizontale.



Neuf onglets

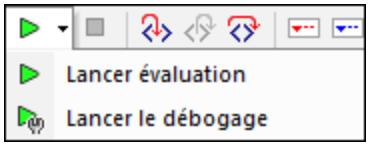
La fenêtre XPath/XQuery a neuf onglets, qui sont situés à gauche de la fenêtre *(voir capture d'écran ci-dessus)*. Le fait de disposer de plusieurs onglets vous permet de travailler avec des expressions différentes dans des onglets différents et comparer les résultats. Cliquer sur l'onglet que vous souhaitez consulter.

Mode Évaluation et Mode Débogage

La Fenêtre XPath/XQuery peut être utilisée en deux modes :

- Mode d'évaluation, dans lequel une expression XPath ou XQuery est évaluée par rapport à un ou plusieurs documents XML/JSON. L'expression est saisie dans le **volet Expression**, et le résultat est affiché dans le **Volet Résultats**. Vous pouvez cliquer sur des nœuds dans le résultat pour vous rendre à ce nœud dans le document XML ou JSON.
- Mode Déboguer, dans lequel vous pouvez déboguer une expression XPath/XQuery telle qu'elle s'applique au document XML actif actuellement. Vous pouvez définir des points d'arrêt et des points de trace, et passer étape par étape à travers l'évaluation. À chaque étape, vous pouvez voir le contenu des variables ainsi que les expressions Watch personnalisées définies pour contrôler les aspects supplémentaires de l'évaluation.

Pour alterner entre les deux modes, veuillez sélectionner la commande appropriée dans le menu déroulant **Lancer l'évaluation/Débogage** situé dans le coin gauche de la barre d'outils de la fenêtre *(voir capture d'écran ci-dessous)*.



Vous trouverez une descriptions de l'utilisation des deux modes dans les sous-sections de cette section.

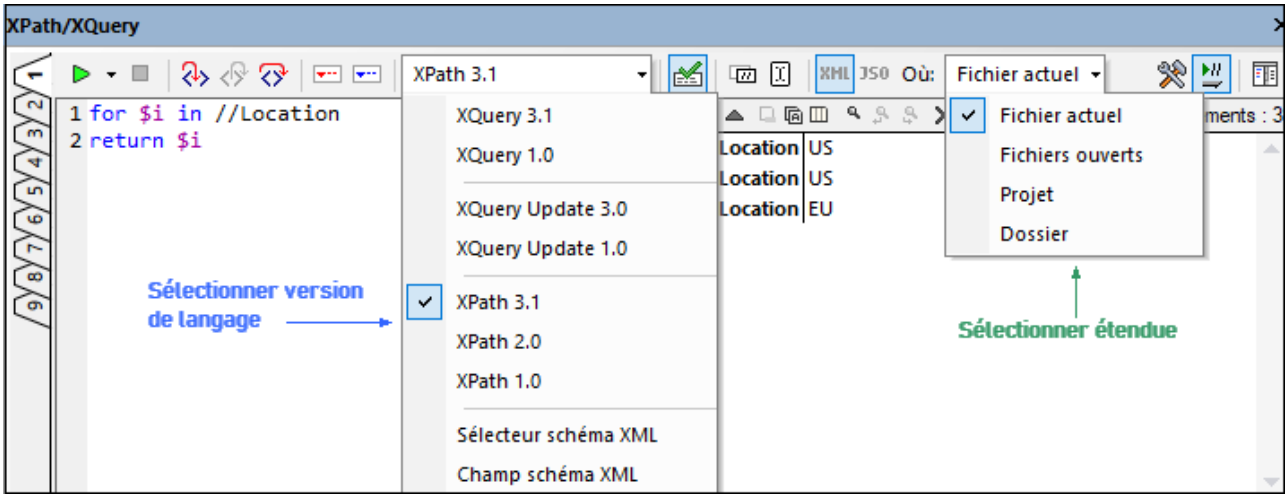
 © 2016-2022 Altova GmbH

Générateur d'expression XPath/XQuery

Dans les deux modes, le Générateur d'expression peut être utilisé pour vous aider à construire des expressions syntactiquement correctes. Allumer ou éteindre le Générateur d'expression avec la touche **Mode générateur** dans la barre d'outils principale .

10.2 Évaluer l'expression

La fenêtre XPath/XQuery vous permet de générer une expression XPath ou XQuery (dans une version de langage que vous pouvez choisir ; *voir capture d'écran ci-dessous*), ensuite, vous pouvez évaluer l'expression dans le cadre d'une étendue que vous spécifiez dans l'option *Où (voir capture d'écran*). L'expression peut être évaluée dans le fichier actuel, ainsi que dans les ensembles suivants de plusieurs fichiers : (i) tous les documents ouverts actuellement ; (ii) les fichiers du projet XMLSpy actuellement actif ; ou (iii) des fichiers d'un dossier sélectionné. Si vous souhaitez uniquement tester l'expression, un fichier convenable en tant qu'étendue est approprié. La fenêtre XPath/XQuery peut néanmoins aussi être utilisée pour trouver des données dans un ou plusieurs fichiers puis de la rapporter dans le volet Résultats ; dans ce cas, choisir un ensemble de fichier approprié dans l'étendue.

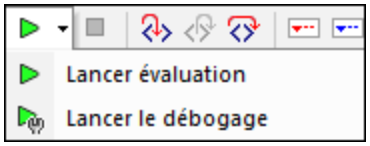


La fenêtre XPath/XQuery comprend une barre d'outils et deux volets : le volet Expression *(à gauche dans la capture d'écran ci-dessus)* et le volet Résultats *(à droite dans la capture d'écran)*.

Procédure d'évaluation

Pour évaluer une expression XPath/XQuery, procéder comme suit *(aussi se référer à la capture d'écran ci-dessus)*:

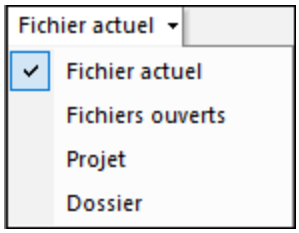
1. *Sélectionner le mode d'évaluation :* Sélectionner **Lancer évaluation** dans le menu déroulant de la commande **Lancer Évaluation/Débogage (F5)** (située en haut à gauche de la barre d'outils ; *voir capture d'écran ci-dessous*).



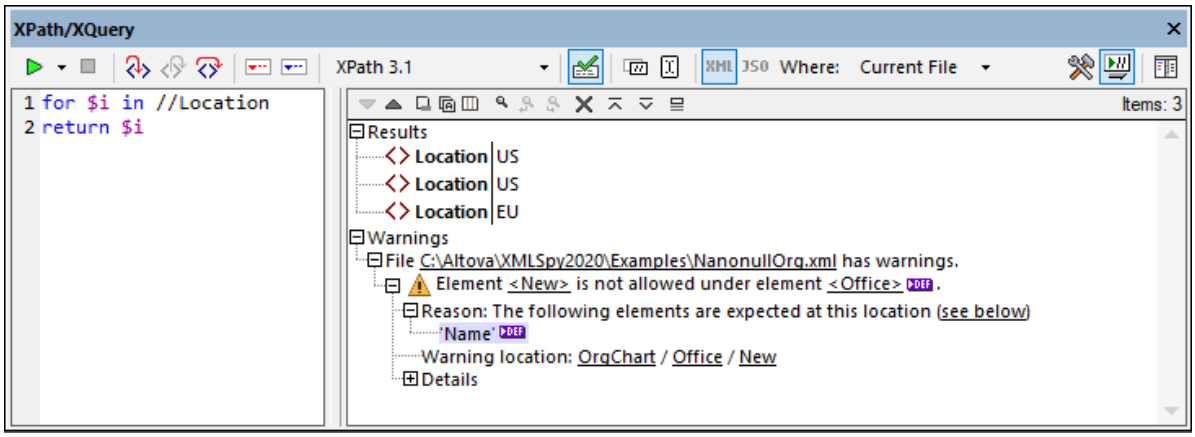
2. *Layout Horizontal/Vertical :* pour changer de layout, cliquer sur la touche **Layout Horizontal/Vertical** (située en haut à droite de la barre d'outils). Le layout par défaut est le layout vertical.
3. *Sélectionner version de langage :* dans la barre d'outils, sélectionner la version de langage pour l'expression que vous souhaitez évaluer; par exemple, XPath 3.1. Le langage par défaut est XPath 3.1. Voir aussi la note ci-dessous intitulée *XML Schema Selector and XML Schema Field*.

© 2016-2022 Altova GmbH

4. *Saisir expression:* dans le volet Expression, saisir l'expression à évaluer. Si vous avez besoin d'aide pour construire une expression, utiliser les assistants à la saisie du Générateur d'expression. Pour plus d'informations concernant les fonctions d'édition du volet Expression,voir aussi la note ci-dessous intitulée *Édition dans le volet Expression*.
5. *Choisir l'étendue de l'évaluation :* dans l'option *Où* de la barre d'outils, choisir les fichiers dans lesquels l'expression doit être évaluée. Les options sont : *Fichier actuel ; Fichiers ouverts ; Projet (XMLSpy) ;* ou *Dossier*. La valeur par défaut est *Fichier actuel*.



   Si *Fichier actuel* est sélectionné, le fichier qui est actif actuellement sera interrogé. En sélectionnant *Fichiers ouverts* l'expression sera évaluée par rapport à tous les fichiers actuellement ouverts dans XMLSpy. Le projet se réfère au projet XMLSpy actif actuellement. (Les dossiers externes dans un projet XMLSpy peuvent être exclus en cochant l'icône *Ignorer dossiers externes*). L'option *Dossier* vous permet de chercher le dossier requis ; le XPath d'expression sera évalué par rapport aux fichiers XML ou JSON dans ce dossier.
6. *Choisir évaluation XML ou JSON :* si l'étendue d'évaluation est le fichier actuel, le mode d'évaluation (XML ou JSON) est déterminé automatiquement par le type de conformité du type de fichier du document (mode JSON pour des fichiers conformes à JSON, mode XML pour des fichiers non-JSON). Ce mode détecté automatiquement ne peut pas être modifié, et les touches sont désactivées. Si l'étendue de l'évaluation est une option à fichier multiple, les deux touches sont activées et vous pouvez sélectionner le mode d'évaluation que vous souhaitez ; la valeur par défaut est celle des deux options précédemment sélectionnée.
7. *Définir le nœud contextuel :* Le nœud contextuel peut être défini soit sur : (i) le nœud racine, soit (ii) sur la sélection actuelle dans le document actif. Vous pouvez basculer entre les deux paramètres par le biais de la touche de barre d'outils **Définir sélection actuelle en tant que l'origine pour XPath/XQuery**. Le paramètre par défaut est le nœud racine.
8. *Validation XML :* si la touche de la barre d'outils **Valider XML** est activée (le paramètre par défaut), alors les fichiers XML en cours d'évaluation seront validés. Les erreurs sont traitées en tant qu'avertissement et sont rapportées dans le volet Résultats *(capture d'écran ci-dessous)*, mais l'évaluation continue.



9. *Évaluer l'expression :* si l'option de bascule de la barre d'outils **Évaluer sur saisie** est sélectionnée, le résultat de l'évaluation est affiché dans le volet Résultats pendant que vous saisissez l'expression. Si cette option n'est pas sélectionnée, l'évaluation doit être lancée explicitement, en cliquant la commande **Lancer Évaluation/Débogage (F5)** (situé en haut à gauche de la barre d'outils).

Touches de barre d'outils utilisées dans la procédure d'évaluation

	**Lancer Évaluation/Débogage (F5)**	Active la sélection du Mode Évaluation et lance l'évaluation
	**Stoppe Évaluation/Débogage (Shift+F5)**	Activé pendant l'évaluation. Cette fonction est utile si l'évaluation dure top longtemps ou se perd dans une boucle sans fin et que vous souahaitez donc interrompre l'évaluation.
	**Layout Horizontal/Vertical**	Passe entre le layout horizontal et vertical
	**Passer au mode Génération**	Passe au mode Générateur d'expression qui vous propose des assistants à la saisie sensibles au contexte pour vous assister dans la construction d'expressions
	**Copier XPath de la sélection actuelle**	Copie le chemin de localisation du nœud dans le document XML dans la dernière position du curseur dans le volet Expression
	**Mode d'évaluation XML/JSON**(bascule entre les modes d'évaluation XML et JSON)	L'icône marquée est l'option active de la paire. Lorsque l'étendue de l'évaluation est plusieurs fichiers, les deux icônes sont activées et une icône peut être sélectionnée. Sinon, le mode d'évaluation est détecté automatiquement conformément au type de fichier ; l'autre icône est désactivée.
	**Définir la sélection actuelle en tant que contexte**	Bascule le contexte d'expression entre le nœud racine et la sélection actuelle
	**Évaluation pendant la saisie**	Active l'évaluation des expressions alors que l'expression est en cours de saisie
	**Valider XML**	Une fois activé, le document XML cible est validé

Sélecteur de schéma XML et Champ de Schéma XML
Les options *Sélecteur de Schéma XML* et *Champ de schéma XML* sont utilisées pour un sous-ensemble étroit de cas particuliers XPath 1.0 et sont utiles lorsque des contraintes d'identité uniques ont été définies dans le Schéma XML. Lorsqu'une de ces options a été sélectionnée, seuls des tests de nom (et le caractère générique *) sont autorisés dans l'expression XPath, et les prédicats et les fonctions XPath ne peuvent pas être utilisés. De plus, en ce qui concerne l'option *Sélecteur de Schéma XML*, seules des expressions situées sur l'axe enfant sont autorisées ; en ce qui concerne l'option *Champ de Schéma XML*, les expressions se trouvant sur l'axe enfant et l'axe attribut sont autorisées. Pour plus d'informations, voir XML Schema: Structures Recommendation de W3C.

Édition dans le volet Expression
Veuillez noter les points suivants concernant l'édition des expressions dans le volet Expression :

© 2016-2022 Altova GmbH

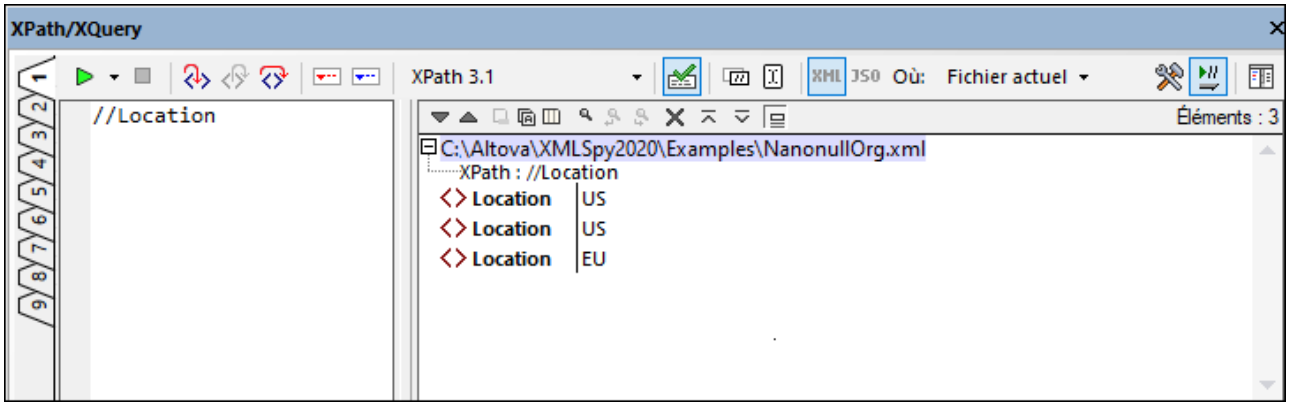
- Pour créer l'expression sur plusieurs lignes (pour une meilleure lecture), utiliser la touche **Retour**.
- Pour augmenter/réduire la taille du texte dans le champ d'expression, cliquer dans le champ d'expression, puis appuyer sur **Ctrl** et tourner la roue. *Veuillez noter que cela s'applique également dans le volet Résultats.*
- Au lieu de saisir manuellement l'expression de chemin de localisation d'un nœud, vous pouvez laisser XMLSpy la saisir pour vous. Pour ce faire : (i) Placer le curseur sur le point dans l'expression XPath dans lequel vous souhaitez saisie le chemin de localisation ; (ii) Placer le curseur dans le nœud que vous souhaitez cibler ; (iii) Dans la barre d'outils, cliquer sur la touche **Copier le XPath de la sélection actuelle**. Cela saisit le chemin de localisation du nœud sélectionné dans l'expression. Le chemin de localisation sera un chemin absolu commençant au niveau du nœud racine du document.

Évaluation XQuery et JSON

- Pour plus d'informations concernant des évaluations XQuery, voir la section, évaluation XQuery . (Les icônes XQ sont destinées à l'évaluation XQuery ; les icônes XQU sont destinées aux exécutions XQuery Update.)
- Pour une description de la requête de documents JSON, voir Expressions XQuery pour JSON et Transformations JSON avec XSLT/XQuery.

Volet Résultats

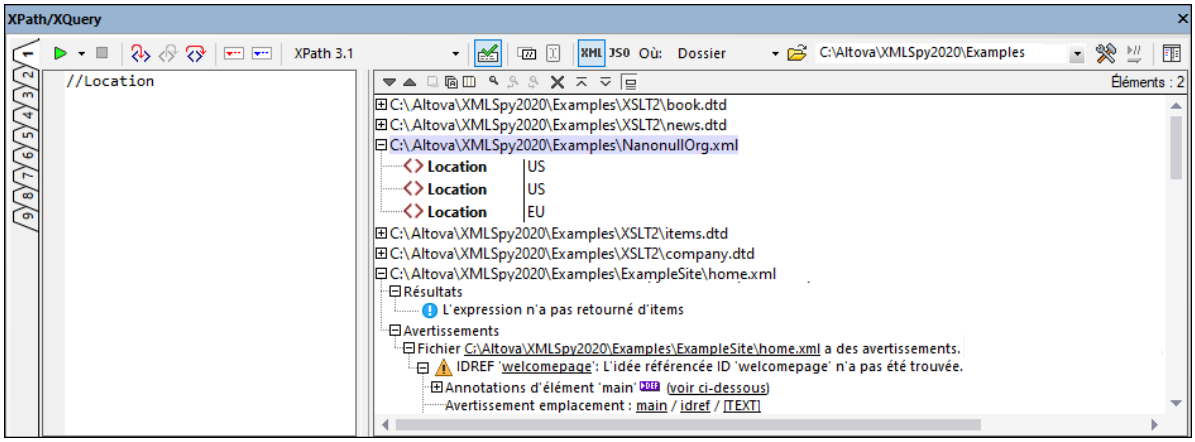
Le volet Résultats est affiché dans la capture d'écran ci-dessous, à droite. Veuillez noter qu'il a sa propre barre d'outils.



Le volet Résultats propose les fonctions suivantes :

- Activer l'icône *Afficher en-tête dans la sortie* si, dans la sortie, vous souhaitez afficher l'emplacement du fichier XML et de l'expression XPath (comme indiqué dans la *capture d'écran ci-dessous*).
- La liste de résultats consiste en deux colonnes : (i) un nom de nœud ou un type de données ; (ii) le contenu du nœud.
- Si l'expression XPath retourne des nœuds (comme des éléments ou des attributs), vous pouvez sélectionner si le contenu des nœuds doit être affiché en tant que la valeur du nœud. Pour ce faire, activer la bascule *Afficher résultat complet*.
- Lorsque le résultat contient un nœud (y compris un nœud de texte), contrairement aux littéraux générés par l'expression, le fait de cliquer sur ce nœud dans le volet Résultats marque le nœud correspondant dans le document XML dans la fenêtre Principale.
- Si l'évaluation est effectuée sur plusieurs fichiers (*spécifiés dans l'option* Où), alors les résultats de chaque fichier sont recensés séparément sous le chemin de ce fichier (*voir capture d'écran ci-*

dessous). Si le mode d'évaluation est XML, alors les fichiers conformes à XML sont évalués, les autres types sont ignorés. Si le mode d'évaluation est JSON, alors les fichiers conformes à JSON sont évalués, les autres types sont ignorés.



- Si les résultats impliquent plusieurs fichiers, cliquer sur un nom de fichier dans la liste de résultats pour ouvrir le fichier dans XMLSpy et en faire le fichier actif.
- Vous pouvez copier les deux colonnes d'une sous-ligne de résultat ou uniquement la colonne de valeur. Pour copier toutes les colonnes, cliquer avec la touche de droite sur une sous-ligne et activer **Copie inclut toutes les colonnes**. (En alternative vous pouvez activer/désactiver la commande par le biais de son icône dans la barre d'outils du volet Résultats.) Ensuite, cliquer avec la touche de droite sur la sous-ligne que vous souhaitez copier et choisir soit **Copier sous-ligne** (pour cette sous-ligne) ou **Copier tout** (pour toutes les sous-lignes).

Barre d'outils du volet Résultats

La barre d'outils du volet Résultats contient des icônes qui propose des fonctions de navigation, de recherche et de copie. Ces icônes sont décrites dans la table ci-dessous de gauche à droite. Les commandes correspondantes sont aussi disponibles dans le menu contextuel des items de la liste de résultat.

Icône	Fonction
Suivant, Précédent	Sélectionne, respectivement, l'item suivant et précédent dans la liste de résultats
Copier la ligne de texte sélectionnée dans le presse-papiers	Copie la colonne de valeur de l'item de résultat sélectionné dans le presse-papiers. Afin de copier toutes les colonnes, activer la commande *La copie contient toutes les colonnes (voir ci-dessous)*
Copier tous les messages dans le presse-papiers	Copie la colonne de valeur de tous les items de résultat dans le presse-papiers, y compris des valeurs vides. Chaque item est copié en tant que ligne séparée
La copie contient toutes les colonnes	Passe entre la copie (i) de toutes les colonnes, ou (ii) uniquement la colonne de valeur. Le séparateur de colonne est un espace unique
Chercher	Ouvre un dialogue *Chercher* pour chercher un string, y compris des caractères spéciaux, dans la liste de résultat
Chercher précédent	Cherche l'occurrence précédente du terme qui a été saisi en dernier dans le dialogue *Chercher*
Chercher suivant	Cherche l'occurrence suivante du terme qui a été saisi en dernier dans le dialogue

© 2016-2022 Altova GmbH

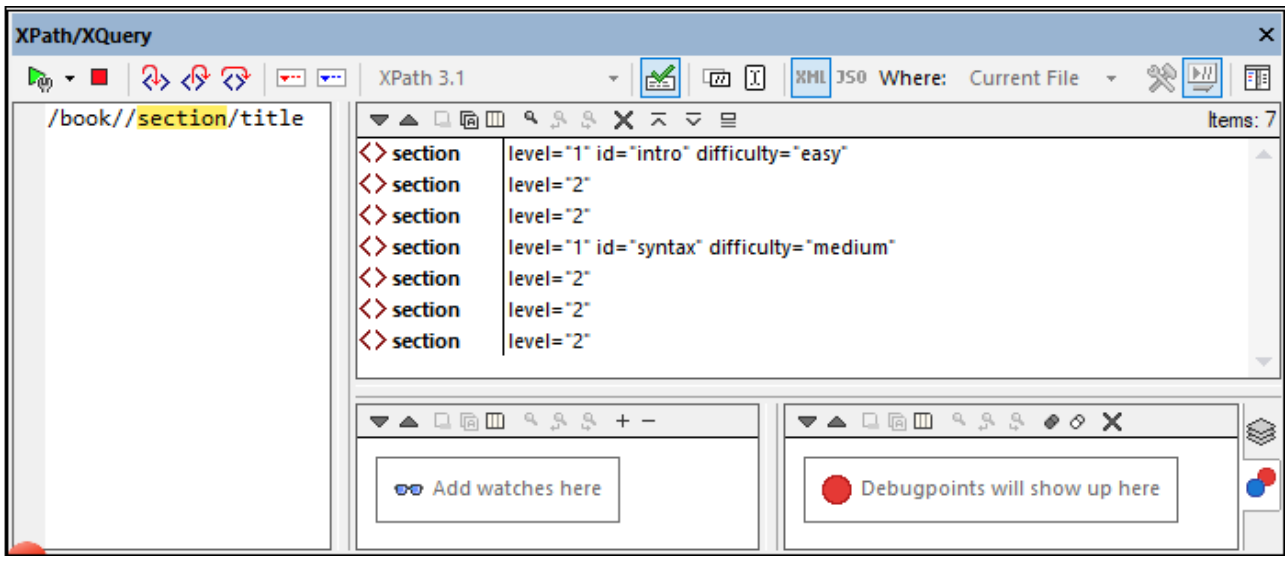
	Chercher
Supprimer	Supprime la liste de résultats
Réduire les résultats multi-ligne en une seule ligne	Si la colonne de valeur d'un item de résultat contient un texte multi-ligne (du texte qui contient un/des caractère/s de saut de ligne), vous pouvez basculer entre un affichage multi-ligne et ligne unique
Afficher le résultat complet	Affiche le contenu entier des nœuds en tant que la valeur du nœud
Afficher l'en-tête dans la sortie	Affiche l'emplacement du/des fichier/s XML et l'expression XPath

10.3 Déboguer l'expression

Le mode Débogage de la fenêtre XPath/XQuery *(voir capture d'écran ci-dessous)* vous permet de déboguer une expression XPath/XQuery dans le contexte du fichier actif.

Dans le mode Débogage, deux volets supplémentaires sont ajoutés au volet Résultat *(voir capture d'écran ci-dessous)*:

- Le volet Pile d'appel et Points de débogage, ils ont chacun un onglet individuel dans le volet
- Le volet Variables et Expressions Watch ; les expressions watch et les variables sont indiquées dans le même volet.



Les volets Expression et Résultat peuvent s'afficher horizontalement ou verticalement. Pour passer entre les deux vues, cliquer sur la touche **Layout Horizontal/Vertical** (en haut à droite dans la barre d'outils de la fenêtre).

Le mode Débogueur propose les fonctions suivantes :

- Vous permet d'entrer pas à pas dans le processus d'évaluation XPath, une étape à la fois, pour voir comment l'expression XPath est évaluée. Utiliser la touche de la barre d'outils **Pas à pas détaillé (F11)**. À chaque étape d'évaluation, la partie de l'expression actuellement évaluée est marquée en jaune *(voir capture d'écran ci-dessus)*, alors que le résultat de l'évaluation de cette étape est affiché dans le volet Résultats. Par exemple, dans la capture d'écran ci-dessus, tous les éléments descendants `section` de l'élément `book` ont été sélectionnés.
- Définir des points d'arrêt à l'endroit où vous souhaitez interrompre l'évaluation et vérifier les résultats à ces endroits. Vous pouvez faire défiler l'évaluation en ne vous arrêtant que dans les points d'arrêt. Utiliser la touche de la barre d'outils **Lancer le débogage (F5)**. Cela est plus rapide que de s'interrompre à chaque étape avec **Pas à pas détaillé (F11)**.
- Définir des points de trace pour consulter un rapport des résultats dans les étapes marquées en tant que points de trace. L'évaluation ne s'interrompra pas (sauf dans les points d'arrêt), mais les résultats de point de trace seront affichés dans une liste dans le volet Résultats.
- Les expressions Watch peuvent être utilisées pour vérifier des information (comme des données de document ou des aspects de l'évaluation). Cela est particulièrement utile dans les points d'arrêt.

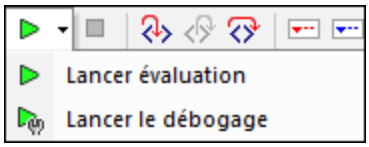
© 2016-2022 Altova GmbH

- Les variables qui se trouvent dans l'étendue, y compris leurs valeurs sont affichées dans le volet Variables et Expressions Watch.
- Les appels de processeur d'une étape d'évaluation sont affichés dans l'onglet Pile d'appel du volet Pile d'appels et Points de débogage.
- Si des points d'arrêt et de trace ont été définis, ceux-ci sont affichés dans l'onglet Points de débogage du volet Pile d'appels et Points de débogage.

Pour plus d'informations concernant ces fonctions, voir leur description ci-dessous.

Configurer le mode Débogage

Pour passer au mode Débogage, cliquer sur **Lancer débogage** dans le menu déroulant de la commande **Lancer évaluation/débogage (F5)** (situé en haut à gauche de la barre d'outils ; *voir capture d'écran ci-dessous*). Dans le mode Débogage, les volets Expressions Watch et Points de débogage apparaîtront. Pour lancer le débogage de l'expression actuelle, cliquer sur **Lancer Débogage** ou sur **F5**.



Veuillez noter que le mode de débogage ne fonctionne qu'avec le fichier actuel, il ne peut pas être utilisé avec plusieurs fichiers. En résultat, l'option *Où* sera définie automatiquement sur *Fichier actuel* lorsque vous passez dans ce mode.

Touches de configuration du mode Débogage

	**Lancer évaluation/débogage (F5)**	Lance le débogueur
	**Layout Horizontal/Vertical**	Alterne entre les layouts horizontaux et verticaux
	**Passer au mode Création**	Passe dans le mode Générateur d'expression qui propose des assistants à la saisie sensibles au contexte pour vous aider à construire des expressions
	**Copier XPath de la sélection actuelle**	Copie le chemin de localisation du nœud dans le document XML dans la dernière position de curseur dans le volet Expression
	**Mode d'évaluation XML/JSON**	L'icône marquée de la paire est l'option active. Le mode d'évaluation est détecté automatiquement conformément au type de fichier ; l'autre icône est désactivée.
	**Définir la sélection actuelle en tant que contexte**	Bascule le contexte d'expression entre le nœud racine et la sélection actuelle
	**Évaluation sur saisie**	Active l'évaluation des expressions pendant la saisie

	**Valider XML**	Une fois activé, les documents XML cibles sont validés

Exécuter le débogage

Les étapes principales pour déboguer une expression XPath/XQuery sont généralement les suivantes :

1. Faire du fichier XML/JSON sur lequel vous souhaitez exécuter l'expression le fichier actif.
2. Saisir l'expression XPath/XQuery dans le volet d'expression.
3. Définir les points d'arrêt ou les points de trace que vous souhaitez. Un point d'arrêt est un point au niveau duquel l'évaluation est interrompue. Un point de trace est un point dans l'évaluation qui est enregistré ; les points de trace permettent donc un chemin traçable des résultats d'évaluation.
4. Si vous cliquez sur **Lancer débogage**, l'évaluation est effectuée dans une étape jusqu'à la fin à moins qu'un point d'arrêt a été marqué dans l'expression. Cliquer sur **Lancer débogage** plusieurs fois pour progresser à travers chaque point d'arrêt à la fin de l'évaluation.
5. Utiliser la fonction Pas à pas détaillé/sortant/principal pour se rendre étape par étape par le biais de l'évaluation.

Touches de débogage

	**Lancer débogueur (F5)**	Lance le débogage. L'évaluation se rend directement à la fin, en s'arrêtant uniquement pour les points d'arrêt
	**Stopper débogage (Shift+F5)**	Quitte l'évaluation et arrête le débogueur
	**Pas à pas détaillé (F11)**	Traiter à travers l'évaluation, une étape à la fois.
	**Pas à pas sortant (Shift+F11)**	Sort en dehors de l'étape de l'évaluation actuelle, et se rend à l'étape parent
	**Pas à pas principal (Ctrl+F11)**	Passe au-dessus des étapes descendantes
	**Insérer/Supprimer point d'arrêt (F9)**	Insère/supprime un point d'arrêt dans l'étape d'expression dans laquelle vous placez le curseur
	**Insérer/Supprimer point de trace (Shift+F9)**	Insère/supprime un point de trace dans l'étape d'expression dans laquelle vous placez le curseur

Étapes de pas à pas détaillé, sortant , principal

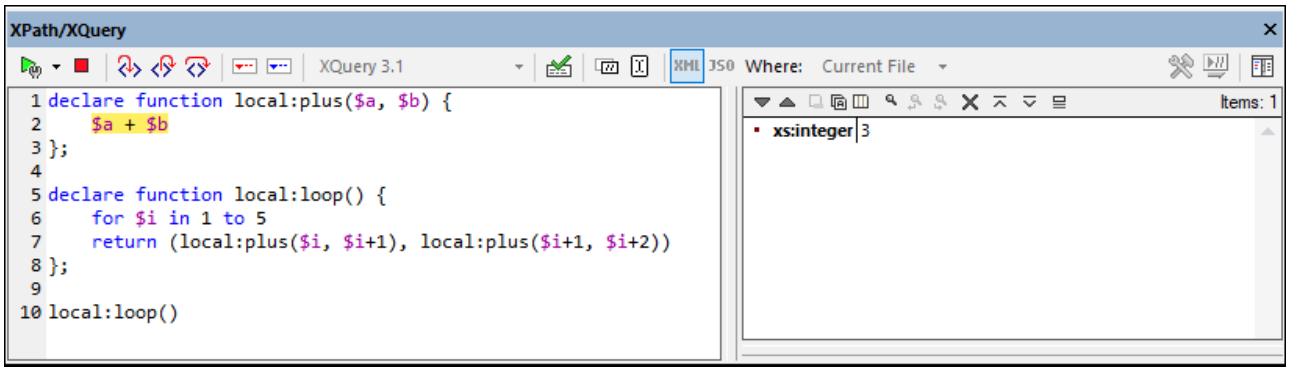
La fonction *Pas à pas détaillé* vous permet de passer étape par étape à travers l'évaluation. Chaque clic de cette commande vous amène à travers l'étape suivante de l'évaluation ; l'étape actuelle est affichée en marquant l'expression *(voir capture d'écran ci-dessous)*. La fonction *Pas à pas sortante* vous permet de vous amener à une étape d'un niveau supérieur en tant que l'étape actuelle, alors que la fonction *Pas à pas principal* passe au-dessus des étapes de niveau inférieur et vous amène dans l'étape suivante du même niveau. Vous pouvez tester la fonction *Pas à pas* en collant l'expression XQuery 3.1 indiquée ci-dessous dans le volet Expression et cliquer sur les trois touches *Pas à pas* pour découvrir comment elles fonctionnent.

© 2016-2022 Altova GmbH

Expression XQuery 3.1 pour tester la fonctionnalité Pas à pas détaillé, sortant, principal

```
declare function local:plus($a, $b) {
        $a + $b
};
declare function local:loop() {
        for $i in 1 to 10
        return (local:plus($i, $i+1), local:plus($i+1, $i+2))
};
local:loop()
```

La capture d'écran ci-dessous montre l'évaluation lorsque le traitement a été interrompu lorsque l'étape addition `$a + $b` a été atteinte pendant le premier passage à travers la boucle ; c'est à dire, lorsque `$i=1`. Au niveau de cette étape d'addition, le résultat montre `3` (en tant que conséquence de `1+2`).



Points d'arrêt

Les points d'arrêt sont des points dans lesquels vous souhaitez que le débogueur stoppe une fois qu'il a été lancé avec **Lancer débogage**. Ils sont utiles lorsque vous souhaitez analyser une partie spécifique de l'expression. Lorsque le Débogueur s'arrête au point d'arrêt, vous pouvez contrôler le résultat et vous pouvez ensuite utiliser la fonction **Pas à pas détaillé** pour afficher les résultats des étapes suivantes de l'évaluation. Pour définir un point d'arrêt, placer le curseur dans l'expression à l'endroit dans lequel vous souhaitez avoir un point d'arrêt, et cliquer sur la touche de la barre d'outils **Insérer/Supprimer Point d'arrêt (F9)**. Le point d'arrêt sera marqué avec une surligne en pointillé rouge. Pour supprimer un point d'arrêt, le sélectionner et cliquer sur **Insérer/Supprimer Point d'arrêt (F9)**.

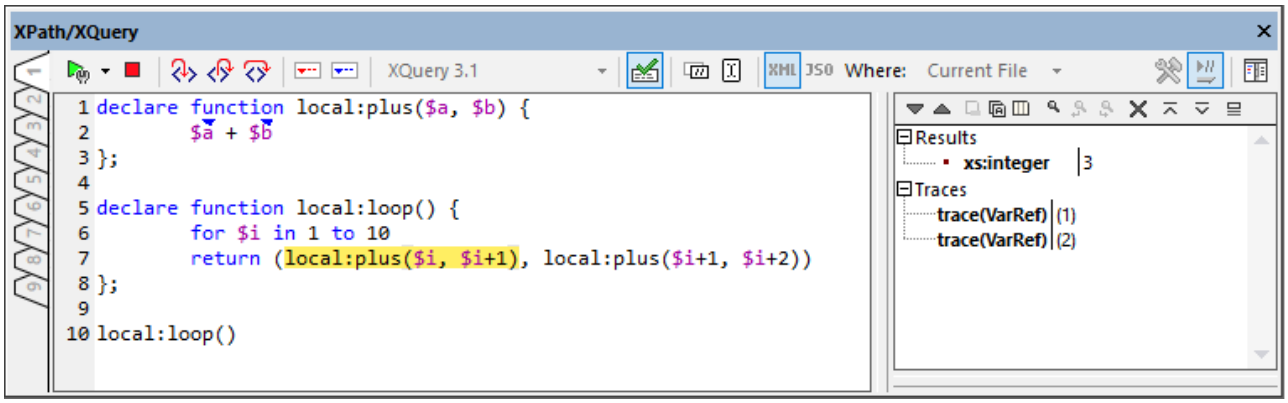
Voir aussi les Points de débogage ci-dessous.

Points de trace

Les points de trace sont des points dans lesquels les résultats sont enregistrés. Ces résultats sont affichés dans l'arborescence *Traces* de l'onglet *Résultat* (*voir capture d'écran ci-dessous*). Cela vous permet de voir tous les résultats d'évaluation des parties individuelles de l'expression. Par exemple, dans la capture d'écran ci-dessous, des points de trace ont été définies dans `$a`, `$b`, et `local:plus($i, $i+1)`; les résultats au niveau de ces points de trace pendant la première itération à travers la boucle sont affichés dans l'arborescence *Traces* : `$a=1`, `$b=2`, et `local:plus($i, $i+1)=3`.

Pour définir un point de trace, placer le curseur au niveau dans lequel vous souhaitez avoir le point de trace, puis cliquer sur la touche de la barre d'outils **Insérer/Supprimer Point trace (Shift+F9)**. Le point de trace

sera marqué avec une surligne en pointillé bleu (*voir capture d'écran ci-dessous*). Pour supprimer un point de trace, le sélectionner et cliquer sur **Insérer/Supprimer Point trace (F9)**.

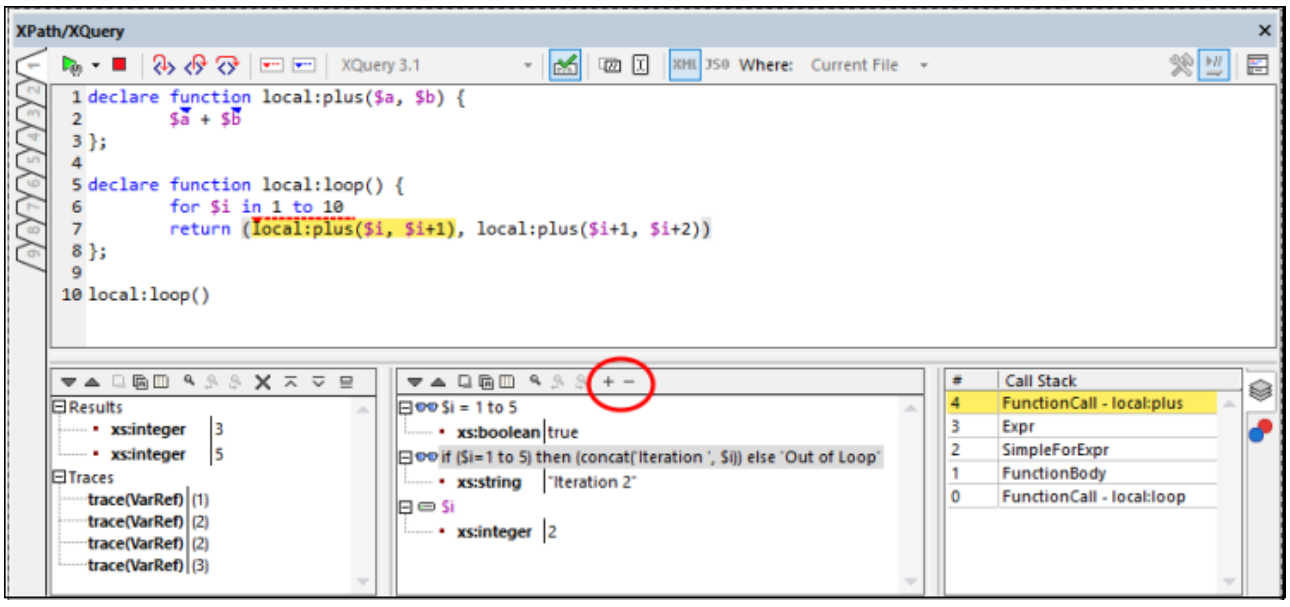


Note: Si le point d'arrêt et le point de trace sont définis sur une partie de l'expression, alors la surligne est composée de pointillés rouge et bleu.

Voir aussi Points de débogage ci-dessous.

Variables, Expressions Watch et Pile d'appels

Les variables et les expressions watch sont affichées dans le volet Variables et Expressions Watch *(volet central inférieur dans la capture d'écran ci-dessous)*.



Variables

Les variables qui ont été déclarées dans l'expression et qui sont incluses dans l'étape d'évaluation actuelle seront affichées avec leurs valeurs actuelles respectives. Par exemple, dans la capture d'écran ci-dessus, le traitement vient d'atteindre l'appel dans la fonction `local:plus`. La variable `$i` se trouve dans la plage contenue dans la fonction `local:loop` et vient d'être incrémentée à `2`. Donc `$i` est affiché avec sa valeur actuelle. Les

variables `$a` et `$b` ne se trouvent pas dans `local:loop` et ne se trouvent donc pas dans la plage ; par conséquent, elles ne sont pas affichées.

Expressions Watch

Les expressions Watch sont des expressions que vous pouvez saisir, soit avant le début de l'évaluation ou pendant une interruption dans une évaluation. Elles peuvent être utilisées pour les objectifs suivants :

- Pour tester des conditions spécifiques. Par exemple, dans la capture d'écran ci-dessus, l'expression watch `$i=1 to 5` est utilisée pour tester si la variable `$i` a une valeur dans la plage donnée. Le résultat `true` nous dit que cette condition a été remplie.
- Pour trouver des données dans le cadre d'un contexte particulier. Par exemple, dans le cadre du contexte d'un élément `Company`, nous pouvez saisir une expression watch `@id` pour consulter ce code de clients de l'entreprise dans le document XML cible.
- Pour générer des données supplémentaires. Par exemple, dans la capture d'écran ci-dessus, l'expression watch `if ($i=1 to 5) then (concat("Iteration ", $i)) else "Out of Loop"` can generate a suitable string to indicate in which iteration of the loop the evaluation currently is.

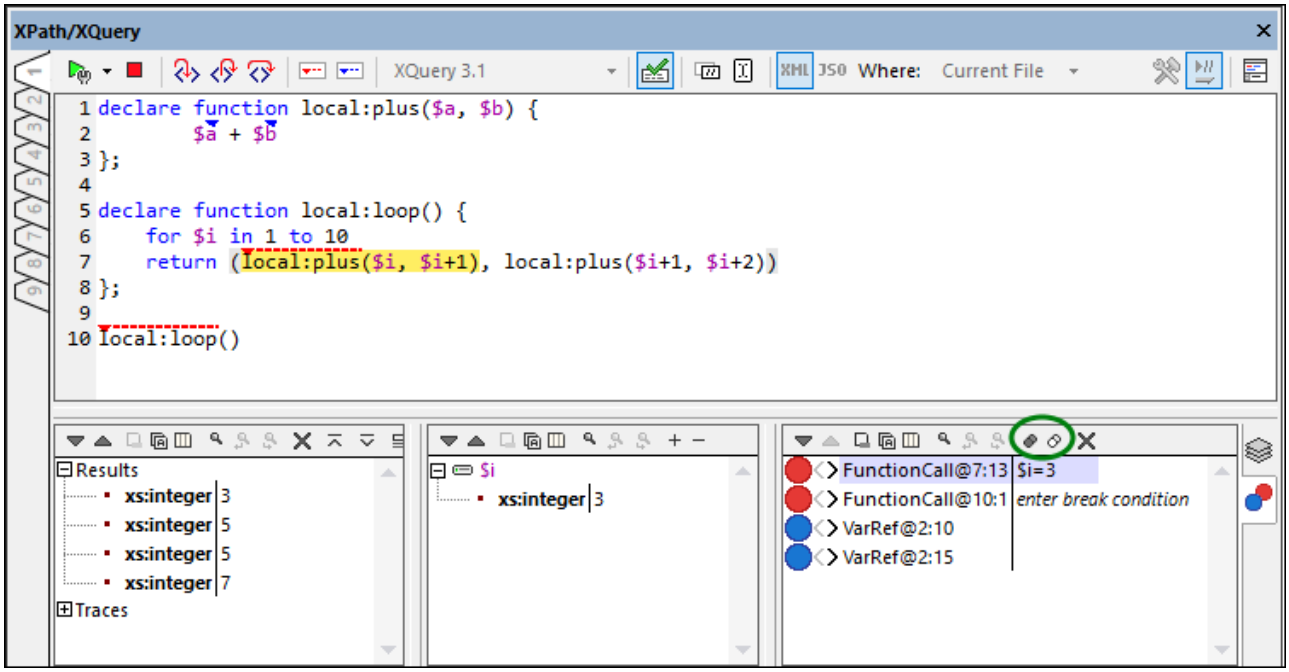
Pour saisir l'expression watch, cliquer sur **Ajouter entrée Watch** dans la barre d'outils du volet *(encerclé en rouge dans la capture d'écran ci-dessus)*, puis saisir l'expression et cliquer sur **Entrée** une fois que vous avez terminé. Pour supprimer une expression watch, la sélectionner et cliquer sur **Supprimer l'entrée watch sélectionnée** dans la barre d'outils. Si, pendant le débogage, l'expression ne peut pas être évaluée correctement pour plusieurs raisons (par exemple, si une de ses variables se trouve en dehors de la plage), alors l'expression watch passe au rouge.

Pile d'appels

L'onglet *Pile d'appel* du volet Pile d'appel et Points de débogage *(volet situé en bas à droite dans la capture d'écran ci-dessus)* affiche les appels de processeur jusque dans ce point dans le débogage. L'appel de processeur actuel est marqué en jaune. Veuillez noter que seuls les appels qui mènent directement à l'étape d'évaluation actuelle sont affichés. Par exemple, dans la capture d'écran ci-dessus, l'étape d'évaluation actuelle est un appel de fonction dans la fonction `local:plus`. Maintenant, bien qu'il s'agisse de la seconde itération de `local:loop`, les appels de processeur de la première itération **ne sont pas** affichés. Cela est dû au fait que ces appels se trouvent sur un niveau parallèle dans l'appel de fonction actuel et n'ont pas mené à la fonction.

Points de débogage

L'onglet Points de débogage du volet Pile d'appel et Points de débogage *(volet inférieur de droite dans la capture d'écran ci-dessous)* montre les points d'arrêt (avec des cercles rouges) et des points de trace (cercles bleus) que vous avez défini dans l'expression. Chaque point de débogage est recensé avec son numéro de ligne et de caractère. Par exemple, `FunctionCall@7:13` signifie qu'il y a un point de débogage sur la ligne 7, caractère 13.



Veuillez noter les fonctions suivantes :

- En ce qui concerne les points d'arrêt, vous pouvez saisir la **condition d'arrêt** en double-cliquant sur *Saisir condition d'arrêt*, en saisissant l'expression pour la condition, et en appuyant sur **Entrée**. Ce point d'arrêt sera activé uniquement lorsque la condition évalue sur `true`. Par exemple, dans la capture d'écran ci-dessus, la condition d'arrêt `$i=3` activera le point d'arrêt sur cet appel de fonction uniquement lorsque la valeur de `$i` est `3`. La capture d'écran montre que l'évaluation s'est interrompue à ce point d'arrêt.
- Vous pouvez activer/désactiver tous les points de débogage en cliquant sur leurs touches de barre d'outils respectives : **Activer tous les points de débogage** et **Désactiver tous les points de débogage** *(touches encerclées en vert dans la capture d'écran ci-dessus)*. Lorsqu'un point de débogage est désactivé pour toutes les évaluations jusqu'à ce qu'il soit activé à nouveau.
- Vous pouvez activer/désactiver des points d'arrêt individuels dans leurs menus contextuels respectifs.

Commandes de la barre d'outils dans les volets

Les volets de la fenêtre XPath/XQuery dans le mode Débogage (*voir capture d'écran ci-dessus*) contient des touches qui fournissent des fonctions de navigation, de recherche et de copie. Ces touches, de gauche à droite, sont décrites dans la table ci-dessous. Les commandes correspondantes sont aussi disponibles dans le menu contextuel des items de la liste de résultat.

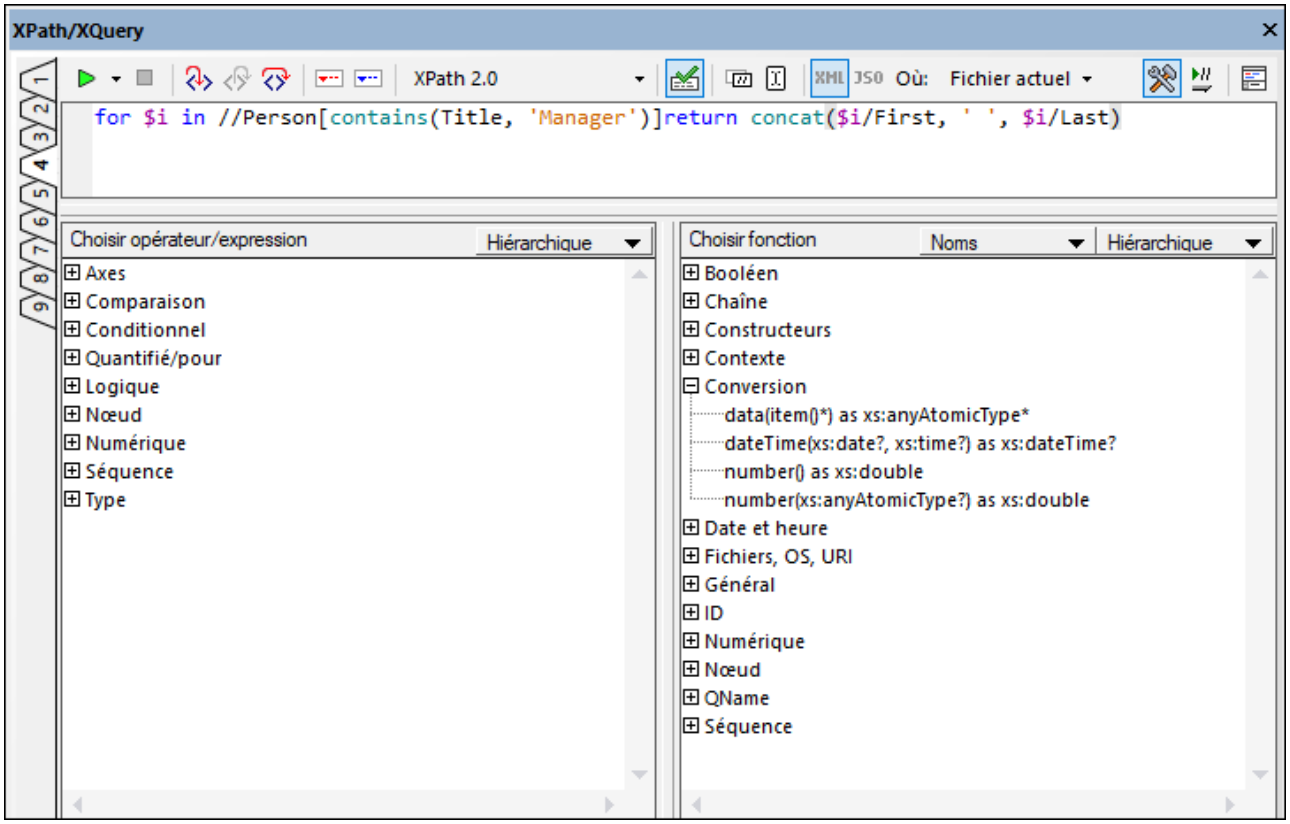
Icône	Fonction
Suivant, Précédent	Sélectionne, respectivement, l'item suivant et précédent dans la liste de résultats
Copier la ligne de texte sélectionnée dans le presse-papiers	Copie la colonne de valeur de l'item de résultat sélectionné dans le presse-papiers. Afin de copier toutes les colonnes, activer la commande *La copie contient toutes les colonnes (voir ci-dessous)*

© 2016-2022 Altova GmbH

Copier tous les messages dans le presse-papiers	Copie la colonne de valeur de tous les items de résultat dans le presse-papiers, y compris des valeurs vides. Chaque item est copié en tant que ligne séparée
La copie contient toutes les colonnes	Passe entre la copie (i) de toutes les colonnes, ou (ii) uniquement la colonne de valeur. Le séparateur de colonne est un espace unique
Chercher	Ouvre un dialogue *Chercher* pour chercher un string, y compris des caractères spéciaux, dans la liste de résultat
Chercher précédent	Cherche l'occurrence précédente du terme qui a été saisi en dernier dans le dialogue *Chercher*
Chercher suivant	Cherche l'occurrence suivante du terme qui a été saisi en dernier dans le dialogue *Chercher*
Supprimer	Supprime la liste de résultats
Réduire les résultats multi-ligne en une seule ligne	Si la colonne de valeur d'un item de résultat contient un texte multi-ligne (du texte qui contient un/des caractère/s de saut de ligne), vous pouvez basculer entre un affichage multi-ligne et ligne unique
Afficher le résultat complet	Affiche le contenu entier des nœuds en tant que la valeur du nœud
Afficher l'en-tête dans la sortie	Affiche l'emplacement du/des fichier/s XML et l'expression XPath

10.4 Générateur d'expression

Le générateur d'expression (ou Mode générateur) est activé/désactivé en cliquant sur l'icône **Mode générateur** de la barre d'outils principale. (*voir la capture d'écran ci-dessous*). Le générateur d'expression peut être activé dans les deux modes (Mode d'évaluation et Mode de débogage). Il a deux volets d'assistants à la saisie : (i) pour des opérateurs et expressions ; et (ii) pour des fonctions (*voir la capture d'écran ci-dessous*). Les items dans les deux volets peuvent être affichés soit par groupe de manière hierarchique, soit en tant que liste plate. Sélectionnez l'option souhaitée dans la liste déroulante en haut à droite de chaque volet (*voir la capture d'écran ci-dessous*). Dans la capture d'écran, les deux volets affichent leurs items dans des groupes hiérarchiques.



Fonctions du générateur d'expression :

- Pour visionner une description de texte d'un item soit dans le volet d'assistants à la saisie, passez sur l'item.
- Chaque fonction est regroupée avec sa signature (c'est-à-dire avec ses arguments, les types de données des arguments, et le type de données de la sortie de la fonction).
- Si plus d'une signature existe pour un seul nom de fonction, chaque signature est recensée en tant qu'une fonction séparée. (Ces variantes sont connues en tant que surcharges de ce nom de la fonction.) Dans la capture d'écran ci-dessus, par exemple, la fonction **`contains`** est affichée deux fois : une fois pour chacune de ses deux signatures.

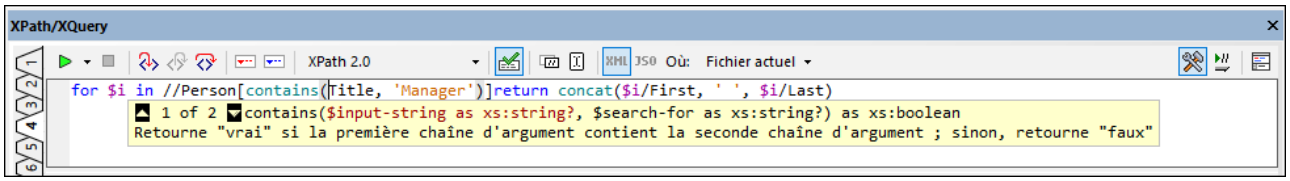
© 2016-2022 Altova GmbH

- Les arguments sont recensés par leurs noms (le cas échéant) ou par leurs types de données. Choisir l'option que vous souhaitez depuis la liste déroulante dans la barre de titre du volet Fonctions (*encerclés en vert dans la capture d'écran ci-dessus*).
- Double-cliquez sur un item dans un des trois volets (opérateur, expression ou fonction), insérez cet item à l'emplacement du curseur dans l'expression. Les fonctions sont insérées avec leurs arguments indiqués par le caractère d'espaces réservés (symboles #).
- Si le (i) texte est sélectionné dans le champ d'édition d'expression, et (ii) qu'un opérateur, une expression ou une fonction qui contient un espace réservé est double-cliquée pour l'insérer, alors le texte qui a été sélectionné est inséré à la place de l'espace réservé. Il s'agit d'un moyen rapide pour insérer un long texte (comme une expression de chemin) dans un opérateur, une expression ou une fonction.
- Vous pouvez insérer le chemin dans un nœud dans le document actif en sélectionnant ce nœud puis en cliquant sur la touche **Copier le XPath de la sélection actuelle** de la barre d'outils principale de la fenêtre.

Touches de configuration utilisées dans la génération d'expression

	**Layout Horizontal/Vertical**	Passe entre le layout horizontal et vertical
	**Passer au mode Génération**	Passe au mode Générateur d'expression qui vous propose des assistants à la saisie sensibles au contexte pour vous assister dans la construction d'expressions
	**Copier XPath de la sélection actuelle**	Copie le chemin de localisation du nœud dans le document XML dans la dernière position du curseur dans le volet Expression
	**Définir la sélection actuelle en tant que contexte**	Bascule le contexte d'expression entre le nœud racine et la sélection actuelle
	**Charger/Enregistrer le snippet XPath/XQuery**	Charge/enregistre respectivement un snippet XPath/XQuery depuis/vers un fichier XQuery

Une fois que vous avez saisi une fonction dans l'expression, en passant sur le nom de la fonction dans le volet Expression affiche la signature de la fonction et une description du texte de la fonction. Si plus d'une signature existe pour une fonction, celles-ci sont indiquées avec un facteur de surcharge en bas de l'écran. Si vous placez le curseur dans le cadre des parenthèses de la fonction et appuyez sur **Ctrl+Shift+Barre espace**, vous pouvez consulter les signatures des surcharges diverses de ce nom de fonction *(voir la capture d'écran ci-dessous)*.



Ouvrir et enregistrer des snippets XPath/XQuery depuis/vers un fichier

Vous pouvez enregistrer une expression XPath/XQuery, ou des snippets plus longs que vous avez saisi dans la fenêtre XPath/XQuery, ensemble avec les paramètres actuels de la fenêtre, vers un fichier XQuery, et vous pouvez charger des snippers XPath/XQuery depuis un fichier XQuery. Pour réaliser ces deux fonctions, cliquez sur leurs icônes respectives (Charger/Enregistrer Snippet), qui sont situés dans la barre d'outils de la fenêtre.



Une fois un snippet XPath/XQuery enregistré dans un fichier, il peut être chargé dans la fenêtre XPath/XQuery de toute instance XMLSpy (version 2022 et plus élevée). Ceci est utile si vous souhaitez utiliser le snippet sur un autre appareil, ou le passer à un autre utilisateur, ou même l'utiliser vous-même plus tard sur le même appareil. Lorsque l'expression est chargée dans la fenêtre XPath/XQuery depuis un fichier XQuery, les paramètres de la fenêtre basculeront automatiquement vers les paramètres de la fenêtre qui ont été enregistrés avec le snippet.

Enregistrer le snippet dans le fichier
Pour enregistrer l'expression XPath/XQuery ou le snippet dans un fichier, suivez les étapes suivantes :

1. Définir les paramètres suivants : (i) le langage d'évaluation ; (ii) le paramètre de champ *Où* ; et (iii) dans le cas du champ *Où* a été défini en tant que *Projet*, si les dossiers externes doivent être ignorés ou non.
2. Saisir l'expression XPath/XQuery ou le snippet que vous voulez enregistrer.
3. Cliquez sur **Enregistrer le Snippet**.
4. Dans le dialogue *Enregistrer en tant que* qui apparaît, sélectionnez le fichier, ou saisissez le nom d'un nouveau fichier dans lequel vous voulez enregistrer le snippet, et cliquez sur **Enregistrer**.

Le snippet sera enregistré dans le fichier, avec les paramètres de fenêtre étant enregistrés en tant que commentaire dans la première ligne du fichier. Ci-dessous, vous trouverez la liste d'un fichier snippet enregistré.

```
(: {"language":"XQuery3","where":"CurrentFile"} :)
pour $i dans //para[count(*)=0] return $i
```

Note : si vous le souhaitez, vous pouvez éditer ce fichier, y compris la ligne de commentaire qui contient les paramètres de fenêtre.

Charger le snippet depuis le fichier
Pour charger un snippet XPath/XQuery depuis un fichier, suivez les étapes suivantes :

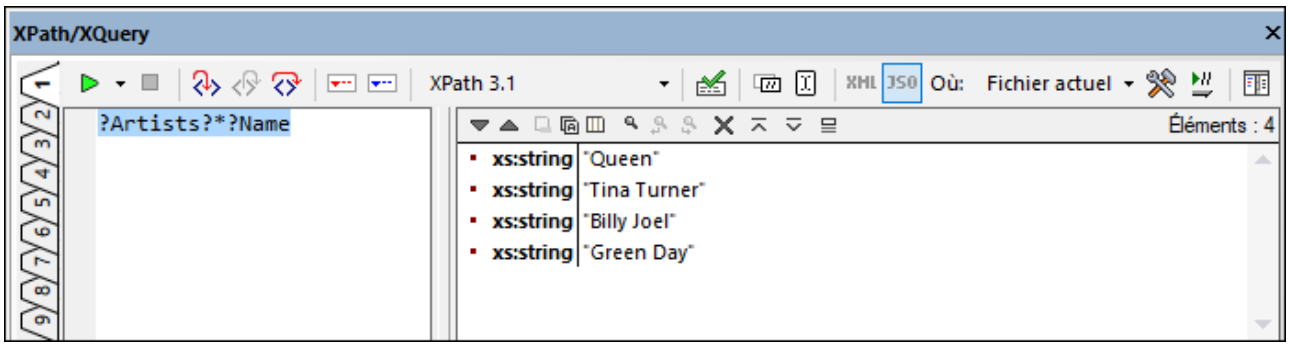
1. Dans l'onglet dans lequel vous souhaitez charger l'expression, cliquez sur **Charger le snippet**.
2. Dans le dialogue *Ouvrir* qui apparait, recherchez le fichier XQuery depuis lequel vous souhaitez charger le snippet et cliquez sur **Ouvrir**.

Le snippet sera chargé et les paramètres de la fenêtre XPath/XQuery changeront à ceux dans le fichier XQuery. Si aucun paramètre n'est stocké dans le fichier, alors les paramètres dans la fenêtre XPath/XQuery ne changeront pas. Si le paramètre Où (Where) n'est pas correctement défini, alors le paramètre de la fenêtre sera mis par défaut dans le *fichier actuel*.

© 2016-2022 Altova GmbH

10.5 Expressions XQuery pour JSON

Des documents JSON peuvent être requêtés en saisissant une expression de requête XPath/XQuery 3.1 dans la Fenêtre de sortie XPath/XQuery (*voir capture d'écran ci-dessous*).



Pour évaluer une expression dans un document JSON, procéder comme suit :

1. Choisir soit l'icône **XPath 3.1** ou **XQuery 3.1**.
2. S'assurer que la fenêtre se trouve dans le mode d'évaluation JSON.
3. Saisir l'expression XPath 3.1 ou XQuery 3.1 .
4. Cliquer sur **Lancer évaluation** (à gauche dans la barre d'outils).

Expressions XQuery 3.1 pour JSON

Puisque les structures de données JSON utilisent généralement des objets et des tableaux, c'est l'opérateur de consultation XQuery 3.1 `?` qui est utilisé pour situer des nœuds dans des objets JSON (qui sont principalement des mappages depuis une perspective XQuery) et des tableaux JSON. Cette méthode pour situer un nœud diffère de la manière dont les expressions de chemin sont écrites pour situer des nœuds dans des documents XML. Dans ceux-ci, l'opérateur "barre oblique" `/` est utilisé pour connecter des étapes dans une expression de chemin (par exemple : `items/*`). Dans des expressions XQuery pour JSON, l'opérateur de barre oblique n'est pas utilisé pour situer des nœuds.

Exemples d'expressions XQuery pour JSON

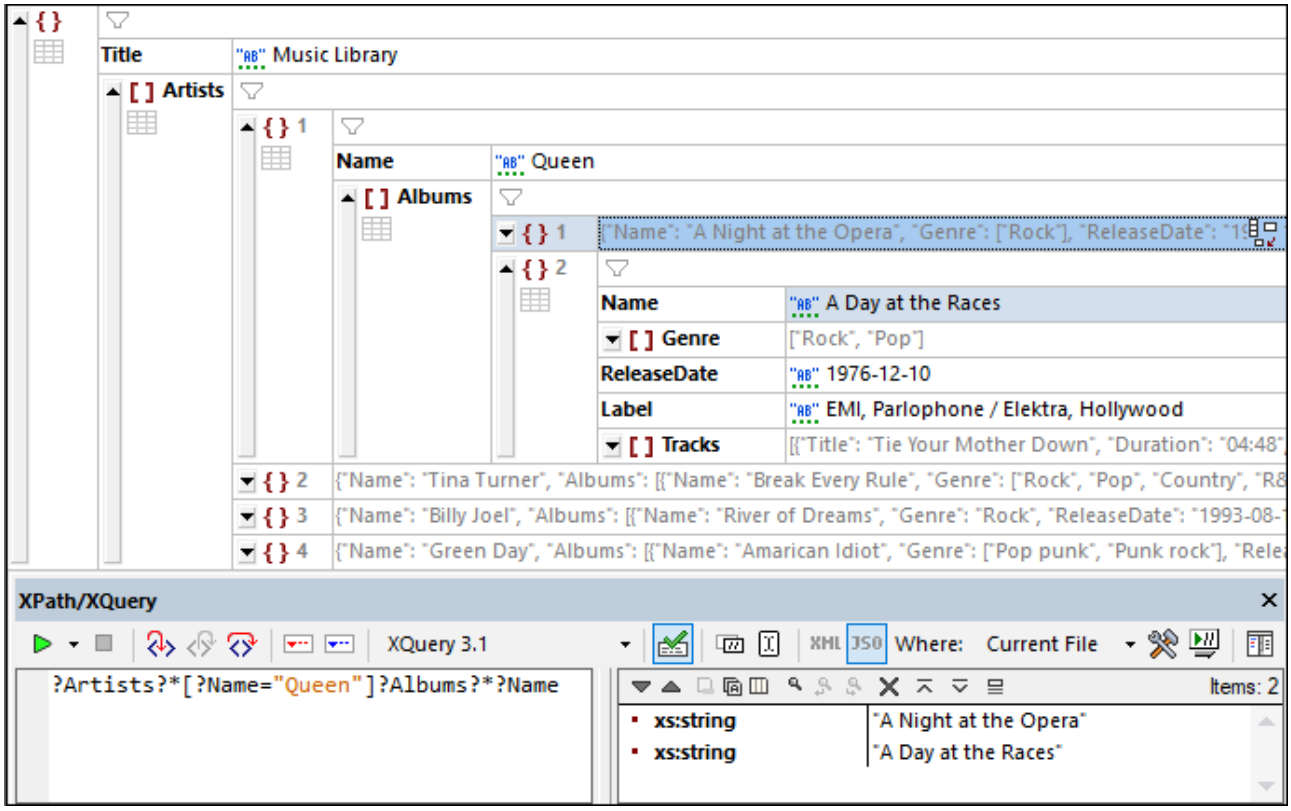
`?items?*`
Signification de l'expression : consultation du nœud enfant `items` puis consultation de tous ses nœuds enfants. Veuillez noter que `items` est censé être un nœud enfant du nœud contextuel.

`?Artists?1?Albums?2?Name`
Signification de l'expression : consultation du nœud enfant `Artists` puis consultation de son premier nœud enfant. Dans ce nœud, consulte le nœud enfant `Albums` puis consulte son second nœud enfant. Retourne maintenant le nœud `Name` de ce second nœud enfant.

`?Tracks?*[contains(?Writer, 'Brian')]`
Signification de l'expression : consultation du nœud enfant `Tracks` puis consultation de tous ses enfants. Lors de la consultation des enfants, consulte tous les enfants de nœud `Writer` de l'enfant, et ne sélecte que ceux qui contiennent le string `'Brian'`. Veuillez noter qu'il y a trois opérateurs de consultation dans cette expression. Chacun est utilisé dans une nouvelle étape, à l'endroit où un nodeset doit être consulté.

`?Artists?*[?Name="Queen"]?Albums?*?Name`

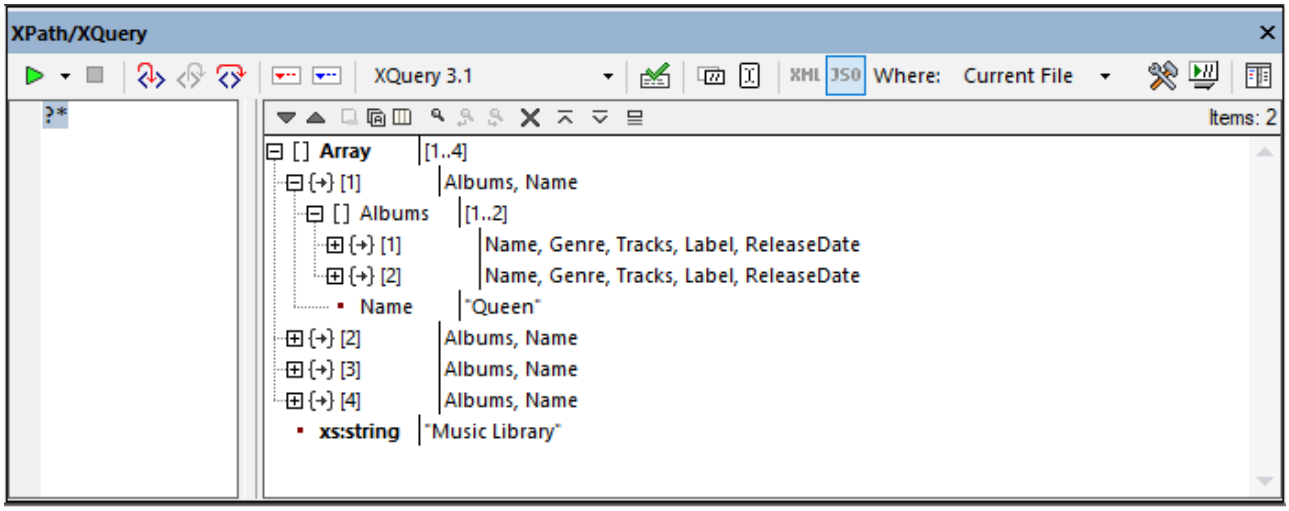
Signification de l'expression : Dans l'objet racine, consulte le nœud enfant **`Artists`** puis consulte tous ses enfant qui ont un nœud **`Name`** avec une valeur de **`"Queen"`**. Dans ces nœuds, consulte tous les nœuds de l'enfant **`Albums`**, puis leurs enfants. Dans ces enfants, consulte (et retourne) les nœuds respectifs **`Name`**. Dans la capture d'écran ci-dessous, cette expression est connue dans la fenêtre XPath/XQuery avec la représentation Mode JSON Grid du document cible JSON.



Volet Résultats pour l'évaluation JSON

Le volet Résultats montre les composants JSON sélectionnés en gras dans le côté gauche du volet *Résultats*, et la valeur du composant dans la côté droit du volet. Dans la capture d'écran ci-dessous, les résultats sont affichés en gras. L'array a été étendu pour montrer ses membres.

© 2016-2022 Altova GmbH



Les mappages et les arrays sont affichés en format abrégé ou un détail selon que l'icône *Afficher le résultat complet* dans la barre d'outils a été activée ou pas. Les mappages et les arrays du côté gauche du volet peuvent être étendus en cliquant sur leurs icônes plus respectives.

10.6 Points importants

Expressions XPath 1.0

- Les fonctions XPath 1.0 doivent être saisies sans préfixe d'espace de noms.
- Les quatre tests de nœud par type sont pris en charge : `node()`, `text()`, `comment()`, et `processing-instruction()`.

Expressions XPath 2.0 et 3.1

- Les strings (par ex. 'Bonjour') et les littéraux numériques (par ex. 256) sont pris en charge. Pour créer d'autres littéraux sur la base des types de Schéma XML, vous utilisez un constructeur à préfixe d'espace de noms (par ex. `xs:date('2004-09-02')`). Le préfixe d'espace de noms que vous utilisez pour les types de Schéma XML doivent être liés à l'espace de noms de Schéma XML : `http://www.w3.org/2001/XMLSchema`, et cet espace de noms doit être déclaré dans votre fichier XML.
- Les fonctions XPath 2.0 et 3.1 utilisées par l'Évaluateur XPath font partie de l'espace de noms `http://www.w3.org/2005/xpath-functions`. Par convention, le préfixe `fn:` est lié à cet espace de noms. Néanmoins, puisque cet espace de noms est l'espace de noms de fonction par défaut utilisé par l'Évaluateur XPath, il est inutile de spécifier un préfixe des fonctions. Si vous utilisez un préfixe, assurez-vous que le préfixe soit lié à l'espace de nom des Fonctions XPath, que vous devez déclarer dans le document. Exemples d'utilisation de la fonction : `current-date()` (dont l'espace de noms des fonctions n'est pas déclaré dans le document XML) ; `fn:current-date()` (dont l'espace de noms des fonctions n'est pas déclaré dans le document XML, ou bien est déclaré dans le document XML et lié au préfixe `fn:`). Vous pouvez omettre le préfixe d'espace de noms même si l'espace de noms des Fonctions a été déclaré dans le document XML avec ou sans préfixe ; cela est dû au fait qu'une fonction utilisée de cette manière dans une expression XPath se trouve dans l'espace de noms par défaut — qui est l'espace de noms par défaut pour les fonctions.
- Les extensions XPath d'Altova se trouvent dans l'espace de noms **`http://www.altova.com/xslt-extensions`**.

Note : Pour résumer le problème d'espace de noms : si vous utilisez des constructeurs ou des types provenant de l'espace de noms de Schéma XML, vous devez déclarer l'espace de noms de Schéma XML dans le document XML et utiliser les préfixes d'espace de noms corrects dans l'expression XPath. Vous n'avez pas besoin d'utiliser un préfixe pour les fonctions XPath.

Types de données dans XPath 2.0 et 3.1

Si vous évaluez une expression XPath 2.0 ou 3.1 pour un document XML qui référence un Schéma XML et est valide conformément à ce schéma, vous devez construire ou utiliser explicitement des types de données qui ne sont pas convertis implicitement au type de données requis par une opération. Dans les Modèles de données XPath 2.0 et 3.1 utilisés par le moteur intégré XPath, toutes les valeurs de nœud **atomisées** provenant du document XML sont attribués au type de données `xs:untypedAtomic`. Le type `xs:untypedAtomic` fonctionne bien avec les conversions de type implicites. Par exemple, l'expression `xs:untypedAtomic("1") + 1` résulte dans une valeur de `2` parce que la valeur `xs:untypedAtomic` est implicitement promue à `xs:double` par le biais de l'opérateur d'addition. Les opérateurs arithmétiques promeuvent implicitement des opérandes à `xs:double`. Les opérateurs de comparaison promeuvent des opérandes à `xs:string` avant de comparer.

© 2016-2022 Altova GmbH

Néanmoins, dans certains cas, il est nécessaire de convertir explicitement dans le type de données requis. Par exemple, si vous avez deux éléments, `startDate` et `endDate`, qui sont définis comme étant de type `xs:date` dans le Schéma XML, alors, par exemple, l'utilisation de l'expression XPath 2.0 `endDate - startDate` affichera une erreur. Mais si vous utilisez `xs:date(endDate) - xs:date(startDate)` ou `(endDate cast as xs:date) - (startDate cast as xs:date)`, l'expression évaluera correctement à une séquence simple de type `xs:dayTimeDuration`.

Note : Les moteurs XPath utilisés par l'évaluateur sont aussi utilisés par le moteur XSLT Altova, donc, les expressions XPath 2.0 ou 3.1 dans les feuilles de style XSLT qui ne sont pas converties implicitement dans le type de données requis doivent être construites explicitement dans le type de données requis.

Longueur de string de caractère et références d'entité

Lorsque les références de caractère et d'entité sont utilisées en tant que le string d'entrée pour la fonction `string-length()`, les références ne peuvent pas être résolues et la longueur du string de texte non résolu est retournée. Dans le cadre d'un environnement XSLT, néanmoins, ces références auraient une signification et la longueur du string résolu est retournée

Prise en charge des fonctions XPath 2.0 et 3.1

Voir les annexes.

11 Authentic

Authentic View (*capture d'écran ci-dessous*) est une représentation graphique de votre document XML. Il permet aux documents XML d'être affichés sans balises et avec un formatage et des fonctions de saisie des données appropriés comme des champs de saisie, des listes de choix, des fenêtres et des boutons radio. Les données saisies par l'utilisateur dans Authentic View sont saisies dans le fichier XML.



Afin de pouvoir consulter et éditer un document XML dans Authentic View, le document XML doit être associé avec une **StyleVision Power Stylesheet (SPS),** qui est créée dans le produit StyleVision d'Altova. Une SPS (un fichier `.sps`) est, à la base, une feuille de style XSLT. Elle spécifie une présentation de sortie pour un fichier XML qui peut inclure des mécanismes de saisie de données. Les utilisateurs de Authentic View peuvent, donc écrire des données en les renvoyant au fichier XML ou à la BD. Une SPS est basée sur un schéma et y est spécifique. Si vous souhaitez utiliser une SPS pour éditer un fichier XML dans Authentic View, vous devez en utiliser un qui se base sur le même schéma que celui sur lequel le fichier XML est basé.

Utiliser Authentic View

- Si un fichier XML est ouvert, vous pouvez passer à Authentic View en cliquant sur la touche **Authentic** située en bas de la fenêtre principale. Si une SPS n'est pas déjà attribuée au fichier XML, vous serez invité de l'y attribuer. Vous devez utiliser une SPS qui se base sur le même schéma que le fichier XML.
- Un nouveau fichier XML est créé et affiché dans Authentic View en choisissant la commande **Fichier | Nouveau** et puis en cliquant sur la touche "Sélectionner une feuille de style StyleVision". Ce nouveau fichier est un fichier modèle associé avec la SPS que vous ouvrez. Elle peut contenir avoir une quantité variable de données de démarrage. Ces données de démarrage sont contenues dans un fichier XML

© 2016-2022 Altova GmbH

(un fichier XML modèle) qui peuvent être associées avec la SPS en option. Une fois que l'Authentic View d'un fichier XML est affiché, vous pouvez y saisir les données et enregistrer le fichier.

- Vous pouvez aussi ouvrir une SPS par le biais de la commande **Authentic | Nouveau document**. Si un Fichier XML Modèle a été attribué à la SPS, alors, les données contenues dans le Fichier XML Modèle est utilisé en tant que données de démarrage du modèle de document créé dans Authentic View.

Dans cette section

Cette section contient un tutoriel Authentic View, qui vous montrera comment utiliser Authentic View. Il sera suivit de la section Éditer dans Authentic View, qui explique chaque fonction d'édition en détail.

Plus d'informations concernant Authentic View

Pour plus d'informations concernant Authentic View, voir (i) la section **Authentic | Interface du Mode Authentic**, qui décrit la fenêtre d'édition Authentic View, et (ii) la section du menu Authentic de la Référence de l'utilisateur de ce manuel.

11.1 Tutoriel Mode Authentic

Dans le Authentic View, vous pouvez éditer les documents XML dans une interface graphique WYSIWYG (*capture d'écran ci-dessous*), comme dans les applications de traitement de texte de style Microsoft Word. De fait, il vous suffit de saisir des données. Aucun besoin de vous soucier du formatage du document, le formatage est déjà défini dans la feuille de style qui contrôle le Authentic View du document XML. La feuille de style (StyleVision Power Stylesheet, abrégée en SPS dans ce tutoriel) est créée par un designer de feuille de style utilisant le produit StyleVision d'Altova.



L'édition d'un document XML dans le Authentic View exige deux actions de la part de l'utilisateur : (i) l'édition de la structure du document (par exemple, l'ajout ou la suppression de parties du document, comme des paragraphes et des en-têtes) ; et (ii) la saisie de données (le contenu de parties de documents).

Ce tutoriel vous entraîne dans les étapes suivantes :

- Ouvrir un document XML dans le Authentic View. L'exigence principale pour l'édition dans le Authentic View est que le document XML soit associé à un fichier SPS.
- Un aperçu de l'interface du Authentic View et une description générale des mécanismes d'édition centraux.
- Éditer la structure de document en insérant et en supprimant les nœuds.
- Saisir des données dans le document XML.
- Saisir (i) des valeurs d'attribut par le biais de l'Assistant à la saisie Attributs, et les (ii) valeurs d'entité.
- Imprimer le document.

© 2016-2022 Altova GmbH

Ce tutoriel a été conçu pour vous aider à vous lancer, il a donc été rédigé de manière intentionnellement simple. Vous trouverez des références supplémentaires et des descriptions de fonctions plus poussées dans la section interface du Mode Authentic.

Exigences du tutoriel

Tous **les fichiers** dont vous aurez besoin pour le tutoriel se trouvent dans le dossier `Examples` de votre dossier d'application Altova. Ces fichiers sont :

- `NanonullOrg.xml` (le document XML que vous allez ouvrir)
- `NanonullOrg.sps` (la StyleVision Power Stylesheet à laquelle le document XML est lié)
- `NanonullOrg.xsd` (le Schéma XML sur lequel le document XML et la StyleVision Power Stylesheet sont basés et auxquels ils sont liés)
- `nanonull.gif` and `Altova_right_300.gif` (deux fichiers d'image utilisés dans le tutoriel)

Note : au cours du tutoriel, nous vous demanderons de regarder le texte du document XML (contrairement au Authentic View du document). Si l'édition de produit Altova que vous utilisez ne comprend pas un Mode Texte (comme c'est le cas avec Authentic Desktop et Authentic Browser), utiliser un **éditeur de texte** courant comme Wordpad ou Notepad pour consulter le texte du document XML.

Attention : nous vous recommandons d'utiliser une copie de `NanonullOrg.xml` pour le tutoriel, ainsi vous pourrez toujours récupérer l'original, le cas échéant.

11.1.1 Ouvrir un document XML dans le Mode Authentic

Dans le Authentic View, vous pouvez éditer un document XML existant ou créer et éditer un nouveau document XML. Dans ce tutoriel, vous allez ouvrir un document XML existant dans le Authentic View (décrit dans cette section) et apprendre comment l'éditer (sections suivantes). En outre, vous trouverez dans cette section une description pour créer un nouveau document XML pour une édition dans le Authentic View.

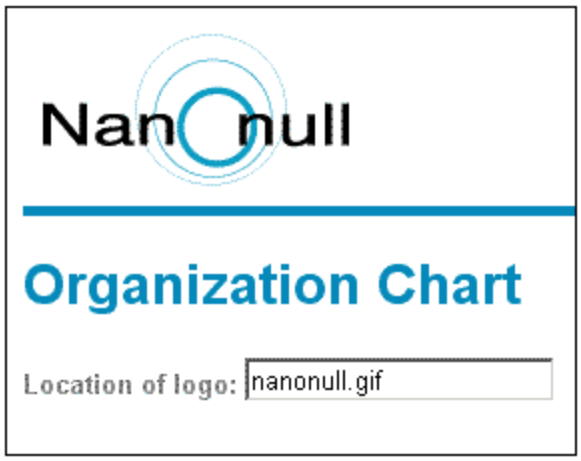
Ouvrir un document XML existant

Le fichier que vous allez ouvrir est nommé `NanonullOrg.xml`. Il se trouve dans le dossier `Examples` de votre application Altova. Vous pouvez ouvrir `NanonullOrg.xml` de deux manières :

- Cliquer sur **Fichier | Ouvrir** dans votre produit Altova, puis chercher `NanonullOrg.xml` dans le dialogue qui apparaît et cliquer sur **Ouvrir**.
- Utiliser Windows Explorer pour localiser le fichier, cliquer avec la touche de droite et choisir votre produit Altova en guise d'application avec laquelle ouvrir le fichier.

Le fichier `NanonullOrg.xml` s'ouvre directement dans Authentic View (*capture d'écran ci-dessous).* Cela s'explique par le fait que

1. Une StyleVision Power Stylesheet (SPS) a déjà été attribuée au fichier et que
2. Dans le dialogue Options (Outils | Options), sous l'onglet Affichage, l'option pour ouvrir les fichiers XML dans le Authentic View, dans le cas où un fichier SPS a été attribué, a été cochée. (Sinon, le fichier s'ouvrira dans le Mode Texte.)



Attention : c'est la SPS qui définit et contrôle comment un document XML est affiché dans le Authentic View. Sans une SPS, il ne peut pas y avoir de Authentic View du document.

Créer un nouveau document XML sur la base d'une SPS

Vous pouvez aussi créer un nouveau document XML qui se base sur une SPS. Vous avez deux possibilités : par le biais de la commande de menu **Fichier | Nouveau** et par le biais de la commande de menu **Authentic | Nouveau document**. Dans les deux cas, une SPS est sélectionnée.

Via Fichier | Nouveau

1. Sélectionner **Fichier | Nouveau**.
2. Dans le dialogue Créer un nouveau document, chercher la SPS désirée.

Si un fichier de modèle XML a été attribué à la SPS, les données dans le Fichier de modèle XML sont utilisées en tant que données de départ du modèle de document XML créé dans Authentic View.

Via Authentic | Nouveau document

1. Sélectionner **Authentic | Nouveau document.**
2. Dans le dialogue Créer un nouveau document, chercher la SPS désirée.

Si un Fichier de modèle XML a été attribué à la SPS, les données dans le Fichier de modèle XML sont utilisées en tant que données de départ du modèle de document XML créé dans Authentic View.

11.1.2 L'interface du Mode Authentic

L'interface d'édition du Authentic View consiste en une fenêtre principale dans laquelle vous saisissez et éditez les données de document, et trois assistants à la saisie. L'édition d'un document est très simple. Si vous souhaitez voir les balises du document, activez les balises. Ensuite, vous pouvez commencer à saisir du contenu dans votre document. Pour modifier la structure du document, vous pouvez utiliser soit le menu contextuel soit l'Assistant à la saisie Éléments.

© 2016-2022 Altova GmbH

Afficher les balises de nœud XML (balises de document)

Un document XML est essentiellement une hiérarchie de nœuds. Par exemple :

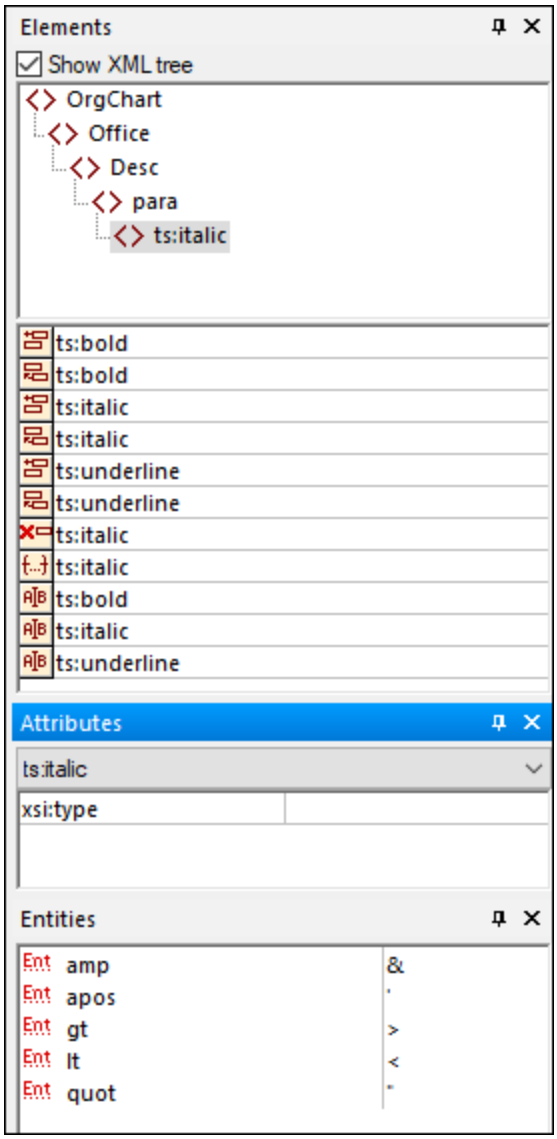
```
<DocumentRoot>
   <Person id="ABC001">
       <Name>Alpha Beta</Name>
       <Address>Some Address</Address>
       <Tel>1234567</Tel>
   </Person>
</DocumentRoot>
```

Par défaut, les balises de nœud ne sont pas affichées dans le Authentic View. Vous pouvez activer les balises de nœud en sélectionnant l'item de menu **Authentic | Afficher grande balise** (ou l'icône de barre d'outils ). Les grandes balises contiennent les noms des nœuds respectifs. En alternative, vous pouvez sélectionner des petites balises (sans nom de code dans la balise) et des balises mixtes (un mélange de grandes balises, de petites balises et de balises sans balises. Cela est défini par le designer de la feuille de style ; la balise mixte par défaut pour le document est "sans balise").

Vous pouvez consulter le texte du document XML dans le Mode Texte de votre produit Altova ou dans un éditeur de texte.

Assistant à la saisie

Il existe trois assistants à la saisie dans l'interface (*capture d'écran ci-dessous*), situés par défaut le long de la bordure droite de la fenêtre d'application. Il s'agit des assistants à la saisie Éléments, Attributs et Entité.



Assistant à la saisie Éléments

L'assistant à la saisie Éléments affiche les éléments qui peuvent être insérés et supprimés avec référence à l'emplacement actuel du curseur ou à la sélection dans la Fenêtre principale. Veuillez noter que l'assistant à la saisie est sensible au contexte ; son contenu change selon l'emplacement du curseur ou de la sélection. Le contenu de l'assistant à la saisie peut être modifié d'une autre manière : lorsqu'un autre nœud est sélectionné dans l'arborescence XML de l'assistant à la saisie Éléments, les éléments pertinents pour ce nœud sont affichés dans l'assistant à la saisie. L'assistant à la saisie Éléments peut être agrandi pour montrer l'arborescence XML en cochant la case Afficher Arborescence XML située en haut de l'assistant à la saisie (*voir capture d'écran ci-dessus*). L'arborescence XML montre la hiérarchie des nœuds depuis le nœud d'élément de niveau supérieur jusque dans le nœud sélectionné dans la Fenêtre principale.

Assistant à la saisie Attributs

L'assistant à la saisie Attributs affiche les attributs de l'élément sélectionné dans la Fenêtre principale et les valeurs de ces attributs. Les valeurs de l'attribut peuvent être saisies dans l'assistant à la saisie Attributs. Les nœuds de l'élément depuis l'élément de niveau supérieur jusqu'à l'élément sélectionné sont disponibles pour la sélection dans la liste de choix de l'assistant à la saisie Attributs. La sélection d'un élément depuis la liste

© 2016-2022 Altova GmbH

déroulante de la liste de choix entraîne l'affichage des attributs de cet élément dans l'assistant à la saisie, où ils peuvent ensuite être édités.

Assistant à la saisie Entités
L'assistant à la saisie Entités n'est pas sensible au contexte et affiche toutes les entités déclarées pour le document. Double-cliquer sur une entité pour l'insérer au niveau de l'emplacement du curseur. La description de l'ajout d'entités à un document est traitée dans la section interface du Mode Authentic.

Menu contextuel

Cliquer avec la touche de droite sur un emplacement dans le document du Authentic View pour ouvrir un menu contextuel pertinent à cet emplacement (du nœud). Le menu contextuel propose des commandes qui vous permettent de :

- Insérer des nœuds à cet emplacement, ou avant ou après le nœud sélectionné. Les sous-menus affichent les listes des nœuds qui sont autorisés au niveau des emplacements d'insertion respectifs.
- Supprimer le nœud sélectionné (si cela est autorisé par le schéma) ou tout élément d'ancêtre amovible. Les nœuds qui peuvent être supprimés (conformément au schéma) sont recensés dans un sous-menu.
- Insérer des entités et des sections CDATA. Les entités déclarées pour le document sont recensées dans un sous-menu. Les sections CDATA peuvent uniquement être insérées dans du texte.
- Couper, copier, coller (y compris coller en tant que XML ou texte), et supprimer du contenu de document.

Note: pour plus de détails concernant l'interface, voir interface du Mode Authentic

11.1.3 Opérations de nœud

Il existe deux types principaux de nœuds que vous aurez l'occasion de rencontrer dans un document XML Authentic View : **les nœuds d'élément** et **les nœuds d'attribut**. Ces nœuds sont marqués par des balises que vous pouvez activer. Il existe d'autres nœuds dans le document, comme les nœuds de texte (qui ne sont pas balisés) et des nœuds de section CDATA (qui sont balisés, afin de les délimiter du texte environnant).

Les opérations de nœud décrites dans cette section se réfèrent uniquement aux nœuds d'élément et aux nœuds d'attribut. Lorsque vous tentez les opérations décrites dans cette section, il est conseillé d'activer les grandes balises.

Note : il est important de garder en tête que seuls des **éléments de même niveau ou de niveau plus élevé** peuvent être insérés avant ou après l'élément sélectionné. Les éléments de même niveau sont des **frères**. Les frères d'un élément paragraphe seront d'autres éléments paragraphes, mais ils pourraient aussi être des listes, une table, une image, etc. Des frères peuvent se produire avant ou après un élément. Les éléments d'un niveau plus élevé sont des éléments **ancêtres** et les frères d'ancêtres. Pour un élément paragraphe, des éléments ancêtres peuvent être une section, un chapitre, un article, etc. Un paragraphe dans un fichier XML valide aurait déjà des ancêtres. C'est pourquoi l'ajout d'un élément de niveau supérieur dans le Authentic View crée le nouvel élément en tant que frère de l'ancêtre pertinent. Par exemple, si un élément section est inséré après un paragraphe, il est créé en tant que frère de la section qui contient l'élément de paragraphe actuel.

Effectuer des opérations de nœud

Les opérations de nœud peuvent être effectuées en sélectionnant une commande dans le menu contextuel ou en cliquant sur l'entrée d'opération de nœud dans l'Assistant à la saisie Éléments. Dans certains cas, un élément ou un attribut peut être ajouté en cliquant sur le lien Ajouter le nœud dans le Authentic View du document. Dans les cas particuliers des éléments définis en tant que paragraphes ou d'item de liste, si vous appuyez sur la touche Entrée lorsque vous vous trouvez dans un tel élément, vous créerez un nouvel élément frère de ce type. Cette section décrit aussi comment les nœuds peuvent être créés et supprimés en utilisant les mécanismes Appliquer l'élément, Supprimer le nœud, et Effacer l'élément.

Insérer les éléments

Des éléments peuvent être insérés dans les emplacements suivants :

- L'emplacement de curseur dans un nœud d'élément. Les éléments disponibles pour l'insertion à cet endroit sont recensés dans un sous-menu de la commande **Insérer** du menu contextuel. Dans l'assistant à la saisie Éléments, les éléments qui peuvent être insérés dans un emplacement sont indiqués par l'icône . Dans le document `NanonullOrg.xml`, placer le curseur dans l'élément `para`, et créer des éléments `bold` et `italic` à l'aide du menu contextuel et de l'assistant à la saisie Éléments.
- Avant ou après l'élément sélectionné ou un de ses ancêtres, si autorisé par le schéma. Sélectionner l'élément requis depuis le/s sous-menu/s qui apparait/ssent. Dans l'assistant à la saisie Éléments, les éléments qui peuvent être insérés avant ou après l'élément sélectionné sont indiqués avec les icônes et , respectivement. Veuillez noter que dans l'assistant à la saisie Éléments, vous pouvez uniquement insérer des éléments avant/après l'élément sélectionné ; vous ne pouvez pas insérer avant/après un élément ancêtre. Pour essayer cette commande, placez tout d'abord le curseur dans l'élément `para` puis dans la table recensant les employés.

Ajouter le lien de nœud

Si un élément ou un attribut est inclus dans le design de document, et qu'il n'est pas présent dans le document XML, un lien `Ajouter lien` s'affichera dans l'emplacement du document à l'endroit où ce nœud est spécifié. Pour voir ce lien, sélectionner, dans la ligne contenant le texte *Emplacement du logo*, le nœud `@href` dans l'élément `CompanyLogo` et le supprimer (en appuyant sur la touche **Supprimer**). Le lien `add @href` apparaît dans l'élément `CompanyLogo` qui a été édité (*capture d'écran ci-dessous*). Cliquer sur le lien pour ajouter le nœud `@href` au document XML. La fenêtre de saisie dans les balises `@href` apparaît parce que le design spécifie que le nœud `@href` sera ajouté comme cela. Vous devrez tout de même saisir la valeur (ou le contenu) du nœud `@href`. Saisir le texte `nanonull.gif`.



 © 2016-2022 Altova GmbH

Si le modèle de contenu d'un élément est ambigu, par exemple s'il spécifie qu'une séquence des éléments enfant peut apparaître dans n'importe quel ordre, alors le lien `add...` apparaît. Veuillez noter qu'aucun nom de nœud n'est spécifié. Cliquer sur le lien pour faire apparaître une liste des éléments qui peuvent être insérés validement.

Note : le lien Ajouter nœud apparaît directement dans le modèle du document ; il n'y a pas d'entrée correspondante dans le menu contextuel ou dans l'assistant à la saisie Éléments.

Créer de nouveaux éléments avec la touche Entrée

Dans les cas où un élément a été formaté en tant que paragraphe ou item de liste (par le designer de feuille de style), appuyer sur la touche Entrée lorsque vous vous trouvez dans un tel nœud pour insérer un nouveau nœud de ce type après le nœud actuel. Vous pouvez essayer ce mécanisme dans le document `NanonullOrg.xml` en vous rendant à la fin d'un nœud `para` (juste avant sa balise de fin) et en appuyant sur **Entrée**.

Appliquer des éléments

Dans les éléments de contenu mixte (ceux qui contiennent du texte et des éléments enfant), certains contenus de texte peuvent être sélectionnés et un élément enfant autorisé peut y être appliqué. Le texte sélectionné devient le contenu de l'élément appliqué. Pour appliquer des éléments, dans le menu contextuel, sélectionner **Appliquer** puis sélectionner depuis les éléments applicables. (Si aucun élément ne peut être appliqué au texte sélectionné, la commande **Appliquer** n'apparaît pas dans le menu contextuel). Dans l'assistant à la saisie Éléments, les éléments qui peuvent être appliqués pour une sélection sont indiqués par l'icône . Dans le document `NanonullOrg.xml`, sélectionner le texte se trouvant dans l'élément `para` de contenu mixte et faire des essais en appliquant les éléments `bold` et `italic`.

Le designer de feuille de style peut aussi avoir créé une icône de barre d'outils à appliquer à un élément. Dans le document `NanonullOrg.xml`, les éléments `bold` et `italic` peuvent être appliqués en cliquant sur les icônes grasses et italiques dans la barre d'outils Authentic de l'application.

Supprimer les nœuds

Un nœud peut être supprimé si sa suppression ne rend pas le document invalide. La suppression d'un nœud entraîne également la suppression de tout son contenu. Un nœud peut être supprimé à l'aide de la commande **Supprimer** dans le menu contextuel. Lorsque la commande Supprimer est marquée, un sous-menu contenant tous les nœuds pouvant être supprimés s'ouvre, en commençant par le nœud sélectionné et jusqu'au nœud de niveau supérieur du document. Pour sélectionner un nœud à supprimer, placer le curseur dans le nœud, ou bien marquer le nœud (ou une de ses parties). Dans assistant à la saisie Éléments, les nœuds qui peuvent être supprimés sont indiqués par l'icône . Il est également possible de supprimer un nœud en le sélectionnant et en appuyant sur la touche **Supprimer**. Dans le document `NanonullOrg.xml`, faites des essais en supprimant quelques nœuds à l'aide des mécanismes décrits. Vous pouvez annuler vos changements avec **Ctrl+Z**.

Effacer les éléments

Les nœuds d'élément qui sont les enfants d'éléments à contenu mixte (aussi bien enfants de texte et d'élément) peuvent être effacés. L'élément entier peut être effacé lorsque le nœud est sélectionné ou lorsque le curseur est placé dans le nœud en tant qu'un point d'insertion. Un fragment de texte dans l'élément peut être effacé de la balise de l'élément en marquant le fragment de texte. Une fois la sélection effectuée, sélectionner **Effacer** dans le menu contextuel et puis l'élément à supprimer. Dans l' assistant à la saisie Éléments, les éléments qui peuvent être effacés pour une sélection particulière sont indiqués avec l'icône  (sélection de

point d'insertion) et l'icône (sélection de la plage). Dans le document `NanonullOrg.xml`, essayer le mécanisme d'effacement avec les élément enfants `bold` et `italic` de `para` (qui présente le contenu mixte).

Tables et structure de table

Il y a deux types de table Authentic View :

- *Tables SPS (statique et dynamique)*. La structure générale de la table SPS est déterminée par le designer de feuille de style. Dans le cadre de cette structure large, les seuls changements structurels que vous êtes autorisé à effectuer sont les changements orientés vers le contenu. Par exemple, vous pouvez ajouter de nouvelles lignes à une table SPS dynamique.
- *Tables XML*, dans lesquelles vous décidez de présenter les contenus d'un nœud particulier (par ex. un nœud pour les détails spécifiques aux personnes) en tant que table. Si le designer de feuille de style a activé la création de ce nœud en tant qu'une table XML, vous pouvez déterminer la structure de la table et éditer ses contenus. Les tables XML sont discutées en détail dans la section Tables dans le Mode Authentic.

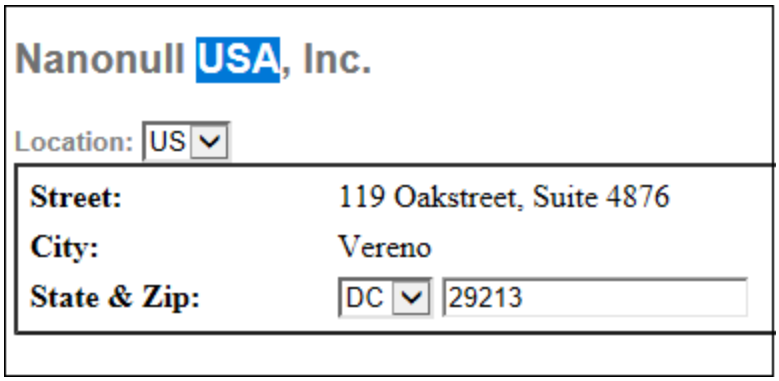
11.1.4 Saisir des données dans le Mode Authentic

Les données sont saisies dans le document XML, directement dans la fenêtre principale du Authentic View. De plus, des données (la valeur de l'attribut) peuvent être saisies dans l'assistant à la saisie Attributs pour les attributs. Les données sont saisies (i) directement en tant que texte, ou (ii) en sélectionnant une option dans un appareil de saisie des données qui est ensuite mappé à une entrée de texte prédéfinie.

Ajouter le contenu de texte

Vous pouvez entrer le contenu d'élément et les valeurs d'attribut directement en tant que texte dans la fenêtre principale du Authentic View. Pour insérer du contenu, placer le curseur à l'emplacement où vous souhaitez insérer le texte, puis le saisir. Vous pouvez aussi copier du texte depuis le presse-papiers dans le document. Le contenu peut aussi être édité à l'aide des mécanismes d'édition standard, comme les touches **Verr.** et **Supprimer**. Par exemple, vous pouvez marquer le texte à éditer et saisir un texte de remplacement avec la touche **Verr.** activée.

Par exemple, pour changer le nom de l'entreprise, dans le champ `Name` de `Office`, placer le curseur après Nanonull, et saisir `USA` pour changer le nom de Nanonull, Inc. en Nanonull USA, Inc.



 © 2016-2022 Altova GmbH

Si le texte est éditable, vous pourrez placer votre curseur dessus et le marquer. Essayez de modifier l'un des **noms de champ** (pas les valeurs de champ), comme "Street", "City", ou "State/Zip," dans le bloc d'adresse. Vous ne pourrez pas placer le curseur dans ce texte car celui-là ne fait pas partie du contenu XML ; il est dérivé de la StyleVision Power Stylesheet.

Insérer des caractères spéciaux et des entités

Lorsque vous saisissez des données, le type suivant de contenu est géré de manière spéciale :

- *Les caractères spéciaux qui sont utilisés pour des balises XML* (esperluette (&), apostrophe ('), supérieur à (>), inférieur à (<) et guillemets (""). Ces caractères sont disponibles en tant qu'entités intégrées et peuvent être saisies dans le document en double-cliquant l'entité respective dans l'assistant à la saisie Entités. Si ces caractères se produisent fréquemment (par exemple dans les listes de code de programme), ils peuvent être saisis dans les sections CDATA. Pour insérer une section CDATA, cliquer avec la touche de droite dans l'emplacement où vous souhaitez saisir la section CDATA, et sélectionner **Insérer Section CDATA** depuis le menu contextuel. Le processeur XML ignore tous les caractères de balise dans les sections CDATA. Cela signifie également que si vous souhaitez insérer un caractère spécial dans une section CDATA, vous devrez saisir ce caractère et non pas sa référence d'entité.
- *Les caractères spéciaux qui ne peuvent pas être saisis par le biais du clavier* doivent être saisis en les copiant depuis le mappage de caractères de votre système vers l'emplacement souhaité dans le document.
- *Une chaîne de texte fréquemment utilisée* peut être définie en tant qu'entité, qui apparaît dans l'Assistant à la saisie Entités. L'entité est insérée dans l'emplacement requis en plaçant le curseur dans chaque emplacement requis et en double-cliquant l'entité dans l'assistant à la saisie. Cela est utile pour l'entretien parce que la valeur de la chaîne de texte est maintenue dans un emplacement ; si la valeur doit être changée, il suffit de changer la définition de l'entité.

Note : Lorsque la balise est dissimulée dans le Authentic View, un élément vide peut facilement être ignoré. Pour vous assurer de ne pas oublier un élément vide, activer les grandes ou les petites balises.

Essayez d'utiliser chaque type de contenu de texte décrit ci-dessus.

Ajouter un contenu par le biais d'un appareil de saisie de données

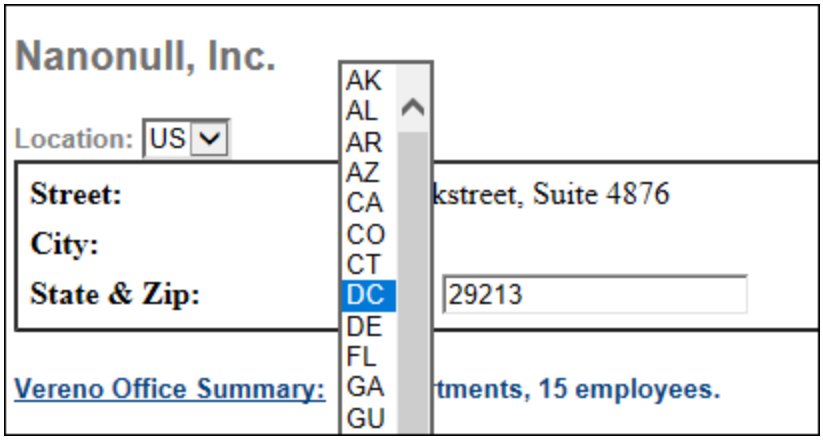
Dans l'édition de contenu que vous avez appris ci-dessus, le contenu est ajouté en saisissant le texte directement en tant que contenu. Vous disposez également d'un autre moyen pour saisir un **contenu d'élément** (ou de valeurs d'attribut) dans Authentic View : par le bais d'appareils de saisie de données.

Ci-dessous, vous trouverez une liste des appareils de saisie de données dans le Authentic View, complétée d'une explication pour saisir les données dans le fichier XML pour chaque appareil.

Appareil de saisie de données	Données dans le fichier XML
Champ de saisie (fenêtre Texte)	Texte saisi par l'utilisateur
Champ de saisie multiligne	Texte saisi par l'utilisateur
Liste de choix	Sélection de l'utilisateur mappé à la valeur
Case à cocher	Sélection de l'utilisateur mappé à la valeur

Bouton radio	Sélection de l'utilisateur mappé à la valeur
Bouton	Sélection de l'utilisateur mappé à la valeur

Dans la table statique contenant les champs d'adresse (*affiché ci-dessous*), il y a deux appareils de saisie de données : un champ d'entrée pour le champ `Zip` et une liste de choix pour le champ State. Les valeurs que vous saisissez dans les champs de texte sont saisies directement en tant que le contenu XML des éléments respectifs. Pour tous les autres appareils de saisie des données, votre sélection est mappée à une valeur.



Voici le texte XML correspondant au Authentic View affiché ci-dessus :

```
<Address>
  <ipo:street>119 Oakstreet, Suite 4876</ipo:street>
  <ipo:city>Vereno</ipo:city>
  <ipo:state>DC</ipo:state>
  <ipo:zip>29213</ipo:zip>
</Address>
```

Veuillez noter que la sélection de liste de choix `DC` est mappée à une valeur de `DC`. La valeur du champ `Zip` est saisie directement en tant que le contenu de l'élément `ipo:zip`.

11.1.5 Saisir des valeurs d'attribut

Un attribut est la propriété d'un élément et un élément peut présenter un nombre quelconque d'attributs. Les attributs ont des valeurs. Vous serez parfois amené à saisir des données XML en tant que valeur d'attribut. Dans le Authentic View, vous pouvez saisir des valeurs d'attribut de deux manières :

- En tant que contenu dans la fenêtre principale si l'attribut a été créé pour accepter sa valeur de cette manière
- Dans l'assistant à la saisie Attributs

Valeurs d'attribut dans la fenêtre principale

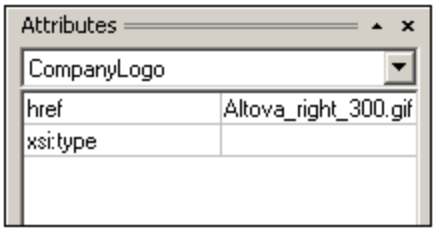
Les valeurs d'attribut peuvent être saisies en tant que texte normal ou en tant que sélection par l'utilisateur qui sera mappée à une valeur XML. Elles seront saisies de la même manière que le contenu d'élément est saisi : voir Saisir des données dans le Authentic View. Dans ces cas, la distinction entre le contenu d'élément et la

valeur d'attribut est réalisée par la StyleVision Power Stylesheet et les données sont gérées de manière appropriées.

Valeurs d'attribut dans l'assistant à la saisie Attributs

Si vous souhaitez saisir ou modifier une valeur d'attribut, vous pouvez aussi le faire dans l'Assistant à la saisir Attributs. Tout d'abord, le nœud d'attribut est sélectionné dans le Authentic View, ensuite la valeur de l'attribut est saisie ou éditée dans l'assistant à la saisie Attributs. Dans le document `NanonullOrg.xml`, l'emplacement du logo est stocké en tant que la valeur de l'attribut `href` de l'élément `CompanyLogo`. Pour changer le logo à utiliser :

1. Sélectionner l'élément `CompanyLogo` en cliquant sur une balise CompanyLogo. Les attributs de l'élément `CompanyLogo` sont affichés dans l'assistant à la saisie Attributs.
2. Dans l'assistant à la saisie Attributs, changer la valeur de l'attribut `href` de `nanonull.gif` à `Altova_right_300.gif` (une image dans le dossier `Examples`).

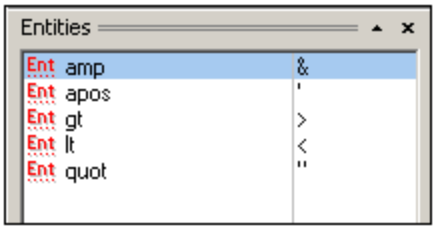


Cela entraîne le remplacement du logo Nanonull par le logo Altova.

Note : les entités ne peuvent pas être saisies dans l'assistant à la saisie Attributs.

11.1.6 Ajouter des entités

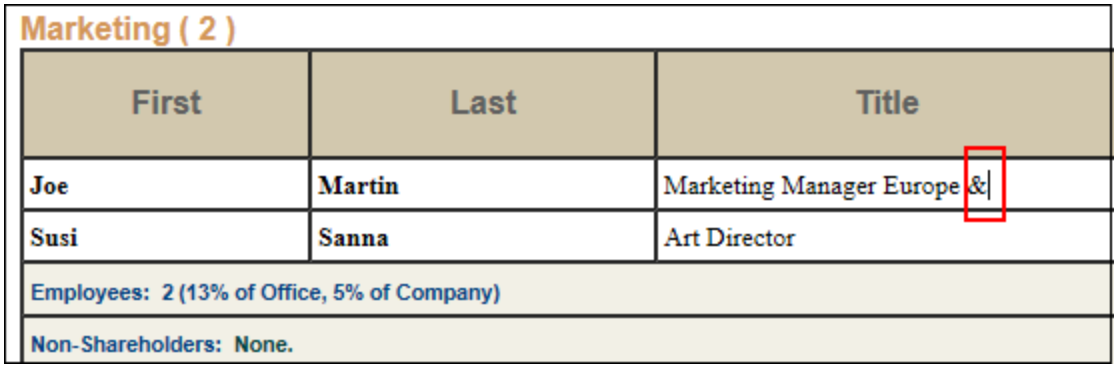
Une entité contenue dans le Authentic View est généralement une donnée XML (mais pas nécessairement), comme un caractère unique ; une chaîne de texte ; et même un fragment de document XML. Une entité peut aussi être un fichier binaire, comme un fichier d'image. Toutes les entités disponibles pour un document particulier sont affichées dans l'assistant à la saisie Entités (*capture d'écran ci-dessous*). Pour insérer une entité, placer le curseur à l'endroit du document où vous souhaitez l'insérer puis double-cliquer sur l'entité dans l'assistant à la saisie Entités. Veuillez noter que vous ne pouvez pas saisir des entités dans l'assistant à la saisie Entités.



Le caractère esperluette (&) a une signification particulière dans XML (tout comme les apostrophes, les symboles inférieur à et supérieur à et les guillemets doubles). Pour insérer ces caractères, les entités sont utilisées de manière à ce qu'elles ne sont pas confondues avec des caractères réservés à XML. Ces caractères sont disponibles en tant qu'entités dans Authentic View.

Dans `NanonullOrg.xml`, changer le titre de Joe Martin (dans Marketing) en Marketing Manager Europe & Asia. Pour ce faire :

1. Placer le curseur à l'endroit où vous souhaitez insérer l'esperluette.
2. Double-cliquer sur l'entité listée en tant que "amp". Cela permet d'insérer une esperluette (*capture d'écran ci-dessous*).



Note : l'assistant à la saisie Entités n'est pas sensible au contexte. Toutes les entités disponibles sont affichées quel que soit l'endroit où le curseur est placé. Cela ne signifie pas qu'une entité peut être insérée à tout endroit du document. Si vous n'êtes pas sûr, valider le document après avoir inséré l'entité : **XML | Valider XML** (**F8**).

Définir vos propres entités

En tant qu'éditeur de document, vous pouvez définir vos propres entités de document. Pour ce faire, voir la description dans la section Définir des entités dans le Authentic View.

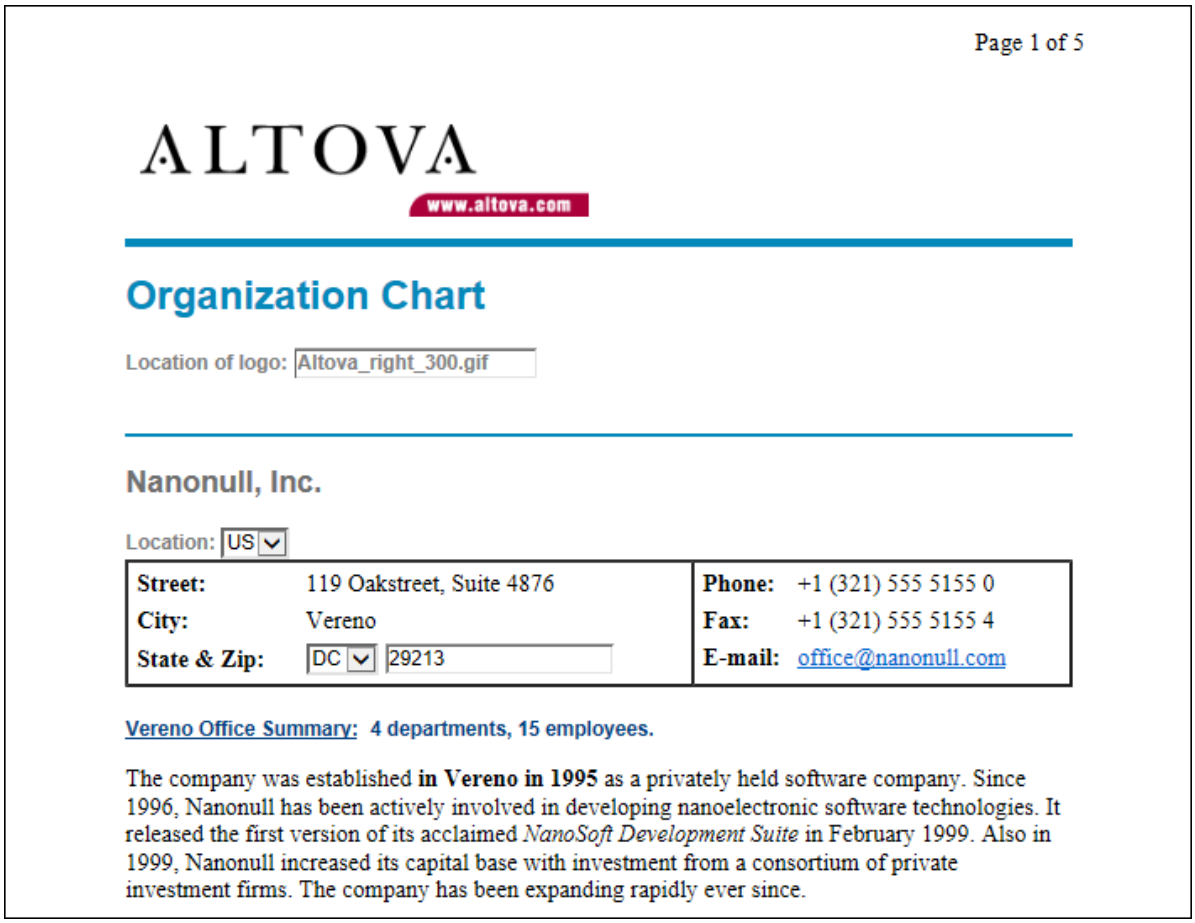
11.1.7 Imprimer le document

Une impression depuis le Authentic View d'un document XML préserve le formatage vu dans le Authentic View.

Pour imprimer `NanonullOrg.xml`, procéder comme suit :

1. Passer au mode Dissimuler balise si vous n'y êtes pas déjà. Vous devrez procéder à cette étape si vous ne voulez pas imprimer les balises.
2. Sélectionner **Fichier | Imprimer aperçu** pour voir un aperçu de toutes les pages. Ci-dessous, vous verrez une partie de l'impression d'une page d'aperçu, réduite de 50%. Veuillez noter que le formatage de la page est le même que celui dans Authentic View.

© 2016-2022 Altova GmbH



3. Pour imprimer le fichier, cliquer sur **Fichier | Imprimer**.

Veuillez noter que vous pouvez aussi imprimer une version du document qui affiche la balise. Pour ce faire, passer au Authentic View pour afficher les petites et les grandes balises, puis imprimer.

11.2 Interface Mode Authentic

Le Authentic View est activé en cliquant sur l'onglet Authentic du document actif. Si aucune SPS n'a été attribuée au document XML, vous serez invité à en attribuer un. Vous pouvez attribuer une SPS à tout moment par le biais de la commande **Authentic | Attribuer une feuille de style Stylevision**.

Cette section fournit :

- Un aperçu de l'interface
- Une description des icônes de la barre d'outils spécifiques à Authentic View
- Une description des modes de consultation disponibles dans la fenêtre principale du Authentic View
- Une description des assistants à la saisie et de la manière de les utiliser
- Une description des menus contextuels disponibles dans des points variés dans le Authentic View du document XML

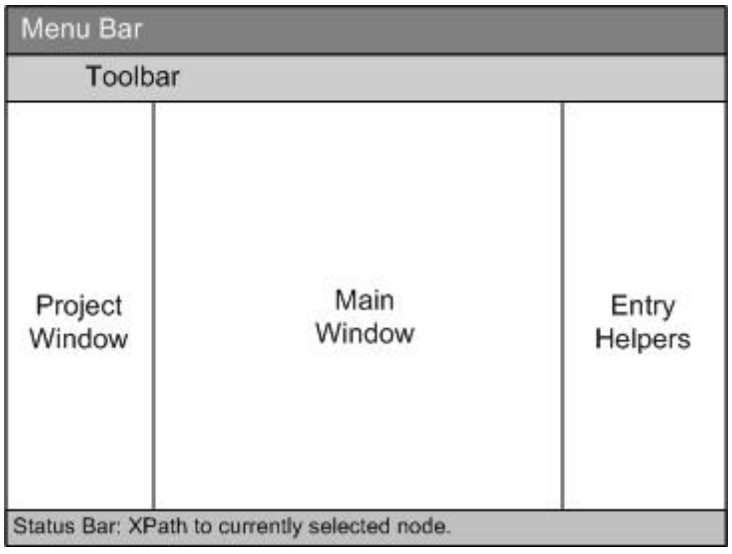
Sources supplémentaires des informations du Authentic View :

- Un tutoriel Authentic View qui vous montre comment utiliser l'interface Authentic View. Ce tutoriel est disponible dans la documentation des produits Altova XMLSpy et Altova Authentic Desktop (voir la section Tutoriel), et en ligne.
- Pour une description détaillée des commandes de menu du Authentic View, voir la section Référence de l'utilisateur de votre documentation de produit.

Site Web Altova : Édition du contenu XML, Création de contenu XML

11.2.1 Aperçu de la GUI

Le Authentic View présente en haut de l'écran une barre de menu et une barre d'outils, et trois volets qui couvrent le reste de l'interface : la fenêtre Projet, la fenêtre Principale et la fenêtre Assistants à la saisie. La disposition est représentée ci-dessous.



© 2016-2022 Altova GmbH

Barre Menu

Les menus disponibles dans la barre de menus sont décrits en détail dans la section Référence de l'utilisateur de votre documentation de produit.

Barre Outils

Les symboles et icônes affichés dans la barre outils sont décrits dans la section, icônes de la barre d'outils du Authentic View.

Fenêtre Projet

Vous pouvez grouper dans un seul projet des fichiers XML, XSL, schéma XML et Entité. Pour créer et modifier la liste des fichiers de projet, utiliser les commandes dans le menu **Projet** (décrit dans la section Référence de l'utilisateur de votre documentation de produit). La liste des fichiers de projet est affichée dans la fenêtre Projet. Un fichier dans la fenêtre Projet peut être accédé en le double-cliquant.

Fenêtre d'info

Cette fenêtre fournit des informations sur le nœud actuellement sélectionné dans le Mode Authentic.

Fenêtre Principale

Il s'agit de la fenêtre dans laquelle le document XML est affiché et édité. Il est décrit dans la section, fenêtre principale Authentic View.

Assistants à la saisie

Vous trouverez trois volets d'assistants à la saisie dans cette fenêtre : Éléments, Attributs et Entités. Les entrées apparaissant dans ces fenêtres sont sensibles au contexte (assistants à la saisie Éléments et Attributs), c.à.d. qu'elles dépendent de l'endroit où le curseur se trouve dans le document. Vous pouvez saisir un élément ou une entité dans le document en double-cliquant son assistant à la saisie. La valeur d'un attribut est saisie dans le champ de valeur de cet attribut dans assistant à la saisie Attributs. Voir la section Assistants à la saisie Authentic View pour plus de détails.

Barre de Statut

La barre de statut affiche le XPath vers le nœud actuellement sélectionné.

Menus contextuels

Il s'agit des menus qui apparaissent lorsque vous cliquez sur la touche de droite dans la fenêtre principale. Les commandes d'édition disponibles sont sensibles au contexte, c.à.d. qu'elles vous permettent de manipuler de la structure et du contenu afférent au nœud sélectionné. De telles manipulations comprennent l'insertion, ajout ou la suppression d'un nœud, l'ajout d'entités ou la découpe et le collage de contenu.

11.2.2 Icônes de la barre outils du Mode Authentic

Les icônes de la barre outils dans le Authentic View sont des raccourcis de commande. Certaines icônes utilisées dans les applications Windows ou les autres produits Altova vous seront déjà familières, d'autres seront nouvelles. Cette section décrit les icônes uniques au Authentic View. Dans la description ci-dessous, les icônes de la même famille sont regroupées.

Afficher/dissimuler les balises XML

Dans Authentic View, les balises pour tous, certains ou aucun des éléments ou attributs XML peuvent être affichés, soit avec leur nom (grandes balises) soit sans nom (petites balises). Les quatre icônes de balise apparaissent dans la barre outils et les commandes correspondantes sont disponibles dans le menu **Authentic**.



	Dissimuler les balises. Toutes les balises XML sont dissimulées sauf celles qui ont été réduites. Double-cliquer sur une balise réduite (qui est la manière habituelle de l'augmenter) dans le mode Dissimuler les balises pour afficher le contenu du nœud et dissimuler les balises.
	Afficher les petites balises. Les balises élément/attribut XML sont affichées sans leurs noms.
	Afficher les grandes balises. Les balises élément/attribut XML sont affichées avec leurs noms.
	Affichage des balises mixtes. Dans la StyleVision Power Stylesheet, chaque élément ou attribut XML peut être spécifié pour être affiché (soit en tant que balise grande ou petite), ou pas. Cela s'appelle un mode de balise mixte étant donné que certains éléments peuvent être spécifiés pour être affichés avec des balises et d'autres sans balises. Ainsi, dans le mode de balise mixte, l'utilisateur du Authentic View voit une balise personnalisée. Veuillez noter, néanmoins que cette personnalisation est créée par la personne qui a conçu la StyleVision Power Stylesheet. Elle ne peut pas être définie par l'utilisateur du Authentic View.

Éditer des structures de tables dynamiques

Les lignes dans une **table SPS dynamique** sont des répétitions d'une structure de données. Chaque ligne représente une occurrence d'un seul élément. Chaque ligne, a donc la même sous-structure XML que la suivante.

Les commandes d'édition des tables dynamiques manipulent les lignes d'une table dynamique SPS. Cela signifie que vous pouvez modifier le nombre et l'ordre des occurrences d'élément. Néanmoins, vous ne pouvez pas éditer les colonnes d'une table SPS dynamique puisque cela signifierait changer la sous-structure des occurrences de l'élément individuel.

Les icônes pour les commandes d'édition de la table dynamique apparaissent dans la barre d'outils, et sont aussi disponibles dans le menu **Authentic**.

© 2016-2022 Altova GmbH



	Apposer une ligne à la table
	Insérer une ligne dans la table
	Reproduire une ligne de table actuelle (des contenus de cellule sont dupliqués)
	Déplacer la ligne actuelle vers le haut d'une ligne
	Déplacer la ligne actuelle vers le bas d'une ligne
	Supprimer la ligne actuelle

Note : ces commande s'appliquent uniquement aux **tables SPS dynamiques**. Elles ne devraient pas être utilisées dans les tables SPS statiques. Les différents types de tables utilisées dans le Authentic View sont décrites dans la section Utiliser des tables dans le Mode Authentic de cette documentation.

Créer et éditer des tables XML

Si vous souhaitez présenter vos données sous la forme d'une table, vous pouvez insérer vos propres tables. Celles-ci sont insérées en tant que tables XML. Vous pouvez modifier la structure d'une table XML et formater la table. Les icônes pour créer et éditer les tables XML sont disponibles dans la barre outils, et sont affichées ci-dessous. Elles sont décrites dans la section icônes d'édition de tables XML.



Les commandes correspondant à ces icônes **ne sont pas disponibles en tant qu'items de menu**. Veuillez aussi noter que pour que vous puissiez utiliser des tables XML, cette fonction doit être activée et configurée convenablement dans la StyleVision Power Stylesheet. Une description détaillée des types de tables utilisés dans le Authentic View et de la manière dont créer et éditer les tables XML est indiquée dans la section Utiliser les tables dans le Mode Authentic.

Icônes de formatage de texte

Le texte dans le Authentic View est formaté en lui appliquant un élément ou un attribut XML qui a le formatage requis. Si ce type de formatage a été défini, le designer de la StyleVision Power Stylesheet peut fournir des icônes dans la barre outils du Authentic View pour appliquer le formatage. Pour appliquer un formatage de texte à l'aide d'une icône de formatage de texte, marquer le texte que vous souhaitez formater, et cliquer sur l'icône appropriée.

Icônes de navigation de ligne de BD

 Les icônes fléchées sont, de gauche à droite, Aller au premier enregistrement dans la BD ; Aller à l'enregistrement précédent ; Ouvrir le dialogue Aller à l'enregistrement n° ; Aller à l'enregistrement suivant et Aller au dernier enregistrement.

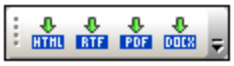
	Cette icône ouvre le dialogue Éditer la requête de base de données dans laquelle vous pouvez saisir une requête. Le Authentic View affiche les enregistrements requêtés.

Édition de la base de données XML

La commande **Sélectionner Nouvelle ligne avec les données XML pour l'édition** vous permet de sélectionner une nouvelle ligne depuis la table pertinente dans une BD XML, comme dans une IBM DB2. Cette ligne apparaît dans le Authentic View, peut y être éditée puis enregistrée dans la BD.

Touches de la barre d'outils Portable XML Form (PXF)

Les touches de barre d'outils PXF suivants sont disponibles dans le Authentic View de XMLSpy et Authentic Desktop:



Cliquer sur les touches individuelles génère une sortie HTML, RTF, PDF et/ou DocX.

Ces touches sont activées lorsqu'un fichier PXF est ouvert dans le Authentic View. Les touches individuelles sont activées si le fichier PXF a été configuré pour contenir la feuille de style XSLT pour ce format de sortie spécifique. Par exemple, si le fichier PXF était configuré pour contenu les feuilles de style XSLT pour HTML et RTF, alors seules les touches de barre d'outils pour les sorties HTML et RTF seront activées alors que celles pour les sorties PDF et DocX (Word 2007+) seront désactivées.

11.2.3 Fenêtre principale du Mode Authentic

Il y a quatre types de modes dans Authentic View: Grandes balises ; petites balises ; balises mixtes et dissimuler toutes les balises. Ces modes vous permettent de visualiser le document avec des niveaux variables d'information de marquage. Pour sauter entre les modes, utiliser les commandes dans le menu **Authentic** ou les icônes dans la barre d'outils (voir la section précédente, icônes de la barre d'outils du Mode Authentic).

Grandes balises

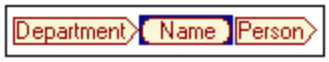
Cette option permet de montrer les balises de démarrage et de fin des éléments et des attributs avec les noms d'élément/attribut dans les balises :



L'élément `Name` dans la figure ci-dessus est **agrandi**, donc la balise de démarrage et de fin sont affichés ainsi que le contenu de l'élément. Un élément/attribut peut être **contracté** en double-cliquant soit sur son balisage

 © 2016-2022 Altova GmbH

de démarrage soit sur son balisage de fin. Pour agrandir l'élément/attribut contracté, double-cliquer sur la balise contractée.

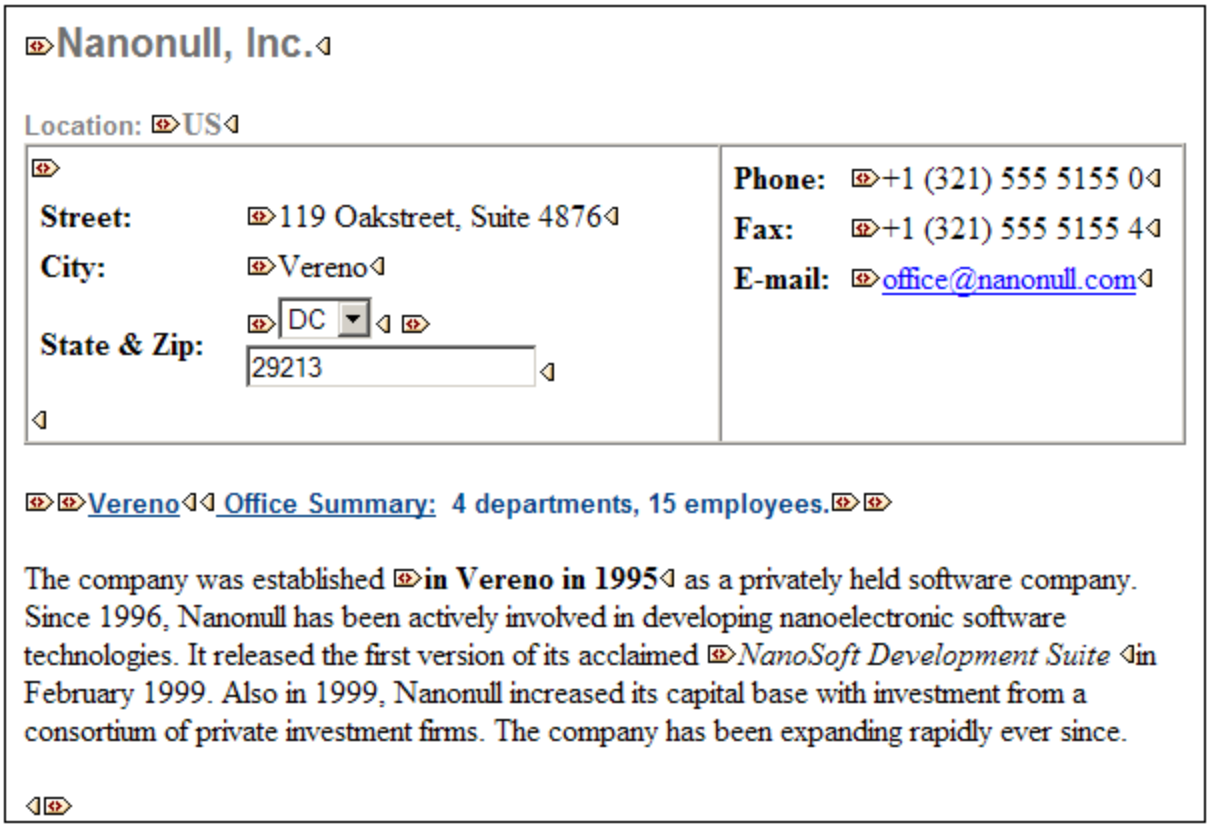


Dans les grandes balises, les attributs sont reconnus par le symbole égale à contenu dans les balises de démarrage et de fin de l'attribut :



Petites balises

Cette option permet de montrer les balises de démarrage et de fin des éléments et des attributs sans les noms :



Veuillez noter que les balises de démarrage contiennent un symbole alors que les balises de fin sont vides. De même, les balises d'élément ont un symbole <> alors que les onglets d'attributs contiennent un signe égal à comme leur symbole (*capture d'écran ci-dessous*).



Pour réduire ou agrandir un élément / attribut, double-cliquer sur la balise appropriée. L'exemple ci-dessous montre un élément réduit (marqué en bleu). Veuillez noter que la forme de la balise de l'élément réduit et celle de la balise de démarrage de l'élément agrandi à sa gauche.



Balisage mixte

Le balisage mixte constitue un balisage personnalisé. La personne ayant conçu la StyleVision Power Stylesheet peut spécifier l'affichage de grandes balises, de petites balises, ou d'aucune balise pour des éléments/attributs individuels dans le document. L'utilisateur du Authentic View peut consulter ce balisage personnalisé dans le mode de consultation Balisage mixte.

Dissimuler toutes les balises

Toutes les balises XML sont dissimulées. Puisque le formatage vu dans le Authentic View est le formatage du document imprimé, ce mode de consultation est un mode WYSIWYG du document.

Affichage du contenu

Dans le Authentic View, le contenu est affiché de deux manières :

- Texte simple. Vous saisissez le texte et celui-ci devient le contenu de l'élément ou la valeur de l'attribut.

- Appareils de saisie de données. L'affichage contient soit un champ de saisie (fenêtre de texte), un champ d'entrée multiligne, une liste de choix, une case à cocher ou un bouton radio. Dans le cas des champs de saisie et des champs d'entrée multiligne, le texte que vous saisissez dans le champ devient le contenu XML de l'élément ou de la valeur de l'attribut.

  Dans le cas des autres appareils de saisie de données, votre sélection produit une valeur de XML correspondante, qui est spécifiée dans la StyleVision Power Stylesheet. Ainsi, dans une liste de choix, une sélection de type, par exemple, "autorisé" (qui pourrait être disponible dans la liste déroulante de la liste de choix) pourrait mapper vers une valeur XML de "1", ou bien vers "autorisé", ou tout autre chose ; alors que "non autorisé" pourrait mapper vers "0", ou "non autorisé", ou tout autre chose.

Nœuds optionnels

Lorsqu'un élément ou un attribut est **optionnel** (conformément au schéma référencé), une invite de type `add [element/attribute]` est affiché :

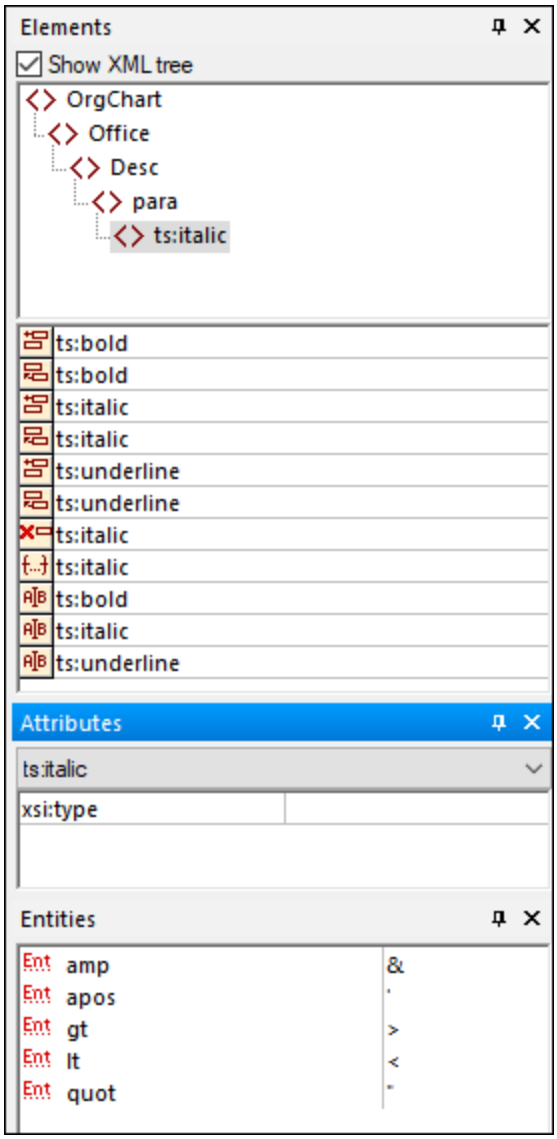
© 2016-2022 Altova GmbH

add synopsis

Cliquer sur l'invite permet d'ajouter l'élément et de placer le curseur pour l'entrée des données. S'il y a plusieurs nœuds optionnels, l'invite de type `add...` est affiché. Cliquer sur l'invite pour afficher un menu des nœuds optionnels.

11.2.4 Assistants à la saisie du Mode Authentic

Il y a trois assistants à la saisie dans le Authentic View : pour Éléments, Attributs et Entités. Ils sont affichés en tant que fenêtre le long du côté droit de l'interface du Authentic View (*voir capture d'écran ci-dessous*).



Les assistants à la saisie Éléments et Attributs sont sensibles au contexte, c. à.d. que ce qui apparaît dans les assistants à la saisie dépend de l'endroit où se trouve le curseur dans le document. Les entités affichées dans les assistants à la saisie Entités ne sont pas sensibles au contexte ; toutes les entités autorisées pour le document sont affichées où que soit le curseur.

Chacun des assistants à la saisie est décrit séparément ci-dessous.

Assistants à la saisie Éléments

L'assistant à la saisie Éléments consiste en deux parties :

- La partie supérieure, contenant une arborescence XML qui peut être activée et désactivée avec la case à cocher **Afficher arborescence XML**. L'arborescence XML montre les ancêtres jusqu'à l'élément de racine du document pour l'élément actuel. Lorsque vous cliquez sur un élément dans l'arborescence XML, les éléments correspondant à cet élément (tel que décrit dans l'item suivant dans cette liste) apparaissent dans la partie inférieure de l'assistant à la saisie Éléments.
- La partie inférieure, contenant une liste qui peut être insérée à l'intérieur, avant et après ; supprimée ; appliquée ou effacée de l'élément sélectionné ou la gamme de texte dans le Authentic View. Vous trouverez une description des choses à faire avec un élément recensé dans l'assistant à la saisie dans l'icône située à gauche du nom d'élément dans l'assistant à la saisie. Les icônes qui se produisent dans les assistants à la saisie Éléments sont recensées ci-dessous avec une description.

Pour utiliser un nœud depuis l'assistant à la saisie, cliquer sur cette icône.



Insérer après Élément

L'élément contenu dans l'assistant à la saisie est inséré après l'élément sélectionné. Veuillez noter qu'il est apposé au niveau hiérarchique correct. Par exemple, si votre curseur se trouve dans un élément `//sect1/para` et que vous apposez un élément `sect1`, le nouvel élément `sect1` sera apposé non en tant qu'un frère suivant `//sect1/para` mais en tant que le frère suivant de l'élément `sect1` qui est le parent de cet élément `para`.



Insérer avant Élément

L'élément contenu dans l'assistant à la saisie est inséré avant l'élément sélectionné. Veuillez noter que, tout comme la commande d'élément Insérer après, l'élément est inséré au niveau hiérarchique correct.



Supprimer Élément

Supprime l'élément et son contenu.



Insérer Élément

Un élément provenant de l'assistant à la saisie peut aussi être inséré dans un élément. Quand un curseur est placé dans un élément, les éléments enfants autorisés de cet élément peuvent être insérés. Veuillez noter que les éléments enfants autorisés peuvent faire partie d'un modèle de contenu éléments uniquement ainsi qu'un modèle de contenu mixte (texte plus éléments enfant).

Un élément enfant autorisé peut être inséré soit lorsqu'une plage de texte est sélectionnée soit lorsque le curseur est placé à un point d'intersection dans le texte.

- Lorsqu'une plage de texte est sélectionnée et qu'un élément a été inséré, la plage de texte devient le contenu de l'élément inséré.
- Lorsqu'un élément est inséré à un point d'insertion, l'élément est inséré à ce point.

 © 2016-2022 Altova GmbH

Une fois qu'un élément a été inséré, il peut être effacé en cliquant sur une des deux icônes Effacer Élément qui apparaît (dans l'assistant à la saisie Éléments) pour ces éléments inline. Laquelle des deux icônes apparaît dépend du fait que vous sélectionnez une plage de texte ou que vous placiez le curseur dans le texte en tant qu'un point insertion (voir ci-dessous).

 *Appliquer Élément*
Si vous sélectionnez un élément dans votre document (en cliquant sur sa balise de démarrage ou de fin dans le Mode Afficher grandes balises) et que cet élément peut être remplacé par un autre élément (par exemple dans un élément de contenu mixte comme `para`, un élément `italic` peut être remplacé par l'élément `bold`), cette icône indique que l'élément contenu dans l'Assistant à la saisie peut être appliqué à l'élément sélectionné (original). La commande **Appliquer élément** peut aussi être appliquée à une plage de texte se trouvant dans le cadre d'un élément de contenu mixte ; la plage de texte sera créée en tant que le contenu de l'élément appliqué.

- Si l'élément appliqué possède un **élément enfant portant le même nom** qu'un enfant de l'élément original et qu'une instance de cet élément enfant existe dans l'élément original, alors l'élément enfant de l'original est retenu dans le contenu de l'élément nouveau.
- Si l'élément appliqué ne possède **pas d'élément enfant portant le même nom** qu'un enfant instancié de l'élément original, alors l'enfant instancié de l'élément original est apposé en tant que frère de tout élément enfant ou d'éléments que le nouvel élément pourrait avoir.
- Si l'élément appliqué possède un **élément enfant pour lequel aucun équivalent n'existe** dans le modèle de contenu de l'élément original, alors cet élément enfant n'est pas créé directement mais une option dans le Authentic View vous permet de l'insérer.

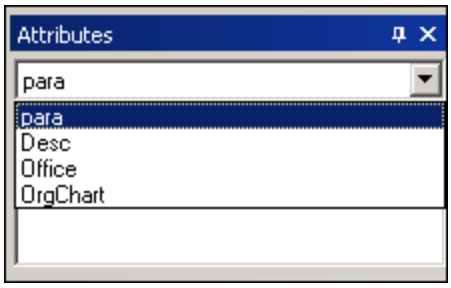
Si une plage de texte est sélectionnée au lieu d'un élément, l'application d'un élément à la sélection créera l'élément appliqué à cet emplacement avec la plage de texte sélectionnée en tant que son contenu. L'application d'un élément lorsque le curseur se trouve à un point d'insertion n'est pas autorisée.

 *Effacer Élément*
Cette icône apparaît lorsque le texte se trouvant dans un élément de contenu mixte est sélectionné. Cliquer sur l'icône pour effacer l'élément des alentours de la plage de texte sélectionnée.

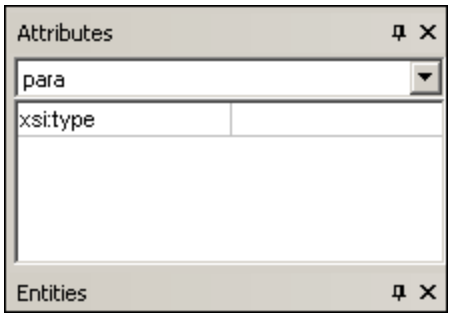
 *Effacer Élément (lorsque le point d'insertion est sélectionné)*
Cette icône apparaît lorsque le curseur est placé dans un élément qui est un enfant d'un élément de contenu mixte. Cliquer sur l'icône pour effacer l'élément inline.

Assistants à la saisie Attributs

L'assistant à la saisie Attributs consiste en une liste de choix déroulante et une liste des attributs. L'élément que vous avez sélectionné (vous pouvez cliquer sur la balise de démarrage ou de fin, ou placer le curseur n'importe où dans le contenu de l'élément pour le sélectionner) apparaît dans la liste de choix. L'assistant à la saisie Attributs affiché dans les graphiques ci-dessous a un élément `para` dans la liste de choix. Cliquer sur la flèche dans la liste de choix pour faire dérouler une liste de tous les **ancêtres remontant jusqu'à l'élément de racine du document** de l'élément `para`, qui, dans ce cas est `OrgChart`.



En dessous de la liste de choix, vous verrez s'afficher une liste des attributs valides pour cet élément, dans ce cas, pour `para`. Si un attribut est obligatoire sur un élément donné, il apparaîtra en gras. (Dans l'exemple ci-dessous, il n'y a pas d'attributs obligatoires sauf l'attribut intégré `xsi:type`.)



Afin de saisir une valeur pour un attribut, cliquer dans le champ de valeur de l'attribut et saisir la valeur. Cela permet de créer l'attribut et sa valeur dans le document XML.

Notez les points suivants :

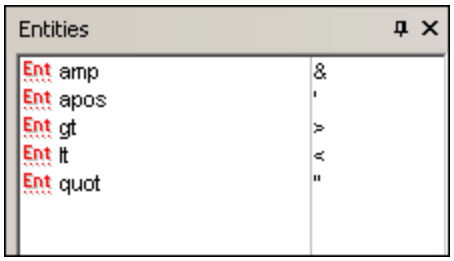
- Dans le cas de l'attribut `xsi:nil`, qui apparaît dans l'assistant à la saisie Attributs lorsqu'un élément nullable a été sélectionné, la valeur de l'attribut `xsi:nil` peut uniquement être saisi en sélectionnant une des valeurs autorisées (`true` ou `false`) depuis la liste déroulante pour la valeur de l'attribut.

- L'attribut `xsi:type` peut être changé en cliquant dans le champ de valeur de l'attribut puis en sélectionnant, depuis la liste déroulante qui apparaît, une des valeurs recensées. Ces valeurs sont les types abstraits disponibles définis dans le Schéma XML sur lequel le document Mode Authentic se base.

Assistants à la saisie Entités

L'assistant à la saisie Entités vous permet d'insérer une entité dans votre document. Les entités peuvent être utilisées pour insérer des caractères spéciaux ou des fragments de texte qui se produisent souvent dans un document (comme le nom d'une entreprise). Pour insérer une entité, placer le curseur à l'endroit du texte où vous souhaitez insérer l'entité, puis double-cliquer l'entité dans l'assistant à la saisie Entités.

© 2016-2022 Altova GmbH



Note : Une entité interne est une entité dont la valeur est définie dans le DTD. Une entité externe est une entité dont la valeur est contenue dans une source externe, par ex. un autre fichier XML. Les entités internes et externes sont recensées dans l'assistant à la saisie Entités. Lorsque vous insérez une entité, qu'elle soit interne ou externe, l'entité, pas sa valeur, est insérée dans le texte XML. Si l'entité est une entité interne, le Authentic View affiche **la valeur de l'entité**. Si l'entité est une entité externe, le Authentic View affiche l'entité, et pas sa valeur. Cela signifie, par exemple, qu'un fichier XML qui est une entité externe sera affichée dans l'affichage du Authentic View en tant qu'une entité ; son contenu ne remplace pas l'entité dans l'affichage Authentic View.

Vous pouvez aussi **définir vos propres entités** dans le Authentic View et celles-ci seront aussi affichées dans l'assistant à la saisie : voir Définir des entités dans la section Éditer dans le Authentic View.

11.2.5 Menus contextuels Mode Authentic

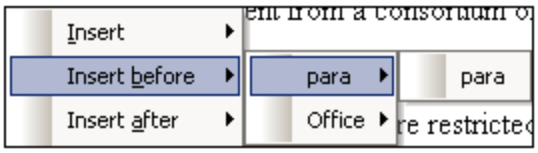
Cliquer avec la touche de droite sur des contenus de document sélectionnés ou sur un nœud pour faire apparaître un menu contextuel avec des commandes pertinentes pour la sélection de l'emplacement du curseur.

Insérer les éléments

La figure ci-dessous montre le sous-menu **Insérer**, qui est une liste de tous les éléments qui peuvent être insérés à l'emplacement actuel du curseur. Le sous-menu **Insérer avant** recense une liste de tous les éléments qui peuvent être insérés avant l'élément actuel. Le sous-menu **Insérer après** recense une liste de tous les éléments qui peuvent être insérés après l'élément actuel. Dans la figure ci-dessous, l'élément actuel est l'élément `para`. Les éléments `bold` et `italic` peuvent être insérés dans l'élément `para` actuel.



Comme indiqué ci-dessous, les éléments `para` et `Office` peuvent être insérés avant l'élément `para` actuel.



Les commandes d'insertion de nœud, de remplacement (**Appliquer**), et la suppression de balise (**Effacer**) qui sont disponibles dans le menu contextuel sont aussi disponibles dans les Assistants à la saisie Mode Authentic et sont décrites en détail dans cette section.

Insérer une entité

Positionner le curseur sur la commande Insérer Entité pour faire apparaître un sous-menu contenant une liste de toutes les entités déclarées. Cliquer sur une entité pour l'insérer dans la sélection. Voir Définir des Entités pour une description de la définition des entités pour le document.

Insérer une section CDATA

Cette commande est activée lorsque le curseur est placé dans du texte. Cliquer dessus pour insérer une section CDATA au niveau d'insertion du curseur. La section CDATA est délimitée par des balises de démarrage et de fin ; pour voir ces balises, vous devez passer à un affichage des grandes ou des petites balises. Dans les sections CDATA, les balises XML et le parsage est ignoré. Les caractères de balisage XML (l'esperluette, l'apostrophe, le signe supérieur à et inférieur à et les guillemets) ne sont pas traités en tant que balises, mais en tant que littéraux. Ainsi, des sections CDATA sont utiles pour des textes comme des listes de code de programme, qui contiennent des caractères de balisage XML.

Supprimer le nœud

Positionner le curseur de la souris sur la commande **Supprimer** pour ouvrir une liste de menu consistant en un nœud sélectionné et tous ses ancêtres inamovibles (ceux qui n'invalideront pas le document) jusqu'à l'élément du document. Cliquer sur l'élément à supprimer. Il s'agit là d'une méthode rapide pour supprimer un élément ou un ancêtre amovible. Veuillez noter qu'en cliquant sur un élément ancêtre, vous supprimerez tous ses descendants, y compris l'élément sélectionné.

Effacer

La commande **Effacer** efface la balise élément autour de la sélection. Si le nœud entier est sélectionné, la balise élément sera effacée pour le nœud entier. Si un segment de texte est sélectionné, la balise élément sera uniquement effacée autours de ce segment de texte.

Appliquer

La commande **Appliquer** applique un élément sélectionné à votre sélection dans la fenêtre principale. Pour plus de détails, voir Assistants à la saisie Mode Authentic.

Copier, Couper, Coller

Il s'agit des commandes Windows standard. Veuillez noter, néanmoins que la commande **Coller** colle du texte soit en tant que XML ou en tant que Texte, selon les spécifications du designer de la feuille de style pour la SPS complète. Pour plus d'informations concernant le fonctionnement des commandes **Copier en tant que XML** et **Copier en tant que Texte**, voir la description de la commande **Coller en tant que** juste en-dessous.

Coller comme

La commande **Coller en tant que** permet de coller en tant que XML ou en tant que texte un fragment XML de Authentic View (qui a été copié dans le presse-papiers). Si le fragment copié est collé en tant que XML, il est collé avec sa balise XML. S'il est collé en tant que texte, seul le contenu de texte du fragment copié est collé (pas la balise XML, si existante). Les situations suivantes sont possibles :

- Un ***nœud entier et ses balises de marquage*** sont marqués dans Authentic View et copiés dans le presse-papiers. (i) Le nœud peut être collé en tant que XML dans n'importe quel endroit où ce nœud peut être placé validement. Il ne sera pas collé dans un emplacement invalide. (ii) Si le nœud est collé

© 2016-2022 Altova GmbH

en tant que texte, seul le *contenu du texte* de ce nœud sera collé (pas la balise) ; le contenu suivant peut être collé dans n'importe quel endroit du document XML où il est possible de coller du texte.

- Un ***fragment de texte*** est marqué dans le Authentic View et copié dans le presse-papiers. (i) Si ce fragment est collé en tant que XML, alors les balises de marquage XML du texte, bien que celles-ci n'ont pas été copiées explicitement avec le fragment de texte, seront collées avec le texte, mais uniquement si le nœud XML est valide à l'endroit où le fragment est collé. (ii) Si le fragment est collé en tant que texte, il peut être collé n'importe où dans le document XML où il est possible de coller du texte.

Note : **I**e texte sera copié dans les nœuds dans les endroits où le texte est autorisé, vous devrez donc veiller à ce que le texte copié n'invalide pas le document. Le texte copié devra donc être : (i) lexicalement valide dans le nouvel emplacement (par exemple des caractères non-numériques dans un nœud numérique serait un exemple invalide), et (ii) ne pas invalider le nœud d'une autre manière (par exemple, quatre chiffres dans un nœud qui n'accepte que des nombres à trois chiffres serait un exemple d'invalidation du nœud).

Note : si le texte collé invalide le document de quelque manière, le texte sera affiché en rouge.

Supprimer

La commande **Supprimer** supprime le nœud sélectionné et son contenu. Un nœud est considéré être sélectionné pour cet objectif en plaçant le curseur dans le nœud ou en cliquant soit sur la balise de démarrage ou de fin du nœud.

11.3 Éditer dans le Mode Authentic

Cette section décrit les fonctions importantes du Authentic View en détail. Les fonctions décrites dans cette section y ont été incluses soit parce qu'elles sont fréquemment utilisées soit parce que les mécanismes ou les concepts impliqués nécessitent des explications.

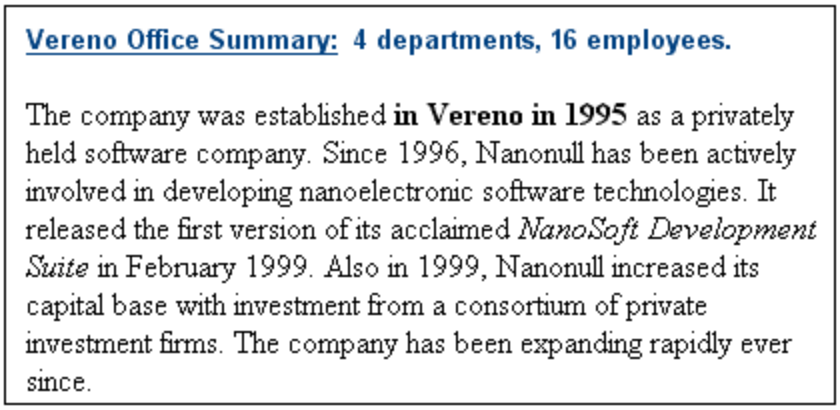
La section explique les fonctions suivantes :

- Il existe trois types distincts de tables utilisées dans le Authentic View. La section Utiliser des tables dans le Authentic View explique les trois types de tables (SPS statique, SPS dynamique et XML), et la manière et le moment de les utiliser. Elle commence avec l'aspect général et conceptuel de la fonction et continue ensuite vers les détails de son utilisation.
- Le sélecteur de date est un calendrier graphique qui saisit les dates dans le format XML correct lorsque vous cliquez sur une date. Voir le Sélecteur de date.
- Une entité est une abréviation pour un caractère spécial ou une chaîne de texte. Vous pouvez définir vos propres entités, qui vous permettent d'insérer ces caractères spéciaux ou chaînes de texte en insérant les entités correspondantes. Voir Définir des entités pour plus de détails.
- Dans les éditions Enterprise et Professional des produits Altova, les utilisateurs du Authentic View peuvent signer des documents XML avec des signatures XML numériques et vérifier ces signatures.
- Quels types de formats d'image peut être affiché dans le Authentic View.

Site Web Altova : Édition du contenu XML, Création de contenu XML

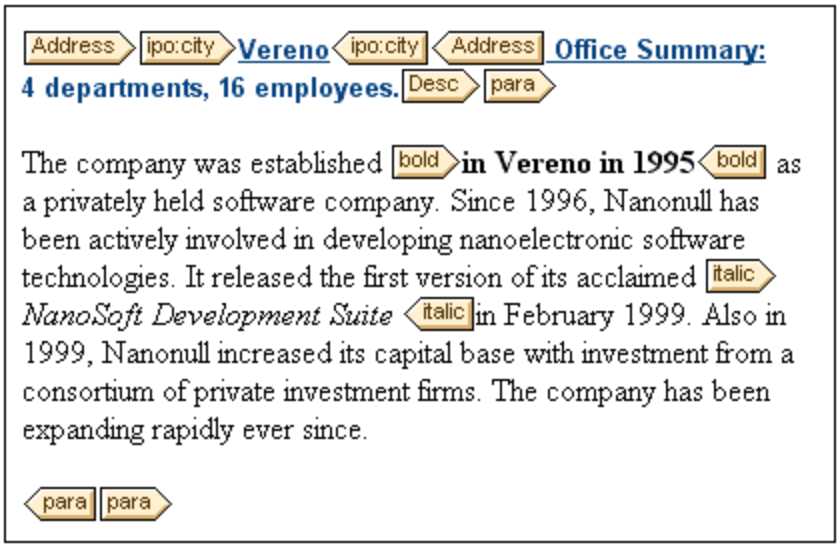
11.3.1 Édition de base

Lorsque vous éditez dans le Authentic View, vous éditez un document XML. Néanmoins, le Authentic View peut dissimuler le balisage structurel du XML du document, affichant ainsi uniquement le contenu du document (*première capture d'écran ci-dessous*). Vous n'aurez donc pas à traiter avec l'aspect technique de XML, et vous pourrez éditer le document comme si vous éditeriez un document texte normal. Si vous le souhaitez, vous pouvez passer au balisage à tout moment pendant l'édition (*seconde capture d'écran ci-dessous*).



Un document éditable Authentic View sans balises XML.

© 2016-2022 Altova GmbH

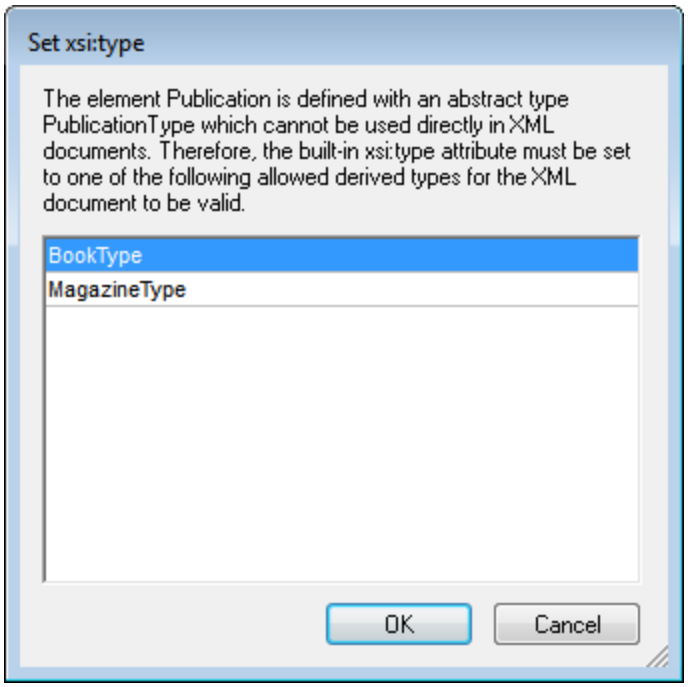


Un document éditable Authentic View avec des balises de marquage XML.

Insérer les nœuds

Très souvent, vous devrez ajouter un nouveau nœud au document XML Authentic. Par exemple, un nouvel élément `Person` peut devoir être ajouté à un document de type carnet d'adresses. Dans ces cas, le Schéma XML permettrait l'ajout du nouvel élément. Il vous suffit de cliquer avec la touche de droite sur le nœud dans le document Mode Authentic avant ou après celui auquel vous souhaitez ajouter le nouveau nœud. Dans le menu contextuel qui apparaît, sélectionner **Insérer avant** ou **Insérer après**, vos besoins. Les nœuds disponibles pour l'insertion à cet endroit dans le document sont recensés dans un sous-menu. Cliquer sur le nœud requis pour l'insérer. Le nœud sera inséré. Tous les nœuds descendants obligatoires sont aussi insérés. Si un nœud descendant est optionnel, un lien cliquable, `Add NodeName`, apparaît afin de vous permettre d'ajouter le nœud optionnel si vous le souhaitez.

Si le nœud ajouté est un élément avec un type abstrait, alors un dialogue (*par exemple comme celui ci-dessous*) apparaîtra contenant une liste des types dérivés disponibles dans le Schéma XML.



La fenêtre dans la capture d'écran ci-dessus s'ouvre lorsqu'un élément `Publication` est ajouté. L'élément `Publication` est de type `PublicationType`, qui est un type complexe abstrait. Les deux types complexes `BookType` et `MagazineType` sont dérivés depuis le `PublicationType` abstrait. C'est pourquoi, lorsqu'un élément `Publication` est ajouté dans le document XML, un de ces deux types concrets dérivés du type abstrait de `Publication` doit être spécifié. Le nouvel élément `Publication` sera ajouté avec un attribut `xsi:type` :

```
<Publication xsi:type="BookType"> ... </Publication>
<Publication xsi:type="MagazineType"> ... </Publication>
...
<Publication xsi:type="MagazineType"> ... </Publication>
```

Sélectionner un des types dérivés disponibles et cliquer sur **OK** permet de :

- Définir le type dérivé sélectionné en tant que la valeur de l'attribut `xsi:type` de l'élément
- Insérer l'élément avec les nœuds descendants définis dans le modèle de contenu du type dérivé sélectionné.

Le type dérivé sélectionné peut être changé ultérieurement en changeant la valeur de l'attribut `xsi:type` de l'élément dans l'Assistant à la saisie Attributs. Lorsque le type de l'élément est changé de cette manière, tous les nœuds du modèle de contenu du type précédent sont supprimés et les nœuds du modèle de contenu du nouveau type sont insérés.

Éditer du texte

Un document Authentic View consiste essentiellement en du texte et des images. Pour éditer le texte dans le document, placer le curseur à l'emplacement où vous souhaitez insérer le texte et effectuer votre saisie. Vous pouvez copier, déplacer et supprimer du texte à l'aide des touches familières (comme la touche **Supprimer**) et les mécanismes glisser et déposer. Une exception est la touche **Entrée**. Puisque le document du Authentic View est préformaté, vous ne devez pas (et ne pourrez pas) ajouter des lignes supplémentaires ou des

© 2016-2022 Altova GmbH

espaces entre les items. La touche **Entrée** dans le Authentic View sert donc à apposer une autre instance de l'élément actuellement en cours d'édition et devrait être utilisé exclusivement à cette fin.

Copier en tant que XML ou en tant que texte

Le texte peut être copié et collé en tant que XML ou en tant que texte.

- Si du texte est collé en tant que XML, le balisage XML est collé avec le contenu de texte des nœuds. Le balisage XML est collé même si uniquement une partie des contenus d'un nœud a été copié. La balise peut uniquement être collée aux endroits autorisés par le schéma.
- Si le texte est collé en tant que texte, la balise XML n'est pas collée.

Afin de coller du texte en tant que XML ou en tant que texte, copier tout d'abord le texte (**Ctrl+C**), cliquer avec la touche de droite à l'emplacement où vous souhaitez coller le texte puis sélectionner la commande de menu contextuel **Coller en tant que | XML** ou **Coller en tant que | Texte**. Si le raccourcis **Ctrl+V** est utilisé, le texte sera collé dans le Mode Coller par défaut de la SPS. Le Mode Coller par défaut aura été spécifié par le concepteur de la SPS. Pour plus de détails, voir la section Menus contextuels.

En alternative, il est possible de déplacer un texte marqué à l'emplacement où il doit être collé. Lorsque le texte est déposé, une fenêtre popup apparaît vous demandant si le texte doit être collé en tant que texte ou en tant que XML. Sélectionner l'option désirée.

Formater du texte

Un des principes de base des systèmes de document XML est que le contenu est gardé séparément de la présentation. Le document XML contient le contenu, alors que la feuille de style contient la présentation (formatage). Dans le Authentic View, le document XML est présenté par le biais de la feuille de style. Cela signifie que tout le formatage que vous verrez dans le Authentic View est produit par la feuille de style. Si vous voyez du texte gras, ce formatage gras aura été défini dans la feuille de style. Si vous voyez une liste ou une table, ce format de liste ou de table aura été défini dans la feuille de style. Le document XML dont vous vous servez pour l'édition dans le Authentic View ne recèle que du contenu ; il ne contient aucune information concernant le formatage. Le formatage est contenu dans la feuille de style. Cela signifie pour vous, l'utilisateur du Authentic View, que vous n'avez pas à (et ne pourrez pas) formater le texte que vous éditez. Vous éditez le contenu. Le formatage qui s'applique automatiquement au contenu que vous éditez est lié à la valeur sémantique et/ou structurelle des données que vous éditez. Par exemple, une adresse e-mail (qui peut être considérée comme une unité sémantique) sera formatée automatiquement d'une certaine manière parce qu'il s'agit d'un e-mail. De la même manière, un titre doit se produire à un emplacement particulier du document (unité aussi bien structurelle que sémantique) et sera formaté automatiquement selon la manière spécifiée par le concepteur dans la feuille de style pour les titres. Vous ne pouvez pas modifier le formatage de l'adresse e-mail ou du titre. Tout ce que vous pouvez faire est d'éditer le contenu de l'adresse e-mail ou du titre.

Dans certains cas, le contenu peut nécessiter une présentation particulière ; par exemple, une chaîne de texte doit être présentée en gras. Dans ces cas, la présentation doit être liée avec un élément structurel du document. Par exemple, une chaîne de texte qui doit être présentée en gras, sera séparée structurellement du contenu environnant par des balises que le concepteur de la feuille de style formatera en gras. Si vous, en tant que l'utilisateur du Authentic View souhaitez utiliser une telle chaîne de texte, vous devrez contenir la chaîne de texte dans l'élément de balise approprié. Pour plus d'information concernant la procédure à suivre, voir la commande Insérer l'élément dans la section Assistant à la saisie Éléments de la documentation.

Utiliser RichEdit dans Authentic View

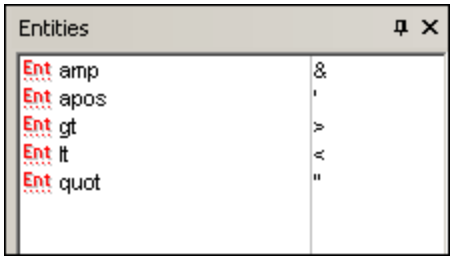
Dans le Authentic View, lorsque le curseur est placé dans un élément qui a été créé en tant qu'un composant RichEdit, les touches et les commandes dans la barre d'outils RichEdit (*capture d'écran ci-dessous*) sont activées. Dans le cas contraire, elles sont grisées.



Sélectionner le texte auquel vous souhaitez appliquer le style et spécifier le style avec les touches et les commandes de la barre d'outils RichEdit. RichEdit permet aux utilisateurs du mode Authentic View de spécifier la police, le poids, la décoration, la taille de la police, la couleur, la couleur de l'arrière-plan et l'alignement du texte. Le texte qui aura été stylisé sera contenu dans les balises de l'élément de style.

Insérer des entités

Dans les documents XML, certains caractères sont réservés pour le balisage et ne peuvent pas être utilisés dans du texte normal. Il s'agit de l'esperluette (`&`), de l'apostrophe (`'`), du caractère inférieur à (`<`), supérieur à (`>`) et des guillemets (`"`). Si vous souhaitez utiliser ces caractères dans vos données, vous devrez les insérer en tant que références d'entité, par le biais de l'Assistant à la saisie Éléments (*capture d'écran ci-dessous*).



XML offre aussi la possibilité de créer des entités personnalisées. Celles-ci peuvent prendre la forme de : (i) caractères spéciaux qui ne sont pas disponibles sur votre clavier, (ii) des chaînes de texte que vous souhaitez réutiliser dans le contenu de votre document, (iii) des fragments de données XML, ou (iv) d'autres ressources comme des images. Vous pouvez définir vos propres entités dans le cadre de l'application Authentic View. Une fois définies, ces entités apparaissent dans l'Assistant à la saisie Éléments et peuvent être insérées comme dans le document.

Insérer des sections CDATA

Les sections CDATA sont des sections de texte dans un document XML que le parseur XML ne traite pas en tant que données XML. Elles peuvent être utilisées pour échapper de grandes sections de texte si vous ne souhaitez pas remplacer des caractères spéciaux par des références d'entités ; cela peut être le cas, par exemple, avec un code de programme ou un fragment XML qui doit être reproduit avec ses balises de marquage. Les sections CDATA peuvent se produire dans un contenu d'élément et sont délimitées par `<![CDATA[` et `]]>` au début et à la fin, respectivement. Par conséquence, la chaîne de texte `]]>` ne devrait pas se produire dans une section CDATA car elle marquerait la fin prématurée de la section. Dans ce cas, le caractère "supérieur à" devrait être échappé par sa référence d'entité (`>`). Pour insérer une section CDATA dans le cadre d'un élément, placer le curseur à l'emplacement désiré, cliquer avec la touche de droite et sélectionner **Insérer Section CDATA** depuis le menu contextuel. Pour voir la balise de section CDATA dans le Authentic View, passer à l'affichage des balises. En alternative, vous pouvez marquer le texte devant être inséré dans une section CDATA, puis sélectionner la commande **Insérer la section CDATA**.

Note : Les sections CDATA ne peuvent pas être insérées dans des champs d'entrée (c'est à dire dans des champs de texte et des champs à texte multiligne). Les sections CDATA peuvent uniquement être

© 2016-2022 Altova GmbH

saisies dans des éléments qui sont affichés dans le Authentic View en tant que composants de contenu de texte.

Éditer et suivre les liens

Un hyperlien consiste en deux parties : le texte de lien et la cible du lien. Vous pouvez éditer le texte de lien en cliquant dans le texte et en l'éditant. Mais vous ne pouvez pas éditer la cible du lien. (La cible du lien est définie par le concepteur de la feuille de style (soit en saisissant une adresse cible statique soit en dérivant l'adresse cible depuis les données contenues dans le document XML)). Vous pouvez vous rendre à la cible du lien depuis le Authentic View en appuyant sur **Ctrl** et en cliquant sur le texte du lien. (Souvenez-vous : si vous cliquez simplement sur le lien, vous aboutirez à l'édition du texte de lien.)

11.3.2 Tables dans le Mode Authentic

Les trois types de table se répartissent dans deux catégories : les tables SPS (statiques et dynamiques) et les tables CALS/HTML.

Les tables SPS sont de deux types : statiques et dynamiques. Les tables SPS sont conçues par le designer de la StyleVision Power Stylesheet à laquelle votre document XML est lié. Vous ne pouvez pas insérer de table SPS dans le document XML, mais vous pouvez saisir des données dans des champs de table SPS et ajouter et supprimer les lignes des tables SPS dynamiques. La section ci-dessous concernant les tables SPS explique les fonctions de ces tables.

Les tables CALS/HTML sont insérées par vous, l'utilisateur du Authentic View. Leur objectif est de vous permettre d'insérer des tables à tout emplacement autorisé dans la hiérarchie du document, le cas échéant. Les fonctions d'édition des Tables CALS/HTML et des icônes d'édition de la table CALS/HTML sont décrites ci-dessous.

11.3.2.1 Tables SPS

Deux types de tables SPS sont utilisées dans Authentic View : les tables statiques et les tables dynamiques.

Tables statiques

Les tables statiques ont une structure et un type de contenu des cellules fixes. Vous, en tant que l'utilisateur de Authentic View, pouvez saisir des données dans les cellules de table, mais vous ne pouvez pas changer la structure de ces tables (donc ajouter des lignes ou des colonnes, etc.) ou modifier le type de contenu d'une cellule. Vous saisissez des données soit en entrant du texte soit en sélectionnant à partir d'options présentées sous la forme de cases à cocher ou de boutons radio ou en tant que liste dans une liste de choix. Une fois avoir saisi les données, vous pouvez les éditer.

Nanonull, Inc.

Street:	119 Oakstreet, Suite 4876	**Phone:**	+1 (321) 555 5155
City:	Vereno	**Fax:**	+1 (321) 555 5155 - 9
State & Zip:	DC 29213	**E-mail:**	office@nanonull.com

Note : Les icônes ou les commandes d'édition des tables dynamiques **ne doivent pas** être utilisées pour éditer des tables statiques.

Tables dynamiques

Les tables dynamiques ont des lignes qui représentent une structure de données répétitives, donc chaque ligne a une structure de données identique (ce qui n'est pas le cas avec des tables statiques). C'est pourquoi vous pouvez effectuer des opérations de ligne : apposer une ligne, insérer une ligne, déplacer une ligne vers le haut, déplacer une ligne vers le bas, et supprimer des lignes. Ces commandes sont disponibles sous le menu **Authentic** et en tant qu'icônes dans la barre outils (affiché ci-dessous).



Pour utiliser ces commandes, placer le curseur n'importe où dans la ligne appropriée et sélectionner la commande nécessaire.

Administration

First	Last	Title	Ext	EMail	Shares	Leave		
						Total	Used	Left
Vernon	**Callaby**	Office Manager	581	v.callaby@nanonull.com	1500	25	4	21
Frank	Further	Accounts Receivable	471	f.further@nanonull.com	0	22	2	20
Loby	Matise	Accounting Manager	963	l.matise@nanonull.com	add Shares	25	7	18
Employees: 3 (20% of Office, 9% of Company)					**Shares: 1500 (13% of Office, 6% of Company)**			
Non-Shareholders: Frank Further, Loby Matise.								

Pour vous déplacer parmi les cellules de la table, utiliser les touches fléchées Haut, Bas, Gauche, Droite. Pour vous déplacer d'une cellule à une autre, utiliser la touche **Tab**. Appuyer sur la touche **Tab** dans la dernière cellule de la dernière ligne crée une nouvelle ligne.

© 2016-2022 Altova GmbH

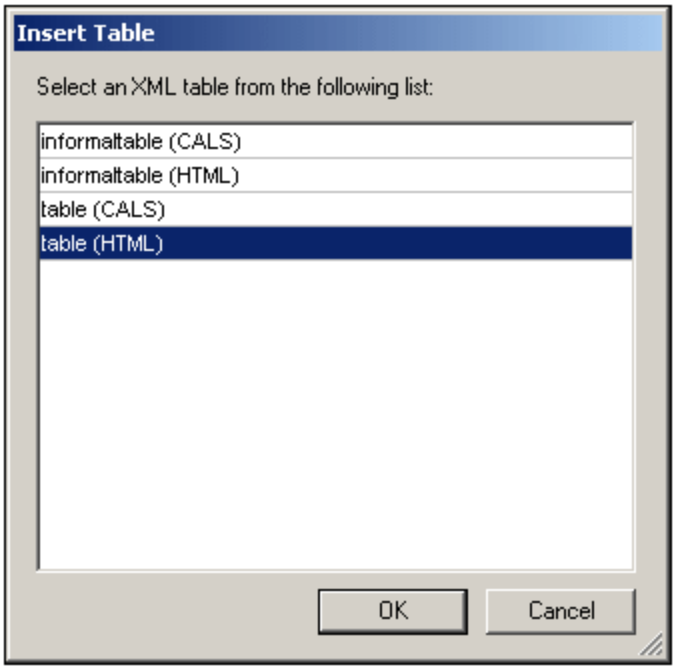
11.3.2.2 Tables CALS/HTML

Les tables CALS/HTML peuvent être insérées par vous, l'utilisateur du Authentic View pour certaines structures de données XML qui ont été spécifiées pour afficher un format de table. Trois étapes doivent être effectuées pour travailler avec des tables CALS/HTML : insérer la table ; la formater ; et saisir les données. Les commandes pour travailler avec les tables CALS/HTML sont disponibles en tant qu'icônes dans la barre d'outils (voir icônes d'édition de table CALS/HTML).

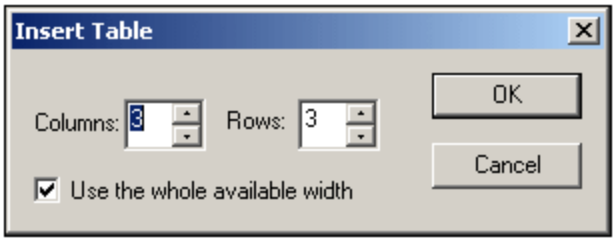
Insérer les tables

Pour insérer une table CALS/HTML, procéder comme suit :

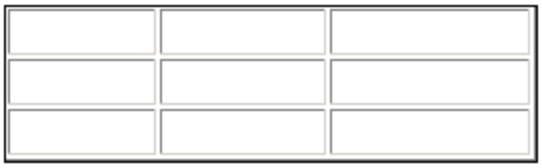
1. Placer votre curseur à l'endroit où vous souhaitez insérer la table, et cliquer sur l'icône . (Veuillez noter que l'endroit où vous pouvez insérer des tables est déterminé par le schéma.) Le dialogue Insérer la table (*capture d'écran ci-dessous*) apparaît. Ce dialogue recense toutes les structures de données d'éléments XML pour lesquelles une structure de table a été définie. Par exemple, dans la capture d'écran ci-dessous, l'élément `informaltable` et l'élément `table` ont tous les deux été définis en tant que table CALS et en tant que table HTML.



2. Sélectionner l'entrée contenant l'élément et le modèle de table que vous souhaitez insérer et cliquer sur **OK**.
3. Dans le dialogue suivant (*capture d'écran ci-dessous*), sélectionner le numéro des colonnes et des lignes et spécifier si un en-tête et/ou un pied de page doit être ajouté à la table et si toute la table doit être étendue sur la largeur totale disponible. Cliquer **OK** une fois terminé.



La table suivante a été créée pour afficher les spécifications données dans le dialogue ci-dessus.



En utilisant les commandes de menu **Table**, vous pouvez ajouter et supprimer des colonnes et créer des joints et des divisions de colonnes et de lignes. Mais pour commencer, vous devez créer une structure large.

Formater les tables et la saisie les données

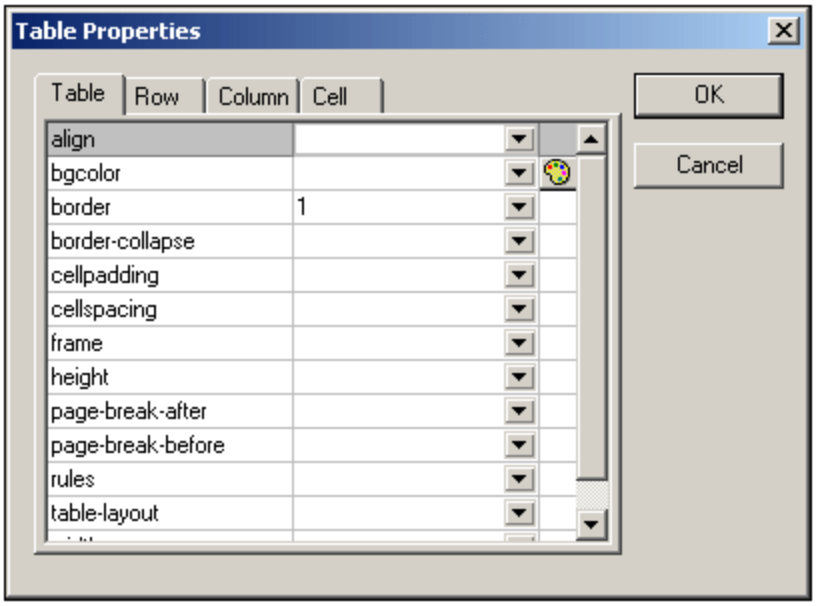
Le formatage de table aura déjà été attribué au design du document. Néanmoins, vous pourrez, dans certaines circonstances, modifier le formatage de table. Ces circonstances sont les suivantes :

- Les éléments correspondants aux éléments de structure de table divers doivent avoir les propriétés de table CALS ou HTML pertinentes définies en tant qu'attributs (dans le schéma XML sous-jacent). Seuls les attributs qui sont définis seront disponibles pour le formatage. Si, dans le design, les valeurs ont été définies pour ces attributs, vous pourrez substituer ces valeurs dans le Authentic View.
- Dans le design aucun attribut `style` contenant les styles CSS doivent avoir été définis. Si un attribut de style contenant des styles CSS a été spécifié pour un élément, l'attribut `style` a une précédence au-dessus de tout autre attribut de formatage défini sur cet élément. En résultat, tout formatage spécifié dans Authentic View sera substitué.

Pour formater une table, une ligne, une colonne ou une cellule, procéder comme suit :

1. Placer le curseur dans n'importe quel endroit de la table et cliquer sur l'icône  (Propriétés de table). Le dialogue Propriétés de table s'ouvre (*capture d'écran ci-dessous*), dans lequel vous pouvez spécifier le formatage pour la table, pour une ligne, une colonne ou une cellule.

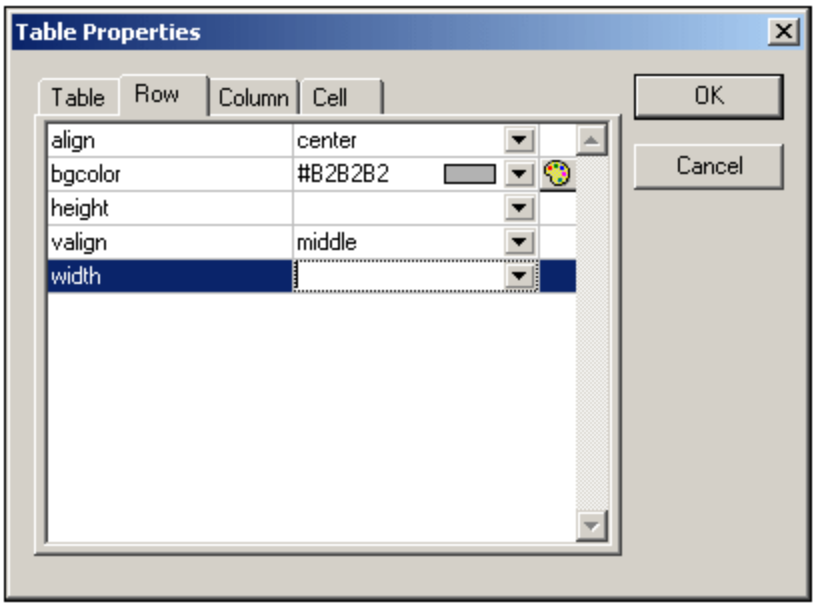
© 2016-2022 Altova GmbH



2. Définir les propriétés d'espacement de cellule et de remplissage de la cellule sur "0". Votre table ressemble maintenant à celle ci-dessous :



3. Placer le curseur dans la première ligne pour la formater et cliquer sur l'icône  (Propriétés de table). Cliquer sur l'onglet **Ligne**.



Puisque la première ligne sera la ligne d'en-tête, définir une couleur d'arrière-plan pour faire ressortir cette ligne par rapport aux autres. Veuillez noter les propriétés Ligne qui ont été définies dans la figure ci-dessus. Ensuite, saisir le texte de l'en-tête de colonne. Votre table devrait ressembler à la figure ci-dessous :

Name	Telephone	Email

Veuillez noter que l'alignement est centré, tel que spécifié.

4. À présent, si, par exemple, vous souhaitez diviser la colonne "Telephone" en deux sous-colonnes "Office" et "Home", dans lequel cas vous devrez diviser la largeur de la colonne "Telephone" en deux colonnes. Tout d'abord, néanmoins, nous allons diviser la portée verticale de la cellule de l'en-tête pour en faire une ligne de sous-en-tête. Placer le curseur dans la cellule "Telephone" et cliquer sur l'icône  (Partager verticalement). Votre table devrait ressembler à la figure ci-dessous :

Name	Telephone	Email

5. À présent, placer le curseur dans la cellule contenant "Telephone", et cliquer sur l'icône  (Partager horizontalement). Ensuite saisir dans les en-têtes de colonne "Office" et "Home". Votre table devrait ressembler à la figure ci-dessous :

Name	Telephone		Email
	Office	Home	

À présent, vous devrez partager la largeur horizontale de chaque cellule dans la colonne "Telephone".

Vous pouvez aussi ajouter et supprimer des colonnes et des lignes et aligner verticalement du contenu de cellule, en utilisant les icônes d'édition de table. Les icônes d'édition de table CALS/HTML sont décrites dans la section titrée Icônes d'édition de table CALS/HTML.

Se déplacer dans les cellules de la table

Pour vous déplacer dans les cellules de la table CALS/HTML, utiliser les touches fléchées Haut, Bas, Gauche, Droite.

Saisir les données dans une cellule

Pour saisir les données dans une cellule, placer le curseur dans la cellule et saisir les données.

© 2016-2022 Altova GmbH

Formater un texte

Les textes contenus dans une table CALS/HTML, comme pour l'autre texte dans le document XML, doivent être formatés à l'aide d'élément ou d'attributs XML. Pour ajouter un élément, marquer le texte et double-cliquer sur l'élément nécessaire dans l'Assistant à la saisie Éléments. Pour spécifier une valeur d'attribut, placer le curseur dans le fragment de texte et saisir la valeur d'attribut nécessaire dans l'Assistant à la saisie Éléments. Une fois avoir formaté les textes d'en-tête en gras, votre table ressemblera à cela :

Name	Telephone		Email
	Office	Home	

Le texte ci-dessus a été formaté en marquant le texte puis en double-cliquant l'élément `strong`, pour lequel un modèle global existe qui spécifie "gras" en tant que le poids de la police. Le formatage de texte est visible immédiatement.

Note : pour afficher le formatage de texte dans Authentic View, un modèle global, avec le formatage de texte nécessaire doit avoir été créé dans StyleVision pour l'élément en question.

11.3.2.3 Icônes d'édition de table CALS/HTML

Les commandes nécessaires pour éditer les tables CALS/HTML sont disponibles sous la forme d'icônes dans la barre outils. Vous trouverez la liste ci-dessous. Veuillez noter que des commandes de menu correspondantes existent pour chacune de ces icônes. Pour une description complète de l'utilisation de Tables CALS/HTML, voir Tables CALS/HTML.

Insérer table



La commande "Insérer la table" insère une **table CALS/HTML** à la position actuelle du curseur.

Supprimer table



La commande "Supprimer la table" supprime la table active actuellement.

Apposer une ligne



La commande "Apposer une ligne" appose une ligne à la fin de la table active actuellement.

Apposer une colonne



La commande "Apposer une colonne" appose une colonne à la fin de la table active actuellement.

Insérer une ligne



La commande "Insérer une ligne" insère une ligne au-dessus de la position actuelle du curseur dans la table active actuellement.

Insérer une colonne



La commande "Insérer une colonne" insère une colonne à gauche de la position actuelle du curseur dans la table active actuellement.

Joindre cellule gauche



La commande "Joindre cellule gauche" joint la cellule actuelle (position actuelle du curseur) avec la cellule à sa gauche. Les balises des deux cellules demeurent dans la nouvelle cellule, les en-têtes de colonne demeurent inchangés et sont concaténés.

Joindre cellule droite



La commande "Joindre cellule droite" joint la cellule actuelle (position actuelle du curseur) avec la cellule à sa droite. Les contenus des deux cellules sont concaténés dans la nouvelle cellule.

Joindre cellule en-dessous



La commande "Joindre cellule en-dessous" joint la cellule actuelle (position actuelle du curseur) avec la cellule en-dessous. Les contenus des deux cellules sont concaténés dans la nouvelle cellule.

Joindre cellule au-dessous



La commande "Joindre cellule au-dessus" joint la cellule actuelle (position actuelle du curseur) avec la cellule au-dessus. Les contenus des deux cellules sont concaténés dans la nouvelle cellule.

Partager la cellule horizontalement



La commande "Partager la cellule horizontalement" crée une nouvelle cellule à droite de la cellule active actuellement. La taille des deux cellules est la même que la cellule originale.

Partager la cellule verticalement



La commande "Partager la cellule verticalement" crée une nouvelle cellule en-dessous de la cellule active actuellement.

Aligner en haut



Cette commande aligne les contenus de la cellule dans le haut de la cellule.

Centrer verticalement



Cette commande recentre les contenus de la cellule.

Aligner en bas



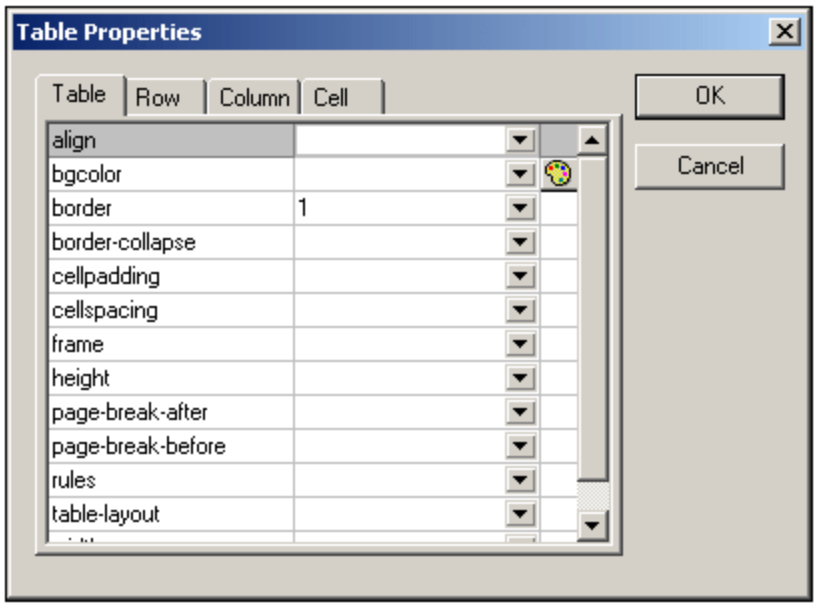
Cette commande aligne les contenus de la cellule dans le bas de la cellule.

© 2016-2022 Altova GmbH

Propriétés de table



La commande "Propriétés de table" ouvre le dialogue Propriétés de table. Cette icône est uniquement active pour les tables HTML, elle ne peut pas être cliquée pour les tables CALS.



11.3.3 Éditer une BD

Dans le Authentic View, vous pouvez éditer des tables de bases de données (BD) et réenregistrer les données dans une BD. Cette section contient une description complète des fonctions d'interface à votre disponibilité lorsque vous éditez une table de BD. Les points généraux suivants doivent être notés :

- Le nombre d'enregistrements dans une table de BD qui est affiché dans Authentic View peut avoir été limité délibérément par le concepteur de la StyleVision Power Stylesheet pour rendre le design plus compact. Dans ces cas, seul ce nombre limité d'enregistrements sera chargé initialement dans Authentic View. À l'aide des icônes de navigation de ligne de table de BD (*voir* Parcourir une table de BD), vous pouvez charger et afficher les autres enregistrements dans la table de BD.
- Vous pouvez effectuer une requête de la BD pour afficher certains enregistrements.
- Vous pouvez ajouter, modifier et supprimer des enregistrements BD et réenregistrer vos changements dans la BD. Voir Modifier une table de BD.

Pour ouvrir une StyleVision Power Stylesheet basée sur BD dans le Mode Authentic, cliquez sur **Authentic** | **Éditer les données de base de données**, et chercher la StyleVision Power Stylesheet requise.

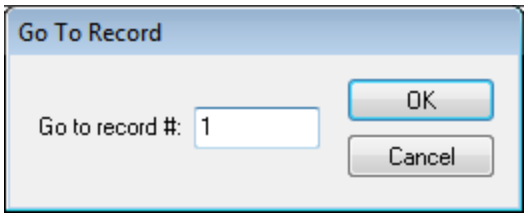
Note : Dans le Mode Authentic, les données provenant d'une base de données SQLite ne peuvent pas être éditées. Lorsque vous tentez d'enregistrer des données SQLite depuis le Mode Authentic, une fenêtre de messages vous informera de cette limitation.

11.3.3.1 Parcourir une table de BD

Les commandes permettant de parcourir des lignes de table de BD sont disponibles sous la forme de touches dans le document Authentic View. Généralement, un volet de navigation comportant quatre ou cinq touches accompagne chaque table de BD.



Les icônes fléchées sont, de gauche à droite, Aller au premier enregistrement dans la table de BD ; Aller à l'enregistrement précédent ; Ouvrir le dialogue Aller à l'enregistrement (*capture d'écran ci-dessous*) ; Aller à l'enregistrement suivant et Aller au dernier enregistrement.



Pour parcourir une table de BD, cliquer sur la touche requise.

Bases de données XML

Dans le cas des BD XML, comme les IBM DB2, une cellule (ou ligne) contient un seul document XML, donc une seule ligne est chargée dans le Authentic View à la fois. Pour charger un document XML qui se trouve dans une autre ligne, utiliser la commande de menu **Authentic | Sélectionner Nouvelle ligne avec les données XML pour l'édition**

11.3.3.2 Requêtes BD

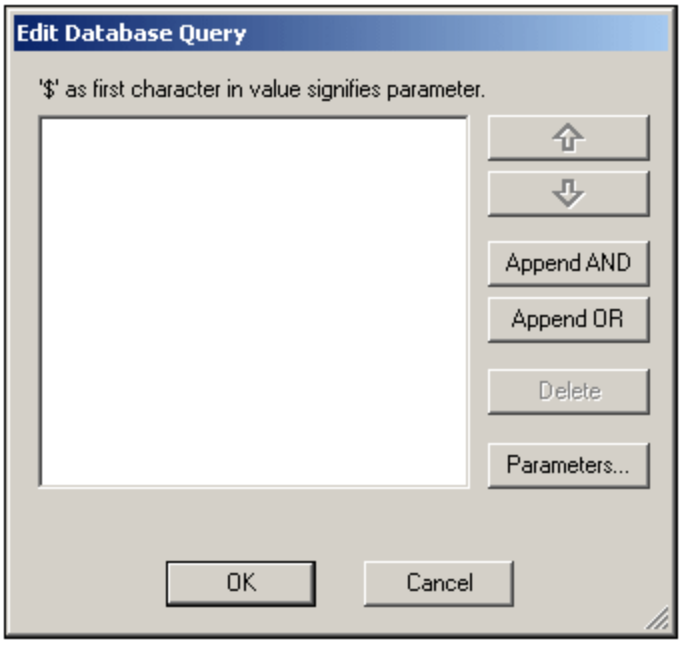
Une requête BD vous permet de requêter les enregistrements d'une table affichée dans le Authentic View. Une requête est effectuée pour une table individuelle et seule une requête peut être effectuée pour chaque table. Vous pouvez effectuer une requête à tout moment pendant l'édition. Si des modifications non enregistrées se trouvent dans votre document Authentic View au moment de la soumission de la requête, vous serez invité à enregistrer **tous** les changements effectués dans le document ou bien d'éliminer **tous** les changements. Veuillez noter que même les changements effectués dans d'autres tables seront enregistrés/éliminés. Une fois avoir soumis la requête, la table est rechargée à l'aide des conditions de requête.

Note : Si vous voyez apparaître un message vous indiquant que trop de tables sont ouvertes, vous pourrez réduire le nombre de tables ouvertes en utilisant une requête pour filtrer certaines des tables.

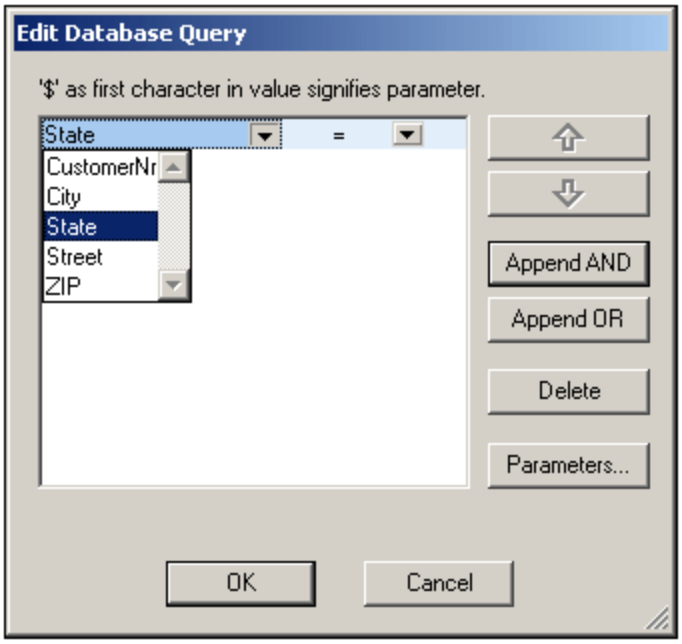
Pour créer et soumettre une requête :

1. Cliquer sur la touche Requête  pour la table désirée afin d'ouvrir le dialogue Éditer la requête de base de données (*voir capture d'écran*). Ce bouton apparaît généralement en haut de chaque table BD ou en-dessous. Si la touche Requête n'apparaît pas pour une table, c'est que le concepteur de la StyleVision Power Stylesheet n'a pas activé la fonction de requête BD pour cette table.

© 2016-2022 Altova GmbH



2. Cliquer sur la touche **Apposer ET** ou **Apposer OU**. Cela vous permettra d'apposer un critère vide pour la requête (affiché ci-dessous).



3. Saisir l'expression pour le critère. Une expression consiste en : (i) un nom de champ (disponible depuis la liste de choix associée) ; (ii) un opérateur (disponible depuis la liste de choix associée) ; et (iii) une valeur (à saisir directement). Pour plus de détails concernant la construction d'expressions voir la section Expressions dans les critères.
4. Si vous souhaitez ajouter encore un critère, cliquer sur la touche **Apposer ET** ou **Apposer OU** selon l'opérateur logique (ET et OU) que vous souhaitez utiliser pour joindre les deux critères. Ensuite,

ajouter le nouveau critère. Pour plus de détails concernant les opérateurs logiques, voir la section Réordonner les critères dans les requêtes BD.

Expressions dans les critères

Les expressions dans les critères de requête BD consistent en un nom de champ, un opérateur et une valeur. Les **noms de champ disponibles** sont les éléments enfants de la table de données de niveau supérieur sélectionnée ; les noms de ces champs sont recensés dans une liste de choix (*voir capture d'écran ci-dessus*). Les **opérateurs** que vous pouvez utiliser sont regroupés ci-dessous :

`=`	Égale à
`<>`	N'est pas égale à
`<`	Inférieur à
`<=`	Inférieur ou égale à
`>`	Supérieur à
`>=`	Supérieur ou égale à
`LIKE`	Phonétiquement semblable à
`NOT LIKE`	N'est pas phonétiquement semblable à
`IS NULL`	Est vide
`NOT NULL`	N'est pas vide

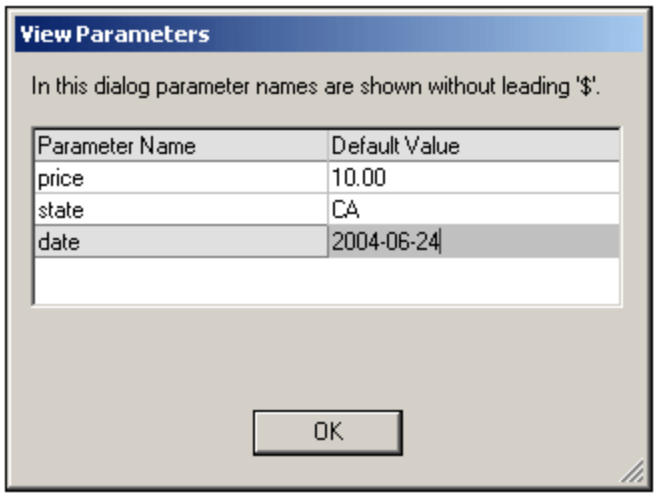
Si IS NULL ou NOT NULL a été sélectionné, le champ de valeur est désactivé. Les **Valeurs** doivent être saisies sans guillemets (ou tout autre délimiteur). Les valeurs doivent présenter le même formatage que celui du champ de BD correspondant ; sinon l'expression évaluera à `FALSE`. Par exemple, si un critère pour un champ de type de données `date` dans une BD MS Access a une expression `StartDate=25/05/2004`, l'expression évaluera à `FALSE` parce que le type de données `date` dans une BD MS Access a un format `YYYY-MM-DD`.

Utiliser des paramètres avec les requêtes BD

Vous pouvez saisir le nom d'un **paramètre** en tant que la valeur d'une expression lorsque vous créez des requêtes. Les paramètres sont des variables qui peuvent être utilisées au lieu des valeurs littérales dans des requêtes. Lorsque vous les saisissez dans une expression, sa valeur est utilisée dans l'expression. Les paramètres qui sont disponibles ont été définis par le concepteur de SPS dans la SPS et peuvent être consultés dans le dialogue Afficher paramètres (*voir capture d'écran ci-dessous*). Une valeur par défaut a été attribuée aux paramètres dans la SPS. Celle-ci peut être contournée en passant une valeur au paramètre par le biais de la ligne de commande (si et quand le document de sortie est compilé par le biais de la ligne de commande).

Afin de consulter les paramètres définis pour la SPS, cliquer sur la touche **Paramètres** dans le dialogue Éditer la requête de base de données. Le dialogue **Afficher les paramètres** s'ouvre (*voir capture d'écran*).

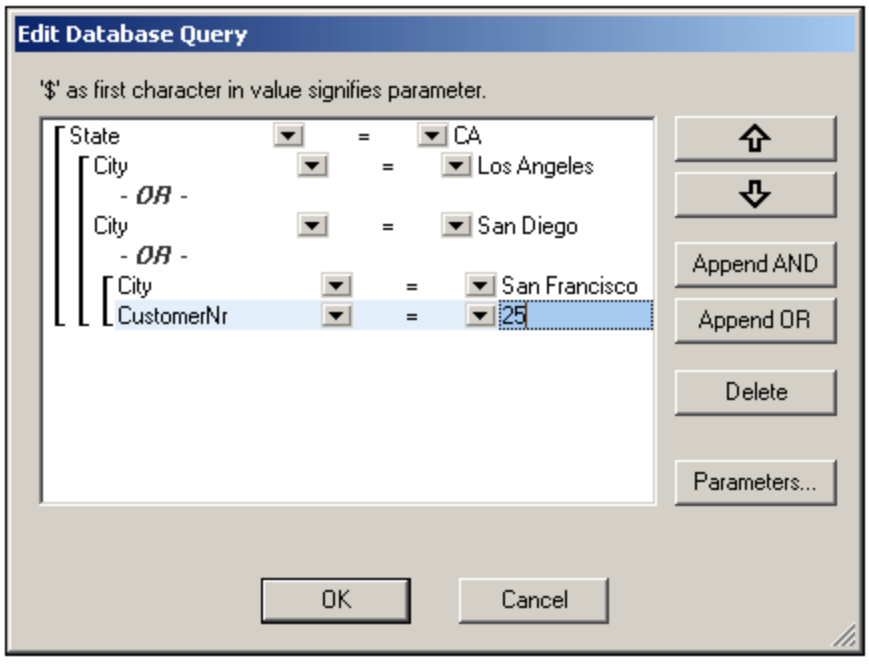
© 2016-2022 Altova GmbH



Le dialogue Afficher paramètres contient **tous** les paramètres qui ont été définis pour la feuille de style dans la SPS et les paramètres doivent être édités dans le design de la feuille de style.

Réordonner les critères dans les requêtes BD

La structure logique de la Requête de BD et les relations entre deux critères ou ensembles de critères est indiquée graphiquement. Chaque niveau de la structure logique est indiquée par un crochet. Deux critères ou ensembles de critères adjacents indiquent l'opérateur ET, alors que si deux critères sont séparés par le mot `OU`, alors l'opérateur OU sera indiqué. Les critères sont présentés en retrait, afin de permettre une lecture plus claire de la structure logique de la requête de BD.



La requête de BD affichée dans la capture d'écran ci-dessus peut être représentée dans le texte sous la forme suivante :

```
State=CA AND (City=Los Angeles OR City=San Diego OR (City=San Francisco AND
CustomerNr=25))
```

Vous pouvez réorganiser la requête de BD en déplaçant un critère ou un ensemble de critères vers le haut ou vers le bas par rapport aux autres critères dans la requête de BD. Pour déplacer un critère ou un ensemble de critères, procéder comme suit :

1. Sélectionner le critère en cliquant dessus, ou sélectionner un niveau entier en cliquant sur le crochet qui représente ce niveau.
2. Cliquer sur la touche Haut ou Bas dans le dialogue.

Les points suivants doivent être notés :

- Si le critère adjacent dans la direction du mouvement se trouve au même niveau, les deux critères échangent leur place.
- Un ensemble de critères (c.à.d. un critère dans un crochet) change de position dans le même niveau ; il ne change pas de niveau.
- Un critère individuel change de position dans le même niveau. Si le critère adjacent se trouve plus en avant ou en arrière (donc, pas au même niveau), alors le critère sélectionné se déplacera vers l'extérieur/intérieur, **un niveau à la fois**.

Pour supprimer un critère dans une requête de BD, sélectionner le critère et cliquer sur **Supprimer**.

Modifier une requête BD

Pour modifier une requête de BD:

1. Cliquer sur la touche Requête . Le dialogue Éditer la requête de base de données s'ouvre. Vous pouvez maintenant éditer les expressions dans un des critères recensés, ajouter le nouveau critère, réordonner le critère ou supprimer le critère dans la requête de BD.
2. Cliquer sur **OK**. Les données provenant de la BD sont automatiquement rechargées dans Authentic View pour réfléchir les modifications de la requête de BD.

11.3.3.3 Modifier une table de BD

Ajouter un enregistrement
Pour ajouter un enregistrement dans une table de BD :

1. Placer le curseur dans la ligne de table de BD et cliquer sur l'icône  (pour apposer une ligne) ou sur l'icône  (pour insérer une ligne). Cela crée un nouvel enregistrement dans le fichier temporaire XML.
2. Cliquer sur la commande **Fichier | Enregistrer** pour ajouter le nouvel enregistrement dans la BD. Dans Authentic View, une ligne pour le nouvel enregistrement est apposée à l'affichage de table de BD. L'item `AltovaRowStatus` pour cet enregistrement est défini sur `A` (pour Ajouter).

Lorsque vous saisissez des données pour le nouvel enregistrement, celui-ci sera saisi en gras et souligné. Cela vous permet de différencier entre les enregistrements ajoutés et les enregistrements existants ; si les enregistrements existants n'ont pas été formatés avec ces propriétés de formatage de texte. Les erreurs de type de données sont marquées par un affichage rouge.

© 2016-2022 Altova GmbH

Le nouvel enregistrement est ajouté à la BD lorsque vous cliquez sur **Fichier | Enregistrer**. Une fois qu'un nouvel enregistrement a été enregistré sur la BD, son champ `AltovaRowStatus` est initialisé (indiqué par `---`) et l'enregistrement est affiché dans Authentic View en tant qu'un enregistrement normal.

Modifier un enregistrement

Pour modifier un enregistrement, placez le curseur à l'endroit nécessaire dans la table de BD et éditez l'enregistrement comme requis. Si le nombre d'enregistrements affichés est limité, vous devrez éventuellement chercher l'enregistrement requis (voir Parcourir une table BD).

Lorsque vous modifiez un enregistrement, les entrées dans tous les champs de l'enregistrement sont soulignées et l'item `AltovaRowStatus` de toutes les instances primaires de cet enregistrement est configuré sur `U` (pour Updated (mis à jour)). Toutes les instances secondaires de cet enregistrement ont leur `AltovaRowStatus` configuré sur `u` (minuscule). Les instances primaires et secondaires d'un enregistrement sont définies par la structure de la BD et correspondent au schéma XML généré. Par exemple, si une table Adresse est contenue dans une table Client, alors la table Adresse peut se produire dans le Document de Design dans deux types d'instanciation : en tant que la table Adresse elle-même et dans le cadre des instanciations de la table Client. Quel que soit l'un des deux types qui est modifié, il s'agira donc du type qui a été modifié primairement. Les autres types (il peut y avoir plus qu'un seul autre type) sont les types secondaires. Les erreurs de type de données sont marquées par un affichage en rouge.

Les modifications sont enregistrées dans la BD en cliquant sur **Fichier | Enregistrer**. Après qu'un enregistrement modifié ait été enregistré sur la BD, son champ `AltovaRowStatus` est initialisé (indiqué avec `---`) et l'enregistrement est affiché dans Authentic View en tant qu'un enregistrement normal.

Notez les points suivants :

- Si un seul champ d'un enregistrement est modifié dans Authentic View, c'est tout l'enregistrement qui sera mis à jour lorsque les données seront enregistrées sur la BD.
- La valeur de date `0001-01-01` est définie en tant que valeur `NULL` pour certaines BD et peuvent déclencher un message d'erreur.

Supprimer un enregistrement

Pour supprimer un enregistrement :

1. Placer le curseur dans la ligne représentant l'enregistrement à supprimer et cliquer sur l'icône . L'enregistrement à supprimer est marqué par un texte barré. `AltovaRowStatus` est configuré comme suit : les instances primaires de l'enregistrement sont définies sur `D` ; les instances secondaires sur `d` ; et les enregistrements indirectement supprimés sur `X`. Les enregistrements indirectement supprimés sont des champs dans l'enregistrement supprimé qui sont conservés dans une table séparée. Par exemple, une table Adresse peut être incluse dans une table Client. Si un enregistrement Client est supprimé, son enregistrement Adresse correspondant sera supprimé indirectement. Si un enregistrement Adresse dans la table Client est supprimé, l'enregistrement Adresse dans la table Client sera supprimé primairement, mais le même enregistrement sera supprimé secondairement dans une table Adresse indépendante si cela a été instancié.
2. Cliquer sur **Fichier | Enregistrer** pour enregistrer des modifications dans la BD.

Note : Enregistrer les données sur la BD réinitialise la commande Annuler, vous ne pourrez donc pas annuler des actions qui ont été effectuées avant l'enregistrement.

11.3.4 Travailler avec des dates

Vous disposez de deux manières pour éditer les dates dans Authentic View:

- Les dates sont saisies ou modifiées à l'aide du Sélectionneur de date.
- Les dates sont saisies ou modifiées en saisissant la valeur.

La méthode que l'utilisateur d'Authentic View utilisera est définie dans la SPS. Les deux méthodes sont décrites dans les deux sous-sections de cette section.

Note sur les formats de date

Dans le document XML, les dates peuvent être stockées dans un des divers types de données. Chacun de ces types de ces types de données exige que la date soit stockée dans un format lexical particulier pour que le document XML soit valide. Par exemple, le types de données `xs:date` exige un format lexical de type `YYYY-MM-DD`. Si la date dans un nœud `xs:date` est saisie dans un autre format que celui-ci, le document XML sera invalide.

Afin de vous assurer que la date est saisie dans le format correct, le concepteur de la SPS peut inclure le Sélecteur de date graphique dans son design. Cela permet d'assurer que la date sélectionnée dans le Sélecteur de date est saisie dans le format graphique correct. S'il n'y a pas de Sélecteur de données, le Authentic View devra veiller à saisir la date dans le format lexical correct. Valider le document XML peut fournir des astuces utiles concernant le format lexical requis.

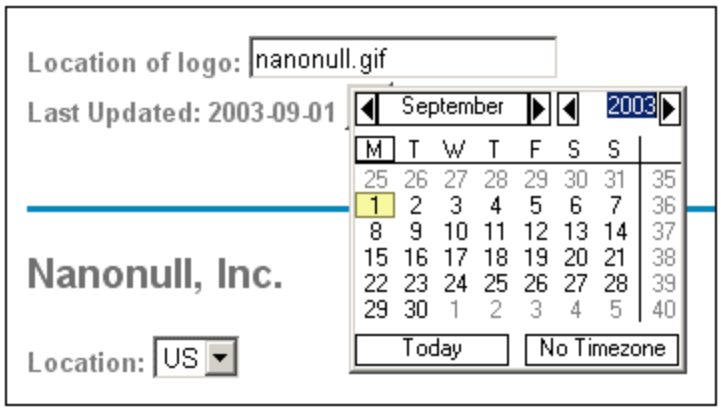
11.3.4.1 Sélectionneur de date

Le Sélectionneur de date est un calendrier graphique utilisé pour saisir des dates dans un format standard dans le document XML. Le fait de disposer d'un format standard est important pour le traitement des dates dans le document. L'icône du Sélectionneur de date apparaît près du champ de date qu'elle modifie (*voir capture d'écran*).



Afin d'afficher le Sélectionneur de date (*voir capture d'écran*), cliquer sur l'icône de Sélectionneur de date.

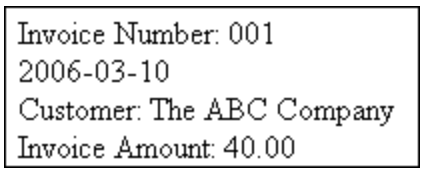
© 2016-2022 Altova GmbH



Pour sélectionner une date, cliquer sur la date, le mois, ou l'année désirée. La date est saisie dans le document XML et la date dans l'affichage est modifiée conformément. Vous pouvez aussi saisir un fuseau horaire le cas échéant.

11.3.4.2 Entrée de texte

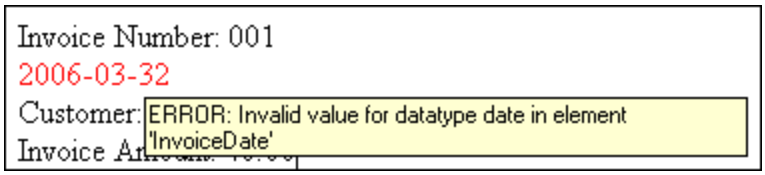
Pour les champs de date qui ne possèdent pas de Sélectionneur de date (*voir capture d'écran*), vous pouvez éditer la date directement en saisissant la nouvelle valeur.



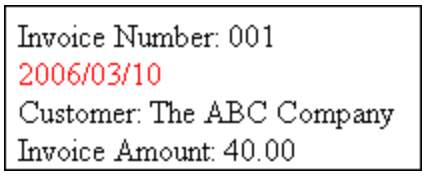
Erreurs

Les types d'erreur suivants seront marqués d'un indicateur :

- Si vous éditez une date et que vous la modifiez de manière à ce qu'elle sorte de la plage valide, la date passe au rouge pour vous avertir de l'erreur. Si vous placez le curseur de la souris sur la date invalide, un message d'erreur apparaît (*voir capture d'écran*).



- Si vous essayez de modifier le format de la date, la date passe en rouge pour vous avertir de l'erreur. (Dans la capture d'écran ci-dessous, les barres obliques sont utilisées au lieu des traits d'union).



11.3.5 Définir les entités

À propos des entités

Vous pouvez définir des entités à utiliser dans le Authentic View, que votre document soit basé sur un DTD ou un Schéma XML. Une fois définies, ces entités sont affichées dans l'Assistant à la saisie Entités et dans le sous-menu **Insérer entité** du menu contextuel. Lorsque vous double-cliquez sur une entité dans l'Assistant à la saisie Entités, cette entité est insérée à l'endroit du point d'insertion du curseur.

Une entité est utile si vous utilisez une chaîne de texte, un fragment XML, ou certaines des autres ressources externes dans plusieurs emplacements de votre document. Vous définissez l'entité, constituée en principe d'un nom bref qui symbolise les données requises dans le dialogue Définir les entités. Une fois avoir défini une entité, vous pourrez l'utiliser dans plusieurs endroits de votre document. Cela vous permet d'économiser du temps et facilite considérablement l'entretien.

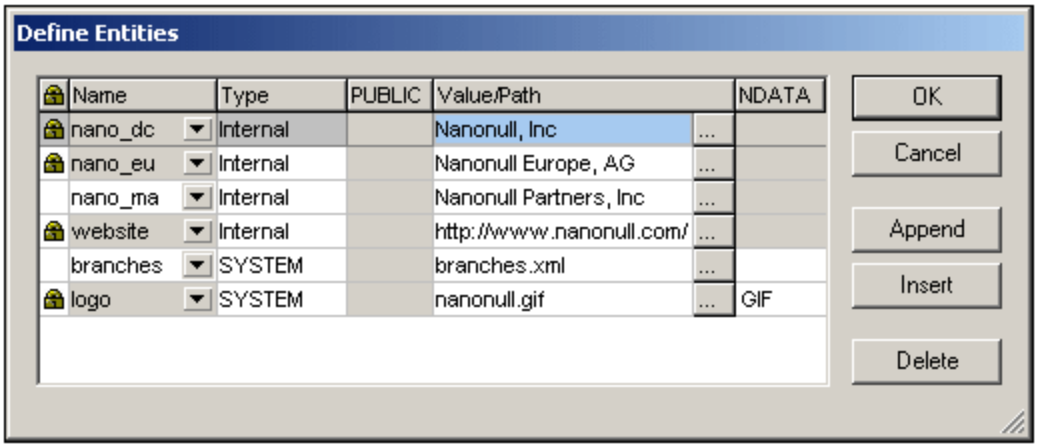
Types d'entités
Il existe deux types d'entités que vous pouvez utiliser dans votre document : une **entité parsée,** soit des données XML (soit une chaîne de texte ou un fragment d'un document XML), ou une **entité non parsée**, soit des données non-XML comme un fichier binaire (généralement un graphique, un son ou un objet multimédia). Chaque entité porte un nom et une valeur. Dans le cas des entités parsées, l'entité est un espace réservé pour les données XML. La valeur de l'entité est soit les données XML elles-mêmes ou une URI qui pointe vers un fichier `.xml` qui contient les données XML. Dans le cas des entités non parsées, la valeur de l'entité est une URI qui pointe vers le fichier de données non-XML.

Définir les entités

Pour définir une entité :

1. Cliquer sur **Authentic | Définir les entités XML**. Cela permet d'ouvrir le dialogue Définir les entités (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



2. Saisir le nom de votre entité dans le champ Nom. Il s'agit du nom qui apparaîtra dans l'Assistant à la saisie Entité.
3. Saisir le type d'entité depuis la liste déroulante dans le champ Type. Les types suivants sont possibles : une entité **Interne** est une entité pour laquelle le texte à utiliser est stocké dans le document XML lui-même. La sélection de **PUBLIC** ou de **SYSTEM** spécifie que la ressource est située en-dehors du fichier XML, et qu'elle sera située avec l'utilisation d'un identificateur public ou un identificateur de système respectivement. Un identificateur de système est une URI qui donne l'emplacement de la ressource. Un identificateur public est un identificateur indépendant de l'emplacement qui permet à certains processeurs d'identifier la ressource. Si vous spécifiez aussi bien un identificateur public que système, l'identificateur public résout à l'identificateur système, et l'identificateur système est utilisé.
4. Si vous avez utilisé PUBLIC en tant que Type, saisir l'identificateur public de votre ressource dans le champ PUBLIC. Si vous avez sélectionné Internal ou SYSTEM en tant que votre Type, le champ PUBLIC est désactivé.
5. Dans le champ Valeur/Chemin, vous pouvez saisir un des items suivants :

   - Si le type d'entité est Interne, saisir la chaîne de texte que vous souhaitez en tant que la valeur de votre entité. Ne pas saisir de guillemets pour délimiter la saisie. Les guillemets que vous saisissez seront traités en tant que partie de la chaîne de texte.
   - Si le type d'entité est SYSTEM, saisir l'URI de la ressource ou sélectionner une ressource sur votre réseau local en utilisant la touche Parcourir. Si la ressource contient des données parsées, il doit s'agir d'un fichier XML (il doit donc avoir une extension `.xml`). En alternative, la ressource peut être un fichier binaire, comme un fichier GIF.
   - Si le type d'entité est PUBLIC, vous devrez saisir en plus un identificateur dans ce champ.

6. L'entrée NDATA indique au processeur que cette entité ne doit pas être parsée mais envoyée au processeur approprié. Le champ NDATA doit donc contenir certaines valeurs pour indiquer que l'entité est une entité non-parsée.

Fonctions de dialogue

Le dialogue Définir entités vous permet d'effectuer les tâches suivantes :

- Apposer des entités
- Insérer des entités
- Supprimer des entités
- Trier des entités par la valeur alphabétique de toute colonne en cliquant sur l'en-tête de colonne ; cliquer une fois pour trier dans l'ordre ascendant, cliquer deux fois pour trier dans l'ordre descendant.
- Redimensionner la fenêtre de dialogue et la largeur des colonnes.

- Verrouillage. Une fois qu'une entité est utilisée dans le document XML, elle est verrouillée et ne peut pas être éditée dans le dialogue Définir entités. Les entités verrouillées sont indiquées par un symbole de cadenas dans la première colonne. Le verrouillage d'une entité permet de garantir que le document XML soit valide par rapport aux entités. (Le document serait invalide si une entité est référencée mais pas définie.)
- Les entrées double sont marquées.

Limitations des entités

- Une entité contenue dans une autre entité n'est pas résolue, soit dans le dialogue, le Authentic View, ou la sortie XSLT, et le caractère esperluette d'une telle entité est affiché dans sa forme échappée : `&`.
- Les entités non-parsées qui ne sont pas des fichiers d'image ne sont pas résolues dans le Authentic View. Si une image dans le design est définie pour lire une entité non parsée externe et dont l'URI est configurée pour être un nom d'entité (par exemple : `'logo'`), alors ce nom d'entité peut être défini dans le dialogue Définir entités (*capture d'écran ci-dessus*) en tant qu'une entité non-parsée externe avec une valeur qui résout à l'URI du fichier image (comme cela a été fait pour l'entité `logo` dans la capture d'écran ci-dessus).

11.3.6 Signatures XML

Une SPS peut être conçue avec une signature XML configurée pour le Authentic View. Lorsque les signatures XML sont activées dans la SPS, l'utilisateur du Authentic View peut signer numériquement le fichier Authentic XML par le biais de l'activation de la signature. Une fois que le document a été signé, toute modification entraînera une vérification de la signature. Lorsqu'un document Authentic XML signé est ouvert dans le Authentic View de tout produit Altova, le processus de vérification sera exécuté sur le document et le résultat de la vérification sera affiché dans une fenêtre.

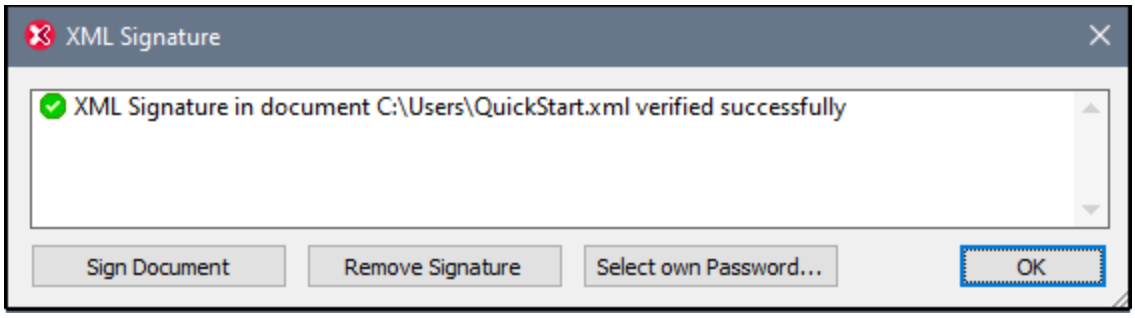
Note : les signatures XML peuvent être utilisées et seront vérifiées, dans le Mode Authentic des éditions Enterprise et Professional des produits Altova suivants : Authentic Desktop, Authentic Browser, XMLSpy et StyleVision.

Actions de signature XML

Les actions d'utilisateur du Mode Authentic suivantes sont possibles :

- *Choisir le certificat/mot de passe :* les signatures sont authentifiées avec un certificat ou un mot de passe. L'objet de l'authentification (certificat ou mot de passe) est exigé tout d'abord lorsque la signature est créée et à nouveau lorsqu'il est vérifié. Si un document XML Authentic a une SPS avec activation de la signature qui lui est attribuée, la SPS peut spécifier un certificat ou un mot de passe par défaut pour la signature. Qu'un certificat ou un mot de passe par défaut ait été spécifié ou pas, la signature peut être configurée pour permettre à l'utilisateur du Authentic View de sélectionner son propre certificat ou un mot de passe. L'utilisateur du Authentic View peut y recourir à tout moment dans le dialogue Signature XML (*capture d'écran ci-dessous*). La sélection d'un certificat ou d'un mot de passe personnel annule le certificat ou le mot de passe par défaut. Le certificat ou le mot de passe personnel est stocké dans la mémoire et est utilisé pour la session actuelle. Si, après qu'un certificat ou un mot de passe personnel ait été sélectionné, l'utilisateur du Authentic View ferme le fichier ou l'application, la SPS repasse à ses paramètres par défaut pour le certificat ou le mot de passe.

© 2016-2022 Altova GmbH



- *Signer le document:* le document XML Authentic peut être signé automatiquement ou manuellement. La signature automatique aura été spécifiée dans la configuration de la signature par le concepteur de la SPS et entraîne la signature automatique du document XML Authentic lors de l'enregistrement. Si l'option de signature automatique n'a pas été activée, le document peut être signé manuellement. Pour ce faire, cliquer sur l'icône dans la barre d'outils Signature XML ou avec la commande **Authentic | Signature XML**, et, dans le dialogue Signature XML qui apparaît (*capture d'écran ci-dessus*), cliquer sur la touche **Signer Document**. Veuillez noter que la signature du document avec une signature intégrée nécessiterait que le schéma permette l'élément `Signature` en tant que le dernier élément enfant de l'élément racine (document). Sinon, le document sera invalide par rapport au schéma. Lors de la signature du document, l'objet d'authentification et le placement de la signature sont déterminés conformément à la configuration de la signature. Vous devez vous assurer que vous avez accès à l'information d'authentification. Pour plus d'information à ce propos, consulter votre concepteur de SPS.

- *Vérifier le document XML Authentic :* si une les Signatures XML sont activées dans la SPS, le processus de vérification sera exécuté sur la signature à chaque fois que le document XML du Authentic View est chargé. Si le mot de passe ou l'information de la clé de certificat n'est pas enregistrée avec la SPS et la signature, respectivement, l'utilisateur du Authentic View sera invité à saisir le mot de passe ou à sélectionner un certificat pour la vérification. Veuillez noter que si une signature intégrée est générée, elle sera enregistrée avec le fichier XML lorsque celui-ci est enregistré. La signature générée doit être supprimée explicitement (par le biais de la touche **Supprimer la signature** du dialogue Signature XML ; *voir capture d'écran ci-dessus*) si vous ne souhaitez pas l'enregistrer avec le fichier XML. De même, si une signature détachée est générée, elle aussi doit être supprimée si cela n'est pas requis.

11.3.7 Images dans le Mode Authentic

Authentic View vous permet de spécifier des images qui seront utilisées dans le document de sortie final (HTML, RTF, PDF et Word 2007). Vous devriez noter que certains formats d'image ne sont pas toujours pris en charge dans certains formats ou certaines applications. Par exemple, le format SVG est pris en charge dans PDF, mais pas dans RTF et nécessiterait un add-on de navigateur pour pouvoir être consultés dans HTML. Ainsi, lorsque vous sélectionnez un format d'image, veuillez vous assurer de sélectionner un format qui est pris en charge dans les formats de sortie de votre document. La plupart des formats d'image sont pris en charge sur tous les formats de sortie (*voir la liste ci-dessous*).

Authentic View est basé sur Internet Explorer, et est capable d'afficher la plupart des formats d'image que votre version d'Internet Explorer peut afficher. Les formats d'image utilisés communs suivants sont pris en charge :

- GIF
- JPG

- PNG
- BMP
- WMF (Microsoft Windows Metafile)
- EMF (Enhanced Metafile)
- SVG (uniquement pour la sortie PDF)

Chemins relatifs

Les chemins relatifs sont résolus relativement au fichier SPS.

11.3.8 Séquences de touche dans le Mode Authentic

Touche Entrée

Dans le Authentic View la touche **Entrée** est utilisée pour ajouter des éléments supplémentaires lorsqu'elle se trouve dans certains emplacements de curseur. Par exemple, si le chapitre d'un livre peut (conformément au schéma) contenir certains paragraphes, une pression sur **Entrée** dans le texte du paragraphe entraîne l'ajout immédiat d'un nouveau paragraphe après le paragraphe actuel. Si un chapitre peut contenir un titre et plusieurs paragraphes, appuyer **Entrée** dans le chapitre, mais en dehors de tout élément de paragraphe (y compris dans l'élément du titre) entraînera l'ajout d'un nouveau paragraphe après le chapitre actuel (dans le cas où plusieurs chapitres sont autorisés par le schéma).

Note : la touche **Entrée ne permet pas** d'insérer une nouvelle ligne. Cela est le cas lorsque le curseur se trouvant dans un nœud de texte, comme un paragraphe.

Utiliser le clavier

Le clavier peut être utilisé de la manière standard, pour la saisie et la navigation. Veuillez noter les points spéciaux suivants :

- La touche **Tab** déplace le curseur vers l'avant, s'arrêtant avant et après les nœuds, et soulignant les contenus de nœud ; elle passe au-dessus des contenus statiques.
- Les hyperliens `add...` et `add Node` sont considérés comme des contenus de nœud et sont soulignés lorsque vous les parcourez avec la touche Tab. Ils peuvent être activés soit en appuyant sur la barre espace soit sur la clé **Entrée**.

© 2016-2022 Altova GmbH

11.4 Authentic Scripting

La fonction **Authentic Scripting** permet une plus grande flexibilité et interactivité dans les designs SPS. Ces designs peuvent être crées ou édités dans les éditions StyleVision Enterprise et Professional, et peuvent être consultés dans le Mode Authentic des éditions Enterprise et Professional des produits Altova.

Vous trouverez une liste complète des prises en charge pour cette fonction dans les produits Altova dans le tableau ci-dessous. Veuillez noter néanmoins que dans la version de confiance d'Authentic Browser plug-in, le scripting interne est désactivé pour des raisons de sécurité.

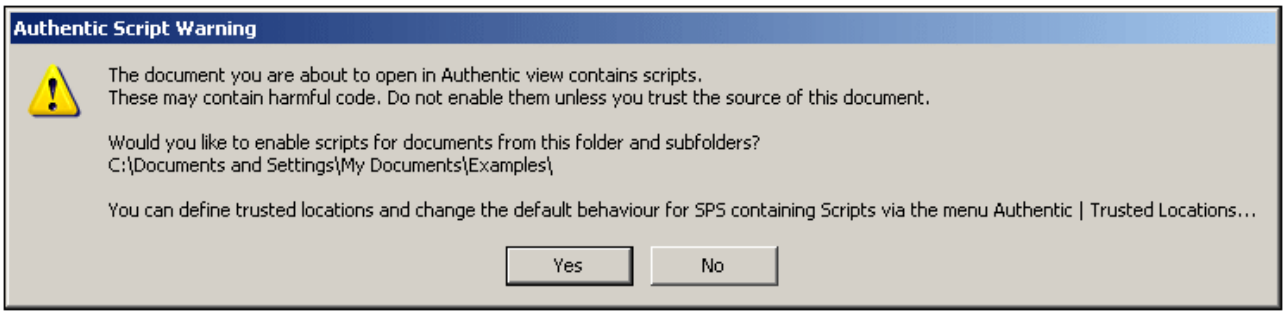
Produit Altova	Création de scripts Authentic	Activation de scripts Authentic
StyleVision Enterprise	Oui	Oui
StyleVision Professional	Oui	Oui
StyleVision Standard *	Non	Non
XMLSpy Enterprise	Non	Oui
XMLSpy Professional	Non	Oui
XMLSpy Standard	Non	Non
AuthenticDesktop Enterprise	Non	Oui
Authentic Browser Plug-in Enterprise Trusted **	Non	Oui
Authentic Browser Plug-in Enterprise Untrusted	Non	Oui

* *Pas d'AuthenticView*
** *Designs scripted affichés. Aucune exécution macro interne ou gestion d'événement. Événements externes déclenchés.*

Les scripts Authentic se comportent de la même manière que dans tous les produits Altova, aucun code spécifique au produit ou paramètres n'est donc nécessaire.

Dialogue d'avertissement de script Authentic

Si un fichier PXF ou un fichier XML lié à une SPS contient un script et que le fichier est ouvert ou qu'il passe au Mode Authentic, un dialogue d'avertissement (*capture d'écran ci-dessous*) s'ouvre.



Vous pouvez choisir parmi une des options suivantes :

- Cliquer sur **Oui** pour ajouter le dossier contenant le fichier vers la liste d'emplacement de confiance pour les scripts Authentic. Par conséquent, tous les fichiers contenus dans le dossier de confiance seront ouverts dans le Mode Authentic sans l'affichage préalable d'un dialogue d'avertissement. La liste des Emplacements de confiance peut être accédée par la commande de menu **Authentic | Emplacements de confiance**, puis modifié.
- Cliquer sur **Non** pour ne pas ajouter le dossier contenant le fichier dans la liste des Emplacements de confiance. Le fichier sera affiché dans le Mode Authentic et les scripts seront désactivés. Le dialogue d'Avertissement de script Authentic apparaîtra à chaque fois que ce fichier est ouvert dans le Mode Authentic. Afin d'ajouter ultérieurement le dossier du fichier dans la liste des Emplacements de confiance, ouvrir le dialogue Emplacements de confiance avec la commande de menu **Authentic | Emplacements de confiance**, et ajouter le dossier ou le modifier le cas échéant.

Pour une description du dialogue Emplacements de confiance, voir la description de la commande de menu **Authentic | Emplacements de confiance** dans la Référence de l'utilisateur.

Note : Lorsque XMLSpy est accédé par le biais de son interface COM (voir Référence du programmeur pour voir comment procéder), **le contrôle de sécurité n'est pas effectué** et le **dialogue Avertissement de script Authentic n'est pas affiché**.

Comment fonctionne Authentic Scripting

Le designer du design SPS peut utiliser l'Authentic Scripting de deux manières pour rendre les documents Authentic interactifs :

- En attribuant des scripts pour les actions définies par l'utilisateur (macros) afin de concevoir des éléments, des boutons de barre d'outils et des items de menu contextuel.
- En ajoutant à l'événement de design des handlers qui réagissent aux événements du Mode Authentic.

Tout le scripting nécessaire pour rendre les documents Authentic interactifs est réalisé dans la GUI de StyleVision (éditions Enterprise et Professional). Les formulaires, les macros et les gestionnaires d'événement sont créés au sein de l'interface de l'Éditeur de Scripting de StyleVision et ces scripts sont enregistrés dans la SPS. Ensuite, dans le Mode Design de StyleVision, les scripts sauvegardés sont attribués aux éléments de design, aux boutons de barre d'outils et aux menus contextuels. Lorsqu'un document XML basé sur la SPS est ouvert dans un produit Altova qui prend en charge l'Authentic Scripting (*voir table ci-dessus*), le document présentera la flexibilité et l'interactivité supplémentaire créée pour lui.

Documentation pour Authentic Scripting

La documentation pour l'Authentic Scripting est disponible dans la documentation de StyleVision. Elle peut être consultée en ligne dans la page de Documentation de produit du site Internet Altova.

© 2016-2022 Altova GmbH

12 HTML et CSS

XMLSpy propose des fonctions d'édition intelligentes pour les documents HTML et CSS. Ces deux types de document peuvent être édités dans le Mode Texte, et le document HTML actif peut être prévisualisé dans le Mode Navigateur.

Les fonctions d'édition intelligentes de chaque type de document sont décrites séparément dans les sous-sections de cette section : HTML et CSS.

12.1 HTML

Les documents HTML peuvent être édités dans le Mode Texte, et la page éditée peut ensuite être consultée immédiatement dans le Mode Navigateur. Le Mode Texte propose tout une série de fonctions d'édition HTML très utiles. Elles sont décrites en détails dans le Mode Texte, mais les fonctions principales, ainsi que les options spécifiques à HTML sont regroupées ci-dessous.

Niveau de prise en charge

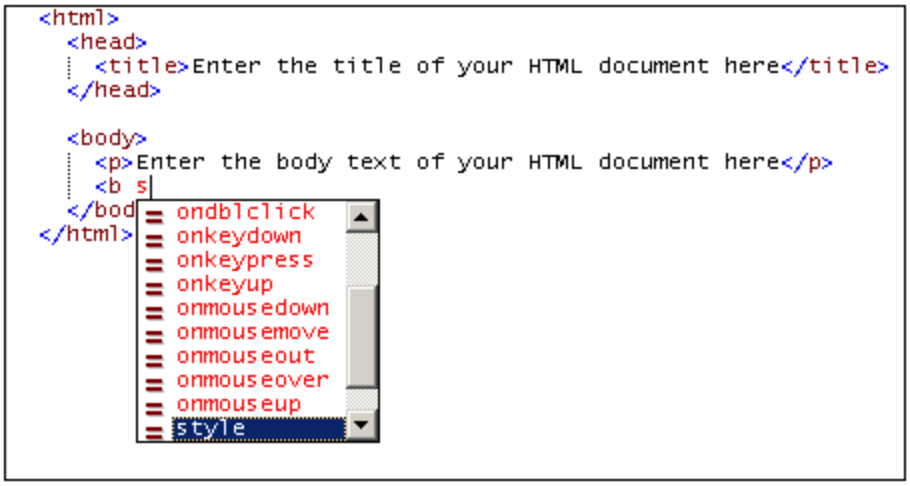
XMLSpy prend en charge HTML 4.0 et HTML 5.0. Les assistants à la saisie et l'édition intelligente sont disponibles pour les versions HTML respectives. Ces fonctions sont décrites ci-dessous.

Assistants à la saisie

Les Assistants à la saisie Éléments, Attributs et Entités sont disponibles lorsqu'un document HTML est actif. Les Assistants à la saisie sont sensibles au contexte, les items affichés dans les assistants à la saisie sont ceux disponibles à l'emplacement du curseur actuel. Utiliser les assistants à la saisie HTML tel que décrit dans le Mode Texte.

Saisie automatique

Lorsque vous saisissez un texte de balisé dans votre document HTML, XMLSpy vous propose une assistance de saisie automatique. Une fenêtre popup contenant une liste de tous les nœuds disponibles au niveau d'insertion du curseur s'affiche. Lorsque vous effectuez votre saisie, la sélection passe automatiquement à la correspondance la plus proche de la liste (*voir capture d'écran ci-dessous*). Cliquer sur l'item sélectionné pour l'insérer dans le point d'insertion du curseur.



Le remplissage automatique pour les éléments apparaît lorsque vous saisissez le crochet gauche de la balise de nœud. Lorsque la balise de démarrage d'un nœud d'élément est saisie dans le document, la balise de fin est aussi insérée automatiquement. Cela garantie une bonne formation.

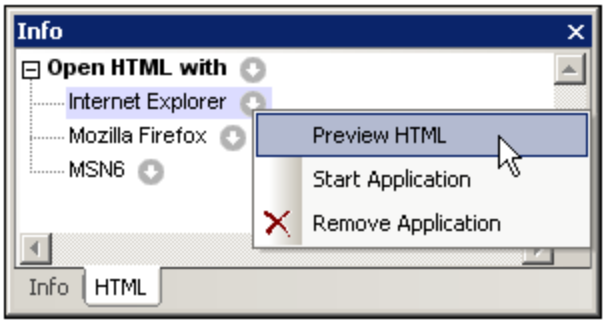
Le remplissage automatique pour les attributs apparaît lorsque vous saisissez un espace après le nom de l'élément dans une balise de démarrage. Lorsque vous cliquez sur le nom d'un attribut dans la fenêtre pop up du remplissage automatique, l'attribut est saisi avec des guillemets et le curseur est placé entre les guillemets.

© 2016-2022 Altova GmbH

Les assistants à la saisie Entité contiennent des entités de caractère provenant des ensembles d'entité HTML 4.0 et HTML 5.0, Latin-1, caractères spéciaux et symboles.

Fenêtre Info HTML

La fenêtre info HTML (*capture d'écran ci-dessous*) rassemble dans une liste les applications qui peuvent être utilisées pour accéder rapidement au fichier HTML actif. Par exemple, si un fichier HTML est activé dans XMLSpy, double cliquer l'item Mozilla Firefox dans la fenêtre Info HTML pour lancer une instance de Mozilla Firefox et charger le document HTML actif s'y trouvant.



Veuillez noter les points suivants :

- L'icône située à droite de l'item *Ouvrir HTML avec* permet d'ajouter des applications à la liste *Ouvrir HTML*. Tous les navigateurs installés sur le système ou toutes les autres applications (comme un éditeur de texte), peuvent être ajoutés par le biais des commandes de menu accessibles par l'icône *Ouvrir HTML avec*. Les applications associées sont généralement des applications de navigateur ou d'éditeur.
- Une fois qu'une application a été ajoutée à la liste *Ouvrir HTML* (sauf lorsqu'elles ont été ajoutées avec la commande **Ajouter navigateurs installés**), vous pourrez la renommer dans la liste *Ouvrir HTML* en la sélectionnant, en appuyant sur **F2** et en modifiant le nom.
- Les icônes situées à droite de chaque application contenue dans la liste *Ouvrir HTML* ouvre chacune un menu contenant des commandes pour : (i) ouvrir l'application ; (ii) ouvrir l'application et charger le fichier HTML lié ; (iii) supprimer l'application de la liste. Double-cliquer sur le nom de l'application pour ouvrir le fichier HTML lié dans cette application.
- Les applications qui ont été ajoutées ou supprimées de la liste *Ouvrir HTML* sont aussi ajoutées à ou supprimées de la fenêtre Info CSS.

Attribuer un DTD

Pour les documents XHTML, un DTD ou un Schéma XML peut être attribué par le biais du menu **DTD/Schéma**, qui vous permet de parcourir le DTD ou le fichier de schéma nécessaire. Un document XHTML peut être édité exactement comme un document XML.

Commandes Mode Navigateur

Les commandes du Mode Navigateur sont disponibles dans le menu **Navigateur**.

12.2 CSS

Les documents CSS peuvent être édités à l'aide des fonctions d'édition intelligentes du Mode Texte. Ces fonctions, telles qu'elles s'appliquent à l'édition des documents CSS, sont décrites ci-dessous.

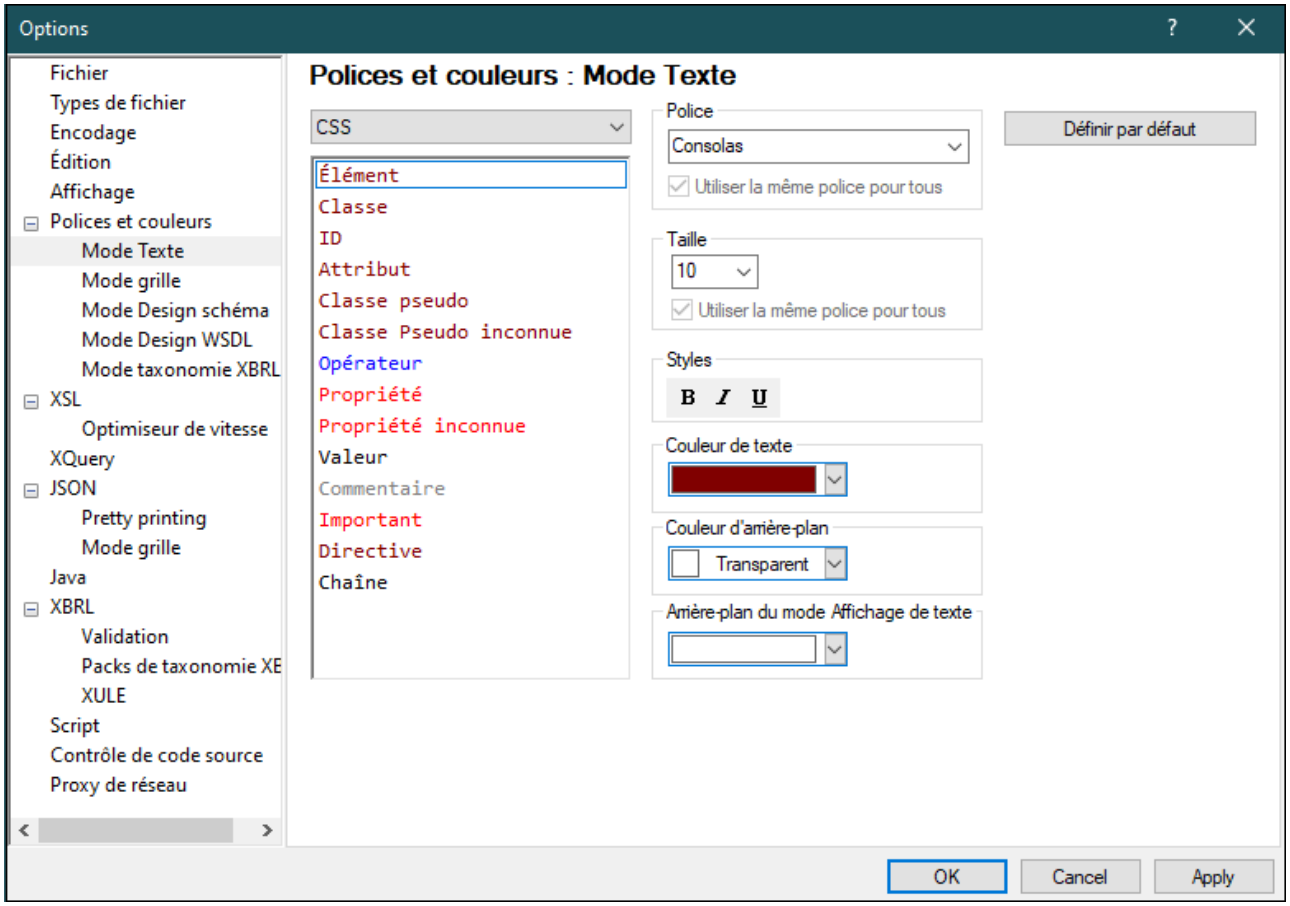
Coloration de la syntaxe

Une règle CSS consiste en un sélecteur, en une ou plusieurs propriétés et les valeurs de ces propriétés. Ces trois composants peuvent être divisées en plusieurs sous-catégories plus spécifiques ; par exemple, un sélecteur peut être une classe, une pseudo-classe, une ID, un élément ou un attribut. De plus, un document CSS peut contenir d'autres items que des règles : par exemple, des commentaires. Dans le Mode Texte, chacun des items de catégorie peut être affiché dans une couleur différente (*capture d'écran ci-dessous*) selon les paramètres que vous avez réglés dans le dialogue Options (*voir ci-dessous*).



Vous pouvez définir les couleurs des composants CSS variés dans la section Polices du dialogue Options (*capture d'écran ci-dessous*). Dans la liste de choix en haut à gauche, choisissez CSS, puis sélectionnez la couleur nécessaire (dans le panneau Styles) pour chaque item CSS.

© 2016-2022 Altova GmbH



Pliage de source

Le pliage de source est la capacité d'agrandir et de réduire chaque règle CSS, indiquée dans la marge de pliage de source par un signe +/−. La marge peut être activée et désactivée dans le dialogue Paramètres de Mode Texte. Lorsqu'une règle est réduite, cela est indiqué visuellement par des points de suspension. En passant avec la souris au-dessus de ces points, le contenu de la règle réduite est affiché dans une fenêtre popup. Si le contenu est trop grand pour une fenêtre popup, cela est indiqué par des points de suspension en bas de la fenêtre popup.

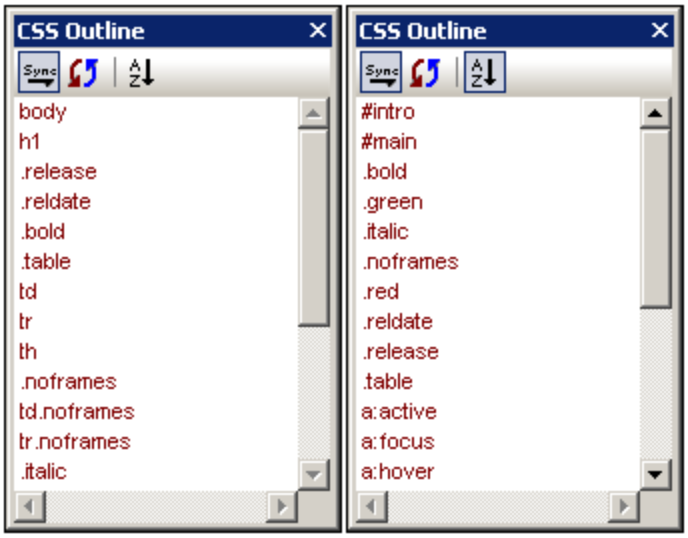
L'icône **Basculer tous les plis** dans la barre d'outils Texte active **toutes** les règles dans leur forme agrandie ou réduit toutes les règles à l'élément de document de niveau supérieur.

Note : la paire d'accolades qui délimitent une règle (*capture d'écran ci-dessus*) passe en gras lorsque le curseur est placé soit avant soit après une des accolades. Cela indique clairement où la définition d'une règle particulière commence et se termine.

Bordure CSS

L'assistant à la saisie de la bordure CSS (*capture d'écran ci-dessous*) propose une bordure en termes de ses sélecteurs. Cliquer sur un sélecteur dans la Bordure CSS pour la marquer dans le document. Dans la capture d'écran en bas à gauche, les sélecteurs ne sont pas triés et sont rassemblés dans l'ordre dans lequel ils apparaissent dans le document. Dans la capture d'écran de droite, la fonction de Tri Alphabétique a été activée (à l'aide de l'icône de la barre outils), et les sélecteurs sont triés alphabétiquement.

Vous devriez noter les points suivants : (i) Afin d'évaluer l'ordre alphabétique des sélecteurs, toutes les parties du sélecteur sont considérées, y compris les caractères point, barre oblique et double point ; (ii) Si le document CSS contient plusieurs sélecteurs regroupés pour définir une seule règle (par ex. `h4, h5, h6 {...}`), chaque sélecteur dans le groupe sera rassemblé séparément.



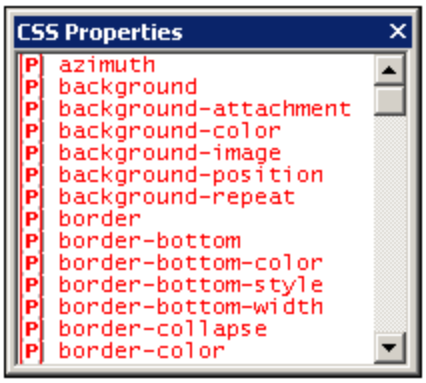
Les icônes dans la barre outils de l'assistant à la saisie Bordure CSS, de gauche à droite, ont la fonction suivante :

	Active et désactive automatiquement la synchronisation (avec le document). Lorsque l'auto-synchronisation est allumée, les sélecteurs sont saisis dans l'assistant à la saisie pendant que vous les saisissez dans le document.
	Synchronise l'assistant à la saisie avec l'état actuel du document.
	Active et désactive le tri par ordre alphabétique. Lorsqu'il est désactivé, les sélecteurs sont regroupés dans leur ordre d'apparence dans le document. Lorsqu'ils sont triés alphabétiquement, les sélecteurs ID apparaissent en premier parce qu'ils sont préfacés par un dièse (par ex. `#intro`).

Assistant à la saisie Propriétés

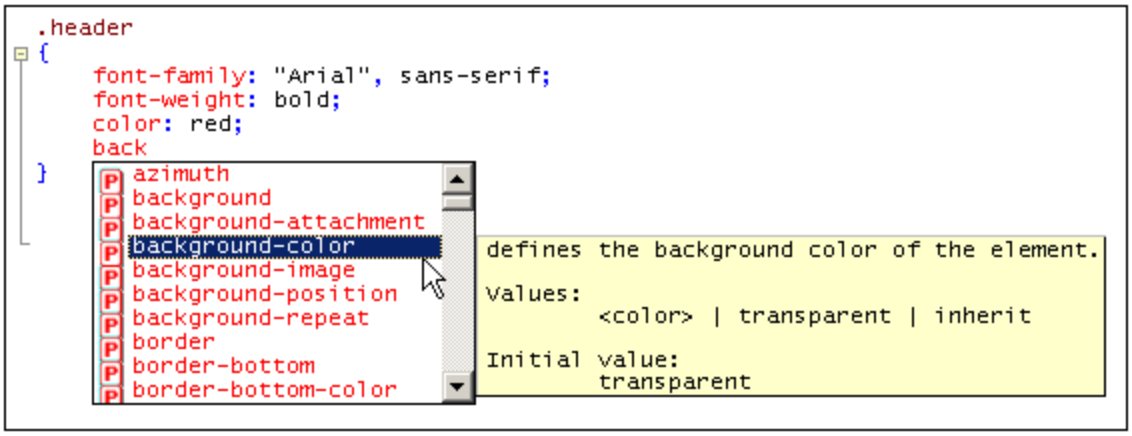
L'assistant à la saisie Propriétés (*capture d'écran ci-dessous*) fournit une liste de toutes les propriétés CSS, classées par ordre alphabétique. Une propriété peut être insérée au niveau d'insertion du curseur en double-cliquant la propriété.

© 2016-2022 Altova GmbH



Saisie automatique des propriétés et info-bulles pour les propriétés

Lorsque vous commencez à saisir le nom d'une propriété, XMLSpy vous propose une liste de propriétés commençant avec les lettres que vous avez déjà saisies (*capture d'écran ci-dessous*). En alternative, vous pouvez placer le curseur à tout autre endroit dans un nom de propriété et appuyer sur **Ctrl+Espace** pour afficher la liste des propriétés CSS.



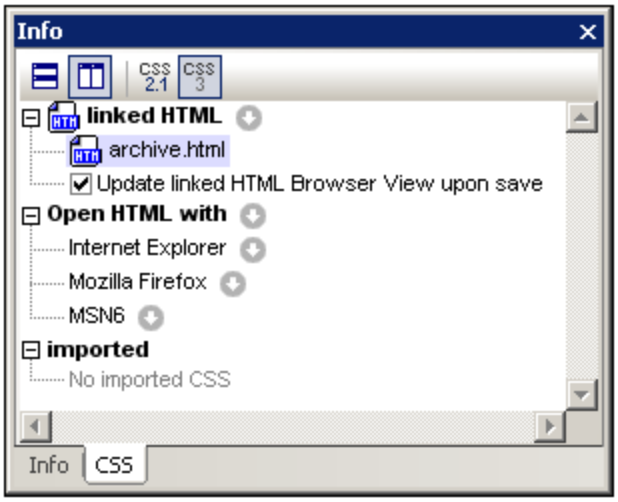
Vous pouvez consulter une info-bulle contenant une définition d'une propriété et ses valeurs possibles en faisant défiler la liste ou en parcourant la liste avec les touches fléchées Haut et bas de votre clavier. L'info-bulle de la propriété marquée s'affiche. Pour insérer une propriété, appuyez soit **Entrer** lorsqu'elle est sélectionnée, soit cliquer dessus.

Fenêtre Info CSS

Lorsqu'un fichier CSS est actif, la fenêtre Info CSS (*capture d'écran ci-dessous*) est activée. La fenêtre Info CSS fournit les fonctions suivantes :

- Elle vous permet de passer entre CSS 2.1 et CSS 3.0. Les assistant à la saisie et les fonctions d'édition intelligentes de la GUI seront activées selon la version CSS sélectionnée dans la barre outils de la Fenêtre Info.
- Elle permet de lier le fichier le fichier CSS à un fichier HTML. Cette fonction vous permet de modifier le document CSS et de consulter immédiatement les effets des modifications. De plus, le fichier HTML

lié peut être ouvert dans plusieurs navigateurs avec la fenêtre Info CSS, permettant donc de consulter dans plusieurs navigateurs des changements dans le document CSS.
- La fenêtre Info CSS recense les feuilles de style CSS importées, vous conférant donc un aperçu de la structure d'importation de la feuille de style CSS active.



Veuillez noter les points suivants :

- La barre outils de la fenêtre Info contient des icônes pour CSS 2.1 et CSS 3.0. Choisir la version que vous souhaitez pour pouvoir changer d'assistant à la saisie et les fonctions d'édition intelligente pour la version CSS sélectionnée.
- Seul un fichier HTML peut être lié au document CSS actif. Pour ce faire, cliquer sur l'icône à droite de l'item *HTML lié*, puis choisir la commande **Définir le lien HTML** et chercher le fichier HTML requis. Le fichier HTML lié sera regroupé sous l'item *HTML lié* dans la fenêtre Info (*voir capture d'écran ci-dessus*). La création de ce lien ne modifie par le document CSS ou le document HTML d'aucune manière. Le lien sert à définir un fichier HTML auquel le document CSS actif peut être appliqué pour les tests.
- Double-cliquer la liste du fichier HTML lié pour ouvrir le fichier HTML dans XMLSpy.
- L'icône de la barre outils vous permet de disposer en mosaïque horizontale et verticale le document CSS et le fichier HTML.
- Lorsque des changements du document CSS sont sauvegardés, le fichier HTML qui est ouvert dans XMLSpy peut être mis à jour automatiquement. Pour activer ces mises à jour automatiques, cocher la case *Mettre à jour le navigateur HTML lié*. Veuillez noter que ces mises à jour n'auront lieu que si le fichier HTML contient une référence au document CSS édité.
- Pour modifier le fichier lié HTML, sélectionner un autre fichier HTML avec la commande **Définir le lien HTML**.
- Pour supprimer le lien vers le fichier HTML, cliquer sur l'icône située à droite de l'item *HTML lié* et choisir la commande **Supprimer lien**.
- L'icône située à droite de l'item *Ouvrir HTML avec* permet d'ajouter des applications à la liste *Ouvrir HTML avec*. Tous les navigateurs installés sur le système ou toute autre application (comme un éditeur de texte), peut être ajouté avec les commandes de menu accédées depuis l'icône *Ouvrir HTML avec*. Les applications associées sont généralement des applications de navigateur ou d'éditeur.
- Une fois qu'une application a été ajoutée à la liste *Ouvrir HTML avec* (sauf lorsqu'elles sont ajoutées avec la commande **Ajouter les navigateurs installés**), son nom dans la liste *Ouvrir HTML avec* peut être changé en la sélectionnant, en appuyant sur **F2**, et en éditant le nom.
- Les icônes situées à droite de chaque application regroupées dans la liste *Ouvrir HTML avec* permettent chacune d'ouvrir un menu contenant des commandes pour : (i) ouvrir l'application; (ii) ouvrir l'application et charger le fichier HTML lié ; (iii) supprimer l'application de la liste. Double-cliquer sur un nom d'application pour ouvrir le fichier HTML lié dans cette application.

© 2016-2022 Altova GmbH

- Les applications ajoutées à ou supprimées de la liste *Ouvrir HTML avec* sont aussi ajoutés ou supprimés de la fenêtre Info HTML.
- L'item *Importé* affiche une liste des fichiers CSS importés par le document CSS actif.

13 JSON, Schéma JSON

JSON (*JavaScript Object Notation*) est un format de stockage et d'échange de données qui utilise la syntaxe JavaScript, et, comme le format XML, est un format de texte uniquement lisible par les humains. Puisque le texte JSON peut être lu et utilisé par tout langage de programmation, il est largement utilisé en tant que format d'échange de données, en particulier sur le web.

En tant que partie de sa fonction d'IDE, XMLSpy propose une prise en charge des documents de données JSON (documents d'instance) pour l'édition et la validation et pour la création de documents de Schéma JSON syntaxiquement et lexicalement corrects.

XMLSpy propose également une prise en charge pour Avro et le schéma Avro.

JSON5

JSON5 est une extension de JSON qui ajoute certaines extensions ECMAScript 5 (voir json5.org pour plus d'informations). JSON5 est un sous-ensemble strict de JavaScript, n'ajoute aucun nouveau type de données aux types JSON existants et fonctionne avec tous les contenus JSON existants.

Toutes les fonctions XMLSpy qui sont disponibles pour les documents d'instance JSON sont aussi disponibles pour les documents d'instance JSON5. Néanmoins, veuillez noter les différences principales entre JSON5 et JSON, et de la manière dont XMLSpy gère les deux formats :

- JSON5 n'est pas un successeur officiel de JSON. Il utilise donc sa propre extension de fichier : **`json5`**.
- Par défaut, XMLSpy reconnaît les fichiers comportant l'extension de fichier **`.json`** comme des documents d'instance JSON, et ceux comportant l'extension de fichier **`.json5`** comme des documents d'instance JSON5.
- Les schémas JSON5 peuvent être validés par rapport aux schémas JSON. Les documents d'instance JSON, qui peuvent être des représentations des instances Avro, peuvent être validées par rapport aux schémas JSON (instances JSON) ET aux instances Avro de schéma. *Voir la section Valider des documents JSON pour plus d'informations à ce sujet.*

Dans cette documentation, le terme *instances JSON* réfère aux documents d'instance JSON et JSON5 sauf indications contraires. Voir aussi la section Différences entre JSON5 et JSON.

JSON et schéma JSON dans XMLSpy

Les deux types de document (instance JSON et schéma JSON) sont rédigés en format JSON, et doivent adhérer aux règles JSON de bonne formation et de validité. Les deux types de document (instance et schéma) présentent généralement l'extension de fichier **`.json`**. Les instances JSON peuvent être éditées dans le Mode Texte et le Mode Grille, et les documents de schéma JSON peuvent être édités dans ces deux modes également comme dans le Mode de schéma JSON, qui est un éditeur de schéma graphique.

XMLSpy fourni la prise en charge suivante pour travailler avec l'instance JSON et les documents de schéma JSON :

- Dans le Mode Texte, coloration syntaxique et contrôles syntaxiques ; remplissage automatique dans les schémas JSON et dans les documents d'instance si ceux-ci ont des associations de schéma, marge pliante ; marquages structurels. Toutes ces fonctions facilitent et accélèrent l'édition d'instances JSON et des documents de schéma JSON valides. Le Mode Texte fournit une validation aussi bien de l'instance que des documents de schéma.

© 2016-2022 Altova GmbH

- Dans le Mode Grille, une structure de grille tabulaire qui vous aide à mieux visualiser la structure du document. Vous pouvez éditer directement dans le Mode Grille. Vous pouvez aussi passer entre le Mode Texte et le Mode Grille selon vos besoins d'édition. Le Mode Grille permet la validation aussi bien de l'instance que des documents de schéma.
- La validation d'instance JSON dans le Mode Texte et le Mode Grille. La validation est effectuée par rapport à un schéma JSON qui est attribué dans la fenêtre Info.
- Le Mode Schéma JSON affiche des schémas JSON dans une mise en page graphique. Cela permet l'utilisation de fonctions glisser/déposer (outre la saisie de texte) pour une création rapide de schémas JSON. Les assistants à la saisie dans le cadre de l'affichage permettent d'effectuer des entrées pour compléter l'édition. De plus, le schéma est constamment contrôlé en termes de validité et les erreurs sont signalées.

Instances JSON : ouvrir des documents d'instance existants et créer de nouveaux documents d'instance

- Dans la section **Options | Types de fichier**, vous pouvez définir le mode par défaut (Mode Texte ou Mode Grille) pour ouvrir les documents d'instance JSON/JSON5. Les documents JSON/JSON5 existants seront ouverts dans le mode de démarrage par défaut que vous sélectionnez. Vous pouvez passer entre le Mode Texte et le Mode Grille à tout moment.
- Pour créer un nouveau document d'instance JSON ou JSON5, cliquer sur **Fichier | Nouveau** et sélectionner respectivement, `json: JavaScript Object Notation` ou `json5: JSON with ECMAScript 5 extensions`. Vous serez invité à choisir, en option, un fichier de schéma JSON ou (pour JSON, pas pour JSON5) Avro pour le nouveau fichier d'instance. Si vous attribuez un schéma, l'attribution sera saisie dans la fenêtre Info. Le nouveau document d'instance sera ouvert dans le Mode Texte ou le Mode Grille, selon les paramètres dans la section **Options | Types de fichier**.

Schémas JSON : ouvrir des schémas existants et créer de nouveaux schémas

- Un document de schéma JSON existant s'ouvre dans le Mode Schéma JSON. Vous pouvez passer au Mode Texte ou au Mode Grille à tout moment.
- Pour créer un nouveau document de schéma JSON, cliquer sur **Fichier | Nouveau**, et sélectionner `json: JSON Schema`. Le nouveau document de schéma JSON s'ouvrira dans le Mode Schéma JSON, avec le mot-clé **`$schema`** au début du document. Vous pouvez passer au Mode Texte ou au Mode Grille à tout moment.

Les modes différents (Texte, Grille, et schéma JSON) sont décrits dans les sous-sections de cette section.

Dans cette section

Cette section est organisée dans les sujets suivants :

- JSON Data explique les bases des documents JSON
- Schéma JSON décrit ce qu'est un schéma JSON et comment il fonctionne
- JSON Lines et JSON Comments propose des information concernant deux spécifications supplémentaires JSON prises en charge par XMLSpy
- Documents JSON dans le Mode Texte vous montre comment travailler avec les fonctions pertinentes JSON du Mode Texte
- Documents JSON dans le Mode Grille décrit comment éditer les documents JSON dans le Mode Grille
- Mode JSON Schema explique les fonctions JSON-schéma-édition du mode et la manière dont vous pouvez l'utiliser lors de la création de vos projets JSON
- Valider les données/documents JSON décrit comment attribuer un schéma JSON à un document JSON et comment valider des documents JSON

- Insérer les fragments JSON décrit comment insérer rapidement des fragments de texte JSON dans votre document JSON depuis des sources externes
- Transformations JSON avec XSLT/XQuery décrit comment interroger des documents JSON avec XPath/XQuery 3.1
- Expressions XQuery pour JSON confère une introduction large concernant l'utilisation de XQuery avec des documents JSON
- Générer un Schéma JSON depuis une instance JSON décrit la fonction permettant de générer un schéma depuis une instance
- Générer une Instance JSON depuis un Schéma JSON décrit comment générer une instance depuis un schéma
- Convertir entre JSON et XML décrit comment convertir entre JSON et XML dans XMLSpy

© 2016-2022 Altova GmbH

13.1 Données JSON

Cette section contient une brève description de la manière dont sont structurées les données JSON. Les données JSON sont généralement stockées dans un document (d'instance) JSON mais peuvent aussi être stockées en tant que fragment de données JSON dans un document d'un autre type. Un fragment de données JSON ou un document est une structure de données JSON, qui est largement définie comme ci-dessous.

XMLSpy prend en charge également **JSON5**, qui est une extension de JSON qui rajoute des extensions minimales ECMScript 5. Voir json5.org pour plus d'informations.

Objets et tableaux JSON

Un document JSON (généralement enregistré avec l'extension de fichier **.json**) est établi sur les structures de données centrales suivantes :

Objet

Un **objet** est délimité par des accolades et est une collection désordonnée de zéro ou plus de paires de **key:value**. Ces paires **key:value** sont les **propriétés de l'objet**. La clé doit toujours être une chaîne et doit donc toujours être contenue dans des guillemets. La clé (aussi nommée le nom de la propriété) est séparée de sa valeur par un double-point. Une valeur de propriété peut être de tout type de données JSON (*voir liste ci-dessous*). Une propriété est séparée de la suivante par une virgule. La liste ci-dessous est un exemple d'un objet présentant trois propriétés (dont toutes ont des valeurs de type atomique) :

```
{
  "emailtype": "home",
  "emailaddress": "contact01.home@altova.com",
  "citycode": 22
}
```

Tableau

Un **tableau** est délimité par des crochets, et est une liste triée de zéro ou plus d'**items** séparés par des virgules. Ces items peuvent être de tout type de données JSON (*voir liste ci-dessous*).

Exemple d'un tableau contenant deux objets

Le tableau consistant en deux objets (chacun contenu dans des accolades). Le tableau lui-même est indiqué par des crochets.

```
[
      {
      "emailtype": "home",
      "emailaddress": "contact01.office@altova.com",
      "citycode": 22
      },
      {
      "emailtype": "office",
      "emailaddress": "contact01.office@altova.com",
      "citycode": 22
      }
  ]
```

Exemple de tableaux qui sont les valeurs des propriétés d'un objet

La liste ci-dessous représente un objet contenant trois paires `key:value`. Chaque valeur est un tableau qui contient un tuple (séquence). (Un tuple peut être considéré comme étant un tableau unidimensionnel). Les trois items dans chaque tuple sont des types atomiques.

```
{
      "x": [ 1, 2, "abc" ],
      "y": [ 3, 4, "def" ],
      "z": [ 5, 6, "ghi" ]
}
```

Types de données JSON

Les valeurs de propriété d'objet et les items de tableau peuvent être constitués des types suivants :

- `string` (doit être contenu dans des guillemets). Une chaîne peut être spécifiée en plus pour posséder un format, comme un format `date-heure` ou `e-mail`
- `number`: un nombre avec une partie fractionnelle ; inclut des entiers
- `integer`: un nombre sans partie fractionnelle ; un sous-ensemble du type `number`
- `boolean` (`true/false`, non contenu dans des guillemets)
- `object`: lorsqu'utilisé avec un autre objet, permet d'imbriquer des données
- `array`: fournit la possibilité de construire des structures plus complexes que ce qui est autorisé par les objets
- `null` (`null`, non contenu dans des guillemets)

Exemple of JSON data

Voici un exemple d'un fragment de données JSON. Veuillez noter comment le document est structuré en objets et en tableaux. De même, notez le type de données des valeurs-clé ; les valeurs de chaîne sont entre guillemets, les autres types apparaissent en vert.

```
{
  "first": "Jason",
  "last": "Jones"
  "isManager": true,
  "age": 35,
  "address": {
    "street": "Jason Avenue",
    "city": "Jasonville",
    "state": "JS",
    "postcode": "JS12 ON34"
  },
  "phone": [
    {
      "type": "home",
      "number": "12 3456-7890"
    },
    {
      "type": "office",
```


© 2016-2022 Altova GmbH

```
      "number": "789 012-34567"
    }
  ],
  "children": [],
  "partner": null
}
```

Certaines différences entre JSON5 et JSON

JSON5 est un sous-ensemble strict de JavaScript, s'ajoute aux nouveaux types de données JSON et fonctionne avec tous les contenus JSON existants. Vous trouverez ci-dessous quelques-unes des différences principales :

- JSON5 prend en charge les commentaires. Les commentaires sont délimités comme suit : `// comment //` ou `/* comment */`.
- Dans JSON5, les paires de clé `key:value` de doivent pas être contenues entre guillemets.
- Dans JSON5, les chaînes peuvent être rédigées sur plusieurs lignes.
- Les documents JSON5 peuvent être validés par rapport aux schémas JSON mais pas par rapport aux schémas.

13.2 Schéma JSON

De même qu'un schéma XML spécifie la structure et le contenu d'un document XML, un schéma JSON spécifie comment sont organisées les données JSON dans un document JSON. Il spécifie quels champs de données sont attendus et comment les valeurs sont représentées. Vous trouverez la spécification de Schéma JSON et plus d'informations concernant le schéma JSON ici.

Un schéma JSON est lui-même un objet JSON. Lexicalement, le schéma complet est contenu dans des accolades (*voir liste ci-dessous*), qui sont les délimiteurs des objets JSON. Le schéma est écrit dans la syntaxe JSON et est généralement enregistré dans un fichier avec une extension `.json`. Il est indiqué en tant qu'un schéma JSON, par le mot-clé **`$schema`**, qui doit être le premier mot-clé de l'objet de niveau supérieur. Ce mot-clé devrait avoir une valeur comme l'une de celles indiquées ci-dessous :

- Versions vers draft-07: **`"http://json-schema.org/draft-N/(hyper-)schema#"`**, où **`N`** est le nombre de la version (**`04`**, **`06`**, or **`07`**).
- Versions depuis draft-2019-09 ultérieures : **`"https://json-schema.org/draft/YYYY-MM/(hyper-)schema"`**, où **`YYYY`** et **`MM`** sont, respectivement, l'année et le mois du draft, par exemple, **`2019-09`**.

Ici, vous trouverez un exemple comment le mot-clé `$schema` est utilisé.

```
{
   "$schema": "https://json-schema.org/draft/2020-12/schema",
   ...
}
```

Note : Bien que le mot-clé `$schema` puisse avoir la valeur **`"http://json-schema.org/schema#"`** qui spécifie la dernière version du schéma, il vaut mieux utiliser une URL qui identifie la version spécifique (puisque les fonctions plus récentes ne sont pas compatibles en arrière). Pour plus d'informations, voir Version de Schéma JSON.

Dans XMLSpy, vous pouvez créer des schémas JSON graphiquement dans le Mode de schéma JSON. Vous trouverez une description de la procédure à suivre dans la section Mode Schéma JSON. Outre les fonctions d'édition de schéma disponibles dans le Mode de schéma JSON, les fonctions liées au schéma suivantes sont disponibles :

- Validation avec le Validateur JSON de XMLSpy : attribuer un schéma JSON à un document d'instance JSON, et valider le document d'instance depuis XMLSpy. Voir Valider les documents JSON pour plus d'informations.
- Définir les options de validation JSON.
- Générer un schéma JSON depuis une instance JSON : si un document d'instance JSON existe déjà, vous pouvez générer un schéma JSON depuis le document. Vous pouvez ensuite éditer le schéma le cas échéant.
- Convertir entre JSON et XML: vous pouvez convertir entre des documents des deux formats.

Terminologie

Ci-dessous, vous trouverez des définitions des termes de schéma JSON communs utilisés dans la GUI et dans cette documentation.

Terme	Définition

© 2016-2022 Altova GmbH

Schéma	L'objet de schéma de niveau supérieur dans un document de schéma JSON ; le fichier de schéma.
Objet	Un type JSON contenant zéro ou plus de propriétés.
Propriété	Une paire `key:value` d'un objet. Sa valeur peut être de n'importe quel type de données JSON.
Mots-clés	La partie `key` de la paire d'un objet `key:value`. Il s'agit toujours d'une chaîne.
Sous-schémas	Un objet qui est l'enfant d'un opérateur ou d'une dépendance.
Définition	La description complète de tout type JSON. Les définitions peuvent être globales ou locales.
Tableau	Une liste ordonnée de zéro ou plus d'`items` de tout type de données JSON séparée par des virgules.
Types atomiques	Les types de données JSON `string`, `number`, `integer`, `boolean` et `null`.
Sélecteurs de type	Les types `any` et `multiple`, qui sélectionnent les types any et multiple, respectivement
Opérateurs	Les sélecteurs d'occurrence qui peuvent être ajoutés en tant qu'enfants des définitions. *Voir la section Opérateurs*.

Types de données JSON

Les valeurs de propriété d'objet et les items de tableau peuvent être constitués des types suivants :

- `string` (doit être contenu dans des guillemets). Une chaîne peut être spécifiée en plus pour posséder un format, comme un format `date-heure` ou `e-mail`
- `number`: un nombre avec une partie fractionnelle ; inclut des entiers
- `integer`: un nombre sans partie fractionnelle ; un sous-ensemble du type `number`
- `boolean` (`true/false`, non contenu dans des guillemets)
- `object`: lorsqu'utilisé avec un autre objet, permet d'imbriquer des données
- `array`: fournit la possibilité de construire des structures plus complexes que ce qui est autorisé par les objets
- `null` (`null`, non contenu dans des guillemets)

13.3 JSON Lines and JSON Comments

XMLSpy prend en charge les documents JSON Lines (JSONL) et JSON with Comments (JSONC), ce qui signifie que la validation et l'édition intelligente de ces documents est disponible de la même manière que pour les autres types de documents JSON. Cette section présente les fonctions principales de ces types de documents JSON.

JSON Lines

JSON Lines (JSONL) est un format de stockage des données structuré, où chaque enregistrement est séparé de l'autre par une newline ; ce qui signifie que chaque enregistrement se trouve sur sa propre ligne. Par conséquence, les enregistrements peuvent être traités un à la fois, ce qui rend le format très utile lorsque du traitement de données comme des fichiers journaux.

Exemple de document JSON Lines

```
["Team", "Played", "Won", "Drew", "Lost", "Points"]
["USA", 2, 1, 1, 0, 4]
["France", 3, 1, 1, 1, 4]
["Germany", 1, 0, 1, 0, 1]
["USA", 1, 0, 0, 1, 0]
```

Pour plus d'exemples, voir http://jsonlines.org/examples/.

Les fichiers JSON Lines sont reconnus en tant que tels dans XMLSpy si le fichier a une extension `.jsonl`.

JSON with Comments

Les documents JSON autres que les documents JSON5 ne permettent pas de commentaires. Le format JSON with Comments (JSONC) a été introduit pour permettre des commentaires dans des documents JSON. Les commentaires suivants sont utilisés dans JSONC et pris en charge dans XMLSpy:

- Commentaires en une ligne : préfixé par `//`. Par exemple : `// My comment`
- Commentaires multi-ligne : délimité par `/*` et `*/`. Par exemple : `/* My comment */`

Les fichiers JSONC sont reconnus dans XMLSpy si le fichier a une extension `.jsonc`.

Note : Des commentaires sont aussi autorisés dans les documents JSON5. Les commentaires dans d'autres types de fichiers JSON (outre JSONC et JSON5) peuvent entraîner des erreurs au cours du traitement.

© 2016-2022 Altova GmbH

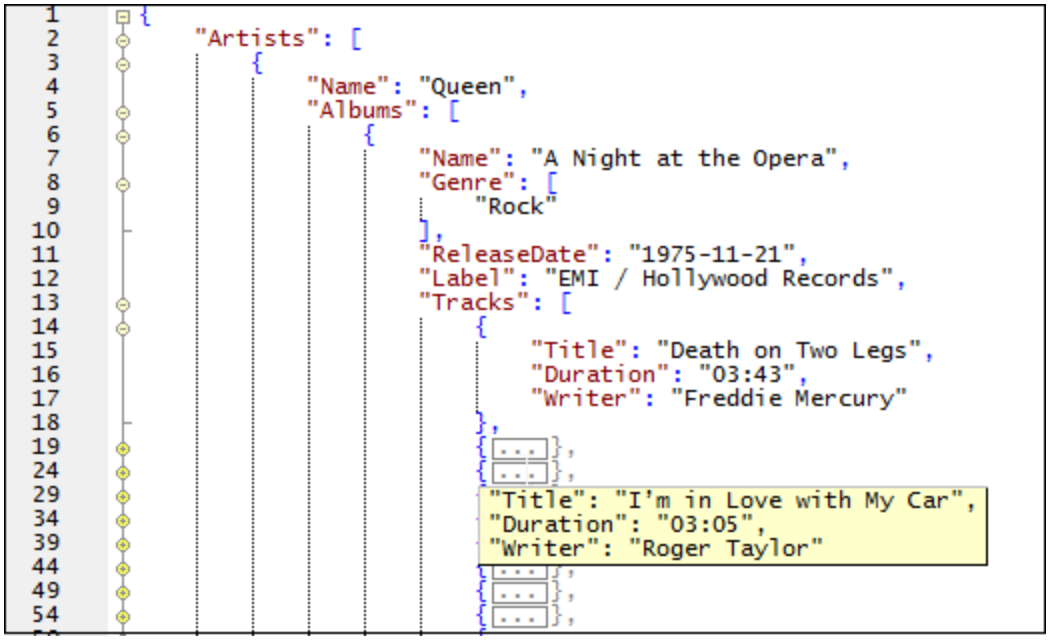
13.4 Mode Texte JSON

Site web d'Altova : Éditeur JSON

Les documents d'instance de schémas JSON, schémas Avro, et JSON/JSON5 (y compris les instances de données Avro dans le format JSON) peuvent être édités en utilisant les fonctions d'édition intelligentes du mode Texte. Ces fonctions comprennent : marges pliantes, marquage structurel, coloration de la syntaxe, vérification de la syntaxe, enregistrer des strings d'image encodés en Base64 dans leurs formats d'image et saisie semi-automatique. XMLSpy permet aussi une conversion entre JSON et XML dans les deux sens et vous permet de générer un schéma JSON/JSON5 depuis une instance JSON.

Marges pliantes

Le pliage de source est activé sur des mots-clés JSON et des définitions, et réfère à la capacité d'agrandir et de réduire ces nœuds. Ceux-ci sont indiqués dans la marge de pliage de source avec un signe +/- (*voir capture d'écran ci-dessous*). La marge peut être ouverte ou fermée dans le dialogue Paramètres de Mode Texte. Lorsqu'un nœud est réduit, cela est indiqué visuellement par une ellipse (*voir capture d'écran ci-dessous*). Si le curseur de la souris est placé au-dessus d'une ellipse, le contenu du nœud réduit est affiché dans une fenêtre popup (*capture d'écran ci-dessous*). Si le contenu est trop grand pour une fenêtre popup, cela est indiqué par une ellipse en bas de la fenêtre.



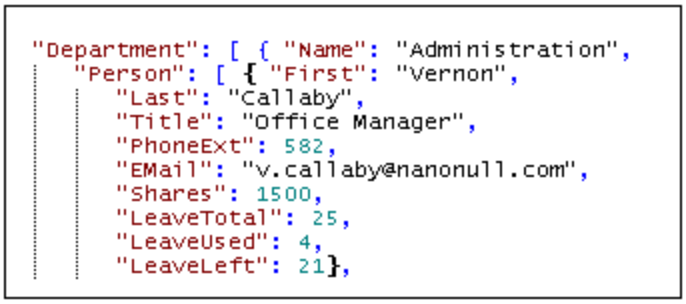
L'icône **Basculer tous les plis** dans la barre d'outils Texte bascule tous les nœuds dans leur forme agrandie ou réduit **tous** les nœuds au niveau de l'élément du document de niveau supérieur.
Les options suivantes sont disponibles en cliquant sur l'icône +/- du nœud :

Cliquer [-]	Réduit le nœud.
Cliquer [+]	Agrandit le nœud de manière à ce que les nœuds descendants s'affichent sous leur forme agrandie ou réduite selon l'état dans lequel ils étaient avant la réduction du nœud.

Shift+Click [−]	Réduit tous les nœuds descendants, mais laisse le nœud qui a été cliqué dans sa forme agrandie.
Ctrl+Click [+]	Agrandit le nœud cliqué ainsi que tous ses nœuds descendants.

Marquage structurel

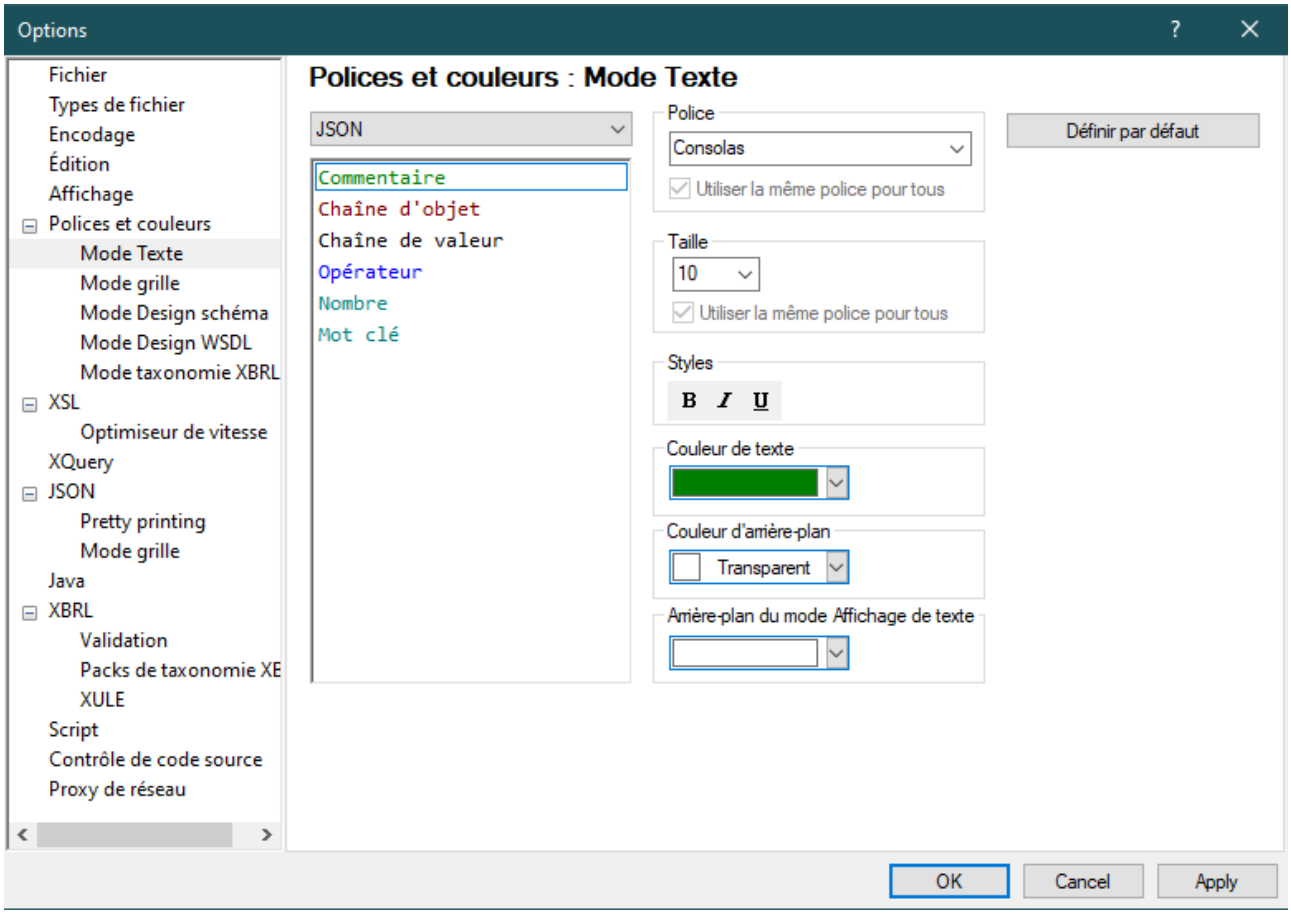
La paire d'accolades ou de crochets qui délimitent un objet ou un tableau JSON, respectivement, (*voir capture d'écran ci-dessous*) passe en gras lorsque le curseur est placé soit avant soit après une des accolades ou un des crochets. Cela indique où la définition d'un élément particulier commence et se termine.



Coloration syntaxique

Un document JSON (instance/schéma JSON ou Avro) ainsi qu'un document JSON5 est constitué de chaînes d'objets, de chaînes de valeur, d'opérateurs, de numéros et de mots-clés. Dans le Mode Texte, chaque catégorie d'item peut être affichée dans une couleur différente (*voir capture d'écran ci-dessus*) conformément aux paramètres que vous définissez dans le dialogue Options (*voir capture d'écran ci-dessous*). Vous pouvez définir les couleurs des différents composants JSON dans la section Polices d'écriture du dialogue Options (*voir capture d'écran ci-dessous*). Dans la liste de choix en haut à gauche, sélectionner *JSON*, puis sélectionner la couleurs souhaitée (dans le panneau Styles) pour chaque item JSON.

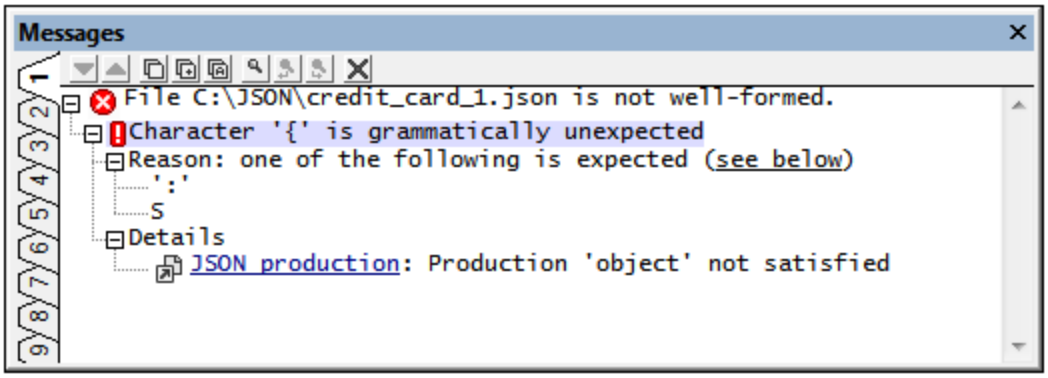
© 2016-2022 Altova GmbH



Note : syntaxe JSON5—mais pas la syntaxe JSON—permet les commentaires. Les commentaires dans JSON5 sont délimités comme suit : `// comment //` ou `/* comment */`.

Vérification de la syntaxe

La syntaxe d'un document JSON (JSON ou instance/schéma Avro) peut être vérifiée en sélectionnant la commande **XML | Vérifier la bonne formation (F7)**. Les résultats de la vérification de bonne formation sont affichés dans la fenêtre Messages (*voir capture d'écran ci-dessous*).



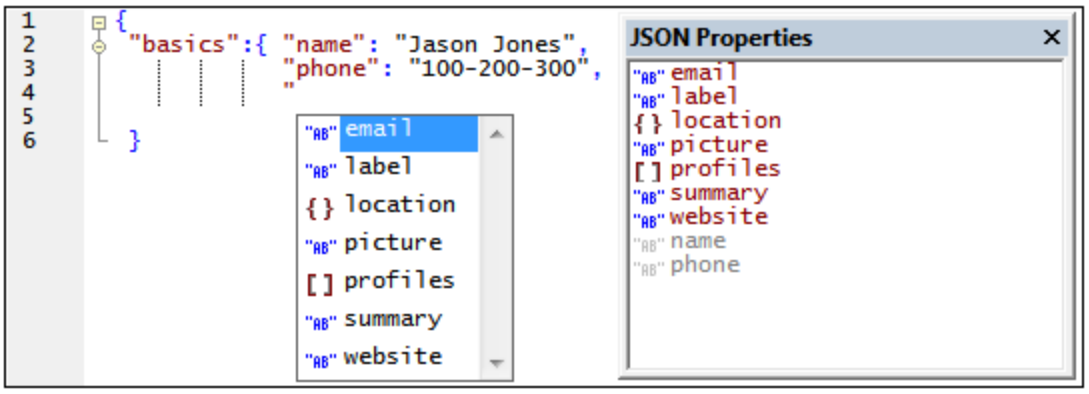
Le message d'erreur dans la capture d'écran au-dessus indique une erreur dans le document : Une accolade qui s'ouvre apparaît à l'emplacement où une colonne est attendue.

Saisie semi-automatique

Le remplissage automatique est activé lorsque le document JSON (JSON instance/schéma ou schéma Avro) en cours d'édition est associé avec un schéma.

- Si le document est un schéma JSON, le remplissage automatique sera basé sur la version de schéma indiquée par le mot-clé `$schema`. Pour plus d'informations, voir aussi Version de schéma JSON.
- Si le document est une instance JSON/JSON5, alors un schéma JSON doit être attribué à l'instance pour que le remplissage automatique soit activé.
- Si le document est un document de données Avro en format JSON, alors un schéma Avro doit être attribué à l'instance pour que le remplissage automatique fonctionne.
- Si le document est un schéma Avro, il sera automatiquement associé avec un schéma pour le schéma Avro, et le remplissage automatique sera basé sur ce schéma.

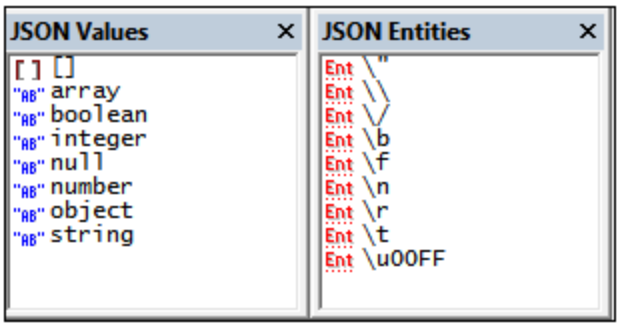
Le remplissage automatique vous offre les options de saisie disponibles à l'emplacement du curseur. La fonction utilise (i) des fenêtres pop-up dans la fenêtre principale et (ii) des assistants à la saisie (*voir la capture d'écran ci-dessous*). Les pop-ups et assistants à la saisie affichent chacun des entrées qui sont valides à l'emplacement du curseur. Pour se déplacer d'une entrée à l'autre dans la liste pop-up, utilisez les touches de direction. Si le schéma contient une description de l'entrée (dans le mot-clé de l'entrée `description` dans le schéma), la description sera affichée à côté de l'entrée pop-up en surbrillance. Sélectionnez une entrée depuis la fenêtre pop-up ou double-cliquez dans l'assistant d'entrée pour l'insérer.



Dans le document d'instance affiché dans la capture d'écran ci-dessus, la fenêtre pop-up et l'assistant à la saisie Propriétés de JSON sont affichés lorsque le curseur est situé après les guillemets qui indiquent le début d'un nom de propriété. L'assistant à la saisie affiche toutes les propriétés autorisées à cet endroit ; les propriétés qui ont déjà été saisies sont grisées et désactivées. La fenêtre pop-up n'affiche que les propriétés qui sont autorisées à cet endroit.

Il existe deux autres assistants à la saisie : Les Valeurs JSON et Entités JSON (*voir la capture d'écran ci-dessous*). Ils montrent, respectivement, les valeurs autorisées des paires `key:value` et des entités pour l'échappement des caractères dans les chaînes JSON. L'assistant à la saisie Valeurs de JSON dans la capture d'écran ci-dessous montre les valeurs autorisées pour le mot-clé `type` pendant l'édition d'un schéma JSON. La dernière saisie de l'assistant de saisie Entités JSON, `\u00FF`, est un espace réservé qui est synonyme d'un caractère Unicode. Remplacez la partie surlignée en bleu par le code du caractère Unicode que vous souhaitez.

© 2016-2022 Altova GmbH



D'autres saisies semi-automatique contextuelles ou des conseils incluent les choses suivantes lorsque celles-ci sont spécifiées dans le schéma : énumérations, descriptions, occurrences requises et valeurs par défaut.

Enregistrer un string encodé en Base64 en tant qu'image

Pour enregistrer un string encodé en Base64 dans son format d'image, cliquez avec la touche de droite sur le texte encodé et sélectionnez la commande **Enregistrer sous Image**. Dans le dialogue qui apparaît, sélectionnez l'emplacement sous lequel vous souhaitez enregistrer l'image et saisissez un nom pour le fichier d'image. L'extension du fichier d'image (`.png`, `.gif`, `.svg`, etc.) sera détectée automatiquement depuis l'encodage Base64 et apparaîtra dans le dialogue Enregistrer. Une fois terminé, cliquer **Enregistrer**.

Cette action peut également être exécutée par le biais de la commande de menu **Editer | Enregistrer sous Image**.

13.5 Mode JSON Grille

Le Mode JSON Grille vous permet de voir la structure du document JSON (instance/schéma JSON ou schéma Avro) dans une grille pour éditer le document plus facilement dans une grille et de restructurer des blocs de texte structuré. Cela vous apporte un aperçu et des possibilités d'édition qui ne sont pas toujours présentes dans le Mode Texte, en particulier dans le cas de documents longs et à structure complexe. Dans le cas des documents JSON, une telle complexité peut se produire sous la forme de tableaux et d'objets imbriqués à des niveaux multiples. Par exemple, comparer le texte JSON recensé ci-dessous (tel qu'il apparaîtrait dans le Mode Texte) et sa représentation dans le Mode Grille (comme affiché dans la capture d'écran ci-dessous).

Note : une prise en charge est disponible dans l'**Enterprise Edition uniquement**.

Liste de code JSON dans le Mode Texte

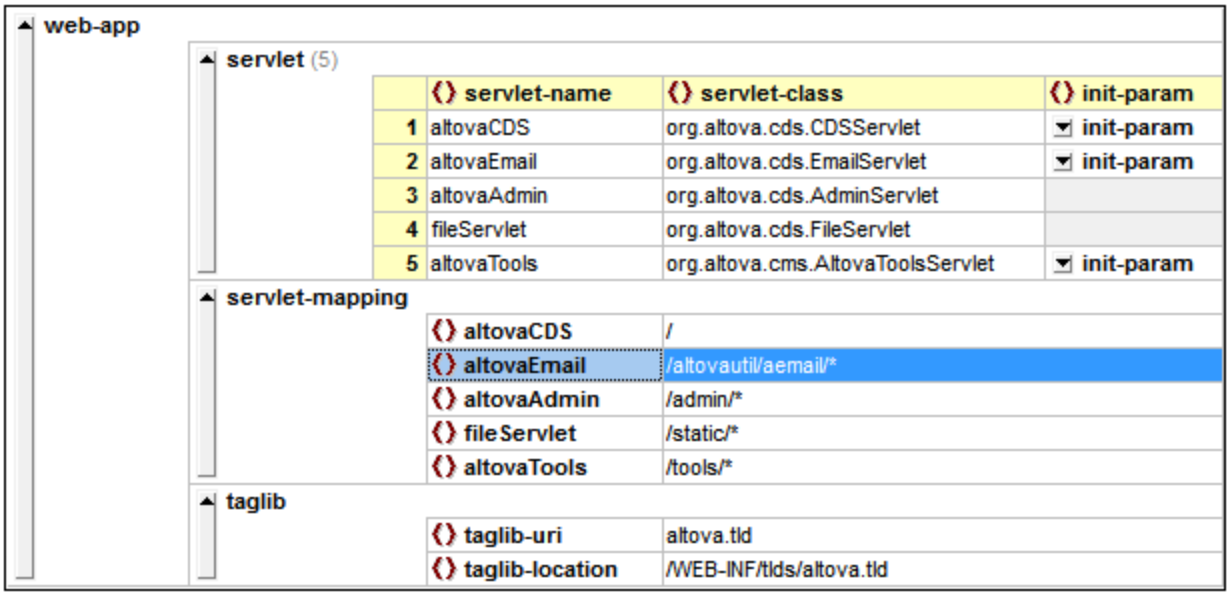

```
{
  "web-app": {
  "servlet": [
    {
        "servlet-name": "altovaCDS",
        "servlet-class": "org.altova.cds.CDSServlet",
        "init-param": {
        "configGlossary:installationAt": "Philadelphia, PA",
        "configGlossary:adminEmail": "ksm@pobox.com",
        "configGlossary:poweredBy": "Altova",
        "configGlossary:poweredByIcon": "/images/altova.gif",
        "configGlossary:staticPath": "/content/static",
        "templateProcessorClass": "org.altova.WysiwygTemplate",
        "templateLoaderClass": "org.altova.FilesTemplateLoader",
        "templatePath": "templates",
        "templateOverridePath": "",
        "defaultListTemplate": "listTemplate.htm",
        "defaultFileTemplate": "articleTemplate.htm",
        "useJSP": false,
        "jspListTemplate": "listTemplate.jsp",
        "jspFileTemplate": "articleTemplate.jsp",
        "cachePackageTagsTrack": 200,
        "cachePackageTagsStore": 200,
        "cachePackageTagsRefresh": 60,
        "cacheTemplatesTrack": 100,
        "cacheTemplatesStore": 50,
        "cacheTemplatesRefresh": 15,
        "cachePagesTrack": 200,
        "cachePagesStore": 100,
        "cachePagesRefresh": 10,
        "cachePagesDirtyRead": 10,
        "searchEngineListTemplate": "forSearchEnginesList.htm",
        "searchEngineFileTemplate": "forSearchEngines.htm",
        "searchEngineRobotsDb": "WEB-INF/robots.db",
        "useDataStore": true,
        "dataStoreClass": "org.altova.SqlDataStore",
        "redirectionClass": "org.altova.SqlRedirection",
        "dataStoreName": "altova",
```


 © 2016-2022 Altova GmbH

```
      "dataStoreDriver": "com.microsoft.jdbc.sqlserver.SQLServerDriver",
      "dataStoreUrl": "jdbc:microsoft:sqlserver://LOCALHOST:1433;DatabaseName=goon",
      "dataStoreUser": "sa",
      "dataStorePassword": "dataStoreTestQuery",
      "dataStoreTestQuery": "SET NOCOUNT ON;select test='test';",
      "dataStoreLogFile": "/usr/local/tomcat/logs/datastore.log",
      "dataStoreInitConns": 10,
      "dataStoreMaxConns": 100,
      "dataStoreConnUsageLimit": 100,
      "dataStoreLogLevel": "debug",
      "maxUrlLength": 500
    }
    }, {
      "servlet-name": "altovaEmail",
      "servlet-class": "org.altova.cds.EmailServlet",
      "init-param": {
        "mailHost": "mail1"
        "mailHostOverride": "mail2"
      }
    }, {
      "servlet-name": "altovaAdmin",
      "servlet-class": "org.altova.cds.AdminServlet"
    }, {
      "servlet-name": "fileServlet",
      "servlet-class": "org.altova.cds.FileServlet"
    }, {
      "servlet-name": "altovaTools",
      "servlet-class": "org.altova.cms.AltovaToolsServlet",
      "init-param": {
        "templatePath": "toolstemplates/",
        "log": 1,
        "logLocation": "/usr/local/tomcat/logs/AltovaTools.log",
        "logMaxSize": "",
        "dataLog": 1,
        "dataLogLocation": "/usr/local/tomcat/logs/dataLog.log",
        "dataLogMaxSize": "",
        "removePageCache": "/content/admin/remove?cache=pages&id=",
        "removeTemplateCache": "/content/admin/remove?cache=templates&id=",
        "fileTransferFolder": "/usr/local/tomcat/webapps/content/fileTransferFolder",
        "lookInContext": 1,
        "adminGroupID": 4,
        "betaServer": true
      }
    }
  ],
  "servlet-mapping": {
    "altovaCDS": "/",
    "altovaEmail": "/altovautil/aemail/*",
    "altovaAdmin": "/admin/*",
    "fileServlet": "/static/*",
    "altovaTools": "/tools/*"
  },
```

```
      "taglib": {
        "taglib-uri": "altova.tld",
        "taglib-location": "/WEB-INF/tlds/altova.tld"
      }
    }
  }
```

Alors que la structure de document dans le Mode Texte (*liste ci-dessus*) est difficile à lire et requiert une lecture approfondie, la structure dans le Mode Grille (*capture d'écran ci-dessous*) est bien plus lisible.



De plus, vous pouvez modifier la structure de manière très simple en ajoutant, supprimant, ou déplaçant des objets dans la grille. Des blocs entiers de texte peuvent être réorganisés, (par exemple, en les triant ou en les déplaçant). Le contenu également peut être édité dans le Mode Grille. Cela est très simple à faire, grâce à la disponibilité de commandes in-cell dans les cellules individuelles.

Barre d'outils Affichage Grille

Les commandes liées à l'édition dans le Mode Grille peuvent être accédées rapidement par le biais de la barre d'outils Affichage Grille *(capture d'écran ci-dessous)*—qui est située dans l'espace de la barre d'outils tout en haut de la fenêtre d'application. Pointez sur l'icône de la barre d'outils pour voir ses nom et raccourci.



Note : Ces commandes sont souvent disponibles dans le menu JSON.

© 2016-2022 Altova GmbH

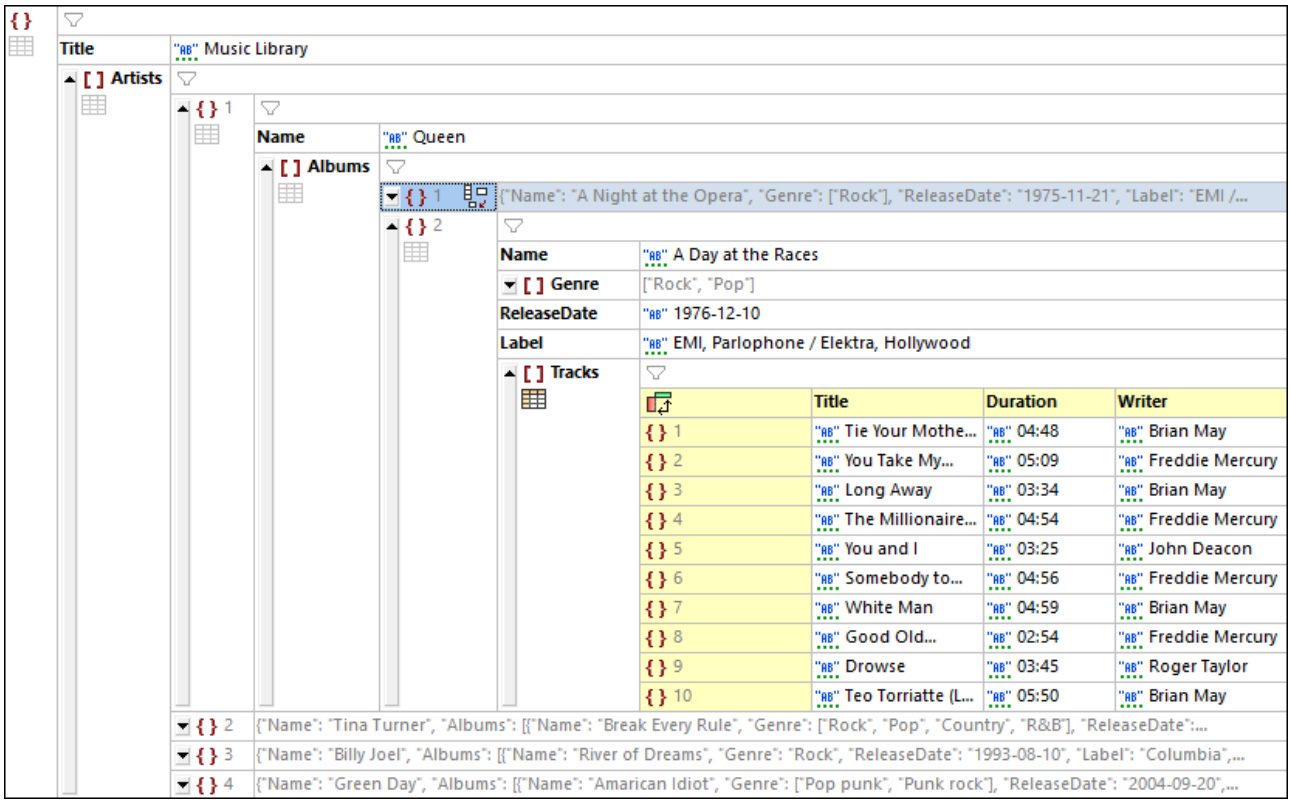
Dans cette section

Cette section est organisée dans les rubriques ci-dessous. Ces descriptions expliquent les fonctions spécifiques au Mode JSON Grid.

- Comment le document JSON est affiché dans le Mode JSON Grille
- Éditer la structure du document
- Éditer le contenu du document
- Glisser-déposer dans le Mode Grille
- Utiliser l'affichage de table
- Utiliser les filtres XQuery
- Utiliser les formules XQuery
- Affichage de l'image dans le Mode Grille
- Graphiques dans le Mode Grille
- Menu contextuel dans le Mode Grille
- Paramètres du Mode Grille

13.5.1 Affichage du document dans le Mode Grille

Le Mode JSON Grid affiche le document JSON hiérarchiquement dans une grille (*voir capture d'écran ci-dessous*).



Chaque ligne de la grille affiche une des structures de données suivantes, chacune d'entre elle est affichée dans la table ci-dessous avec son symbole. Veuillez noter que les arrays et les objets peuvent contenir des composants enfants qui peuvent être des objets, des arrays, ou des valeurs atomiques.

{}	Un objet *(voir définition)*. Les objets contiennent des paires `key:value`.
[]	Un array *(voir définition)*. Les arrays contiennent des items, qui sont généralement des objets ou des valeurs.
	Une paire `key:value` *(voir définition)*.

Dans la capture d'écran ci-dessus, nous voyons un objet racine qui contient deux paires `key:value` : (i) **`"Title":"Music Library"`**, (ii) **`"Artists":[Array]`**. L'array lié à **`Artists`** contient quatre items qui sont des objets.

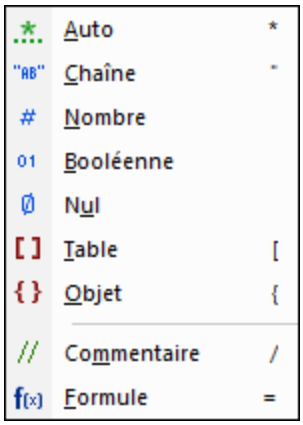
Veuillez noter les points suivants :

- Vous pouvez zoomer en avant et en arrière dans la grille avec **Ctrl + Roue de la souris**, ou **Ctrl+[Plus]** et **Ctrl+[Moins]**.
- Les objets, array, et items d'array du même niveau sont numérotés, en commençant par 1. Dans la capture d'écran ci-dessus, par exemple, dans le cadre de l'array `Artists`, les objets sont numérotés de `1` à `4`. Donc également dans les arrays `Albums` et `Tracks`. Veuillez noter que cette numérotation n'est pas contenue dans le document lui-même, mais est constitue une fonction Mode JSON Grid pour vous aider à voir la structure du document.
- Les objets et arrays peuvent être agrandis ou réduits en appuyant sur l'icône à tête de flèche située à gauche du symbole de la structure *(voir capture d'écran ci-dessus)*. Lorsqu'une structure de données est réduite, son contenu est affiché en tant que texte dans une seule ligne ; par exemple, voir les trois dernières lignes de grille de la capture d'écran ci-dessus.
- Si vous choisissez plusieurs composants au même niveau, vous pouvez les agrandir/réduire tous en appuyant sur **Shift** et en cliquant sur une des têtes de flèche des composants sélectionnés.
- Lorsque des arrays et des objets sont agrandis, veuillez noter : (i) comment les contenus des arrays et des objets commencent dans la ligne suivante en partant depuis son symbole, (iii) comment le nom d'un array (sa clé) se trouve sur la même ligne que le symbole d'array, alors que les items de l'array commencent dans la ligne suivante. Lorsqu'un array ou un objet est réduit, néanmoins, son contenu est affiché sur la même ligne.
- Veuillez noter que les paires `key:value` sont représentées sur une ligne, sauf lorsqu'elles font partie de Affichage table (dans ce cas, la clé sera l'en-tête de colonne). Pour consulter un exemple de Affichage table, voir les contenus de l'array **`Tracks`** dans la capture d'écran ci-dessus.

Symboles de types de données (ou types)

Les valeurs de propriété d'objet et les items d'array peuvent être de types différents ; chaque type est indiqué par un symbole comme indiqué dans la capture d'écran ci-dessous. (Dans la capture d'écran ci-dessus, veuillez noter que le type de la valeur de toutes les paires `key:value` est le type `string`.)

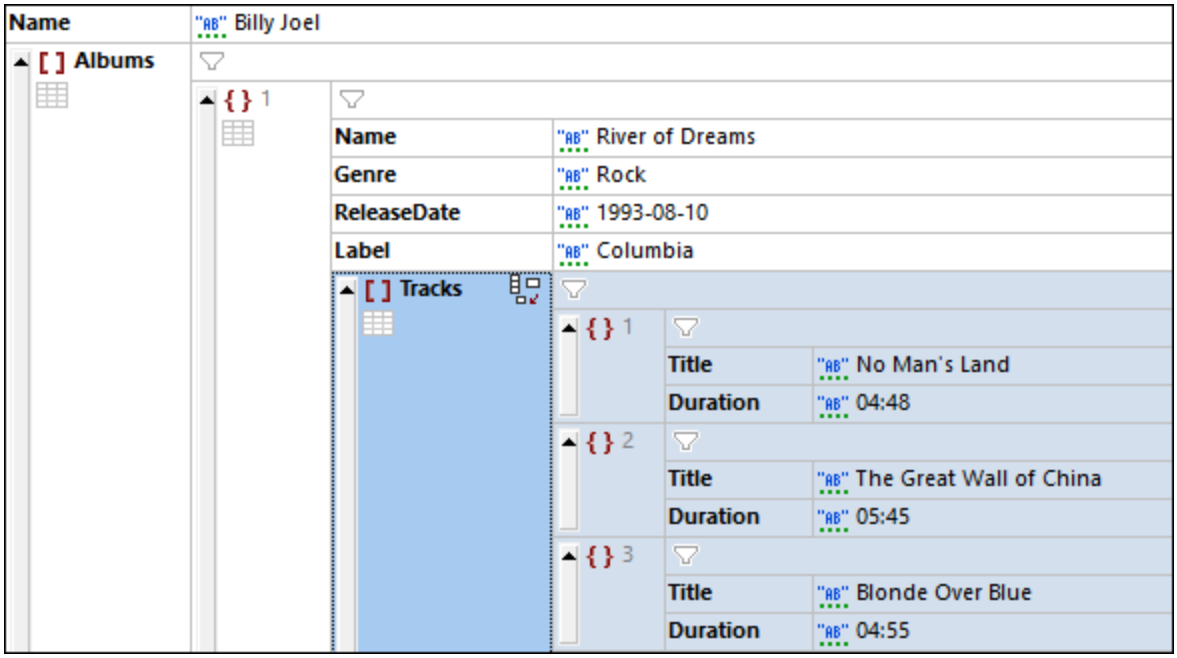
© 2016-2022 Altova GmbH

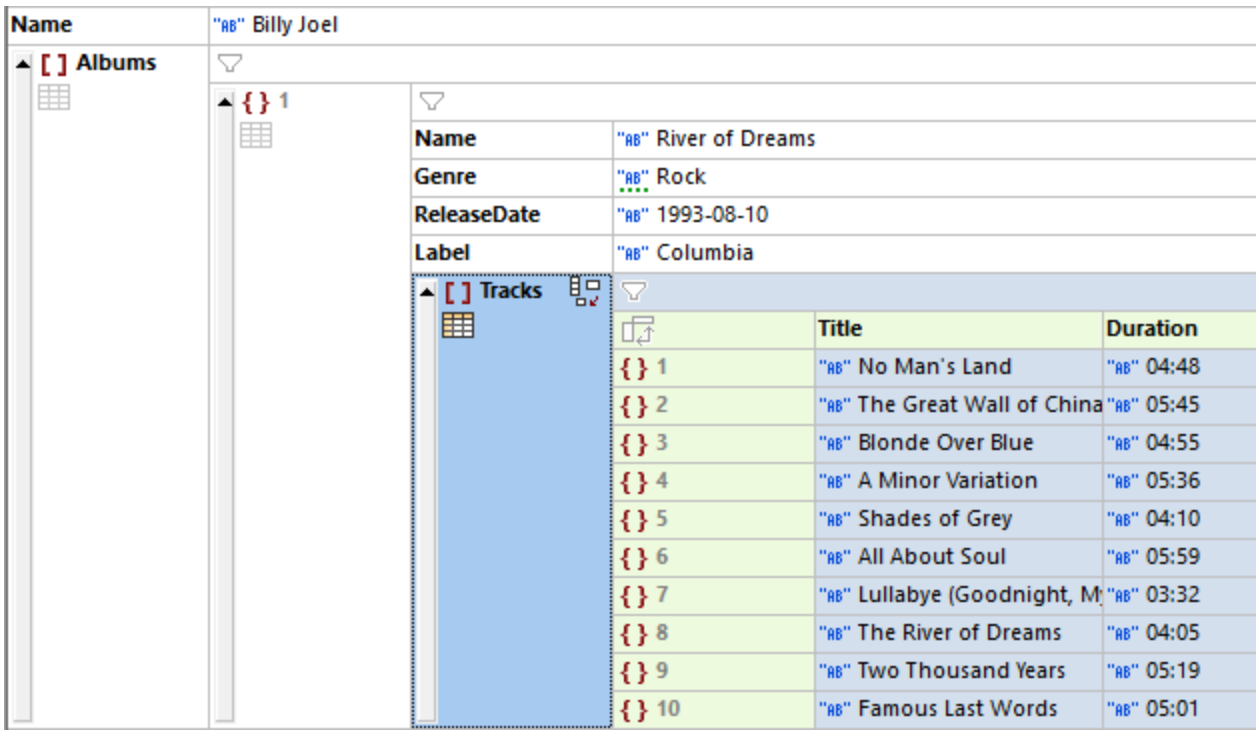


Note : Les commentaires sont pris en charge dans JSONC et JSON5. Les Formules sont spécifiques à XMLSpy.

Affichage Table

Les objets et les arrays qui contiennent au moins un objet ou array peuvent être consultés soit en tant que liste *(marquée dans la capture d'écran à gauche)* soit en tant que table *(marquée dans la capture d'écran à droite)*. L'affichage peut passer entre table et liste pour des objets et arrays individuels.





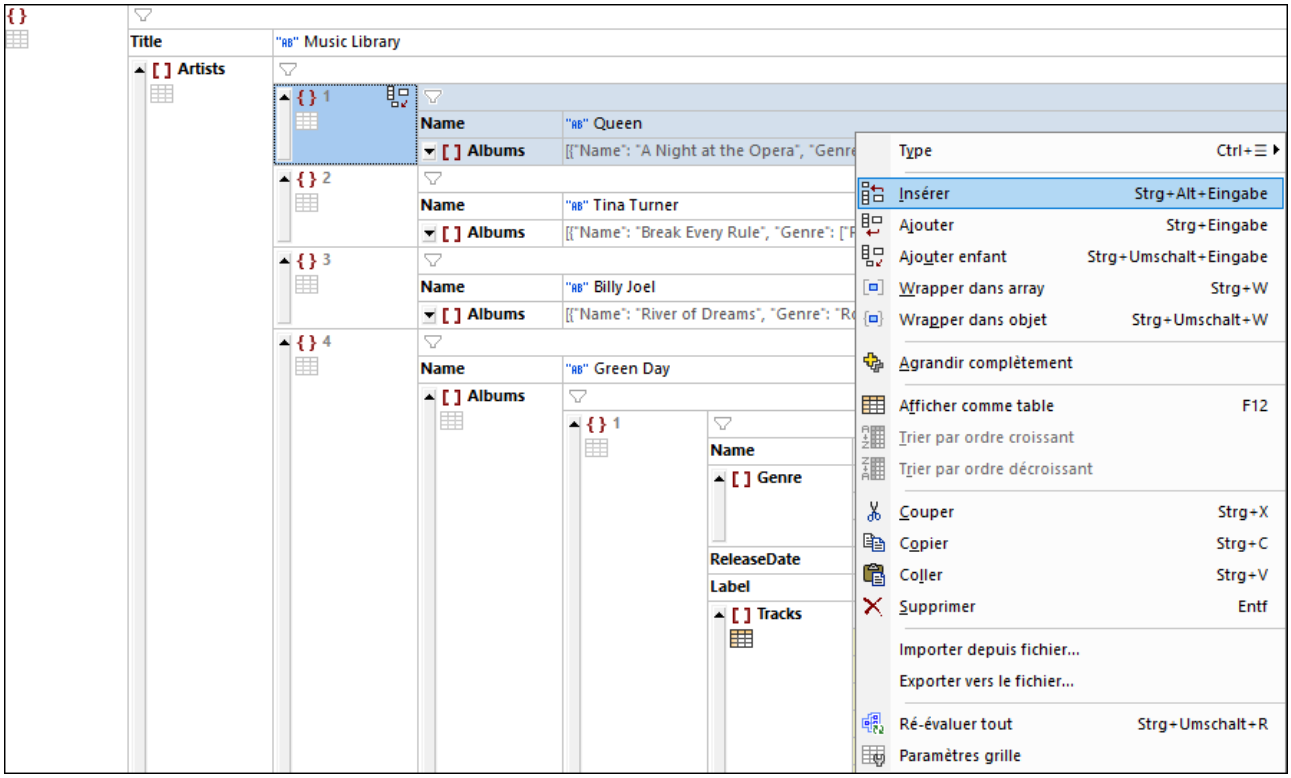
Pour une description plus détaillée d'un Affichage Table du Mode JSON Grid, voir la section Affichage table.

© 2016-2022 Altova GmbH

13.5.2 Éditer la structure du document

Vous pouvez éditer la structure du document JSON en ajoutant de nouveau composants, et en supprimant ou en déplaçant les blocs de grille sélectionnés. Vous pouvez éditer la structure de document d'une des manières suivante :

- Choisir un composant et cliquer avec la touche de droite pour utiliser les commandes dans son menu contextuel *(voir capture d'écran ci-dessous).* Les commandes sont décrites dans la section Menu contextuel.
- Utiliser les touches d'édition qui sont disponibles dans les cellules. Ces touches sont décrites dans la section Touches de commande dans les cellules ci-dessous.
- Choisir un composant et le glisser/déposer dans un nouvel emplacement pour modifier la structure.



Les raccourcis des commandes pour construire une structure sont recensés dans la table ci-dessous.

Commande	Raccourci
Insérer (Frère) après	Ctr+Enter
Insérer (Frère) avant	Ctr+Shift+Enter
Apposer (Frère)	Ctr+E
Ajouter enfant	Ctr+Alt+Enter
Wrapper dans Array	Ctr+W

Wrapper dans Objet	Ctr+Shift+W

Note : Si un composant est ajouté, par exemple, en tant que propriété (inséré, apposé ou ajouté en tant qu'enfant) et que vous souhaitez changer son type en tant qu'objet ou array, le faire en cliquant sur le symbole de type du composant et choisir le type que vous souhaitez.

Touches de commande dans les cellules

/	S'il est coloré l'affichage de Table est activé, sinon il est désactivé. Cliquer pour changer d'affichage
	Inverser les colonnes et les lignes
/	S'il est coloré, un filtre pour la table, l'objet, ou l'array est actif, sinon, le filtre est désactivé. Cliquer pour activer/désactiver. Pour éditer l'expression, double-cliquer
	Dans un array : apposer un item d'array dans la liste ou la table Dans un objet: apposer une paire `key:value` (en tant que ligne dans une liste, ou (dans un affichage de table) en tant que cellule d'une nouvelle colonne de table)
	Ajouter une paire vide `key:value` ; le type de la valeur est `string` par défaut
	Choisir le type de données d'une valeur de propriété ou saisir un commentaire ou une formule

13.5.3 Éditer le contenu du document

Éditer le contenu dans le Mode JSON Grid est simple : Double-cliquer à l'intérieur de la cellule qui doit être éditée et éditer le contenu tel que requis. De plus, le Mode JSON Grid propose les fonctions d'édition suivantes :

- Fonction de sélection du type ainsi que détection automatique du type, et conversions des valeurs
- Saisie semi-automatique dans les documents JSON qui sont basés dans le schéma JSON

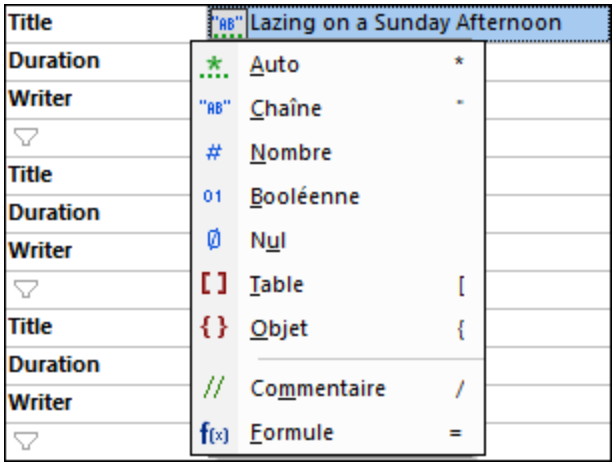
Sélection de type et détection automatique des types

Le Mode Grid JSON distingue entre les deux catégories de types :

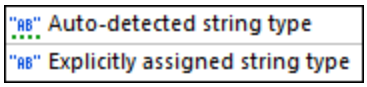
- *Types simples :* String, Nombre, Booléen, Nul
- *Types simples spéciaux :* Auto (qui est détecté depuis la valeur), Commentaire (JSONC, JSON5)
- *Types de conteneur :* Objet, Array
- *Types spéciaux de conteneur :* Formule

Le type d'une valeur est indiqué par un symbole qui se trouve devant la valeur *(voir la capture d'écran ci-dessous)*. Pour changer un type, il faut cliquer sur son symbole et sélectionner le type souhaité depuis le menu qui apparaît. Les symboles et raccourcis de types sont affichés dans la capture d'écran ci-dessous :

© 2016-2022 Altova GmbH



Les types détectés automatiquement ont des points verts en-dessous *(voir la capture d'écran ci-dessous)*. Un type explicite est un type que vous attribuez.



Les actions liées au type se produisent dans deux situations :

- Lorsqu'un document JSON est chargé : tous les types simples sont convertis en `Auto`, qui sont détectés automatiquement depuis les valeurs. Par exemple, **`"MyString"`** est automatiquement détecté en tant que type String, **`123`** en tant que type Nombre, **`true`** en tant que type Booléen, et **`null`** en tant que type Null. Dans les cas de strings ambigus, choisir le type explicitement.
- Lorsqu'une nouvelle structure de données ou une valeur est saisie : les types simples sont détectés automatiquement et le type est attribué automatiquement. Vous pouvez changer le type ultérieurement le cas échéant.

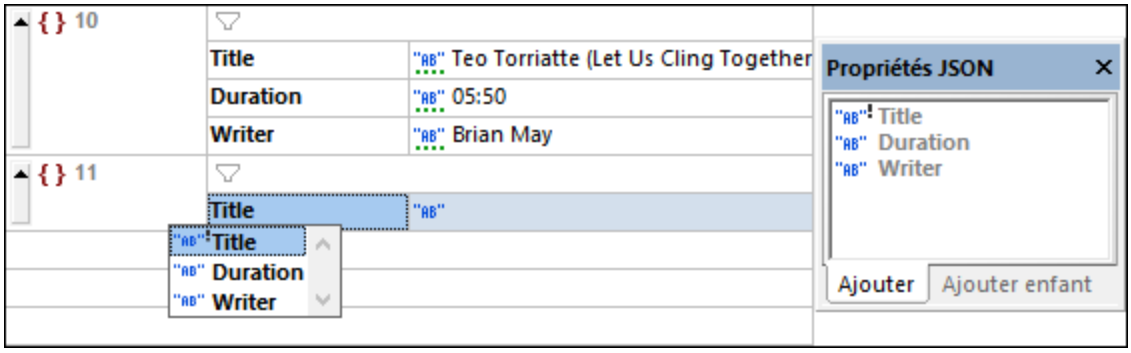
Note : les paramètres du Mode Grid JSON vous permettent de spécifier (i) la procédure à suivre pour changer les types lorsque de multiples cellules sont sélectionnées, et (ii) comment les valeurs de types atomiques devraient être traitées lorsque le type est changé en array ou objet.

Remplissage automatique

Le remplissage automatique est activé lorsque le document JSON (JSON instance/schéma ou schéma Avro) en cours d'édition est associé avec un schéma.

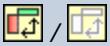
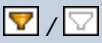
- Si le document est un schéma JSON, le remplissage automatique sera basé sur la version de schéma indiquée par le mot-clé **`$schema`**. Pour plus d'informations, voir aussi Version de schéma JSON.
- Si le document est une instance JSON/JSON5, alors un schéma JSON doit être attribué à l'instance pour que le remplissage automatique soit activé.
- Si le document est un document de données Avro en format JSON, alors un schéma Avro doit être attribué à l'instance pour que le remplissage automatique fonctionne.
- Si le document est un schéma Avro, il sera automatiquement associé avec un schéma pour le schéma Avro, et le remplissage automatique sera basé sur ce schéma.

Le remplissage automatique vous offre les options de saisie disponibles à l'emplacement du curseur. La fonction utilise (i) des fenêtres pop-up dans la fenêtre principale et (ii) des assistants à la saisie (*voir la capture d'écran ci-dessous*). Les pop-ups et assistants à la saisie affichent chacun des entrées qui sont valides à l'emplacement du curseur. Pour se déplacer d'une entrée à l'autre dans la liste pop-up, utilisez les touches de direction. Sélectionnez une entrée depuis la fenêtre pop-up ou double-cliquez dans l'assistant d'entrée pour l'insérer.



Boutons de commande dans la cellule

Dans le Mode JSON Grid, les touches dans les cellules de la grille proposent des fonctions qui sont disponibles dans le contexte de la cellule respective. L'utilisation de ces commandes de touche est une manière rapide et aisée d'effectuer des opérations utilisées habituellement. Les commande de touche « en cellule » disponibles sont recensées dans la table ci-dessous. Des commandes liées à l'écran sont affichées avec un fond bleu. Les commandes pour éditer le document (structure et contenu) sont affichées avec un fond beige. Pour les commandes supplémentaires dans un contexte de cellule particulier, utiliser le menu contextuel de la cellule. Note : Le nom d'une commande de touche d'une cellule est affiché en tant qu'infobulle lorsque vous passez avec la souris.

	Optimise les largeurs des colonnes de la grille conformément au contenu de la cellule.
	Si coloré, Affichage Table est activé, autrement il est désactivé. Cliquer pour passer à l'affichage
	Une commande bascule en haut à gauche de la cellule de la table. Bascule des lignes vers les colonnes et vice versa
	Si coloré, un filtre pour la table, un objet, ou un array est activé, autrement le filtre est désactivé. Cliquez pour désactiver/activer. Pour éditer l'expression, double-cliquez sur celle-ci
	Commande bascule pour word-wrapper les cellules. Si sélectionné, word-wrap est activé. Par défaut, seulement les contenus des items sont encapsulés. Si vous voulez en plus encapsuler les noms des items, choisissez cette option depuis la liste déroulante de l'icône.
	Dans un tableau : Apposer un item d'array dans la liste ou la table Dans un objet : Apposer une paire `key:value` (comme ligne dans une liste, ou (dans l'affichage de table) comme une cellule d'une nouvelle colonne de table)
	Ajouter une paire `key:value` vide ; le type de la valeur est un `string` par défaut
	Sélectionner le type de données d'une valeur de propriété, ou entrer un commentaire ou

© 2016-2022 Altova GmbH

	une formule

Note : pour une description d'autres commandes de la barre d'outils, voir la section Menu contextuelo dans le Mode Grille.

Valider sur modification

Le mode *Valider sur Édition* est activé par défaut. Lorsqu'il est activé, des vérifications de bonne formation et de validation sont effectuées pendant que vous modifiez un document dans le Mode JSON Grid. Pour que la validation d'un document JSON soit effectuée (en plus des vérifications de bonne forme), un Schéma JSON doit être attribué au document JSON. Les erreurs sont indiquées en affichant du texte erroné en rouge et en marquant l'emplacement avec un point d'exclamation rouge. Voir Valider des documents JSON pour plus d'informations.

Le mode *Valider sur Édition* peut être activé ou désactivé soit (i) via la commande de menu **XML | Valider sur Édition**, (ii) via la touche de la barre d'outils **Valider sur Édition**, ou (iii) via l'option *Sur Édition* des Paramètres de validation du dialogue Options.

Chercher et Remplacer

Vous pouvez utiliser les commandes **Chercher (Ctrl+F)** et **Remplacer (Ctrl+H)** du menu Éditer pour trouver du texte dans le Mode JSON Grid et le remplacer. Les résultats sont en surbrillance orange, avec les composants contenu également en surbrillance orange. Dans le cas de cellules contenant des strings de texte long, seule la cellule est en surbrillance.

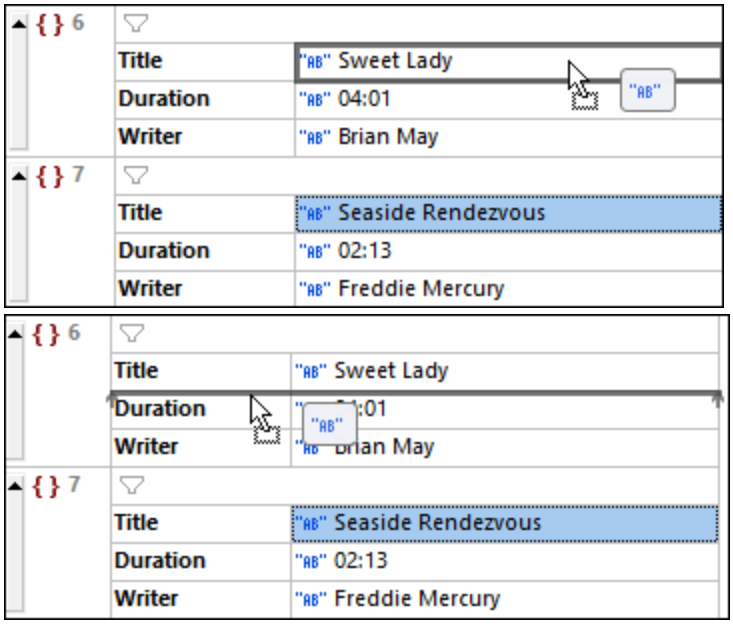
13.5.4 Glisser-déposer dans le Mode Grille

Le Mode Grille propose une superposition pour glisser très utile qui vous permet de glisser un fragment de document JSON dans le Mode Grille depuis un document qui est ouvert dans XMLSpy, une application externe ou même un site web.

Si vous placez le curseur sur un nœud cible, une superposition pour glisser apparait, qui non seulement fournit des informations sur ce qui qui a été glissé mais également sur la manière dont l'item sera créé lorsqu'il est déposé. Par exemple, dans la capture d'écran ci-dessous, la valeur `Seaside Rendezvous` est déposée (déplacée). Le popup `AB` indique que l'item est une valeur string.

- Si vous passez sur un champ de valeur (comme dans la capture d'écran à gauche), le champ entier est marqué, indiquant que le valeur du champ marqué sera remplacée par la valeur du champ glissé.
- Si vous passez sur un nom de nœud (tel que `Durée` dans la capture d'écran à droite), une ligne d'insertion apparait, indiquant que le string de valeur sera déposé à cet endroit pour créer un nœud de valeur d'une paire key–value. Dans ce cas, la clé sera un string vide.

Note : vous pouvez toujours annuler chaque étape avec **Ctrl+Z** ou la commande menu **Éditer | Annuler**.

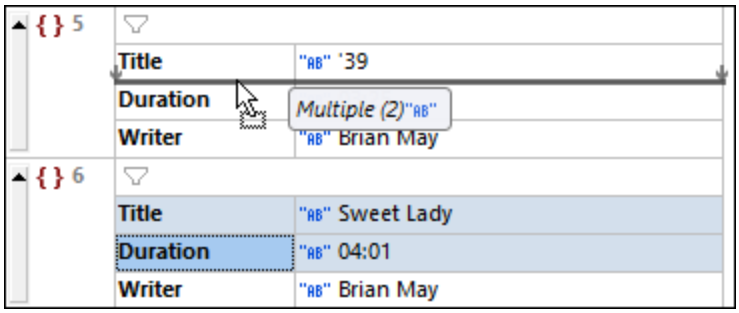


Information contenant une superposition pour glisser

Les informations suivantes concernant une superposition pour glisser seront affichées.

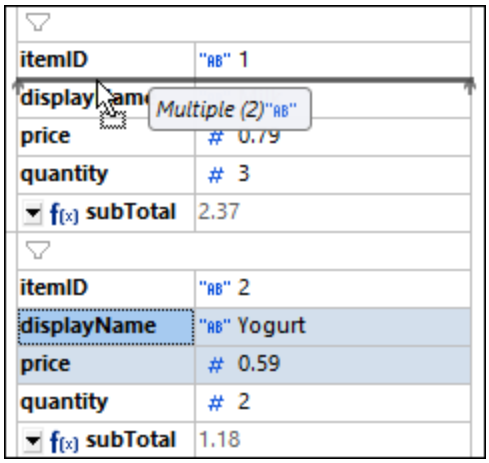
Mode Grille normal

- La valeur glissée dans un champ de valeur fait que la valeur source remplace la valeur dans le champ cible (*voir la capture d'écran ci-dessus gauche*).
- La valeur déposée dans une arborescence en tant que nœud crée le nœud de valeur d'une paire key–value (*voir la capture d'écran ci-dessus droite*). La clé sera un string vide.
- La superposition pour des nœuds multiples d'un seul type indique le type de nœud qui est déposé et le nombre de ces nœuds. Pour sélectionner un nœud (et pas sa valeur), veuillez noter que vous devez cliquer sur le nom du nœud. Dans la capture d'écran ci-dessous, l'information dans la superposition pour glisser indique que les deux paires key–value de type string seront déposées.



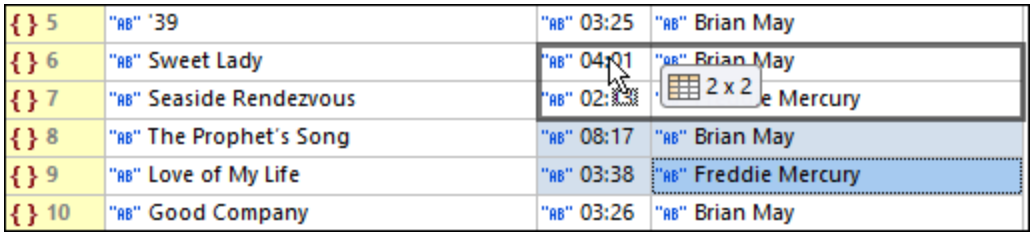
- La superposition pour les nœuds multiples indique que les nouveaux nœuds seront créés avec un type string par défaut (*voir la capture d'écran ci-dessous*). Vous pouvez modifier le type de nœuds glissés en cliquant sur les icônes respectives *Éditer Type*. Pour sélectionner un nœud (et pas sa valeur), veuillez noter que vous devez cliquer sur le nom du nœud.

© 2016-2022 Altova GmbH



Affichage de la table

- Lorsque les valeurs sont glissées, le nombre de cellules sélectionnées est indiqué par **`colonne x lignes`** (*voir la capture d'écran ci-dessous*). La matrice des cellules glissées remplacera une matrice correspondante. La cellule cible (la cellule sur laquelle vous déposez la matrice) recevra la valeur de la cellule en haut à gauche de la matrice. Les autres cellules glissées rempliront les cellules vers la droite et vers le bas de la cellule cible. Par exemple, dans la capture d'écran ci-dessous, les cellules bleues sont glissées et déposées sur la cellule 6 **`Durée`**. Ceci a pour conséquence que celles-ci remplacent les cellules qui ont été marquées. Si la matrice des cellules glissées dépasse les limites de la table, le nombre approprié de colonnes et/ou de lignes est ajouté pour accommoder les cellules glissées. Dans ce cas, les limites qui seront élargies sont indiquées par des pointillés.



- Lorsque la cible est l'arborescence du nœud —et non la cellule—alors le nœud ajouté est indiqué, ensemble avec (i) les colonnes qui seront ajoutées (dans des crochets), et (ii) le nombre d'instances du nouvel item. Par exemple, dans la capture d'écran ci-dessous, un array est ajouté qui contient deux objets, dont chacun est doté d'une paire key–value **`Titre`**, **`Durée`**, et **`Artiste`**.



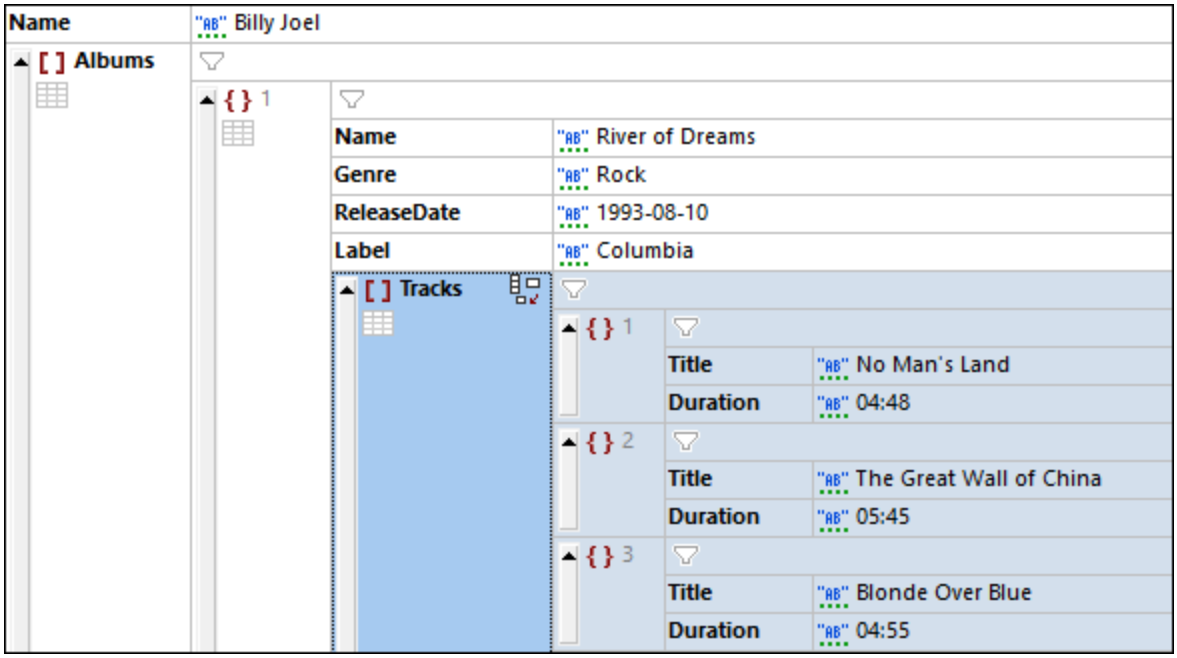
- Une colonne de table peut être déplacée en la sélectionnant (cliquez sur son en-tête à cet égard) et glissée dans l'en-tête de la colonne adjacente à laquelle vous voulez la déplacer. Une ligne d'insertion indiquera de quel côté de la colonne cible la colonne déplacée sera placée.

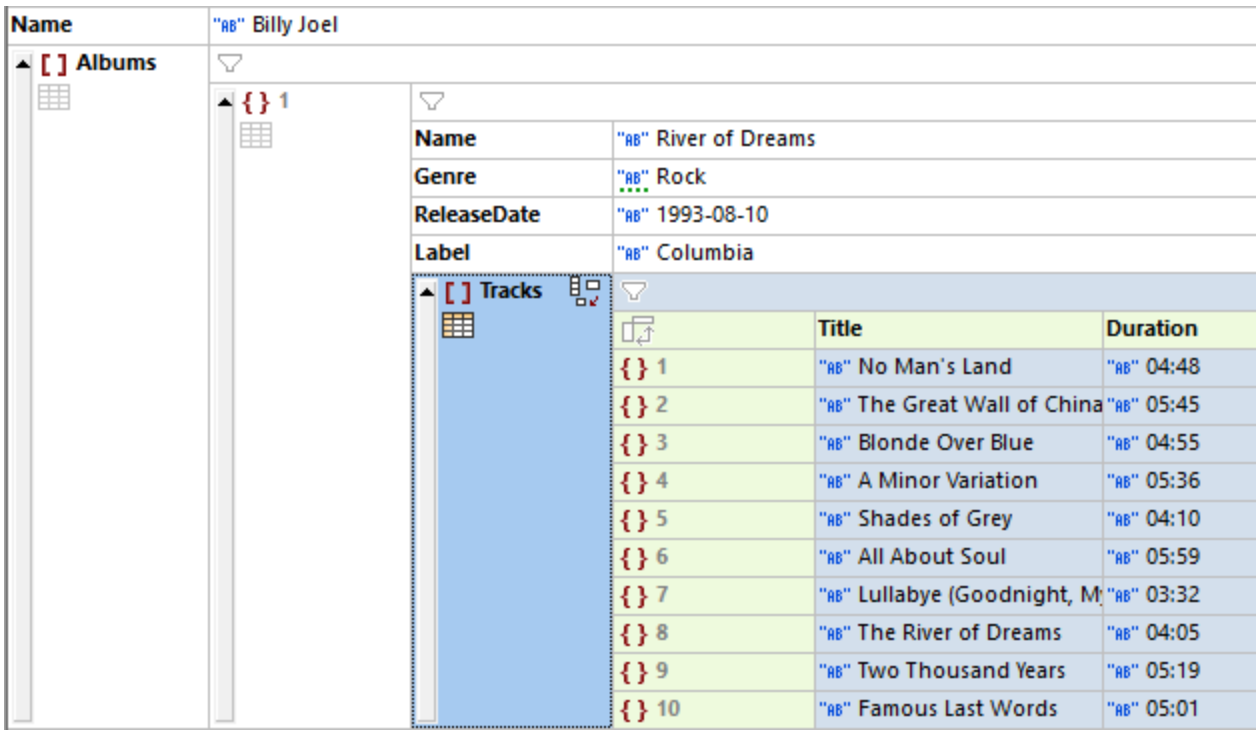
13.5.5 Affichage table

Mode Affichage de la table

Les objets et arrays qui contiennent au moins un objet ou array peuvent être affichés soit comme liste *(mis en surbrillance dans la capture d'écran à gauche)* ou comme table *(mis en surbrillance dans la capture d'écran à droite)*. Pour basculer entre ces deux modes, cliquez sur l'icône de la table qui est située en dessous de l'icône de l'objet ou de l'array *(voir la capture d'écran)*. Dans les tables d'array, les items d'array sont affichés en tant que lignes. Dans la capture d'écran en dessous à droite, l'array `Tracks` array est affiché en tant que table. L'array consiste en objets enfants, qui sont affichés en tant que lignes. Chaque propriété d'objet (`Titre` et `Durée` dans la capture d'écran ci-dessous) est affichée en tant que colonne, avec les nom des propriétés communes étant affichés comme en-têtes des colonnes. Pour basculer des lignes et colonnes, cliquez sur l'icône dans la cellule gauche du haut de la table.

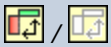
© 2016-2022 Altova GmbH





Les icônes pour afficher et éditer dans l'Affichage de la table

Les icônes affichées ci-dessous sont disponibles dans l'Affichage de la table et proposent des fonctions d'affichage et d'édition.

	Optimise les largeurs des colonnes de la grille conformément au contenu de la cellule.
	Si coloré, Affichage Table est activé, autrement il est désactivé. Cliquer pour passer à l'affichage
	Une commande bascule en haut à gauche de la cellule de la table. Bascule des lignes vers les colonnes et vice versa
	Si coloré, un filtre pour la table, un objet, ou un array est activé, autrement le filtre est désactivé. Cliquez pour désactiver/activer. Pour éditer l'expression, double-cliquez sur celle-ci
	Commande bascule pour word-wrapper les cellules. Si sélectionné, word-wrap est activé. Par défaut, seulement les contenus des items sont encapsulés. Si vous voulez en plus encapsuler les noms des items, choisissez cette option depuis la liste déroulante de l'icône.
	Dans un tableau : Apposer un item d'array dans la liste ou la table Dans un objet : Apposer une paire `key:value` (comme ligne dans une liste, ou (dans l'affichage de table) comme une cellule d'une nouvelle colonne de table)
	Ajouter une paire `key:value` vide ; le type de la valeur est un `string` par défaut
	Sélectionner le type de données d'une valeur de propriété, ou entrer un commentaire ou une formule

Éditer dans l'Affichage de la table

Dans l'Affichage de la table, vous pouvez réaliser des actions d'édition décrites ci-dessous.

Ajouter des lignes dans une table
Pour ajouter des lignes, vous avez le choix entre plusieurs méthodes :

- *Ajouter une ligne enfant à la table :* Sélectionnez la table et, dans le menu contextuel, choisissez la commande **Ajouter Enfant (Ctrl+Shift+Enter**). De manière alternative, cliquez sur l'icône de la table **Apposer Enfant** *(voir la liste des icônes ci-dessus)*. Une ligne sera apposée en bas de table.
- *Apposer une ligne frère à la table lorsqu'une ligne est choisie :* Sélectionnez une ligne et, dans le menu contextuel, choisissez la commande **Apposer (Ctrl+Enter**). Une ligne sera apposée en bas de table.
- *Insérer une ligne frère au-dessus de la ligne sélectionnée :* Sélectionnez une ligne et, dans le menu contextuel, choisissez la commande **Insérer (Ctrl+ Alt+Enter**).

Saisir ou éditer une valeur de propriété
Sélectionnez la cellule de la table dans laquelle la valeur de la propriété est située et saisissez la valeur. Le type de données sera détecté automatiquement si la valeur est un string, un nombre, un booléen ou un null. S'il y a une ambiguïté, le type est défini sur `string`. Vous pouvez écraser la sélection automatique en utilisant l'icône datatype-selection *(voir la liste des icônes ci-dessus)*. De manière alternative, vous pouvez utiliser un raccourci approprié (*voir la table des raccourcis ci-dessous*).

*	Auto- (détecte des string, nombre, booléen, null, et définit en conséquence)
"	String

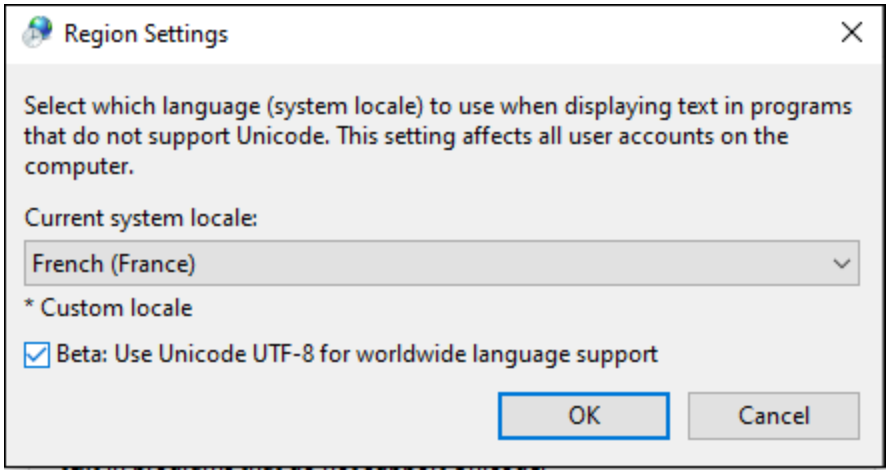
© 2016-2022 Altova GmbH

[	Array
{	Objet
/	Commenter (le document doit être JSONC ou JSON5)
=	Formule (le document doit être JSONC ou JSON5)

Ajouter une cellule de ligne vide dans une nouvelle colonne
Vous pouvez ajouter une nouvelle cellule dans une ligne. La nouvelle cellule fera partie d'une colonne nouvellement créée. Pour ajouter une nouvelle cellule, sélectionnez la ligne de votre choix et, dans le menu contextuel, sélectionnez la commande **Ajouter enfant (Ctrl+Shift+Enter**). De manière alternative, cliquez sur l'icône **Ajouter la cellule de la ligne** *(voir la liste des icônes ci-dessus)*. La cellule de la ligne sera créée dans une colonne. Le type de la valeur dans la cellule est `string` par défaut. Saisissez la valeur de la propriété dans la cellule et le nom de la propriété en tant qu'en-tête de la colonne. Les autres cellules dans la colonne nouvellement créée seront vides.

Trier les lignes de table sur la base des valeurs d'une colonne sélectionnée
Vous pouvez trier les lignes d'une table sur la base des valeurs relatives d'un de ses nœuds enfant (une colonne). Par exemple, vous pouvez trier les lignes sur la base de la colonne *DernierNom* pour vous donner les éléments répétitifs de la table triée par ordre alphabétique. Pour trier une colonne, sélectionnez l'en-tête de colonne et cliquez sur la commande barre d'outils dans le Mode Grid **Trier par ordre croissant** ou **Trier par ordre décroissant**. Ces commandes sont aussi disponibles dans le menu JSON.

Trier l'ordre dans certaines langues, en particulier celles qui n'ont pas d'alphabet latin, peut être bénéfique en activant la prise en charge beta Unicode UTF-8 dans le dialogue des paramètres de la région linguistique de Windows 10 (ou supérieure). Pour ce faire, suivez les étapes suivantes : Allez au dialogue des *Paramètres* de Windows et recherchez les *Paramètres de langue*. Sous *Paramètres associés*, cliquez sur *Paramètres de langue administrative*. Dans le dialogue Région qui apparaît, allez à l'onglet *Administrative* et sous *Language for non-Unicode programs*, cliquez sur *Change system locale*. Dans le dialogue des paramètres de la région qui apparaît (*voir la capture d'écran ci-dessous*), sélectionnez l'option *Beta : Use Unicode UTF-8 for worldwide language support* et cliquez sur **OK**.



Affichage de la table et applications externes

Vous pouvez tirer profit de la structure de la table pour échanger les données entre l'affichage de la table et une application spreadsheet (telle que MS Excel). Pour déplacer les données de l'affichage de la table, suivez les instructions suivantes :

1. Dans l'affichage de la table, sélectionnez les nœuds que vous voulez copier. Procédez en cliquant sur les cellules elles-mêmes, les en-tête colonnes, les en-tête lignes, ou sur la table entière. Si vous cliquez sur la table entière ou les en-tête de colonnes, les en-tête de colonnes sont également copiées ; autrement, elles ne le sont pas. (Dans la capture d'écran ci-dessous, les lignes 1 à 8 sont sélectionnées, ensemble avec leurs en-têtes de colonnes.)



[ ] Tracks	Title	Duration	Writer
{ } 1	"AB" Death on Two Legs	"AB" 03:43	"AB" Freddie Mercury
{ } 2	"AB" Lazing on a Sunday Afternoon	"AB" 01:08	"AB" Freddie Mercury
{ } 3	"AB" I'm in Love with My Car	"AB" 03:05	"AB" Roger Taylor
{ } 4	"AB" You're My Best Friend	"AB" 02:50	"AB" John Deacon
{ } 5	"AB" '39	"AB" 03:25	"AB" Brian May
{ } 6	"AB" Sweet Lady	"AB" 04:01	"AB" Brian May
{ } 7	"AB" Seaside Rendezvous	"AB" 02:13	"AB" Freddie Mercury
{ } 8	"AB" The Prophet's Song	"AB" 08:17	"AB" Brian May

2. Sélectionnez la commande du menu contextuel Copier | Copier en tant que texte séparé par tabulation.
3. Collez les données directement dans le programme de spreadsheet.

	A	B	C
1	Title	Duration	Writer
2	Death on Two Legs	3:43	Freddie Mercury
3	Lazing on a Sunday Afternoon	1:08	Freddie Mercury
4	I'm in Love with My Car	3:05	Roger Taylor
5	You're My Best Friend	2:50	John Deacon
6	'39	3:25	Brian May
7	Sweet Lady	4:01	Brian May
8	Seaside Rendezvous	2:13	Freddie Mercury
9	The Prophet's Song	8:17	Brian May

L'échange de données fonctionne dans les deux sens. Vous pouvez aussi copier des données d'une spreadsheet à l'affichage de la table. Pour ce faire, procédez comme suit :

1. Sélectionnez une plage dans l'application externe et copiez-la dans le presse-papiers (avec **Ctrl+C**).
2. Sélectionnez une cellule individuelle dans l'affichage de la table.
3. Collez les données copiées avec **Ctrl+V**.

Les données seront collées dans la table avec une structure correspondant à la structure originale et en commençant par la cellule choisie dans le Mode Table. Notons les points suivants :

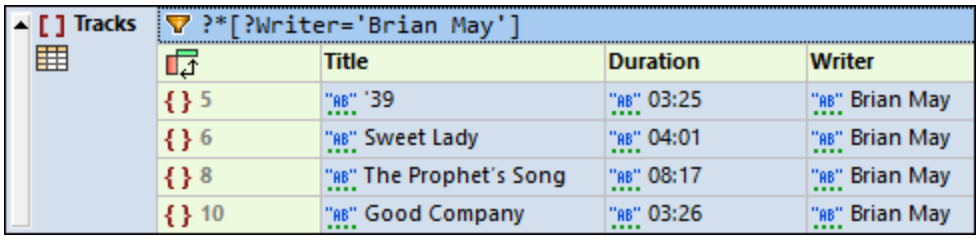
© 2016-2022 Altova GmbH

- Si les données sont copiées dans une cellule qui permet la création de nouvelles lignes et/ou colonnes sans rendre les données existantes invalides, de nouvelles lignes et/ou colonnes seront créées.
- Des lignes sont créées en tant qu'objets nouveaux, alors que les colonnes sont créées en tant que propriétés des objets (ligne) d'une table.
- Si la structure de la table ne peut pas être modifiée de manière valide, les nouvelles données écrasent les données d'origine de la cellule sélectionnée en tant que texte.

Pour plus de tâches complexes liées à l'échange de données, utilisez les fonctions uniques importation/exportation de XMLSpy.

13.5.6 Filtres

Les filtres vous permettent de filtrer l'affichage des objets et arrays. Par exemple, dans la capture d'écran ci-dessous, un filtre (qui est une expression XQuery 3.1) a été appliqué à un array pour que seules les pistes écrites par Brian May sont affichées. En termes JSON, seuls les objets enfants des **`Tracks`** sont affichés qui ont une propriété **`Artiste`** contenant la valeur string `Brian May`. Le filtre de l'expression XQuery recherche tous les objets enfants des **`Tracks`** et sélectionne ceux pour lesquels une recherche de la propriété de l'**`Artiste`** correspond au string `'Brian May'`.



Chaque filtre est exécuté de manière indépendante et n'est pas affecté par d'autres filtres ou des formules dans le document.

Pour toute information sur la construction d'expressions XQuery pour des documents JSON, voir la section Expressions XQuery pour JSON.

Note : lorsque vous saisissez des expressions pour des filtres et des formules en Mode Grid, vous allez éventuellement vouloir afficher une expression sur plusieurs lignes pour une meilleure lisibilité. Pour ajouter une nouvelle ligne dans l'expression, appuyez sur **Ctrl+Enter**.

Configurer un filtre

Pour définir un filtre, cliquez de la touche droite sur l'élément que vous souhaitez filtrer, sélectionnez **Filtre** depuis le menu contextuel qui apparaît, saisissez l'expression XQuery, et cliquez sur **Entrer**. La cellule *Filtre* est indiquée par l'icône *(voir la capture d'écran ci-dessus)*. Vous pouvez activer/désactiver le filtre en cliquant sur l'icône *Filtre*.

Veuillez noter les points suivants concernant les filtres :

- Les filtres peuvent être appliqués uniquement aux objets et arrays.

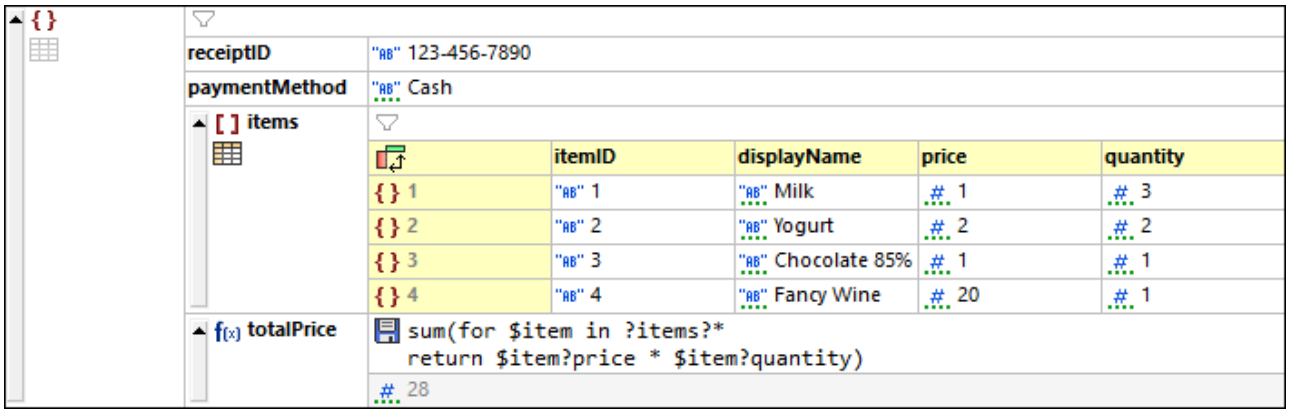
- Le nœud de contexte de l'expression XQuery du filtre est le nœud actuel. Dans la capture d'écran ci-dessus, par exemple, le nœud de contexte de l'expression XQuery du filtre est le nœud `Tracks`.
- Les filtres ne sont **pas stockés** dans le document JSON en tant que contenu JSON, mais peuvent être gardés dans un fichier de métadonnées spécial pur une utilisation ultérieure. Si les filtres sont stockés, il seront automatiquement appliqués à l'objet/array respectif quand le document est rouvert dans le Mode JSON Grid.
- Les filtres peuvent être imbriqués. Un filtre imbriqué peut être appliqué à un contenu filtré du filtre parent.
- Pour ajouter une nouvelle ligne dans une expression, appuyez sur **Ctrl+Enter**.

13.5.7 Formules

Une formule est une expression XQuery 3.1 qui génère un résultat (soit un nodeset ou des calculs) pour un affichage en Mode JSON Grid. Dans la capture d'écran ci-dessous, par exemple, le prix total des items 1 à 4 est calculé et le résultat (`28`) est affiché sur une ligne séparée. Chaque formule est exécutée de manière indépendante et n'est pas affectée par d'autres filtres ou formules dans le document.

Pour toute information sur la construction d'expressions XQuery pour des documents JSON, voir la section Expressions XQuery pour JSON.

Note : lorsque vous saisissez des expressions pour des filtres et des formules en Mode Grid, vous allez éventuellement vouloir afficher une expression sur plusieurs lignes pour une meilleure lisibilité. Pour ajouter une nouvelle ligne dans l'expression, appuyez sur **Ctrl+Enter**.



Si le document est un document JSON5 ou JSONC, alors :

- La formule sera écrite dans le document en tant que commentaire JSON
- Facultativement, **le résultat** de la formule peut être stocké en tant code (non commenté) dans le document JSON. Pour ce faire, cliquez sur l'icône du disque de la formule *(voir la capture d'écran ci-dessous)*. Si la formule n'est pas stockée en tant que contenu JSON, elle sera stockée en tant que document de métadonnées pour l'affichage dans le Mode Grille. Notez que l'icône du disque est uniquement disponible dans les documents en JSON5 et JSONC ; elle n'est pas affichée dans d'autres types de document JSON.

Le document JSON est affiché dans la capture d'écran ci-dessus, y compris la formule stockée et le résultat de la formule.

© 2016-2022 Altova GmbH

*Notez que la **formule** est stockée en tant que commentaire JSON mais le **résultat de la formule** est stockée en tant que code JSON direct. Le code ci-dessous (avec le résultat de la formule étant stocké) est le résultat de l'icône du disque de la formule lorsque vous cliquez dessus.*

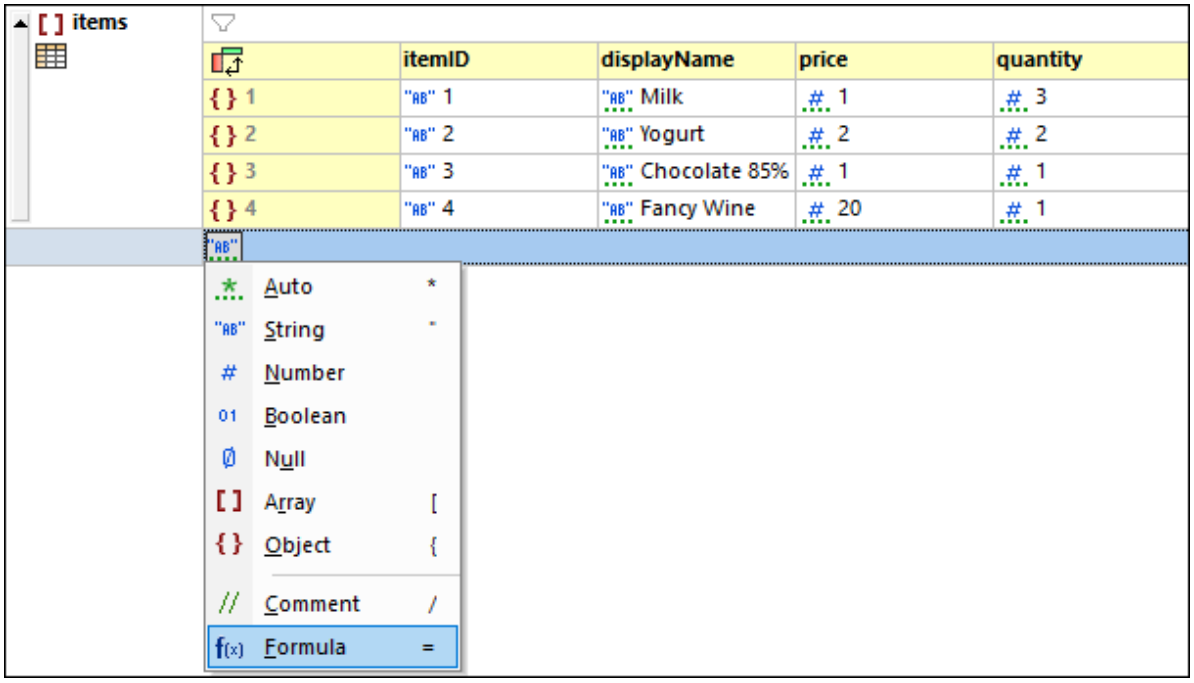
```
{
	"receiptID": "123-456-7890",
	"paymentMethod": "Cash",
	"items": [
		{
			"itemID": "1",
			"displayName": "Milk",
			"price": 1,
			"quantity": 3
		},
		{
			"itemID": "2",
			"displayName": "Yogurt",
			"price": 2,
			"quantity": 2
		},
		{
			"itemID": "3",
			"displayName": "Chocolate 85%",
			"price": 1,
			"quantity": 1
		},
		{
			"itemID": "4",
			"displayName": "Fancy Wine",
			"price": 20,
			"quantity": 1
		}
	],
	//(:altova_xq_embed:)totalPrice(:altova_xq_key:)sum(for $item in ?items?* return
$item?price * $item?quantity)
	"totalPrice": 28
	//(:altova_xq_end:)
}
```

Dans la capture d'écran ci-dessus, la formule résume les membres d'une séquence. Ces membres sont chacun des produits des valeurs de **`prix`** et de **`quantité`** de chaque projet contenu dans le tableau des **`items`**. L'itération de spécifier chaque objet et de l'attribuer, à son tour, à la variable **`$item`** est spécifiée par : **`pour $item dans ?items?*`**. Il est important de noter ici le nœud contextuel, qui est le parent de la formule—et, en conséquence, le parent du nœud des **`items`**. Chaque produit est obtenu en vérifiant le **`prix`** et la **`quantité`** des nœuds enfants de l'objet actuellement dans la variable **`$item`**, et en multipliant ces deux valeurs l'une avec l'autre. Les produits ainsi obtenus sont les membres de la séquence, qui sont ensuite additionnés afin de générer le prix total.

Créer une formule

Pour créer un formule, procéder comme suit :

1. Sélectionnez le nœud pour lequel vous voulez ajouter la formule, soit en tant que frère ou enfant. Cliquez avec la touche droite, et ajoutez le frère ou l'enfant (selon ce que vous voulez). Lorsque vous décidez où vous voulez que la formule apparaisse (en tant que frère ou enfant), n'oubliez pas que le nœud contextuel de l'expression Xquery de la formule sera le nœud parent de la formule. Par exemple, dans l'expression XQuery affichée dans la capture d'écran ci-dessus, le nœud contextuel est le nœud parent de la formule (`Prixtotal`) et de son frère, l'array des `items`. Pour créer la formule dans la capture d'écran ci-dessus, un nœud frère a été apposé dans l'array des `items` *(voir la capture d'écran ci-dessous)*.
2. Changer le type du nœud de la Formule *(voir la capture d'écran ci-dessous)*.



3. Double-cliquez dans la cellule qui contient l'icône `f(x)` et saisissez le nom de la formule *(voir la capture d'écran ci-dessus)*. Si le document est un document JSON5 ou JSONC, alors une icône de disque est affichée. Vous pouvez cliquer sur l'icône pour enregistrer le résultat de la formule vers le document.



4. Par défaut, l'expression XQuery est le string `'XQuery'`, donc le résultat sera le string `XQuery` (affiché dans la cellule ci-dessous sous l'expression). Saisissez votre expression Xquery en double-cliquant dans la cellule de l'expression et tapez l'expression, puis cliquez sur **Enter**. Ainsi, la formule sera-t-elle évaluée et son résultat affiché. Les autres formules ne seront pas modifiées.
5. Si le document contient de multiples formules, cliquez sur la commande de menu **JSON | Tout réévaluer** afin de mettre à jour les résultats de toutes les formules. La commande est particulièrement utile si les formules dans le document recherchent des données qui changent de manière dynamique (par exemple, des taux de change).

Veuillez noter les points suivants pour les formules :

- Le nœud du contexte de l'expression XQuery de la formule est le nœud parent du nœud de la formule.

 © 2016-2022 Altova GmbH

- Les formules sont stockées dans les métadonnées du document (situé dans un fichier de métadonnées JSON au niveau de l'application). Si le document est en JSON5 ou JSONC, la formule est enregistrée en plus en tant que commentaire dans le contenu JSON.
- Le résultat généré par une formule est affiché dans la cellule en dessous de l'expression XQuery. Dans le cas de JSON5 et JSONC, le résultat peut être stocké dans le document en cliquant sur l'icône de disque à côté de l'expression XQuery. Que vous ayez cliquer sur le disque ou non, le résultat de la formule sera calculé et stocké dans les métadonnées du document.

	Le résultat de la formule n'est pas enregistré dans le contenu JSON ; cliquez pour l'enregistrer. *Uniquement dans JSON5 et JSONC.*
	Le résultat de la formule est enregistré dans le contenu JSON ; cliquez pour ne pas l'enregistrer. *Uniquement dans JSON5 et JSONC*

- Lorsque le résultat est un calcul et qu'il est stocké comme contenu, il est stocké en tant que propriété, qui porte le nom que vous avez assigné à la formule. Par exemple, dans la formule `Prixtotal` décrite ci-dessus, le résultat est stocké comme suit : `"totalPrice": 28`.
- Notez la différence : Dans les documents en JSON5 et JSONC, les formules sont enregistrées en tant que commentaires JSON, leurs résultats sont enregistrés en tant que propriétés JSON.

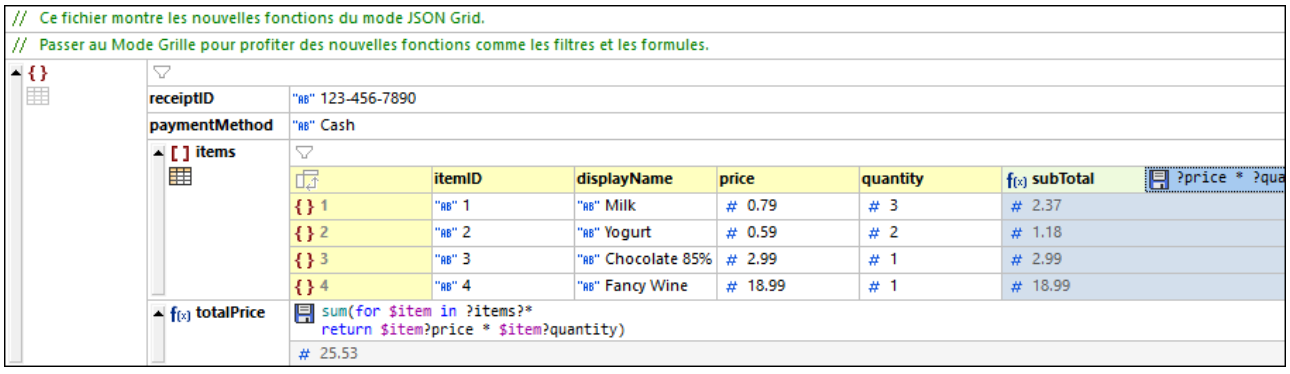
Note : dans le dialogue des paramètres du Mode JSON Grid, vous pouvez spécifier si les formules sont stockées dans les documents JSON5 et JSONC. L'option de stocker les formules dans le document est sélectionnée par défaut.

Note : dans le dialogue des paramètres du Mode JSON Grid, vous pouvez également : (i) définir les options de métadonnées (si les formules doivent être stockées automatiquement dans le fichier de métadonnées, ou stockées uniquement sur demande, ou si elles ne doivent pas du tout être stockées).

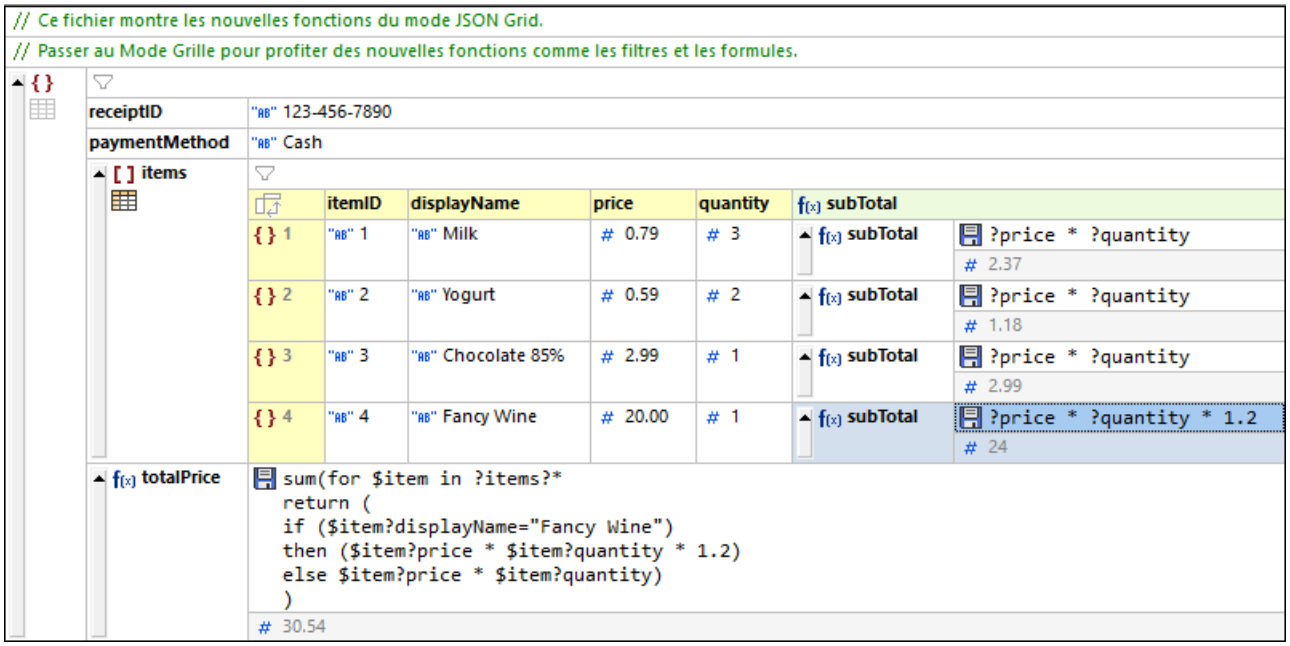
Formules dans les tables

Si toutes les cellules d'une colonne de table (in Affichage Table) contiennent la même formule, la formule est affichée une seule fois—dans l'en-tête de la colonne *(voir la capture d'écran ci-dessous)*. Les *résultats* du calcul de la formule sont, toutefois, affichés dans les cellules respectives.

La formule dans l'en-tête de la colonne est une représentation du Mode Grid. Dans le contenu du document JSON (en Mode Texte), la formule est répétée pour chaque item ligne-table.



Même si une seule formule est différente (telle que dans cellule en surbrillance de la capture d'écran ci-dessous), chaque formule sera affichée dans sa cellule respective.



Réévaluer toutes les formules
Pour mettre à jour les résultats de toutes les formules dans le document, cliquez sur la commande du menu **JSON | Tout réévaluer**.

Procédures pour les actions pertinentes

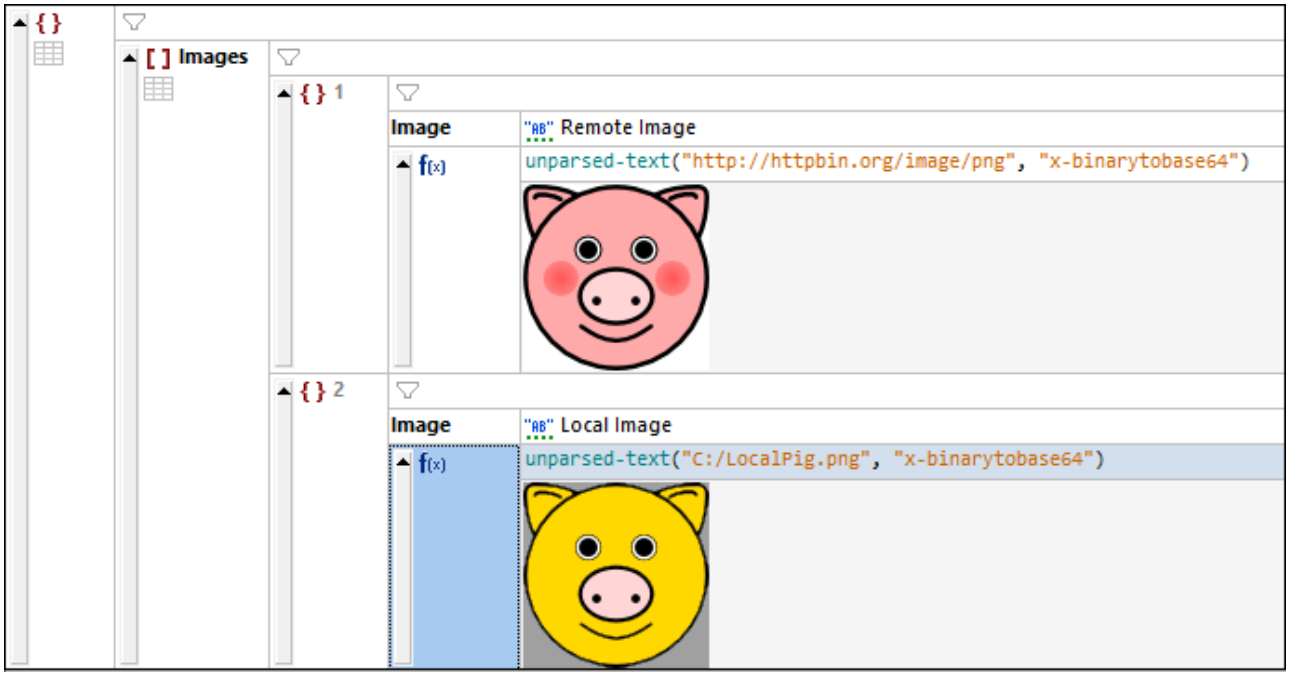
- Ajouter une nouvelle colonne vide à la table comme suit : Basculez vers Affichage de la liste depuis l'Affichage de la table, cliquez avec la touche droite sur n'importe quelle paire `key:value` dans l'affichage de la liste, et apposez ou insérez une nouvelle paire `key:value`par le biais du menu contextuel de l'item. Lorsque vous rebasculez vers l'Affichage de la table, une nouvelle colonne est créée pour la nouvelle `key:value` paire qui a été apposée/insérée. Vous pouvez désormais éditer la colonne dans le Mode Grid.
- Si toutes les formules d'une colonne de table sont les mêmes pour que la formule apparaisse dans l'en-tête, et vous voulez créer maintenant une formule différente pour une cellule individuelle, basculez vers l'Affichage de la liste et éditer la formule de cette cellule. Si vous retournez à Affichage Table, les formules seront affichées dans les cellules individuelles (pour toutes les cellules).
- Pour ajouter une nouvelle ligne dans une expression, appuyez sur **Ctrl+Enter**.

13.5.8 Images

Les images peuvent être affichées dans le JSON Grid View *(voir capture d'écran ci-dessous)*. L'image est convertie automatiquement en une image encodée en Base64 puis affichée.

Pour charger un image saisir l'expression suivante : `unparsed-text("<Image-URL>", "x-binarytobase64")`

© 2016-2022 Altova GmbH



Des URL d'image peuvent être données sous les formes suivantes, y compris en tant que chemins relatifs :

- `http://httpbin.org/image/png`
- `file:///c:/LocalPig.png`
- `C:/LocalPig.png`
- `LocalPig.png`

Enregistrer un string encodé en Base64 en tant qu'image

Pour enregistrer un string encodé en Base64 dans son format d'image, cliquez avec la touche de droite sur l'image ou sa cellule et sélectionnez la commande **Enregistrer sous Image**. (Notez que l'image est en fait encodée en tant que string en Base64 dans le fichier actif mais affichée en mode Grille en tant qu'image.) Dans le dialogue qui apparaît, sélectionnez l'emplacement sous lequel vous souhaitez enregistrer l'image et saisissez un nom pour le fichier d'image. L'extension du fichier d'image (`.png`, `.gif`, `.svg`, etc.) sera détectée automatiquement depuis l'encodage Base64 et apparaîtra dans le dialogue Enregistrer. Une fois terminé, cliquer **Enregistrer**.

Cette action peut également être exécutée par le biais de la commande de menu **Editer | Enregistrer sous Image**.

13.5.9 Graphiques

Des graphiques peuvent être créés dans le Mode JSON Grid en utilisant l'extension Altova XPath/XQuery nommée **`altovaext: graphique`** (*voir capture d'écran ci-dessous*). Cette extension est décrite ci-dessous. Elle est aussi décrite ainsi que d'autres extensions de graphique dans la section Fonctions Graphiques. L'extension **`altovaext:chart`** affichée dans la capture d'écran ci-dessous est utilisée dans une expression XQuery `Let` qui est définie dans une formule de Mode JSON Grid . Le graphique est affiché en tant qu'une

image en-dessous de la formule. (Cet exemple de graphique se trouve dans le fichier **Chart.jsonc**, qui se trouve dans le *dossier Exemples* de votre dossier (Mes) Documents et est aussi accessible par le *projet Exemples*.)

© 2016-2022 Altova GmbH

{ }

[ ] Temperatures

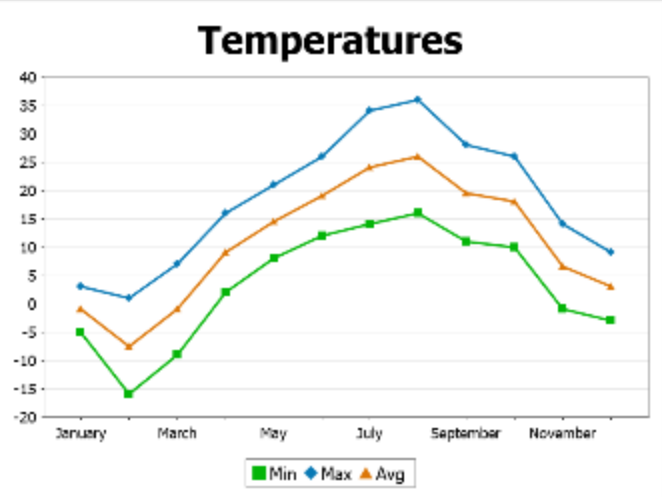
	Month	Min	Max
{ } 1	"AB" January	# -5	# 3
{ } 2	"AB" February	# -16	# 1
{ } 3	"AB" March	# -9	# 7
{ } 4	"AB" April	# 2	# 16
{ } 5	"AB" May	# 8	# 21
{ } 6	"AB" June	# 12	# 26
{ } 7	"AB" July	# 14	# 34
{ } 8	"AB" August	# 16	# 36
{ } 9	"AB" September	# 11	# 28
{ } 10	"AB" October	# 10	# 26
{ } 11	"AB" November	# -1	# 14
{ } 12	"AB" December	# -3	# 9

{ } ChartConfig

// Title is optional, you can remove it	
title	"AB" Temperatures
// Try modifying the kind of this chart.	
kind	"AB" LineChart
width	# 800
height	# 600

f(x) Chart

```
let $temps := ?Temperatures?* return
altovaext:chart(?ChartConfig,
(
    (:name, X-axis,       Y-axis   :)
    ['Min', $temps?Month, $temps?Min],
    ['Max', $temps?Month, $temps?Max],
    (: Calculate average per each min/max
       using mapping operator ! :)
    ['Avg', $temps?Month, $temps ! avg((?Min, ?Max))]
))
```



- *Liste des expressions XQuery pour générer le graphique qui est affiché dans la capture d'écran ci-dessus*

```
let $temps := ?Temperatures?* return
altovaext:chart(?ChartConfig,
(
    (:name, X-axis,       Y-axis   :)
    ['Min', $temps?Month, $temps?Min],
    ['Max', $temps?Month, $temps?Max],
    (: Calculer la moyenne pour chaque min/max
       utiliser l'opérateur de mappage ! :)
    ['Avg', $temps?Month, $temps ! avg((?Min, ?Max))]
))
```

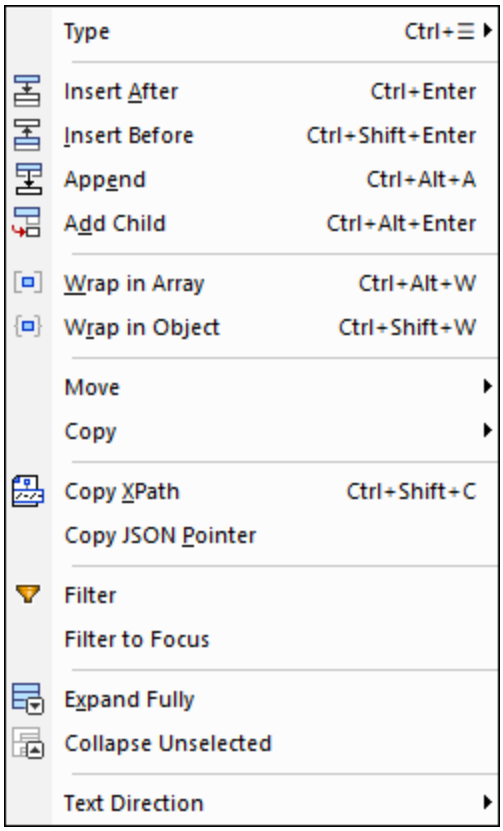
Utiliser l'extension Graphiques Altova

- La fonction d'extension de graphique `altovaext:chart` doit utiliser le préfixe d'espace de noms `altovaext:`.
- La fonction `altovaext:chart` prend deux arguments : (i) information de configuration de graphique et (ii) information de séries de données de graphique.
- L'information de la configuration du graphique est le premier argument du `altovaext:chart`. Il s'agit d'une série non ordonnée de quatre paires de clé-valeur. Ces paires sont pour (i) le titre du graphique (clé est `title`), (ii) le type de graphique, comme des camemberts, des graphiques à lignes, etc. (clé est `kind`; voir Fonctions de Graphiques pour les types disponibles), (iii) largeur de graphique en pixels (uniquement entiers ; clé est `width`), et (iv) hauteur de graphique en pixels (uniquement entiers, clé est `height`). Dans l'exemple de la capture d'écran ci-dessus, l'information de configuration est stockée dans l'objet `ChartConfig`, qui est référencé dans la fonction `altovaext:chart`.
- La série de données de graphique est le second argument de `altovaext:chart`. Chaque série de données est un array de taille 3: (i) le nom de la série, (ii) les valeurs de l'axe x, (ii) les valeurs de l'axe y. Si vous souhaitez créer plusieurs séries (par exemple où chaque série représente une ligne, comme dans l'exemple ci-dessus), créez une séquence de plusieurs arrays. L'exemple ci-dessus contient trois séries de données ; pour les températures minimum, maximum et moyenne. Les données pour les axes X et Y sont référencées depuis l'array nommé `Temperatures`.

Pour plus d'informations, voir la section Fonctions Graphique.

13.5.10 Menu contextuel dans le Mode Grille

Lorsque vous cliquez avec la touche de droite sur un composant dans le Mode JSON Grid, un menu contextuel *(capture d'écran ci-dessous)* apparaît qui vous propose des commandes pour modifier l'affichage et l'édition du document. Le menu contextuel peut aussi être accédé en appuyant sur la touche Menu. Les commandes du menu contextuel sont décrites ci-dessous.



Type
Passer la souris sur la commande **Type** pour afficher un sous-menu dans lequel vous pouvez choisir le type du composant. Vous pouvez aussi accéder au sous-menu du Type avec le raccourci **Ctrl+touche Menu**.

Insérer avant/après, Apposer, Ajouter enfant
Les commandes **Insérer** et **Apposer** ajoutent un item au même niveau. Le nouvel item est ajouté dans le cas de **Insérer Après** et **Insérer Avant**, respectivement, après et avant l'item sélectionné, et dans le cas d'**Apposer** en tant que le dernier frère de l'item sélectionné. La commande **Ajouter enfant** appose un nouvel item en tant qu'enfant. Si un ou plusieurs items enfant existent déjà, le type du nouvel item sera le même que celui du dernier enfant. Si aucun en enfant n'existe, le nouvel item sera une paire `key:value` vide. Par exemple, si l'item sélectionné est un array, avec des objets enfants, alors le nouvel item enfant sera ajouté en tant que le dernier enfant.

Wrapper dans array, Wrapper dans objet
La partie sélectionnée de la table peut être wrappée soit dans un array soit dans un objet.

Déplacer vers le Haut/Bas/Gauche/Droite
S'il est possible de déplacer une composant vers le haut, le bas, la gauche ou la droite de son emplacement actuel dans la grille, la commande correspondante sera activée. Choisir la commande respective pour effectuer le déplacement.

Copier
Ces commandes peuvent être utilisées pour copier la sélection actuelle dans le presse-papier. Seules ces options sont activées qui sont permises dans ce composant.

Commande	Description
Copier en tant que texte JSON	La sélection actuelle est sérialisée en tant que balisage JSON
Copier comme texte séparé par tabulation	La sélection de table actuelle est sérialisée en tant que TSV (valeurs séparées par tab)
Copier en tant qu'Image	L'image actuelle est copiée en tant qu'image

Veuillez noter les points suivants :

- *Copier en tant qu'Image* copie le string encodé en Base64 de l'image sélectionnée. Si le string est collé dans un document où un string Base64-encoded peut être rendu en tant qu'image (par exemple vers une autre table dans l'Affichage Table), il sera rendu. Autrement, il sera collé en tant que string.
- Pour insérer du texte dans une cellule, copier le texte et le coller dans la cellule.
- Pour importer une image depuis un fichier, utilisez la commande **Éditer | Insérer | Fichier Externe Encodé**. Une image est insérée en tant que string encodé enBase64 , et affiche ce string en tant qu'image dans le Mode JSON Grid. Les formats de fichier d'image qui sont pris en charge pour les importations sont : `PNG`, `JPEG`, `BMP`, `GIF`, `TIFF`.

Copier XPath
Cette commande copie une expression localisatrice XPath 3.1 vers le presse-papier, commençant à la racine du document qui situe le nœud sélectionné.

Copier JSON Pointer
Cette commande copie l'expression JSON Pointer vers le presse-papiers qui situe le nœud sélectionné. Par exemple : **`/Artists/1/Albums/1/Tracks`**. Voir **Copy XPointer/JSON-Pointer** pour plus d'informations.

Filtre, Filtre pour Focus
La commande **Filtre** est une commande bascule. Elle ajoute un filtre à l'objet sélectionné ou désactive le filtre. Veuillez noter qu'après avoir ajouté un filtre via cette commande, vous devez saisir une expression de filtre ; autrement, le filtre sera supprimé automatiquement. Après avoir ajouté un filtre, vous pouvez le désactiver en sélectionnant la commande une nouvelle fois ou en cliquant sur le filtre. Pour supprimer le filtre, cliquez sur *Supprimer tous les filtres* dans la barre d'outils.

La commande **Filter pour Focus** (i) construit un filtre sur l'élément racine qui contient une expression XPath pour situer le nœud sélectionné, et (ii) filtre l'affichage du Mode Grille pour montrer le nœud sélectionné (et ses descendants) directement sous l'élément racine. Ceci est utile si vous souhaitez focaliser l'affichage uniquement sur le nœud sélectionné. Si vous souhaitez supprimer le filtre, cliquez sur l'icône Filtre dans la cellule du contenu de l'élément racine.

Agrandir complètement
Cette commande est activée si le composant sélectionné ou un de ses descendants est réduit. Elle agrandit le composant et tous les composants du descendant.

Réduire non sélectionné
Cette commande réduit tous les composants sauf le composant sélectionné et ses composants ancêtres.

Direction du Texte
Cette commande est disponible pour les nœuds contenant un type de string, et change le sens de la lecture pour commencer soit à gauche ou à droite de la cellule. Ceci est utile lorsque vous utilisez des langues comme l'arabe et le hébreux.

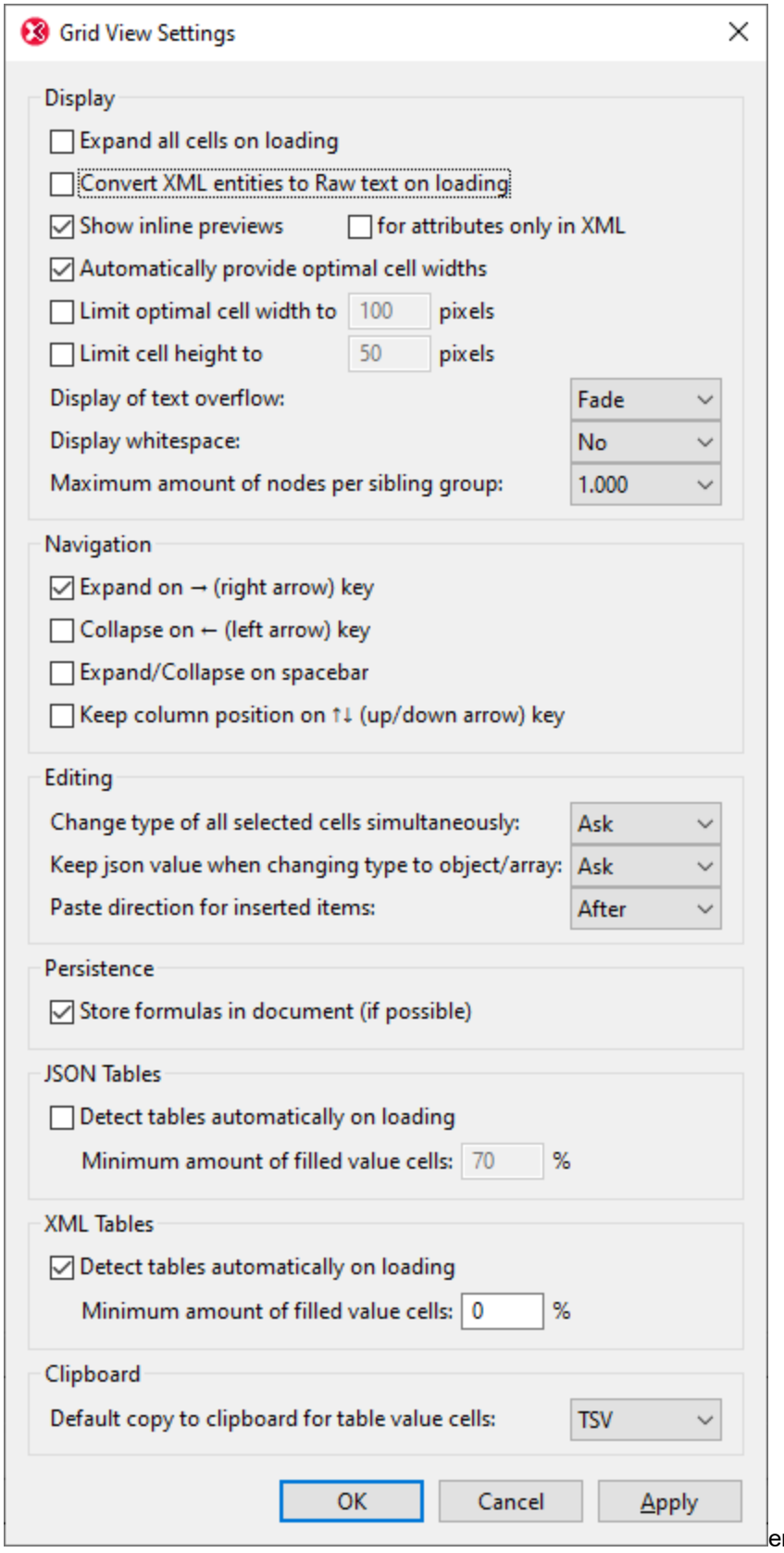
© 2016-2022 Altova GmbH

13.5.11 Paramètres du Mode Grille

Les paramètres du Mode Grid sont définis dans le dialogue des Paramètres du Mode Grid *(capture d'écran ci-dessous)*, auquel on accède via l'icône des paramètres dans la Barre d'outils du Mode View.

	Les paramètres du Mode Grid dans la Barre d'outils du Mode Grid

Les paramètres du Mode Grille sont décrits ci-dessous. Notez que ces paramètres s'appliquent à tous les documents du Mode Grille (XML, JSON, DTD).

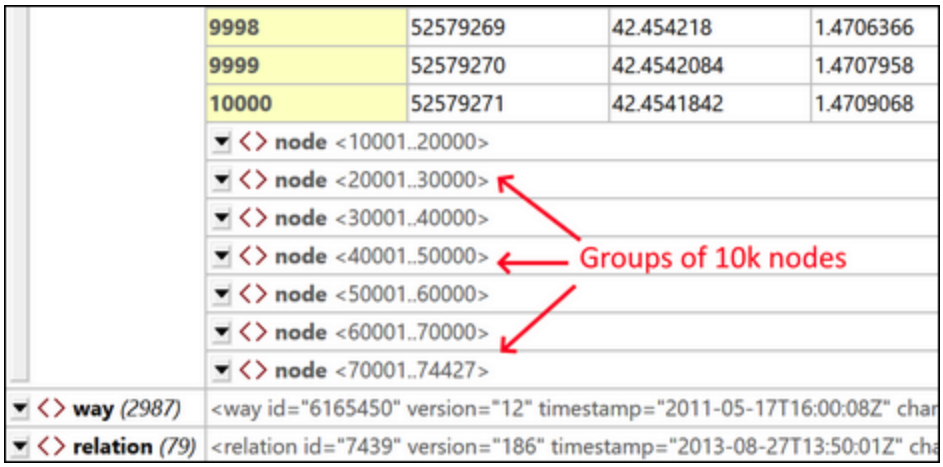


end

© 2016-2022 Altova GmbH

Afficher
Les cases à cocher dans la section *Affichage* sont assez explicites. Veuillez trouver ci-dessous quelques clarifications.

- Si les cellules ne sont pas agrandies lors du chargement, le nœud racine et tous ses descendants sont compressés. Vous devrez étendre chaque nœud au fur et à mesure que vous plongez au cœur du document.
- Si *Convertir les entités XML en texte brut sur chargement* est sélectionné, alors les entités XML seront chargées en Mode Grille comme texte brut de l'entité respective ; elles ne seront pas résolues en leurs représentations de glyphes respectives.
- Si *Afficher aperçus instantanés* n'est pas cochée, alors, au lieu d'un aperçu de la cellule étant affiché, seul le numéro de l'index de l'élément dans la cellule sera affiché. Si des aperçus instantanés sont activés, vous pouvez choisir si vous voulez afficher un aperçu qui contient (i) le contenu et les attributs de l'élément ou (ii) uniquement les attributs. Pour opter pour ce dernier, cochez *pour des attributs uniquement*; pour opter pour le premier, désactivez *pour des attributs uniquement*. Veuillez noter qu'uniquement la première partie du contenu inline d'une cellule sera affichée ; vous pouvez passer sur un onglet de démarrage d'un élément pour voir tout son contenu.
- Lorsque les largeurs optimales sont activées, toute la largeur de la grille est affichée. Pour y parvenir, le texte dans certaines cellules sera renvoyé à la ligne.
- Lorsque du texte dépasse une cellule, ce dépassement peut être affiché soit en tant que texte, soit être indiqué par une ellipse.
- Vous pouvez basculer l'affichage de l'espace blanc dans les cellules de grille on ou off. Un espace est affiché en tant que point centré vertical et un onglet est affiché comme une flèche. Une fin de ligne (end-of-line) est indiquée avec un nouveau saut de ligne à l'intérieur d'une cellule.
- Les nœuds sibling peuvent être affichés en groupes frères de 100 nœuds, 1k ou 10k nœuds *(voir la capture d'écran ci-dessous)*. Ceci est utile pour deux raisons : (i) Si vous souhaitez gagner de l'espace d'affichage dans les fichiers qui ont un grand nombre de nœuds frères, vous pouvez grouper les nœuds frères en grands groupes ; (ii) si vous souhaitez traiter uniquement un sous-ensemble de nœuds frères (gagnant ainsi du temps de traitement), vous pouvez élargir des groupes frères pertinents que vous souhaitez traiter et effectuer l'action de traitement sur ces nœuds uniquement ; par exemple, quand vous utilisez Chercher-et-Remplacer ou quand vous convertissez le type d'un nœud (voir *Édition* ci-dessous).



Navigation
Généralement, vous pouvez utiliser les touches de direction pour naviguer dans la grille. Ces paramètres fournissent des options intelligents pour utiliser les clés.

- *Élargir sur la touche de direction de la droite :* Si un élément de cellule est réduit, appuyer sur la *touche de direction de droite* élargit l'élément dans la cellule. Si l'item de cellule n'est pas réduit, la *touche de direction de droite* vous amène à la prochaine cellule à droite (y compris l'enfant si la prochaine cellule à droite est un enfant). Si l'option n'est pas sélectionnée, la *touche de direction de droite* s'arrête à la cellule réduite. Veuillez noter que la fonction *Étendre avec la touche de direction Flèche droite* ne s'applique pas aux cellules à l'intérieur des tables ; dans les cellules de table, l'action vous amène à la prochaine cellule à droite.
- *Réduire sur la touche de direction de gauche :* Lorsque vous vous déplacez à gauche avec la *touche de direction gauche* alors, à un certain moment, vous devez également déplacer la hiérarchie de document vers le haut. Si cette option est sélectionnée, alors les items qui peuvent être réduits le seront quand vous appuyez sur la *touche de direction de gauche* ; autrement, de tels items ne seront pas réduits bien que le focus basculera vers l'item parent. Veuillez noter que la fonction *Réduire avec la touche de direction de gauche* ne s'applique pas aux cellules des tables ; dans les cellules de table, l'action vous amène à la prochaine cellule à gauche.
- *Élargir/Réduire sur la barre d'espace :* Les fonctions d'espace de barre en tant que bascule pour élargir/réduire un élément. Elle peut donc être utilisée comme touche additionnelle pour naviguer au sein de la grille.
- *Garder la position de la colonne sur les touches de direction Haut/Bas :* Les touches de direction *Haut* et *Bas* vous déplacent, respectivement, vers le haut et le bas dans les cellules de la grille, y compris dans les items parent et enfants - qui sont hiérarchiquement à différents niveaux, et donc dans différentes colonnes. Si cette option est choisie, les niveaux représentés dans les colonnes autres que la colonne actuelle sont ignorés. Ceci fonctionne, par exemple, comme suit. Disons que le curseur se trouve dans la colonne pour l'élément **`subject/course/books/book/title`**. Avec l'option *Garder position de colonne* choisie, vous pouvez utiliser les *touches de direction Haut* et *Bas* pour naviguer uniquement à travers les titres de livres (sans aller dans les colonnes **`book`**, **`books`**, **`course`**, ou **`subject`**, ou toute colonne pour les items descendants de **`Title`**.)

Édition

Les cases à cocher dans la section *Affichage* sont assez explicites. Veuillez trouver ci-dessous quelques clarifications.

- Lorsque vous modifiez le type de cellules multiples sélectionnées, vous avez les options suivantes qui définissent si vous pouvez poursuivre l'action du paramètre : *Toujours, Jamais,* ou *Demander* (pour la décision de l'utilisateur).
- Lorsque vous modifiez un type JSON d'un type atomique à objet ou array, vous avez les options suivantes qui définissent si vous pouvez poursuivre l'action du paramètre : (i) *Demander* (si la valeur du type atomique devrait être maintenue comme valeur d'une paire d'enfants sans nom **`key:value`** ou ignorée), (ii) *Toujours* (maintenir la valeur d'une paire d'enfants sans nom **`key:value`**), (iii) *Jamais* (maintenir la valeur, à savoir ignorer la valeur).
- L'option de la *direction Coller* détermine si une sélection dans le presse-papiers est collée au-dessus ou en dessous de la cellule sélectionnée.

Persistance

Les filtres et formules peuvent être enregistrés dans les méta-données des documents XML et JSON. Les formules sont stockées comme instructions de traitement dans les documents XML et comme commentaires dans certains types de documents JSON. Ces paramètres permettent de spécifier comment procéder avec le stockage des filtres et des formules. Dans les cas des formules, la terminologie fait, *si possible*, référence au fait que les formules ne peuvent pas être enregistrées comme commentaires dans tous les documents JSON (uniquement dans les documents JSON5 et JSONC).

© 2016-2022 Altova GmbH

Tables JSON, tables XML
Si le paramètre pour détecter les tables en Mode Grille est sélectionné automatiquement lors du chargement, vous pouvez sélectionner le pourcentage minimum de cellules de tables remplies qui qualifient les tables à être détectées comme des tables. Si le nombre de cellules de table remplies ne dépasse pas ce niveau, la structure est affichée comme grille normale avec les éléments répétitifs recensés l'un en dessous de l'autre.

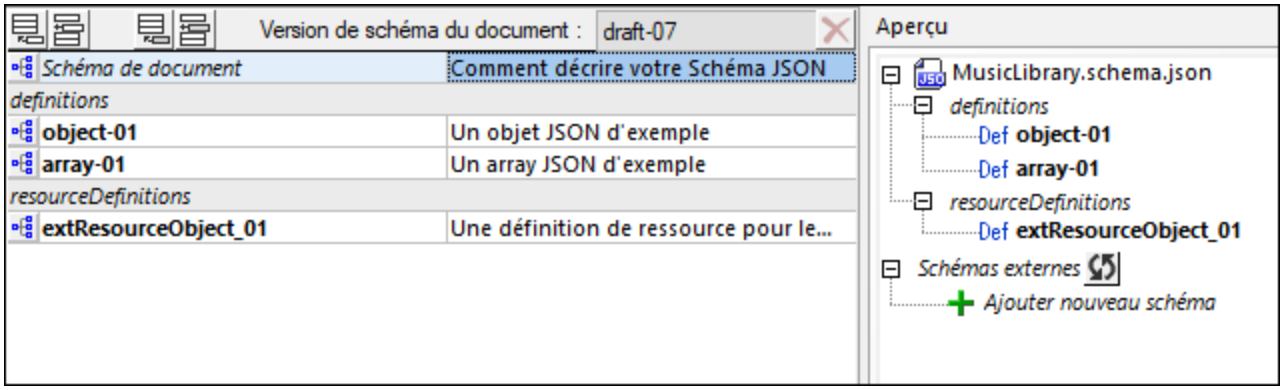
Presse-papiers
Vous pouvez aussi choisir si les contenus du presse-papiers doivent être stockés comme valeurs séparées par tabulation (TSV), ou comme XML/JSON (dépendant du type de document). Ceci est une fonction très utile : Si vous souhaitez coller une table depuis le presse-papiers dans un autre document, le paramètre vous permet de choisir si la table copiée est stockée comme TSV ou avec un marquage. (Pour voir la différence, collez une table dans un éditeur de texte après avoir copié la table vers le presse-papiers dans chacun des formats.)

13.6 Mode Schéma JSON

Le Mode de Schéma JSON peut être utilisé pour consulter et éditer des documents de schéma JSON. Les parties principales de la fenêtre du Mode de Schéma JSON sont :

- Une fenêtre principale qui passe entre un **Grille d'aperçu des définitions** et un **Mode Design**
- Trois fenêtres d'assistants à la saisie (situées par défaut du côté droit de la fenêtre principale) : Aperçu, Détails et Contraintes
- Une fenêtre Messages (située par défaut en-dessous de la fenêtre principale)
- Une fenêtre Info (située par défaut en-dessous à gauche de la fenêtre d'application)

La capture d'écran ci-dessous montre la fenêtre principale et assistant à la saisie Aperçu.

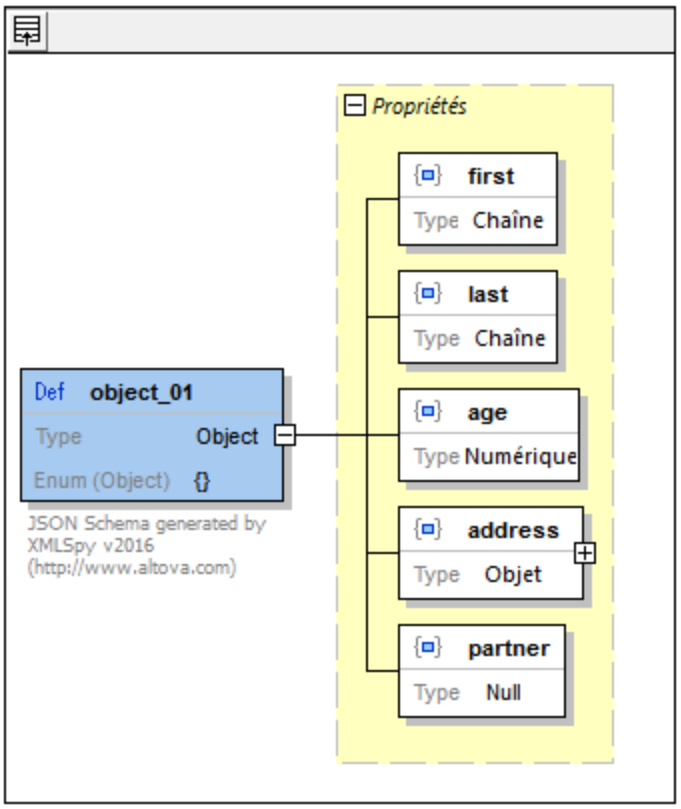


La fenêtre principale

La fenêtre principale passe entre une Grille d'aperçu des définitions (*affichée dans la capture d'écran ci-dessus*) et un Design Mode (*capture d'écran ci-dessous*). La Grille d'aperçu des définitions montre le schéma principal du document actuel (*recensé en tant que "Document Schema"*), plus toute définition que vous rajoutez au schéma. (Une définition est une description d'une structure de données JSON. Dans la capture d'écran ci-dessus, `object_01` et `array_01` sont des définitions, d'un objet et d'un array, respectivement). Les définitions sont aussi regroupées dans l'assistant à la saisie Aperçu (*voir capture d'écran ci-dessus*).

Alors que la Grille d'aperçu des définitions propose une consultation à niveau élevé du schéma JSON, elle ne montre pas se qui se trouve dans toute définition recensée dans l'aperçu. Pour consulter et éditer une définition dans le Mode Design (*capture d'écran ci-dessous*), cliquer sur l'icône de la définition (*voir capture d'écran ci-dessus*) ou double-cliquer sur la définition dans l'assistant à la saisie Aperçu (*voir capture d'écran ci-dessus*).

© 2016-2022 Altova GmbH



Pour retourner à la Grille d'aperçu des définitions depuis le Mode Design, cliquer sur l'icône **Passer à la grille de définitions** située en haut à gauche du Mode Design (*voir capture d'écran ci-dessus*). Pour configurer le Mode Design, cliquer sur la commande de menu **Design de schéma | Configurer Mode**.

Les assistants à la saisie

Les deux modes du Mode Design (Grille d'aperçu des définitions et Mode Design) disposent de trois assistants à la saisie : Aperçu, Détails et Contraintes. Ces assistants à la saisie offrent des mécanismes pour : (i) l'affichage des informations à propos du schéma et de ses définitions, et (ii) la saisie d'informations et de valeurs liées aux définitions. Vous trouverez une description plus détaillée dans la section Assistants à la saisie : Aperçu, Détails, Contraintes.

13.6.1 Version de Schéma JSON

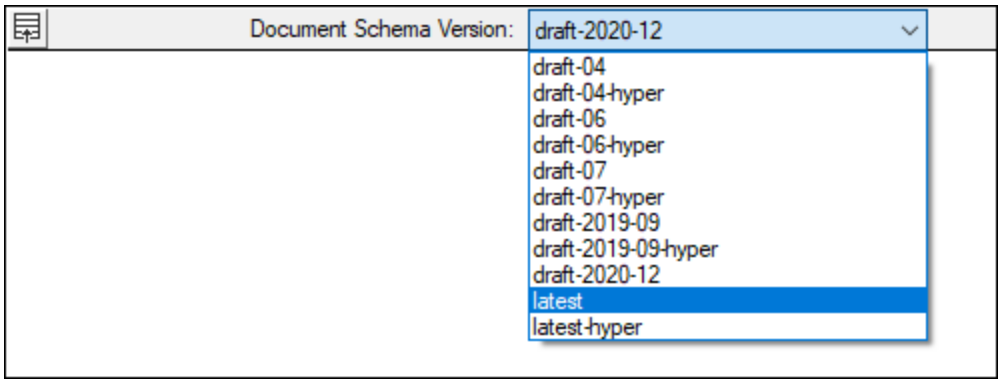
Un schéma JSON est écrit dans la syntaxe JSON et sera généralement enregistré dans un fichier avec une extension `.json`. Il est indiqué en tant qu'un schéma JSON avec le mot-clé **`$schema`**, qui doit être le premier mot-clé de l'objet de niveau supérieur et ont une valeur qui est l'URI de la version schéma JSON que vous souhaitez utiliser. Ici, vous trouverez deux exemple montrant comment utiliser le mot-clé **`$schema`** :

```
{
   "$schema": "http://json-schema.org/draft-07/schema#",
   ...
}
```

```
{
   "$schema": "http://json-schema.org/schema#",
   ...
}
```

Dans le premier exemple ci-dessus, la version de schéma est nommé explicitement. À la place de choisir explicitement une version, vous pouvez utiliser `"http://json-schema.org/schema#"`, comme dans le deuxième exemple ci-dessus. Ceci indique que la version de schéma à utiliser est la version la plus récente (actuellement **<%JSON-SCHEMA-LATEST%>**).

Dans le Mode Schéma JSON, vous pouvez changer la version dans la liste de choix dans la barre en haut de la fenêtre principale *(voir capture d'écran ci-dessous)*.



Fonctions des nouvelles versions de schéma qui ne sont pas définies dans une version plus ancienne

Si vous utilisez une fonction ou une version de schéma plus récente et que vous passez ensuite à une version plus ancienne qui ne prend pas en charge cette fonction, la chose suivante se produit :

- Un message apparaît vous invitant à supprimer/reconvertir la fonction ou garder la fonction
- Si vous le conservez, le composant ou le détail correspondant à la nouvelle fonction est affiché dans un texte de couleur orange. Par exemple, si une valeur a été définie pour le mot-clé **`const`** (nouveauté dans **`draft-06`**) et que vous changez la version de schéma à **`draft-04`**, alors la valeur du mot-clé **`const`** est affichée en orange.

Versions de Schéma JSON

Pour plus d'informations concernant les spécifications de Schéma JSON, en particulier en ce qui concerne les fonctions supplémentaires avec chaque version, vois les liens ci-dessous :

Spécification de Schéma JSON : http://json-schema.org/specification.html
Draft-06 notes de version : http://json-schema.org/draft-06/json-schema-release-notes.html
Draft-07 notes de version : http://json-schema.org/draft-07/json-schema-release-notes.html
Draft 2019-09 (avant Draft-08) : http://json-schema.org/specification-links.html#2019-09-formerly-known-as-draft-8
Draft 2020-12 : http://json-schema.org/specification-links.html#2020-12

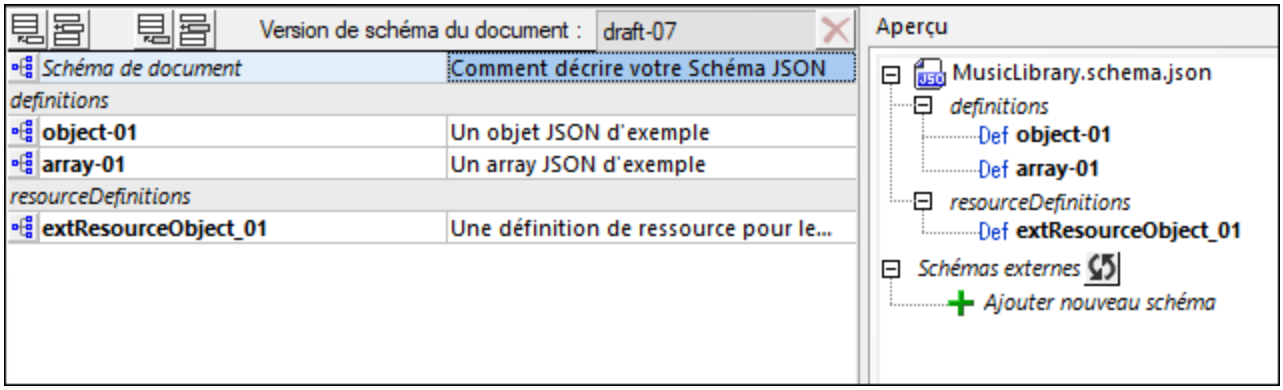
© 2016-2022 Altova GmbH

De plus, les Hyper Schémas suivants sont disponibles : **`draft-04-hyper`**, **`draft-06-hyper`**, **`draft-07-hyper`**, **`2019-09-hyper`**.

Les liens vers les schémas principaux et hyper schémas sont disponibles ici : http://json-schema.org/specification-links.html.

13.6.2 Ajouter des définitions globales

La **Grille d'aperçu des définitions** dans la fenêtre principale (*capture d'écran ci-dessous*) affiche une liste des définitions globales du schéma. Ces définitions globales sont : (i) la définition principale de schéma de document, (ii) les définitions (ou *$defs* dans des schémas ultérieurs) des types JSON globaux, comme des objets, des arrays, des chaînes, etc., qui sont des types de Schéma JSON ; (iii) les définitions de types JSON définis ou externes ; actuellement, seules des définitions qui se produisent dans un conteneur nommé **`resourceDefinitions`** sont disponibles ; il s'agit du conteneur utilisé par l'Azure Resource Manager de Microsoft pour les définitions JSON. Ajouter une nouvelle section *resourceDefinitions* au document de schéma par le biais de l'icône **Ajouter une section de définitions** ou **Insérer une section de définitions** dans la barre outils de la grille (*capture d'écran ci-dessous*).



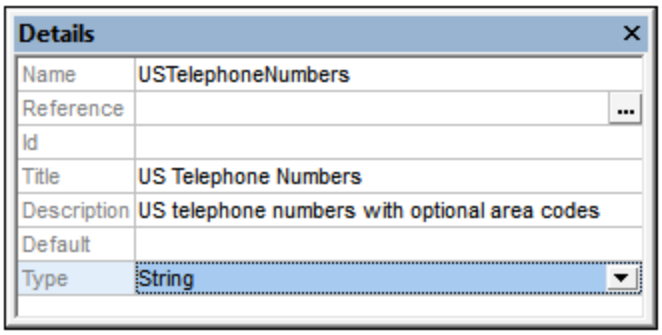
La définition globale d'un type JSON est utile si ce type doit être réutilisé dans le même schéma ou dans un schéma différent. Par exemple, vous pouvez définir un type de chaîne JSON pour des numéros de téléphone US dans un schéma JSON, puis référencer cette définition non seulement depuis l'intérieur du même schéma mais aussi depuis d'autres schémas JSON.

Ajouter une définition, et les actions liées

Les actions suivantes sont disponibles pou ajouter et éditer les définitions dans la Grille d'aperçu des définitions.

- *Pour ajouter une définition :* cliquer sur l'icône **Ajouter une Définition de schéma nommée** ou **Insérer une définition de schéma nommée** située en haut à gauche de la Grille d'aperçu des définitions (*voir capture d'écran ci-dessus*). Une nouvelle définition vide sera créée dans la grille à l'emplacement où vous l'ajoutez ou l'insérez ; elle aura un nom par défaut. La nouvelle définition sera aussi recensée dans l'assistant à la saisie Aperçu en tant que *Def* (*voir capture d'écran ci-dessus*).

- *pour changer le type d'une définition :* chaque nouvelle définition est créée avec un type de `Any`. Vous pouvez changer sont type dans l'assistant à la saisie Détails (*voir capture d'écran ci-dessous, où le type est 'String'*) ou en éditant la définition dans le Mode Design.



- *Pour renommer une définition :* double-cliquer son nom et éditer le nom. En alternative, éditer le champ *Nom* dans Assistant à la saisie Détails.
- *Pour saisir une description de la définition :* éditer le champ *Description* dans l'assistant à la saisie Détails. La description apparaît dans la Grille d'aperçu des définitions à côté du nom de la définition (*voir capture d'écran ci-dessous*). Vous pouvez aussi double-cliquer dans le champ Description de la Grille d'aperçu des définitions pour éditer une description.



- *Pour référencer une définition :* voir la description de l'assistant à la saisie Aperçu et la section Définitions Globales et Locales.
- *Pour éditer une définition :* cliquer sur l'icône de définition dans la Grille d'aperçu des définitions ou double-cliquer sur la définition dans l'assistant à la saisie Aperçu. Cela permet d'ouvrir la définition dans le Mode Design, où elle peut être éditée.

13.6.3 Assistants à la saisie : Aperçu, Détails, Contraintes

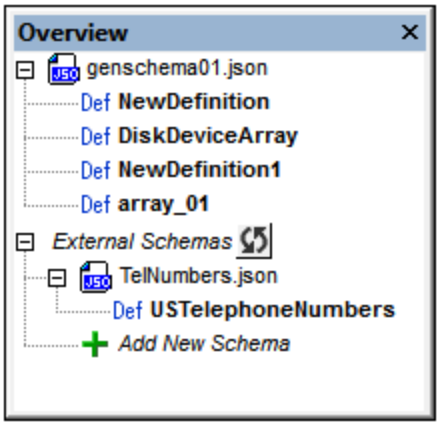
Les assistants à la saisie du Mode Schéma JSON sont situés par défaut dans le côté droit de la fenêtre de l'application. Il s sont disponibles dans les deux modes de la fenêtre principale : (i) Grille d'aperçu des définitions, et (ii) Mode Design. Vous pouvez glisser des fenêtres d'assistants à la saisie par leur barre de titre vers d'autres emplacement dans l'écran et vous pouvez double-cliquer sur la barre de titre d'un assistant à la saisie pour ancrer et détacher cet assistant à la saisie. Pour plus d'informations concernant ces actions, voir la section Assistants à la saisie.

Assistant à la saisie Aperçu

L'assistant à la saisie Aperçu (*capture d'écran ci-dessous*) recense la définition de schéma actuelle et toutes les définitions globales du schéma actuel. Double-cliquer sur une définition pour ouvrir cette définition dans le Mode Design, où elle peut être éditée. Si vous souhaitez utiliser les définitions depuis les schémas externes, il convient tout d'abord d'ajouter le schéma externe et puis de réutiliser la définition que vous souhaitez.

© 2016-2022 Altova GmbH

Ajouter le schéma externe
Ajouter le schéma externe en cliquant sur l'icône **Ajouter nouveau schéma** dans l'assistant à la saisie Aperçu puis chercher le schéma que vous souhaitez ajouter. Une fois qu'un schéma a été ajouté, ses définitions sont affichées dans l'assistant à la saisie Aperçu. La capture d'écran ci-dessous, par exemple, montre que le schéma `TelNumbers.json` a été ajouté et que ce schéma a une définition nommée `USTelephoneNumbers`. Vous pouvez ajouter autant de schémas externes que vous le souhaitez.



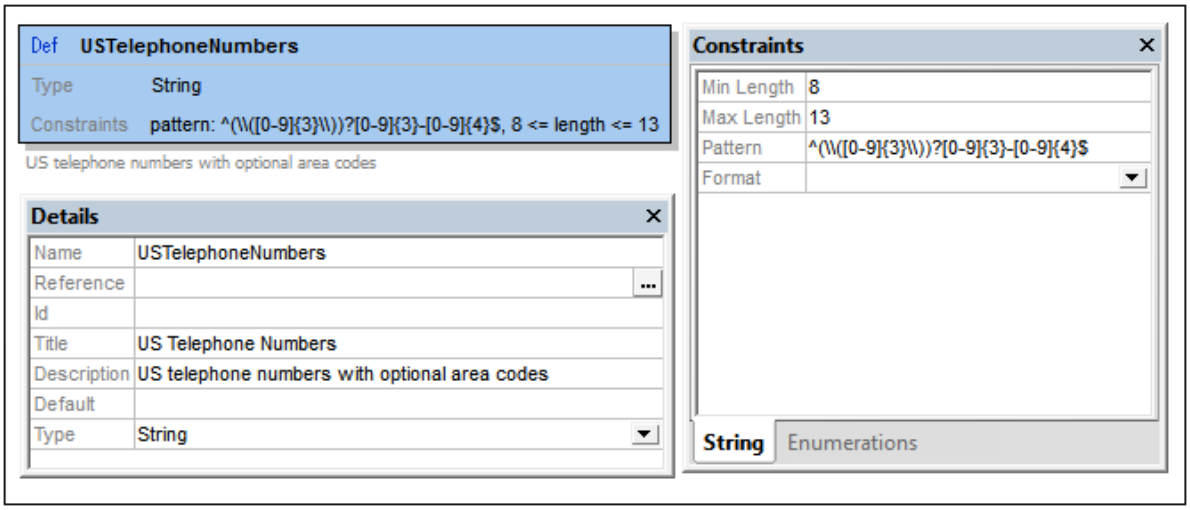
Réutiliser une définition externe
Une fois qu'un schéma externe ait été ajouté, ses définitions deviennent disponibles pour une réutilisation dans les définitions du schéma d'importation. Lorsqu'une définition réutilise une autre définition (en la référençant), elle prend les propriétés de cette définition. Le référencement peut être effectué de deux manières :

- *Dans le Mode Design :* en glissant une définition depuis l'assistant à la saisie Aperçu dans la définition où elle est souhaitée
- *Dans la Grille d'aperçu des définitions ou le Mode Design :* par le biais du champ *Reference* de l'assistant à la saisie Détails de la définition où vous souhaitez une réutilisation. Ceci est expliqué ci-dessous dans la description de l'assistant à la saisie Détails.

Note : L'icône **Réinitialisation** située à côté de l'entrée *Schémas externes* dans la fenêtre Aperçu met à jour tous les schémas externes ajoutés. Veuillez noter que, si aucune définition n'a été réutilisée depuis un schéma externe, ce schéma sera supprimé de la liste lorsque la liste est réinitialisée.

Assistant à la saisie Détails

Les propriétés d'une définition peuvent être saisies dans les fenêtres Détails et Contraintes lorsque la définition est sélectionnée dans l'un des deux modes de la fenêtre principale : Grille d'aperçu des définitions ou Mode Design. La capture d'écran ci-dessous montre la définition de `USTelephoneNumbers` dans le Mode Design, avec les assistants à la saisie Détails et Contraintes. Veuillez noter que l'information contenue dans les deux assistants à la saisie est aussi affichée dans la fenêtre bleue de la définition dans le Mode Design. Les propriétés qui peuvent être définies dans ces deux assistants à la saisie sont regroupées ci-dessous.



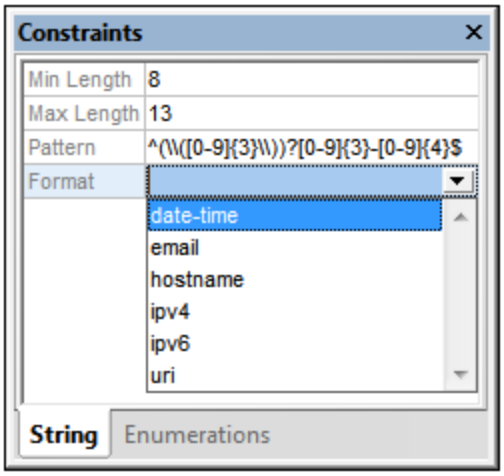
Les détails suivants peuvent être saisis dans l'assistant à la saisie Détails :

- *Nom :* le nom de la définition.
- *Référence :* si vous souhaitez qu'une définition réutilise une autre définition, cliquer sur la touche **Dialogue supplémentaire** du champ *Référence (voir la capture d'écran ci-dessus)*. Cela permet d'afficher le dialogue Éditer référence (*capture d'écran ci-dessous*), qui regroupe toutes les définitions disponibles (depuis le schéma actuel et les schémas externes). Sélectionner la définition que vous souhaitez réutiliser, sélectionner l'option *Chemin relatif* si vous souhaitez un chemin relatif et cliquer sur **OK**. Voir les Défintions globales et locales pour les détails.
- *Type :* sélectionner le type de données de la définition depuis la liste déroulante de la liste de choix. Veuillez noter que le fait de changer le type entraînera une suppression des mots-clés spécifiques au type précédent. Si vous souhaitez retourner aux définitions précédentes, appuyer sur **Annuler (Ctrl-Z)**. Les types sont expliqués dans Données JSON et Sélecteurs de type (Any, Multiple, etc).
- *ID :* il s'agit d'un mot-clé optionnel qui définit un URI pour le schéma. Cet URI peut être utilisé pour référencer le schéma et est utilisé en tant que qu'URI de base pour d'autres références URI à l'intérieur du schéma. La valeur ID doit être un string qui est un URI. Veuillez noter que le validateur JSON d'Altova utilise uniquement un déréférencement canonique. Voir la spécification JSON pour plus d'informations.
- *Ancre (nouveau dans draft-2019-09):* il s'agit d'un mot-clé d'identificateur optionnel qui fournit un fragment de nom brut (et pas un URI comme c'est le cas avec l'ID). La valeur de l'*ancre* doit être un string tel que décrit dans les drafts respectifs.
- *Titre, Description :* les valeurs de ces deux mots-clés sont utilisées à des fins descriptives.
- *Commentaire (nouveau dans draft-07) :* prévu pour rédiger des notes aux mainteneurs de schéma, par rapport à la fonction Description, destiné aux utilisateurs finaux.
- *Const (nouveau dans draft-06) :* une valeur constante, comme une énumération à une valeur.
- *Défaut :* la valeur par défaut de la définition.
- *Lecture seule, Écriture seule (nouveau dans draft-07) :* indiquent, respectivement, les champs de Lecture seule et d'Écriture seule. Un exemple d'un champ en Écriture seule serait un champ de mot de passe.
- *Déprécié (nouveau dans draft-2019-09) :* un indicateur que la définition sélectionnée peut être supprimée dans l'avenir. Les applications peuvent gérer de telles définitions de manière spécifique.

 © 2016-2022 Altova GmbH

Assistant à la saisie Contraintes

Les contraintes d'une définition dépendent de son type. Les contraintes de chaque type sont décrites ci-dessous. (Voir aussi Types atomiques.)



Si un type n'apparaît pas dans la liste ci-dessous, aucune contrainte ne pourra être définie pour lui. Veuillez noter, néanmoins que des énumérations peuvent être définies pour tous les types :

- *Chaîne :* la longueur de la chaîne et le motif de la chaîne ; le motif est spécifié au moyen d'une expression régulière. Dans le champ *Format*, vous pouvez sélectionner un des formats de chaîne définis dans la spécification (*voir capture d'écran ci-dessus, qui montrent les formats disponibles dans draft-04*) ; des formats supplémentaires ont été définis dans des versions ultérieures. *Content Media Type* et *Content Encoding* (deux nouveautés dans draft-07) permettent de sélectionner le type de média et l'encodage de données non-JSON encodée dans un string JSON.
- *Numérique :* la plage des valeurs autorisées
- *Array :* le nombre des items autorisés dans l'array si les items doivent être uniques
- *Objet :* le nombre des propriétés autorisées

L'assistant à la saisie Contraintes pour tous les types a un onglet énumérations. Vous pourrez y spécifier une liste des items autorisés de ce type de définition. De plus, un onglet Exemples est disponible (nouveauté dans draft-06) pour tous les types sauf *Forbidden.* Il s'agit d'une série d'exemples sans effet de validation ; la valeur de `default` est utilisable en tant qu'un exemple sans le répéter sous ce mot-clé.

13.6.4 Définitions Globales et Locales

Les définitions de schéma JSON peuvent être créées globalement ou localement.

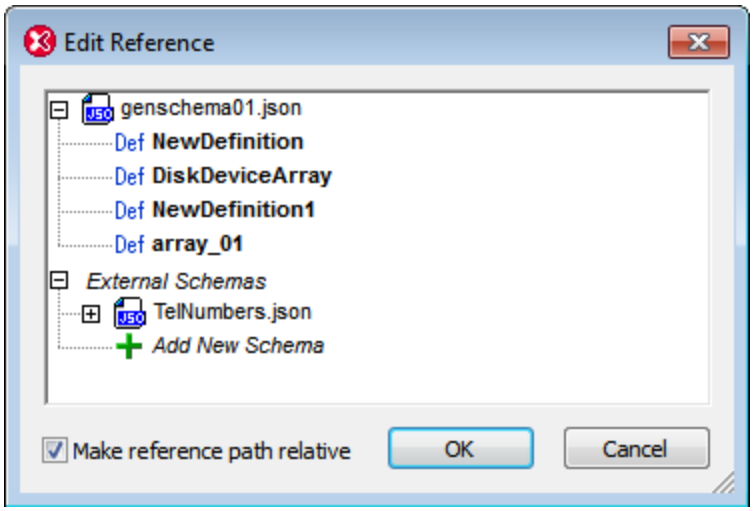
- **Les définitions globales** sont créées dans la Grille d'aperçu des définitions de la fenêtre principale en ajoutant une définition puis en spécifiant ses propriétés. Une définition globale peut être référencée par d'autres définitions dans le même schéma ou dans d'autres schémas. Cela permet la réutilisation de définitions dans votre projet. Toutes les définitions globales du schéma actuel sont affichées dans la Grille d'aperçu des définitions du schéma. Les définitions globales provenant d'autres schémas peuvent

être rendues disponibles pour une réutilisation en ajoutant le schéma externe dans l'assistant à la saisie Aperçu.
- **Les définitions locales** sont créées dans les définitions globales, c'est-à-dire, en ajoutant des définitions descendantes ou frères à une définition globale.

Réutiliser une définition globale

Afin de réutiliser une définition globale, choisir une des deux étapes :

- Dans le Mode Design, glisser la définition globale depuis l'assistant à la saisie Aperçu dans la définition où elle doit être utilisée.
- Dans le Mode Design, cliquez de la touche droite sur la définition pour laquelle vous souhaitez référencer une définition globale et sélectionnez **Éditer la Référence**. (De manière alternative, avec la définition sélectionnée dans le Mode Design, allez à son assistants à la saisie Détails et cliquez sur le bouton **Dialogue supplémentaire** du champ *Référence*.) Dans le dialogue Éditer Référence qui apparait (*capture d'écran ci-dessous*), sélectionnez la référence globale que vous souhaitez référencer. Une autre manière d'accéder au dialogue Editer Référence est de sélectionner la définition référencée, allez à son champ *Référence* dans l'assistant de saisie Détails, et cliquez sur le bouton des champs **Dialogue supplémentaire** de la définition globale sélectionnée que vous souhaitez réutiliser (*voir la description de l'assistants à la saisie Détails pour tout détail*).



Note : une définition peut (i) référencer une définition globale et ne pas contenir de définition locale ou de contrainte locale, ou (ii) référencer les deux, une définition globale et contenir des définitions locales (à partir du *draft-2019-09*). Dans le dernier cas, la référence est connue comme une **référence élargie**. Dans le dialogue Éditer Référence, vous pouvez créer une référence vers une définition globale en tant que référence élargie en vérifiant la case à cocher du dialogue *Référence élargie*. Une définition globale qui est créée en tant que référence élargie est toujours affichée comme dernier item dans la liste des définitions des composants de référencement. Si une valeur existe pour référencer le mot-clé de la `description` du composant, alors cette valeur est affichée en dessous du composant de référencement.

Note : si vous changez le nom d'une définition globale après son référencement par une autre définition dans le même schéma, alors le nom est également modifié dans la référence. Néanmoins, les références provenant d'autres schémas devront être éditées manuellement pour respecter le changement de nom.

© 2016-2022 Altova GmbH

Convertir des définitions locales en définitions globales

Pour convertir une définition locale, cliquez avec la touche de droite dans Mode Design et sélectionnez **Rendre global**. Une définition globale est créée et une référence sera créée dans la définition locale. Étant donné que le nom de la définition globale est généré automatiquement, vous pourrez l'éditer et la modification sera passée à la référence de la définition locale.

Changer une ref vers une définition globale en une définition locale

Une référence à une définition globale peut exister sur les définitions locales et globales. Pour supprimer la référence et rendre ses propriétés locales, cliquer avec la touche de droite sur la définition (locale ou globale) dans le Mode Design et choisir **Rendre local**. Les propriétés de la définition globale sont créées localement sur la définition.

13.6.5 Mode Design

Dans le Mode Design, vous pouvez spécifier la structure et les valeurs autorisées des définitions globales individuelles. Les définitions sont spécifiées via les composants ou mécanismes de GUI suivants :

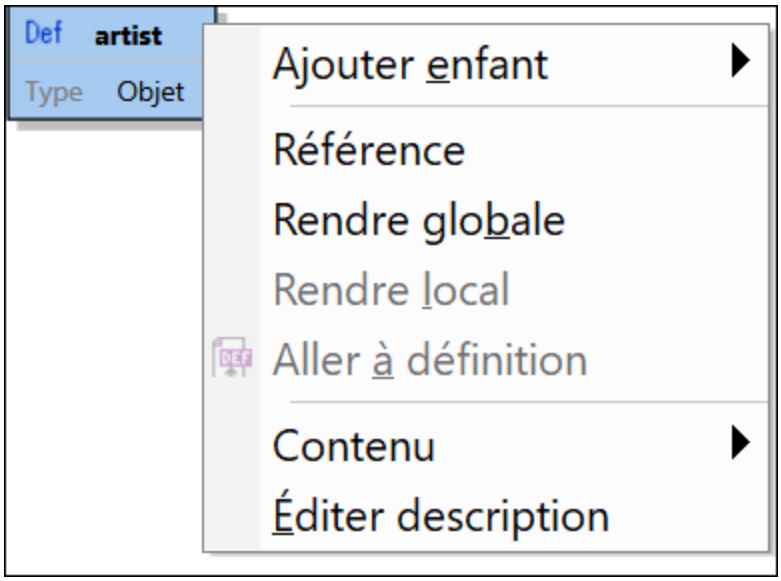
- L'Assistant à la saisie Détails (aussi disponible dans la Grille d'aperçu des définitions)
- L'Assistant à la saisie Contraintes (aussi disponible dans la Grille d'aperçu des définitions)
- Le menu contextuel de la définition (accédé en cliquant avec la touche de droite sur la fenêtre de la définition dans la fenêtre principale)

Les définitions qui peuvent être spécifiées par le biais des assistants à la saisie Détails et Contraintes sont décrites dans la section Assistants à la saisie : Aperçu, Détails, Contraintes. Certaines de ces propriétés peuvent aussi être spécifiées dans la fenêtre de la définition dans la fenêtre principale. Dans cette section, et dans les trois sections suivantes, nous décrivons le mécanisme principalement graphique disponible dans la fenêtre principale.

Note : Si vous souhaitez annuler une modification accidentelle ou non voulue, appuyer sur **Ctrl+Z**.

Menu contextuel

Le menu contextuel d'une définition (*fenêtre bleue dans la capture d'écran ci-dessous*) vous permet de concevoir la structure de la définition et d'éditer ses propriétés.



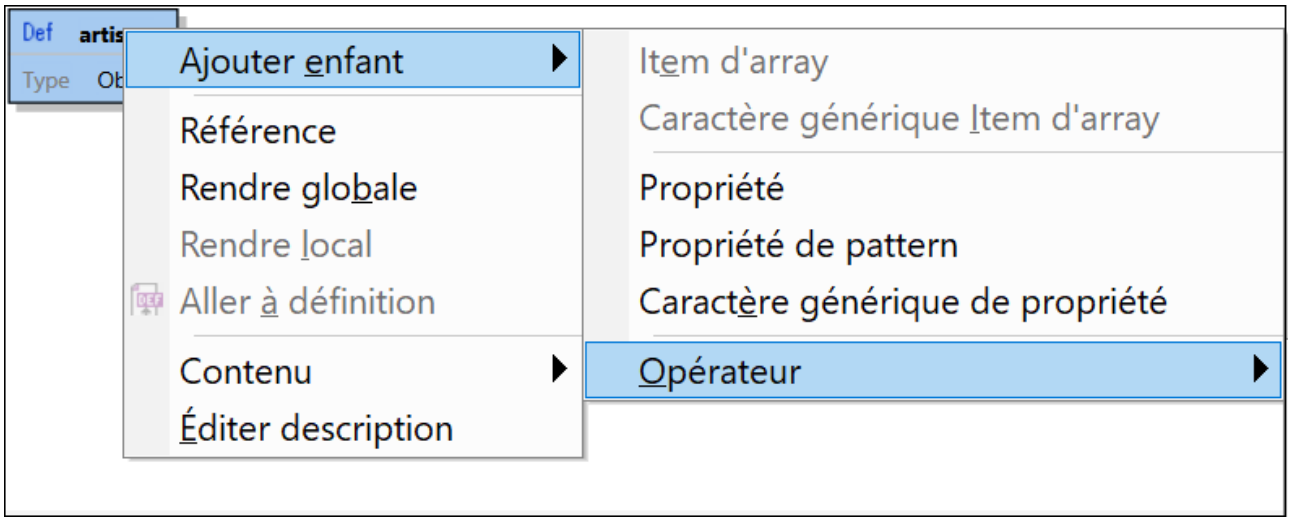
Les commandes suivantes sont disponibles :

- *Ajouter enfant :* l'enfant pouvant être ajouté dépend du type de la définition (voir *Ajouter enfant : créer une structure* ci-dessous).
- *Éditer la Référence :* permet à la définition de référencer une définition globale et de prendre les propriétés de cette définition globale. Le dialogue Éditer la référence que la commande ouvre est la même que celle accédée par l'Assistant à la saisie Détails et est décrite dans la section Assistants à la saisie : Aperçu, Détails, Contraintes.
- *Rendre global :* cette commande est activée lorsque la définition est une définition locale. Elle fait de la définition actuellement sélectionnée une définition globale et ajoute une référence à cette définition globale dans la sélection actuelle.
- *Rendre local:* cette commande est activée lorsque la définition est une définition globale. Elle converti la définition actuellement sélectionnée en une définition locale en créant une référence à la définition globale originale.
- *Aller à la définition :* si la définition sélectionnée est contenue dans une définition qui référence une définition globale, alors cette commande est activée. Cliquer dessus pour vous rendre à la définition globale.
- *Contenu :* la commande **Contenu** affiche un sous-menu contenant des commandes pour couper, copier et réinitialiser les contenus de la définition sélectionnée.
- *Éditer description :* permet l'édition du champ *Description* de la définition.

Ajouter enfant : créer une structure

La structure d'une définition est créée en ajoutant plusieurs niveaux de descendants. Ces niveaux sont créés avec la commande **Ajouter enfant** du menu contextuel. Les enfants qui peuvent être ajoutés à une définition dépendent de son type:

© 2016-2022 Altova GmbH



- *Objets* : prend des propriétés et des opérateurs
- *Arrays* : prend des items d'array et des opérateurs
- *Types atomiques (chaîne, nombre, booléenne, null)* : prend des opérateurs
- *Tout* : prend des propriétés , des items d'array et des opérateurs
- *Multiple* : varie selon les types inclus ; prend l'union d'enfants autorisés pour les types sélectionnés
- *Opérateurs* : permet l'utilisation d'opérateurs logiques pour déterminer la structure

Les structures qui peuvent être créées pour chaque type sont décrites en détail dans les sections auxquelles elles sont liées depuis la liste ci-dessus.

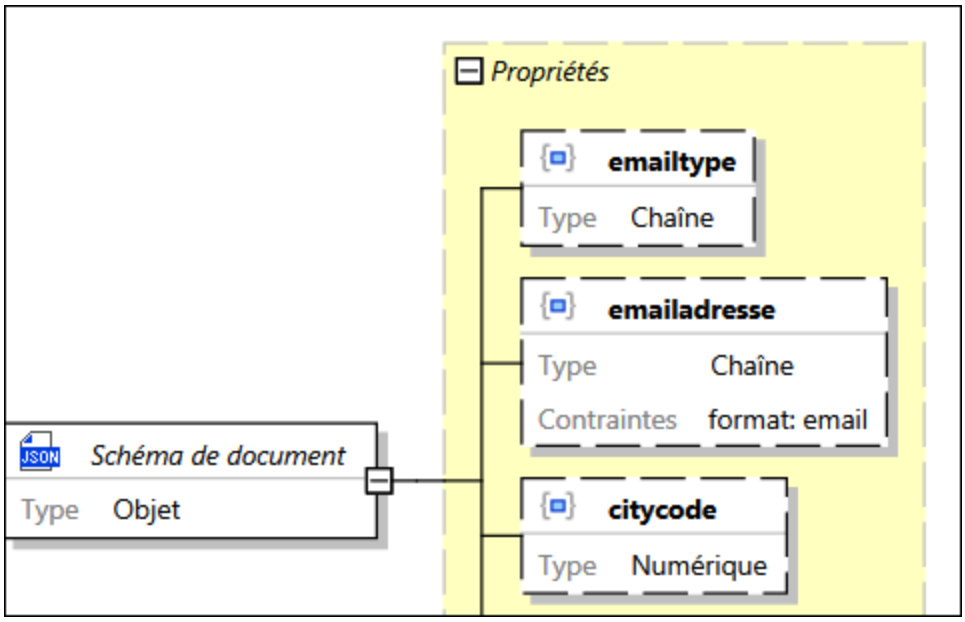
13.6.6 Objets et propriétés

Un objet est contenu dans des accolades et mappe une clé vers une valeur, comme ceci : `"MyKey": Value`. La clé doit toujours être une chaîne et doit donc toujours être contenue dans des guillemets. La valeur peut être de toute type de données JSON. Chaque paire `key:value` est connue en tant qu'une propriété de l'objet (*voir capture d'écran ci-dessous*).

Voici un exemple d'un objet instancié comportant trois propriétés :

```
{
  "emailtype": "home",
  "emailaddress": "contact01.home@altova.com",
  "citycode": 22
}
```

Le schéma pour l'objet ressemblerait à cela dans le Mode Design :



Veuillez noter :

- Chacune des propriétés doit être présente dans l'instance. Cela est indiqué par les bordures pleines des propriétés. Si une propriété est optionnelle, la bordure sera en tirets. Vous pouvez définir si une propriété est requise ou optionnelle dans le menu contextuel de la propriété ou par le biais de l'assistant à la saisie Détails.
- L'ordre dans lequel les propriétés doivent avoir lieu dans l'instance n'est pas, et ne peut pas être, défini dans le schéma. Cela signifie que l'ordre dans lequel les propriétés sont définies dans le schéma n'a pas d'importance.
- Le carré bleu contenu entre des crochets signifie une propriété (par rapport à une propriété de motif ou un caractère générique de propriété, qui sont tous deux symbolisés par d'autres symboles ; *voir ci-dessous*).
- Le type d'une propriété peut être édité en double-cliquant le type dans le diagramme et en sélectionnant une option depuis la liste déroulante qui apparaît. En alternative, le type peut être sélectionné dans l'assistant à la saisie Détails.
- La valeur de contrainte de la propriété `emailaddress` est définie dans l'assistant à la saisie Contraintes.

Propriétés, propriétés de motifs, caractères génériques de propriétés et schémas de noms de propriété

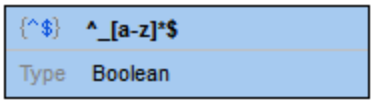
Un objet peut avoir des propriétés, des propriétés de motif, des caractères génériques de propriété et des schémas de nom de propriété (nouveau dans draft-07). Celles-ci peuvent être ajoutées à l'objet par le biais de menus contextuels : (i) de l'objet, (ii) de la fenêtre jaune de propriétés (cliquer avec la touche de droite sur le titre *Propriétés* de la fenêtre), et (iii) des propriétés individuelles. Les propriétés ont été décrites ci-dessus. Nous allons maintenant nous pencher sur les propriétés de motif et les caractères génériques de propriété.

Propriété de motif

Une **propriété de motif** (*capture d'écran ci-dessous*) définit le nom de la propriété en tant qu'expression régulière. Dans la capture d'écran ci-dessous, par exemple, l'expression régulière spécifie que la propriété doit : (i) avoir un nom qui commence avec un tiret bas, et (ii) avoir une booléenne en tant que sa valeur. Il n'y a pas

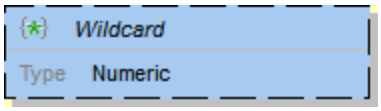
 © 2016-2022 Altova GmbH

de contrainte d'exigence pour une propriété de motif. Vous pouvez ajouter autant de propriétés de motif que vous le souhaitez. Veuillez noter l'icône pour les propriétés de motif.

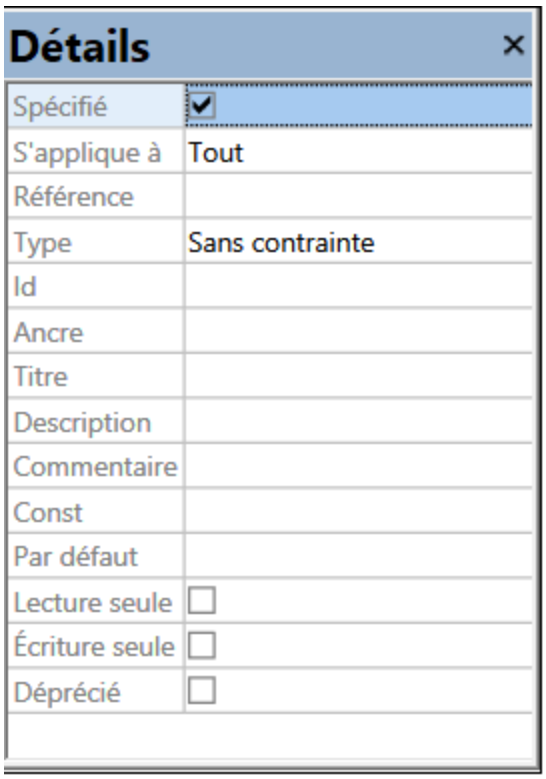


Caratère générique de propriété

Un **caractère générique de propriété** (*capture d'écran ci-dessous*) spécifie que tout nombre de propriétés peut se produire en plus des autres propriétés du set de propriétés de l'objet. Néanmoins, le caractère générique peut définir un type pour ces occurrences. La capture d'écran ci-dessous à gauche montre un caractère générique de propriété qui définit des propriétés avec n'importe quel nom mais possédant des valeurs numériques. Il ne peut y avoir qu'un caractère générique de propriété par objet. Si le caractère générique est défini sur le type *Tout*, néanmoins, vous pourrez définir des contraintes pour chaque type dans l'assistant à la saisie Contraintes. Veuillez noter l'icône pour les caractères génériques de propriété.



À partir de d*raft-2019-09* et ultérieur, les caractères génériques de propriété ont un nouveau mot-clé `unevaluatedProperties`, qui est traité uniquement si le mot-clé `additionalProperties` manque. Les valeurs de ces deux mots-clé sont produites en définissant des valeurs appropriées pour les entrées *Spécifé*, S'applique à, et entrées de *Type* dans l'assistant de saisie Détails du caractère générique (*voir la capture d'écran ci-dessous*).



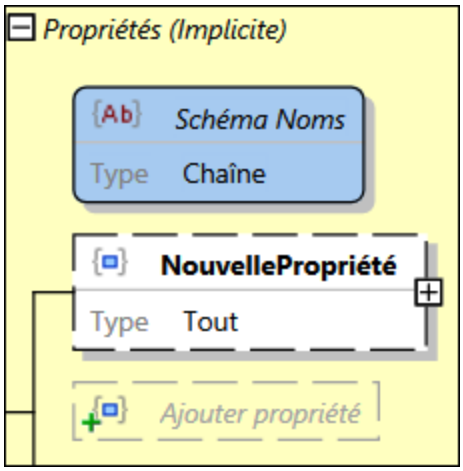
Les effets de ces valeurs sur les mots-clés `unevaluatedProperties` et `additionalProperties` (et, vice versa, l'effet des mots-clés sur les valeurs d'assistant à la saisie de l'éditeur) sont affichés dans la table ci-dessous. La capture d'écran ci-dessus, par exemple, définit `unevaluatedProperties=true`.

addtionalProperties	unevaluatedProperties		Spécifié	S'applique à	Type
--	--	<=>	false	Tous	Sans contrainte
true	ignoré	<=>	true	Tous	Sans contrainte
false	ignoré	<=>	--	--	--
Schéma	ignoré	<=>	true	Tous	Type de schéma
--	true	<=>	true	Non évalué	Sans contrainte
--	false	<=>	true	Non évalué	Interdit
--	Schéma	<=>	true	Non évalué	Type de schéma

Veuillez noter les points suivants :

- Si `addtionalProperties` et `unevaluatedProperties` sont présents, alors `unevaluatedProperties` is est ignoré.
- *Spécifié*=false ne fonctionne qu'avec *Scope*=Tous et *Type*=Sans contrainte.

Schéma de noms de propriété
Un **schéma de noms de propriété** (*capture d'écran ci-dessous*) limite les noms des propriétés de cet objet. (Cette fonction est nouvelle dans nouveauté dans draft-07.) Par exemple, dans la capture d'écran ci-dessous, nous pouvons voir que les noms des propriétés doivent être des strings. De plus, nous pouvons spécifier des limites supplémentaires pour le nom de la propriété par le biais de l'assistant à la saisir Contraintes : par exemple, que le nom de la propriété s'inscrit dans une plage de longueur de caractère bien définie ou qu'il possède un certain motif.



Note : il n'existe pas de paramètres d'occurrence minimum ou maximum pour une propriété de motif ou un caractère générique de propriété. Voir la section concernant la validation de propriété pour comprendre le fonctionnement.

© 2016-2022 Altova GmbH

Comment sont validées les propriétés

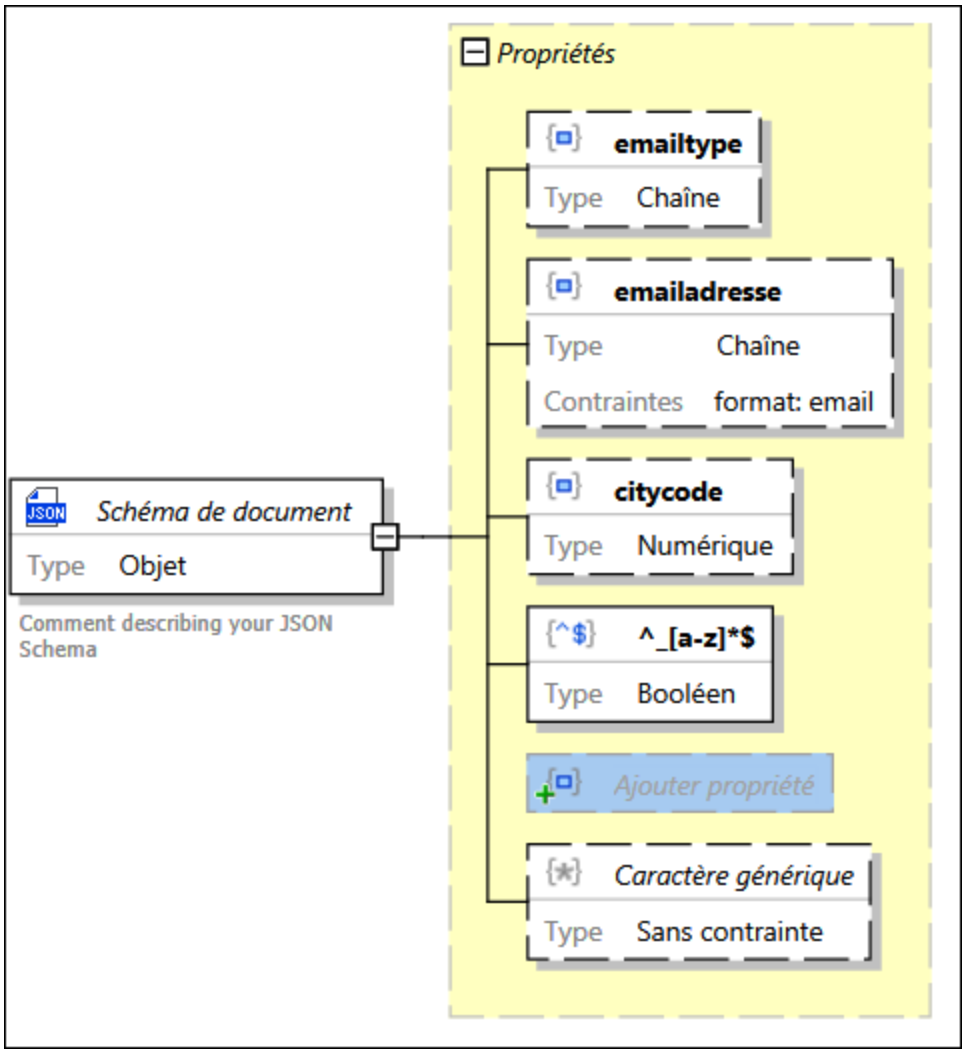
Lorsqu'une propriété est rencontrée dans l'instance, elle est validée comme suit :

1. Le nom de la propriété est contrôlé dans le schéma par rapport à toutes les propriétés nommées de cet objet.
2. Si aucune correspondance n'est trouvée, le nom est contrôlé par rapport à toutes les propriétés de motif contenues dans l'ensemble de propriété de l'objet.
3. Si aucune concordance n'est toujours pas trouvée, le caractère générique est invoqué s'il existe.
4. Si aucune concordance n'est toujours pas trouvée pour le nom, une erreur de validité est rapportée. Si le nom correspond à celui d'une propriété ou d'une propriété de motif, ou si un caractère générique existe, alors la valeur est contrôlée par rapport à la valeur de la définition de la propriété correspondante.
5. Si l'instance de la valeur correspond au type et aux contraintes de la définition de propriété correspondante, la propriété est valide. Sinon elle est invalide.

Exemple

La capture d'écran ci-dessous définit un objet qui :

- doit avoir trois propriétés nommées `emailtype`, `emailaddress`, et `citycode`
- peut avoir une ou plusieurs propriétés avec un nom qui commence avec un tiret bas et une valeur qui est booléenne (voir la propriété de motif dans la capture d'écran ci-dessous)
- peut avoir une ou plusieurs propriétés avec n'importe quel nom et valeur



13.6.7 Propriétés non spécifiées

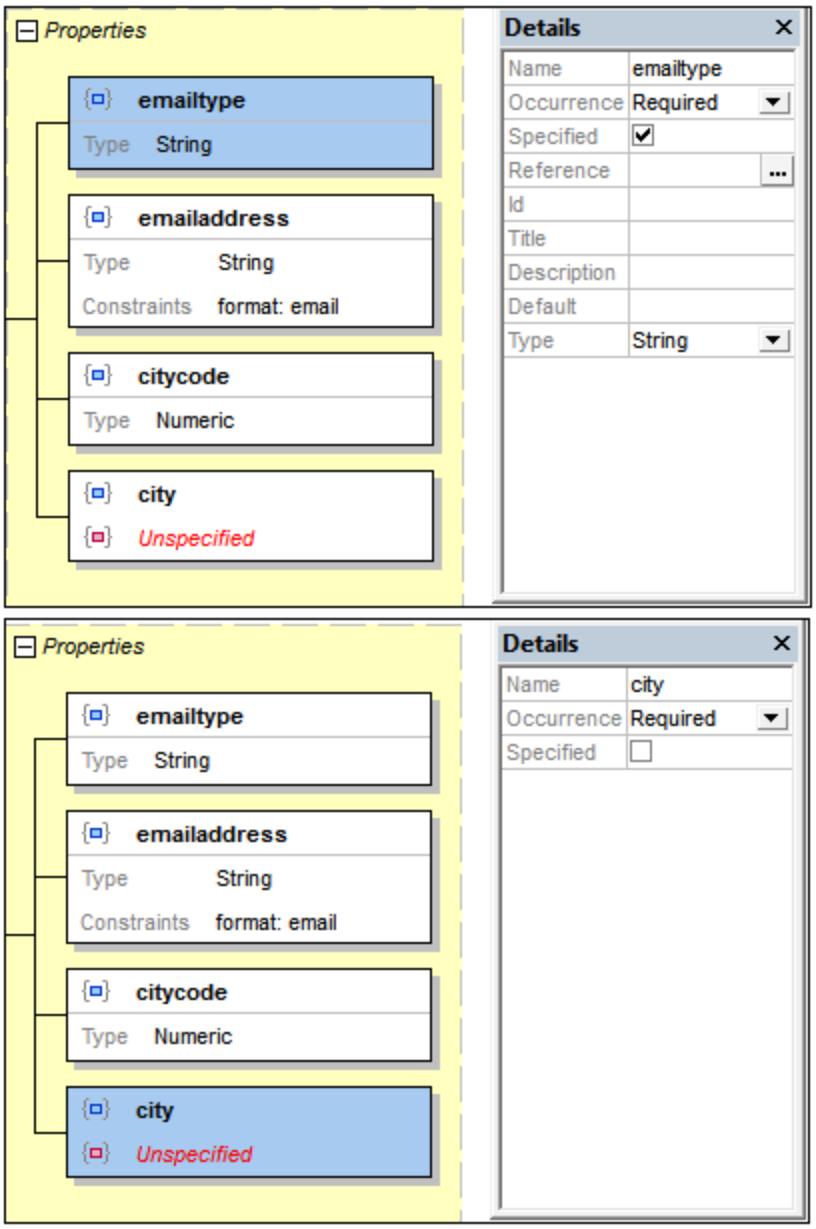
Dans la liste de code ci-dessous, le mot -clé **`required`** spécifie que quatre propriétés sont exigées pour cet objet. Néanmoins, des quatres propriétés requises, seules trois ont été définies. La quatrième propriété, **`city`**, est non-définie. Les propriétés définies sont nommées spécifiées, alors que la propriété non définie est nommée non-spécifiée. Voir les captures d'écran sous la liste.

Liste de code : propriétés spécifiées et non-spécifiées

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "JSON Schema generated by XMLSpy v2016 (http://www.altova.com)",
    "type": "object",
    "properties": {
```

 © 2016-2022 Altova GmbH

```
            "emailtype": {
                "type": "string"
            },
            "emailaddress": {
                "type": "string",
                "format": "email"
            },
            "citycode": {
                "type": "number"
            }
        },
        "required": [
            "emailtype",
            "emailaddress",
            "citycode",
            "city"
        ],
        "additionalProperties": false
    }
```



Dans le Mode Design, la propriété non spécifiée est marquée en rouge parce qu'elle est requise par le schéma, mais n'est pas définie. Bien que le schéma JSON lui-même soit valide, un document d'instance qui est validé par rapport au schéma ne sera pas valide. Cela est dû au fait que : (i) si la propriété `city` n'est pas présente, le document sera invalide parce que la propriété `city` sera requise ; (ii) Si la propriété `city` est présente, le document sera invalide parce que la propriété `city` est non-définie et qu'il n'y a pas de caractère générique de propriété pour permettre sa présence (*voir Spécifier implicitement une propriété ci-dessous*) .

Pour créer une définition pour une propriété non-spécifiée, procéder comme suit :

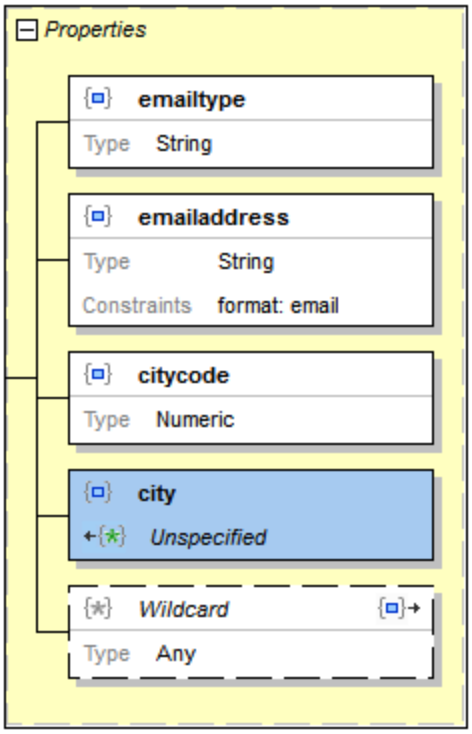
1. Sélectionner la propriété non-spécifiée dans le Mode Design.
2. Dans l'assistant à la saisie Détails, contrôler la case à cocher *Spécifié* (*voir capture d'écran ci-dessus*). En alternative, le marquage *Spécifié* peut être modifié par le biais du menu contextuel.

© 2016-2022 Altova GmbH

3. Modifier la définition de la propriété comme vous le souhaitez.

Spécifier implicitement une propriété

Une propriété peut être spécifiée implicitement en ajoutant une propriété de motif convenable ou un caractère générique de propriété. La capture d'écran ci-dessous montre qu'un caractère générique de propriété a été ajouté. Une propriété d'instance nommée `city` correspondra à ce caractère générique. Donc, dans le schéma, la propriété `city` est soit-disant "spécifiée implicitement" par le caractère générique. Un fichier d'instance contenant la propriété `city` sera valide par rapport à ce schéma.



Veuillez noter les icônes respectives dans la propriété spécifiée implicitement et dans le caractère générique de la propriété. Chaque icône est un lien à l'autre propriété. Double-cliquer sur une icône pour sélectionner l'autre propriété.

13.6.8 Objets et dépendances

Dans le cadre de la définition d'un objet, vous avez la possibilité de spécifier qu'une certaine propriété doit être présente uniquement si une autre propriété est présente. La première propriété est dite dépendante de la deuxième propriété. Voici un scénario contenant une dépendance. Un objet (nommé par exemple `member`) a une propriété appelée `credit_card`, qui est définie en tant qu'optionnelle. La propriété de l'objet `billing_address` peut être rendue dépendante de la propriété `credit_card` : ce n'est que si la propriété `credit_card` est présente que la propriété `billing_address` sera présente.

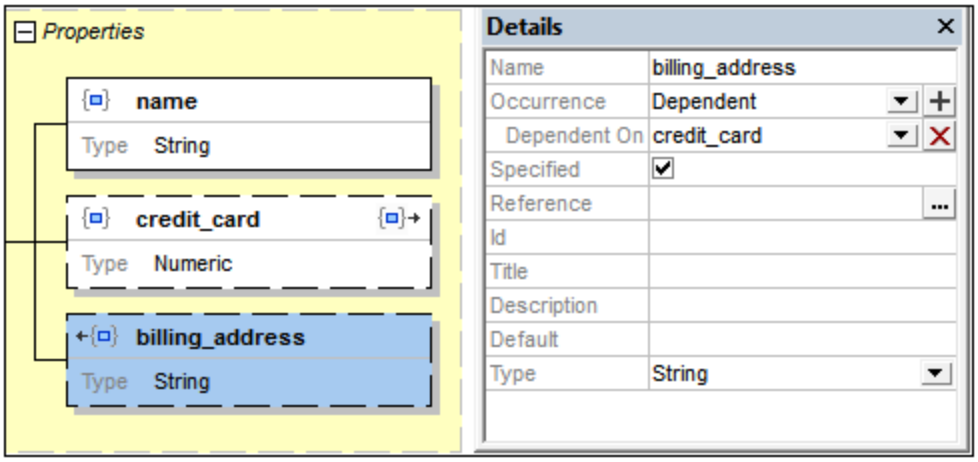
Ce type de dépendance peut être spécifié de deux manières :

- en tant qu'une dépendance de propriété (la structure dépendante est une propriété)

- en tant qu'une dépendance de schéma (la structure dépendante est un schéma)

Dépendances de propriété

La capture d'écran ci-dessous montre un objet possédant une propriété `name` (requise), une propriété `credit_card` (optionnelle), et une propriété `billing_address` (dépendante). La propriété `billing_address` est dépendante de la propriété `credit_card`. Le code de cette définition d'objet JSON est contenu en-dessous de la capture d'écran. Vous trouverez comment créer une dépendance de propriété ci-dessous.



Liste de code d'un objet JSON avec une dépendance de propriété

```
{
   "type": "object",
   "properties": {
      "name": {
      "type": "string"
      },
      "credit_card": {
         "type": "number"
      },
      "billing_address": {
         "type": "string"
      }
   },
   "required": [ "name" ],
   "dependencies": {
      "credit_card": [ "billing_address" ]
   },
   "additionalProperties": false
}
```

Afin de créer une dépendance de propriété, procéder comme suit :

1. Cliquer avec la touche de droite à l'endroit où la dépendance sera basée. (Dans notre exemple, il s'agit de la propriété `credit_card`.)
2. Dans le menu contextuel qui apparaît, sélectionner **Ajouter dépendance | Propriété dépendante**. Une nouvelle propriété est ajoutée avec une valeur Occurrence de *Dependent*.

© 2016-2022 Altova GmbH

3. Définir le nom et la valeur de cette propriété, et ajouter tous les détails supplémentaires ou les contraintes que vous souhaitez.

Pour spécifier une propriété comme étant dépendante d'une autre propriété, procéder comme suit :

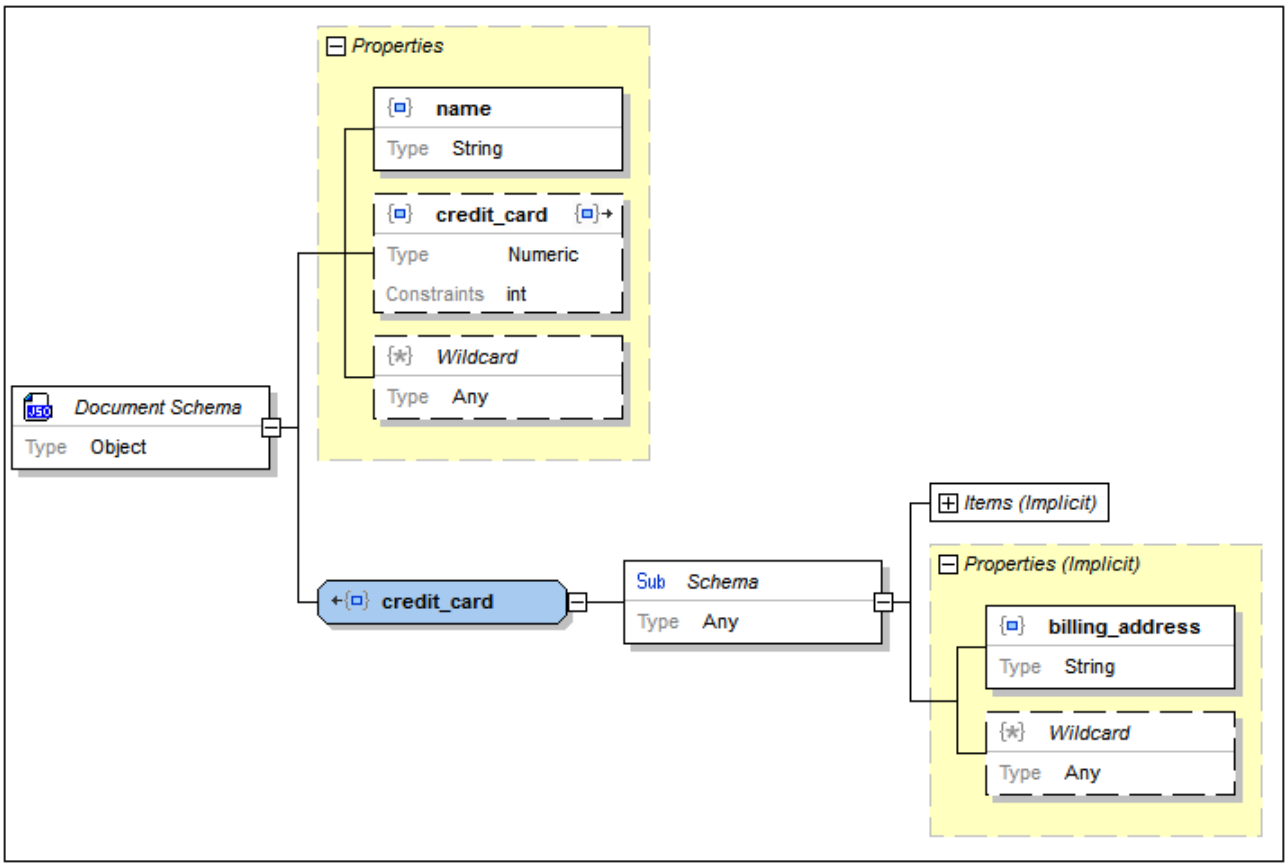
1. Cliquer avec la touche de droite sur la propriété que vous souhaitez rendre dépendante sur une autre propriété. (Dans notre exemple, il s'agit de la propriété `billing_address`.)
2. Dans le menu contextuel qui apparaît, sélectionner **Dépendant**. En alternative, dans l'assistant à la saisie Détails, rendez-vous dans l'entrée *Occurrence*, et choisissez *Dependent* (*voir capture d'écran ci-dessus*).
3. Dans l'assistant à la saisie Détails, cliquer sur l'icône de la liste déroulante de l'entrée *Dependent On*. La liste déroulante affiche toutes les autres propriétés de l'objet. Sélectionnez la propriété sur laquelle vous souhaitez que la propriété actuelle dépende.

Note: Une icône apparaît dans les fenêtres des deux propriétés impliquées dans une dépendance (*voir capture d'écran ci-dessus*). Double-cliquer sur l'icône d'une propriété pour vous amener sur l'autre propriété.

Note: Une propriété peut avoir plusieurs propriétés dépendantes.

Dépendances de schéma

La capture d'écran ci-dessous montre un objet qui décrit la même structure de données d'instance que l'objet discuté dans la section précédente. Les définitions des deux objets, néanmoins, sont différentes. Alors que la définition précédente a utilisé une *dépendance de propriété* pour définir la propriété `billing_address` comme étant dépendante de la propriété `credit_card`, la définition actuelle utilise une *dépendance de schéma* pour définir cette dépendance. Le code de cette définition d'objet JSON est contenu en-dessous de la capture d'écran. La création d'une dépendance de schéma est décrite ci-dessous.



Liste de code d'un objet JSON avec une dépendance de schéma

```
{
   "type": "object",
   "properties": {
      "name": {
         "type": "string"
      },
      "credit_card": {
         "type": "integer"
      }
   },
   "required": [ "name" ],
   "dependencies": {
      "credit_card": {
         "properties": {
            "billing_address": {
               "type": "string"
            }
         },
         "required": [ "billing_address" ]
      }
   }
}
```


© 2016-2022 Altova GmbH

Pour créer une dépendance de schéma, procéder comme suit :

1. Cliquer avec la touche de droite sur la propriété sur laquelle la dépendance se basera. (Dans notre exemple, il s'agit de la propriété `credit_card`.)
2. Dans le menu contextuel qui apparaît, choisir **Ajouter dépendance | Dépendance de schéma**. Une nouvelle définition d'objet est créée. Elle aura le même nom que la propriété à laquelle elle est dépendante (dans notre exemple, `credit_card`), et elle aura un sous-schéma enfant.
3. Définir le sous-schéma de la manière que vous souhaitez en ajoutant des détails ou des contraintes supplémentaires le cas échéant.

Note : Une icône apparaît dans les fenêtres de la propriété et de l'objet impliqué dans une dépendance (*voir capture d'écran ci-dessus*). Double-cliquer sur l'icône dans une fenêtre pour vous rendre dans l'autre fenêtre.

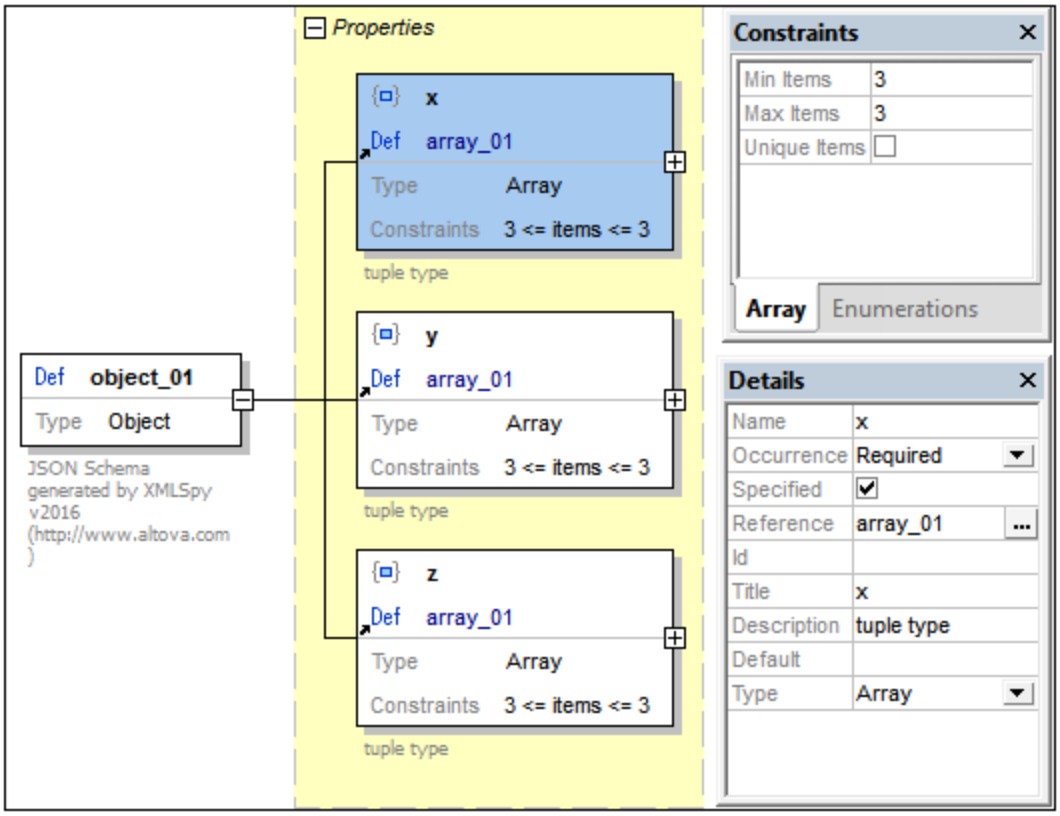
Note : Si vous souhaitez définir plusieurs dépendances, utilisez le sous-schéma dépendant (*voir capture d'écran ci-dessus*).

13.6.9 Tableaux

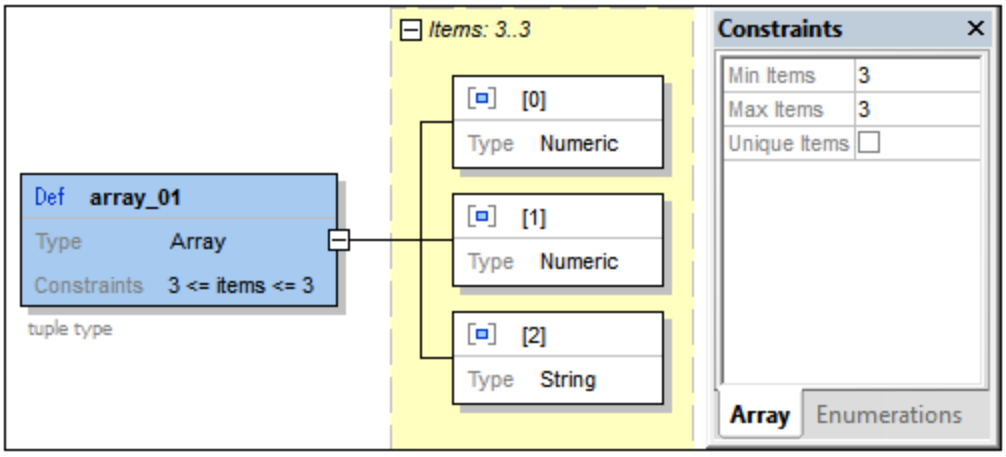
Un tableau est une liste de zéro ou de plusieurs items ordonnés ; il est contenu dans des crochets. Un type est attribué à chaque item de la liste. La liste d'instance ci-dessous est celle d'un objet avec trois propriétés. La valeur de chaque propriété est un tableau (*les délimiteurs sont marqués en jaune*).

```
{
   "x": [ 1, 2, "abc" ],
   "y": [ 3, 4, "def" ],
   "z": [ 5, 6, "ghi" ]
}
```

Les trois tableaux de la liste ci-dessus ont la même définition. Chacun contient trois items ordonnés dans l'ordre suivant : (i) un item de nombre, (ii) un item de nombre, (iii) un item de chaîne. Une description de schéma de cet objet est affiché dans la capture d'écran ci-dessous. Puisque la définition est la même pour les trois tableaux, la définition a été créée dans un tableau global nommé `array_01`. Chacun de ces trois tableaux (`x`, `y`, et `z`) référencent le tableau global `array_01`.



Dans la capture d'écran ci-dessus, le tableau **x** est sélectionné (indiqué par un marquage bleu), et ses détails et contraintes sont affichés dans les assistants à la saisie respectifs (*voir capture d'écran ci-dessus*). Veuillez noter la contrainte sur le nombre d'items autorisés. Le nombre peut être modifié dans l'assistant à la saisie Contraintes et est affiché dans le diagramme. Les items de tableau peuvent être définis dans la définition du tableau lui-même, qui dans ce cas, est la définition globale `array_01` (*capture d'écran ci-dessous*).



Veuillez noter les points suivants :

- La contrainte `unique` spécifie que tous les items dans le tableau doivent être uniques.

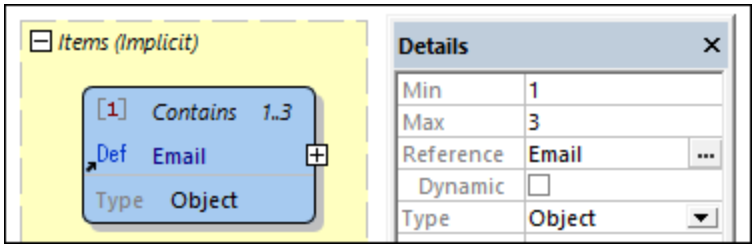
 © 2016-2022 Altova GmbH

- La numérotation des items commence avec `0`.
- Les formulations suivantes dans le diagramme, *3 <= items <= 3* et *Items: 3..3* (*voir capture d'écran ci-dessus*), indiquent toutes les deux les items minimum et maximum autorisés. Dans ce cas, trois items exactement doivent être présents.

Ajouter des items de tableau, des caractères génériques d'item de tableau et le mot-clé contains

Les items de tableau, caractères génériques d'item de tableau et le mot-clé **`contains`** d'un tableau sont ajoutés par le biais du menu contextuel d'une définition ou d'un item de tableau.

- Un caractère générique de tableau permet un plus large éventail d'objets qui peuvent être insérées dans le tableau.
- Le mot-clé **`contains`** spécifie que la valeur du mot-clé **`contains`** doit être un schéma JSON valide et qu'au moins un des éléments du tableau doit être valide par rapport à un objet de schéma référencé. À partir de draft 2019-09 et ultérieur, les mots-clés **`minContains`** et **`maxContains`** ont été introduits. Dans l'assistant d'entrée Détails, ils sont affichés comme les propriétés *Min* et *Max* de la case Contient (*voir la capture d'écran ci-dessous*), et ils définissent combien d'éléments pourraient correspondre avec l'objet de schéma référencé.



13.6.10 Types Atomiques

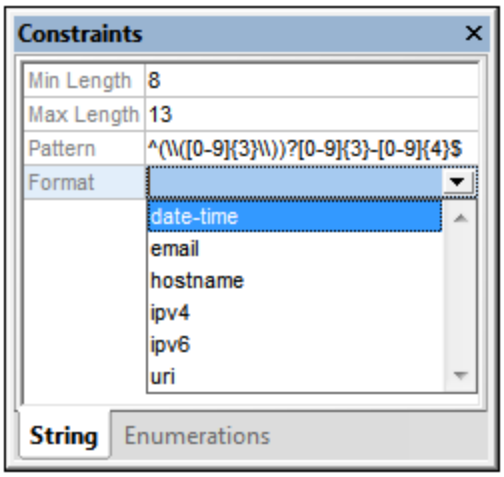
Il existe cinq types atomiques JSON (aka simple or primitive) types: (i) `string`, (ii) `number`, (iii) `integer`, (iv) `boolean` et (v) `null`. Pour spécifier qu'une définition est un de ces types atomiques, procédez à l'une des deux étapes :

- Double-cliquer sur le champ de valeur *Type* dans la fenêtre de définition et sélectionner le type
- Dans l'assistant à la saisie Détails, sélectionner le type provenant de la liste déroulante dans le champ *Type*.

Les contraintes de chaque type atomique sont décrites ci-dessous.

Chaîne

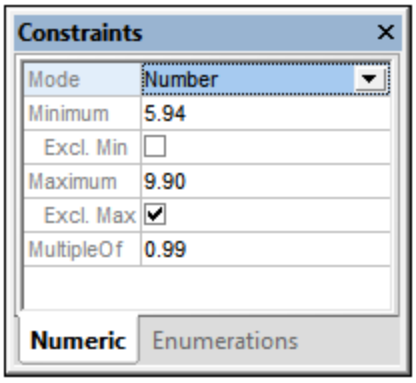
Pour le type de chaîne, vous pouvez spécifier les contraintes suivantes : (i) longueur de la chaîne, (ii) une expression régulière qui décrit le motif de la chaîne, (iii) un format prédéfini depuis la spécification.



Note : dans les paramètres de validation JSON de XMLSpy, vous pouvez spécifier si le format de chaînes dans les documents d'instance JSON doit être validé ou non.

Numérique

Le type numérique désigne un nom collectif pour deux types (`number` et `integer`; *voir capture d'écran ci-dessous*). Le type actuel est défini dans le champ *Mode* (dont le défaut est `number`). La différence entre les deux types est que le type `number` autorise les décimales alors que le type `integer` ne l'autorise pas. Si une valeur existe dans le champ *MultipleOf*, alors la valeur d'instance doit être un entier multiple de la valeur *MultipleOf*.



Les valeurs valides pour le type `number` défini dans la capture d'écran ci-dessus sont : `5.94`, `6.93`, `7.92`, et `8.91`.

Booléenne et Null

Le type `boolean` prend soit `true` ou `false` en tant que ses valeurs. Le type `null` prend `null` en tant que sa valeur. Aucun des types ne prend une contrainte.

© 2016-2022 Altova GmbH

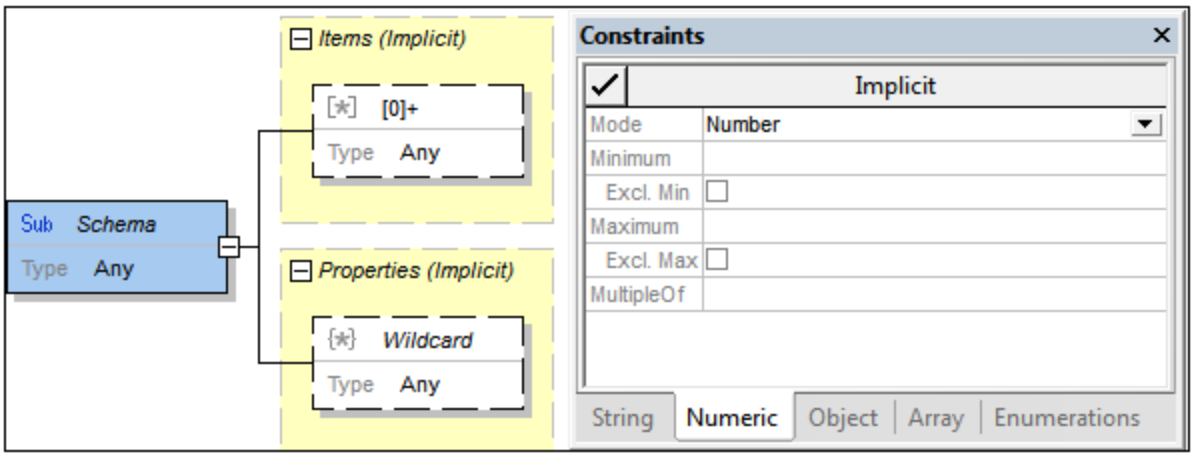
13.6.11 Sélecteurs de type (Any, Multiple)

Dans les listes déroulantes des listes de choix Types que vous trouverez dans le Mode JSON Schema, il y a quatre "types" qui ne sont pas des types JSON : **`any`**, **`multiple`**, **`unconstrained`** et **`forbidden`**. Il s'agit en fait de sélecteurs de type.

- Le sélecteur de type **`any`** sélectionne tout type JSON. Cela signifie que, dans l'instance, tout type JSON sera valide pour cette définition particulière.
- Le sélecteur de type **`multiple`** sélectionne un ou plusieurs types JSON. Cela signifie que si le type d'instance est un des types JSON sélectionné dans le schéma, le type d'instance sera valide pour cette définition particulière.
- Le sélecteur de type **`unconstrained`** (nouveau dans draft-06) ne définit pas de contrainte dans le type JSON. Cela signifie que, dans l'instance, tout type JSON sera valide pour une définition avec ce nom.
- Le sélecteur de type **`forbidden`** (nouveau dans draft-06) interdit tout type JSON, ne permettant ainsi pas l'existence d'une définition portant ce nom.

Le sélecteur de type any

Le type de sélecteur `any` peut être sélectionné dans tous les endroits où il est possible de sélectionner un type. Lorsqu'une définition est ajoutée au schéma, `any` est la sélection de type par défaut. Elle spécifie que l'un des types JSON est valide. Cela signifie que le type d'instance peut validement être un objet, un tableau, ou un des types atomiques (`string`, `number`, `integer`, `boolean`, et `null`).



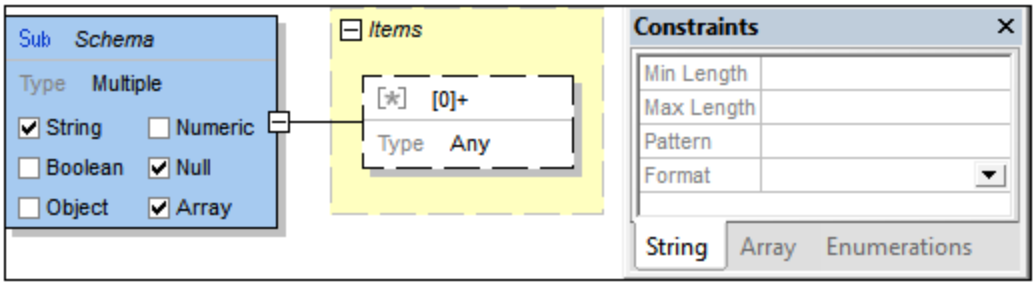
Dans la capture d'écran ci-dessus, le sous-schéma a un type de `Any`. Donc, tous les types JSON sont valides pour cette définition. Les éléments suivants sont implicites et sont mis en place conformément à la UI :

- Puisque les objets sont autorisés, une fenêtre propriétés est créée automatiquement (*voir capture d'écran ci-dessus*). La fenêtre propriétés est définie par défaut pour autoriser toute quantité de propriétés de tout type (par le biais d'un caractère générique de propriété avec un type de `Any`). Vous pouvez modifier les définitions de propriété comme vous le souhaitez.
- Puisque des tableaux sont autorisés, une fenêtre d'item est créée automatiquement (*voir capture d'écran ci-dessus*). La fenêtre d'items de tableau est définie par défaut pour autoriser tout nombre d'items de tableau de tout type (par le biais d'un caractère générique d'item de tableau avec un type de `Any`). Vous pouvez modifier les définitions d'items comme vous le souhaitez.
- Puisque les types string et numeric (nombre et entier) sont autorisés, les contraintes pour ces types atomiques peuvent être définis dans l'assistant à la saisie Contraintes.

C'est pourquoi tous ces types sont définis de manière implicite avec le sélecteur de type `Any`. Afin de changer le type en un type spécifique, sélectionner ce type. Il existe un moyen alternatif de spécifier des objets et des tableaux en tant que type : cliquer avec la touche de droite sur l'objet ou le tableau et sélectionner **Rendre explicite**. Cela fait de ce type le type sélectionné et supprime les autres types ou rend les types d'objet/tableau définis inactifs.

Le sélecteur de type multiple

Le sélecteur de type `multiple` peut être sélectionné partout où un type peut être sélectionné. Il vous permet de sélectionner un ou plusieurs types JSON en contrôlant les types que vous souhaitez autoriser (*voir capture d'écran ci-dessous*). Vous pouvez ensuite spécifier les contraintes pour les types sélectionnés dans l'assistant à la saisie Contraintes.



Dans la capture d'écran ci-dessus, le sous-schéma permet des types de `string`, `null`, et `array`. Les contraintes pour ces types peuvent être définies dans l'assistant à la saisie Contraintes (*voir capture d'écran*).

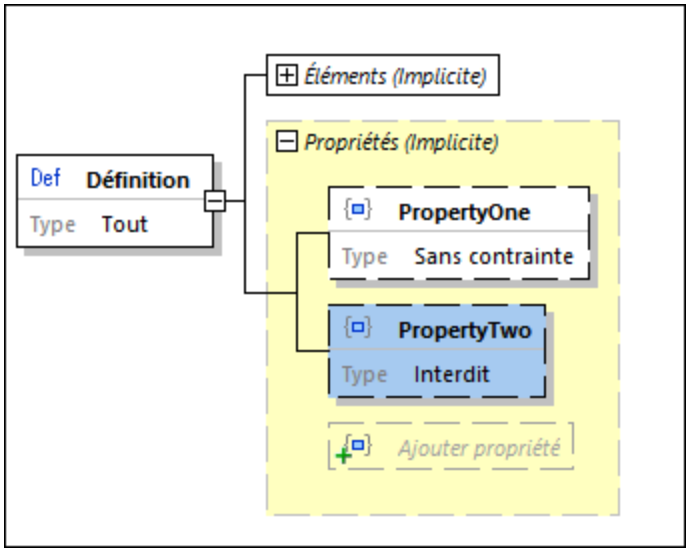
- Les contraintes de chaîne sont définies dans l'assistant à la saisie Contraintes.
- Le type null ne prend plus d'autres contraintes.
- Une fenêtre d'items de tableau est créée automatiquement. Vous pouvez définir le nombre et les types des items de tableau autorisés.

Dans un document d'instance, les types sélectionnés seront autorisés à l'endroit correspondant à celui du sous-schéma.

Les sélecteurs de types unconstrained et forbidden

Les sélecteurs de type `unconstrained` et `forbidden` peuvent être sélectionnés partout où un type peut être sélectionné. Ils vous permettent de spécifier, respectivement, qu'un objet de tout type est autorisé ou bien qu'aucun objet de ce nom n'est autorisé.

© 2016-2022 Altova GmbH



Dans la capture d'écran ci-dessus, une définition a deux propriétés. **`PropertyOne`** peut avoir une valeur de tout type, alors qu'aucune propriété nommée **`PropertyTwo`** n'est autorisée (*voir capture d'écran*). Dans sa forme de texte, cette structure ressemblera à l'extrait de code ci-dessous.

```
"Definition": {
	"properties": {
		"PropertyOne": true,
		"PropertyTwo": false
	}
}
```

13.6.12 BSON (Binary JSON) for MongoDB

La plateforme de données d'application MongoDB stocke des données telles des structures JSON, mais dans une représentation binaire de données. Cette représentation est connue en tant que « JSON binaire » ou BSON. Les principaux avantages d'utiliser BSON pour MongoDB sont :

- Le format binaire de BSON est plus rapide à parser que le texte d'un document JSON.
- Puisque JSON a un datatyping limité, BSON a été doté de plus de types de données (en particulier plus de types de données numériques).

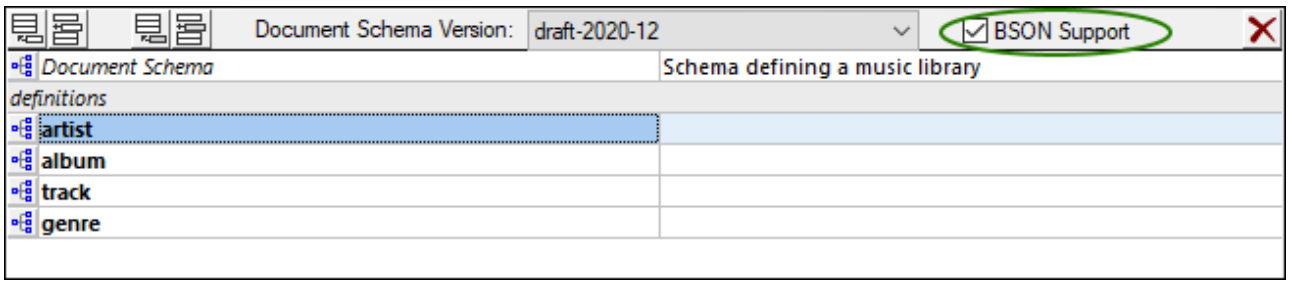
Pour plus d'information sur MongoDB et BSON, voir cette page sur le site web de MongoDB.

Éditer les schémas JSON pour les données BSON

Lors de l'addition de nouvelles données DB et de la modification de données DB, MongoDB permet aux données DB d'être validées par rapport au document de schéma JSON. Toutefois, en raison des types de données BSON supplémentaires, qui ne font pas partie des spécifications de schéma JSON officielles mais les complémentent, les schémas JSON pour BSON sont édités dans XMLSpy via une couche d'édition pour

BSON qui est superposée sur l'éditeur de schéma JSON. Cette couche d'édition vous permet d'ajouter et de modifier des fonctions de schéma spécifiques à BSON - en plus de l'addition aux fonctions JSON standard.

En conséquence, vous pouvez éditer un document de schéma JSON de toute version avec ou sans la couche d'édition BSON. Lorsque le document est édité sans couche d'édition BSON, il est édité en tant que document de schéma JSON. Avec le support BSON, le document de schéma JSON peut définir en plus les fonctions spécifiques à BSON. Pour basculer vers les fonctions d'édition BSON dans le Mode JSON Schema *pour le document actif*, sélectionnez la case à cocher *Support BSON* en haut à droite de la fenêtre principale *(encerclée en vert dans la capture d'écran ci-dessous)*. Veuillez noter que la prise en charge BSON (i) peut être activée indépendamment de la version de schéma JSON qui a été sélectionnée et (ii) s'applique uniquement au document actuel ; elle peut être activée/désactivée pour chaque document séparément.

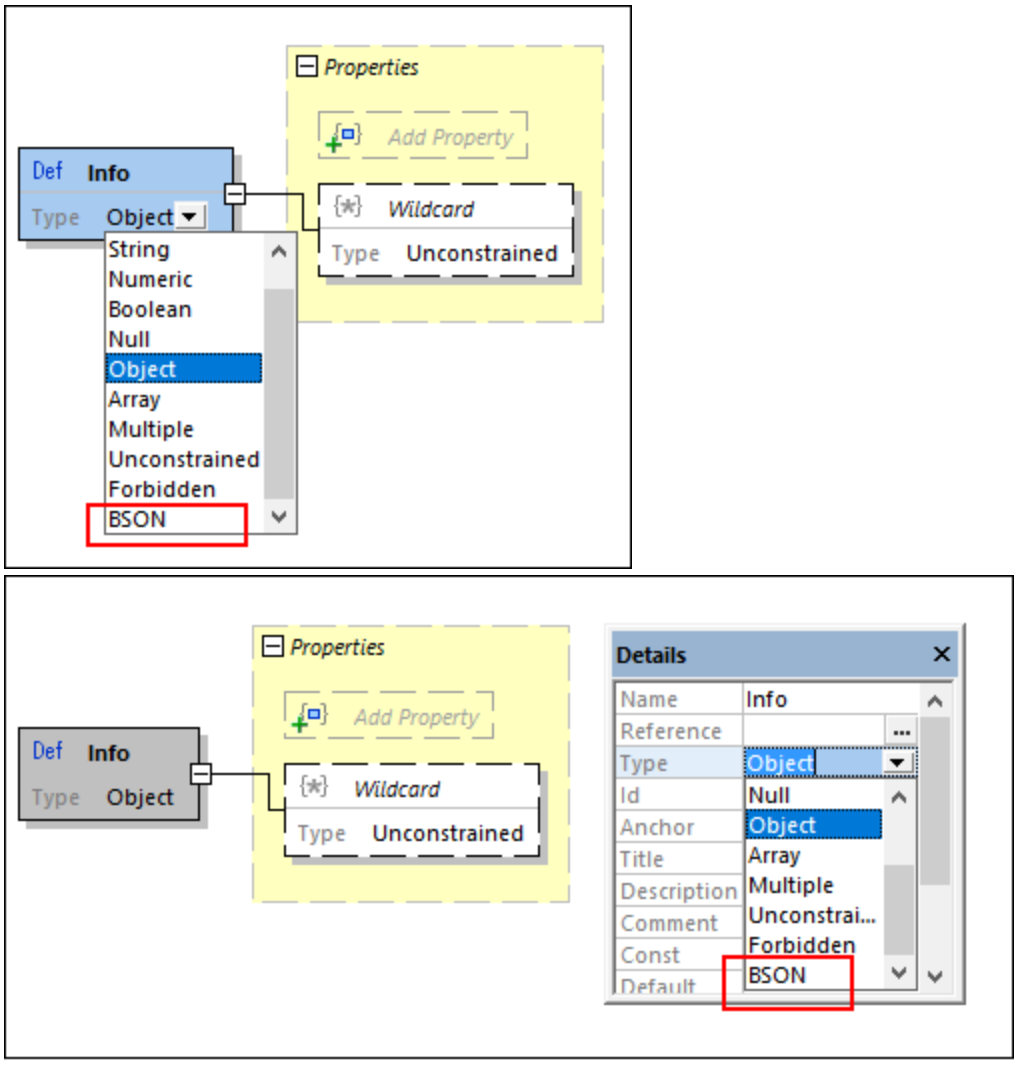


Types BSON

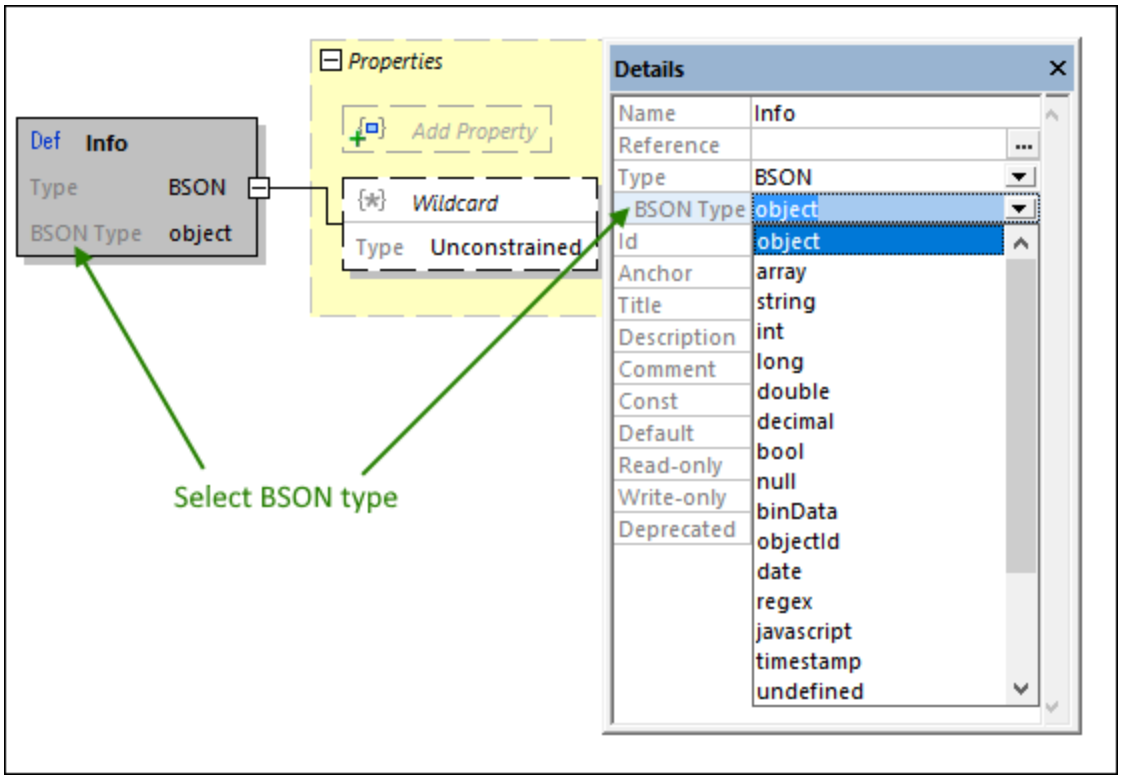
Après avoir activé la couche d'édition BSON, les types de données BSON deviennent disponibles pour les objets JSON, propriétés et items d'array. Spécifier qu'un composant est un type BSON consiste en deux étapes à suivre :

1. Spécifier que le composant est un type de données BSON (et non un type de données JSON) en sélectionnant BSON en tant que base du type JSON. À cette fin, spécifiez le statut dans la zone de liste du sélecteur de type de données du composant (en double-cliquant sur la valeur du type ; *voir la capture d'écran ci-dessous à gauche*) ou l'assistant de saisie Détails du composant (*voir la capture d'écran ci-dessous à droite*).

© 2016-2022 Altova GmbH



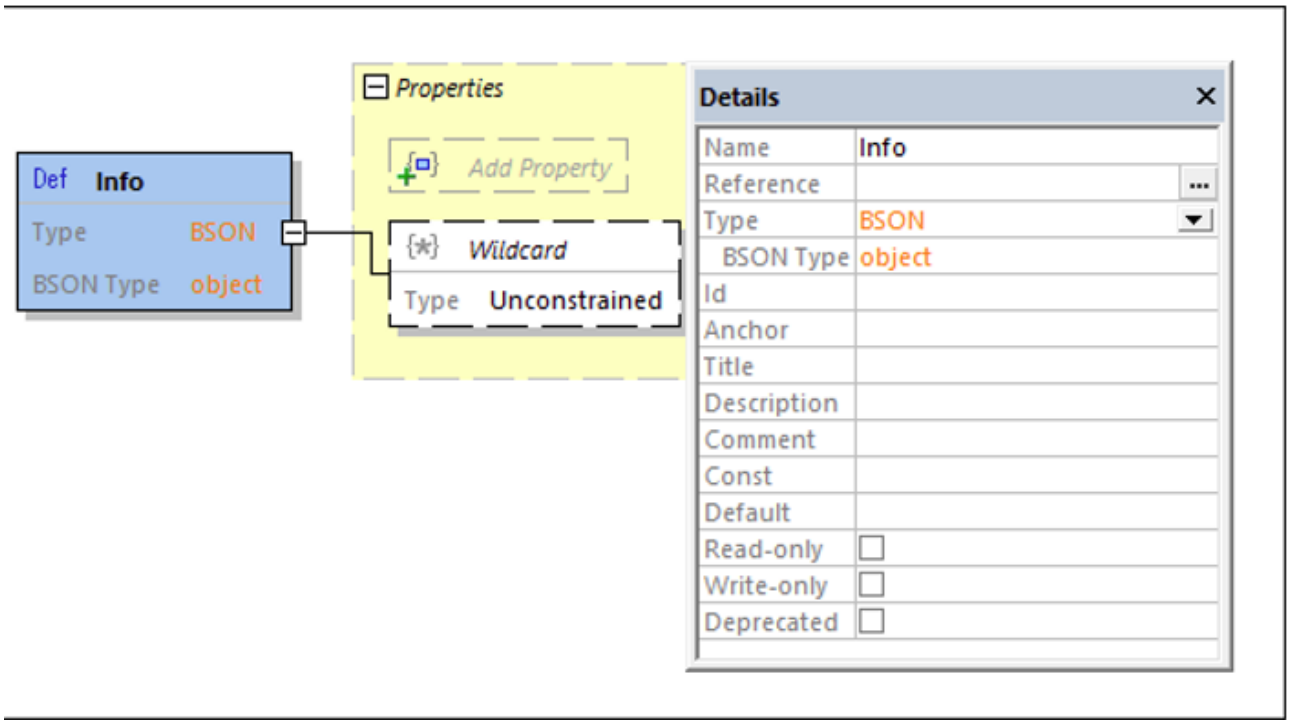
2. Une fois le type de base du composant sélectionné en tant que *BSON*, les types BSON deviennent disponibles pour la sélection. Sélectionnez le type BSON soit dans la sélection du type de données du composant (*touche de direction vers la gauche dans la capture d'écran ci-dessous*) ou dans l'assistant d'entrée Détails (*touche de direction à droite dans la capture d'écran ci-dessous*).



Désactiver la prise en charge BSON

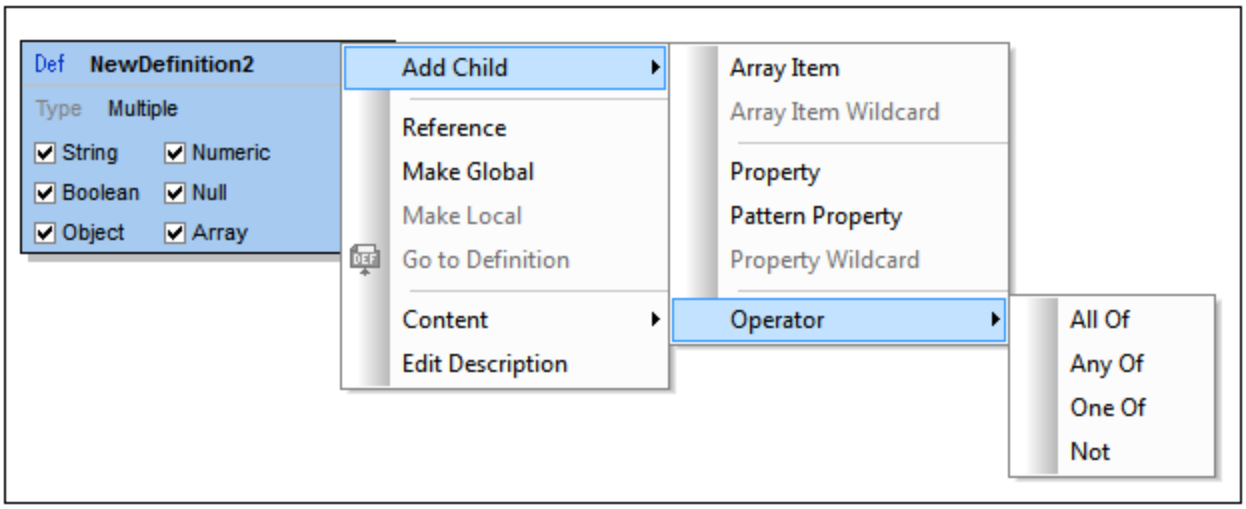
Si vous assignez un type BSON à un composant de schéma JSON (tel que décrit ci-dessus), puis décochez l'option *prise en charge BSON* pour le document, une boîte de message apparaitra. Le message vous informe qu'il existe des types BSON dans le document et vous demande si vous souhaitez supprimer/convertir des types BSON ou les garder. Si vous souhaitez les supprimer/convertir, alors ces types BSON qui peuvent être convertis en types JSON seront convertis tandis que les autres seront supprimés. Si vous souhaitez garder les types BSON, ils seront maintenus - mais affichés en orange dans le Mode Schéma car la prise en charge du type BSON a été supprimée (*voir la capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



13.6.13 Opérateurs

Il y a quatre opérateurs : (i) `allOf`, (ii) `anyOf`, (iii) `oneOf`, et (iv) `not`. Les opérateurs sont utilisés pour spécifier des conditions de validité telles qu'expliquées ci-dessous. Vous pouvez ajouter un opérateur à chaque définition. Pour accéder au sous-menu de l'opérateur, cliquer avec la touche de droite sur la définition avec laquelle vous souhaitez ajouter un opérateur, puis sélectionner **Ajouter enfant | Opérateur** (*voir capture d'écran ci-dessous*).

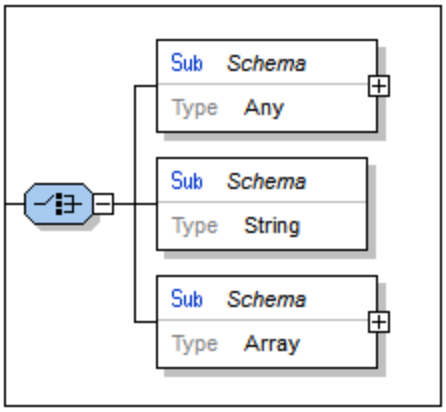


Ces opérateurs spécifient les conditions pour une validation réussie comme suit :

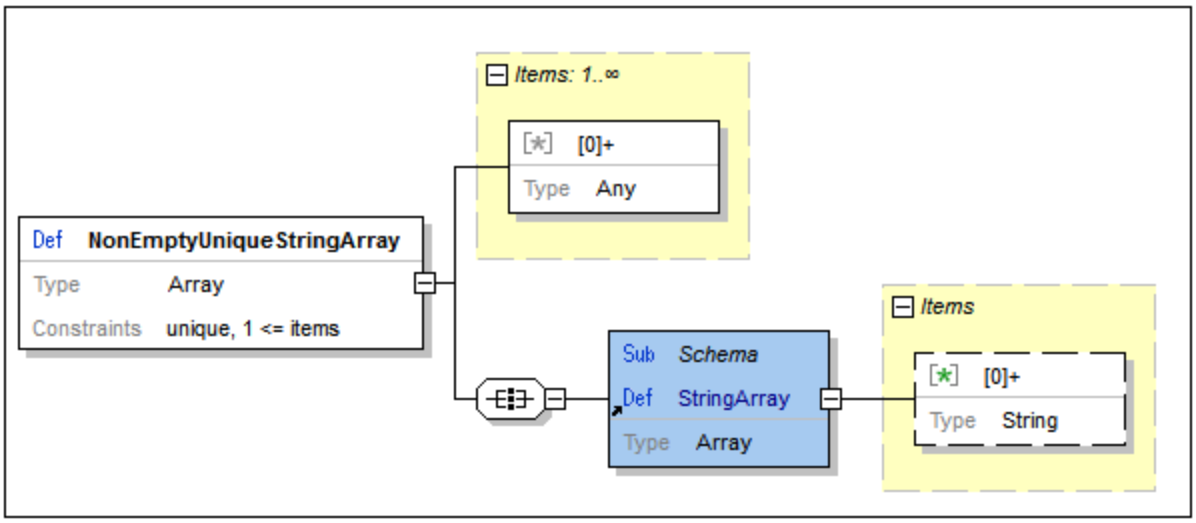
Opérateurs	Icône	Description
	All Of	Contient un ou plusieurs sous-schémas (définitions), ajoutés en tant qu'enfants de l'opérateur. Une instance est valide si elle est valide par rapport à tous ces sous-schémas.
	Any Of	Contient un ou plusieurs sous-schémas (définitions), ajoutés en tant qu'enfants de l'opérateur. Une instance est valide si elle est valide par rapport à au moins un de ces sous-schémas.
	One Of	Contient un ou plusieurs sous-schémas (définitions), ajoutés en tant qu'enfants de l'opérateur. Une instance est valide si elle est valide par rapport à exactement un de ces sous-schémas.
	Not	Contient exactement un sous-schéma (définition), ajouté en tant qu'un enfant de l'opérateur. Une instance est valide si elle est invalide par rapport à définition donnée.

La capture d'écran ci-dessous montre un opérateur *One Of* qui contient trois sous-schémas enfants (définitions). Pour que l'instance soit valide, elle doit avoir une structure de données JSON (à cet endroit de la structure du document) qui correspond à une des définitions des trois sous-schémas.

© 2016-2022 Altova GmbH



Les opérateurs peuvent être utiles pour spécifier l'héritage et les restrictions. La capture d'écran ci-dessous, par exemple, montre comment utiliser l'opérateur *All Of* pour définir un tableau contenant des chaînes uniques non-vides.

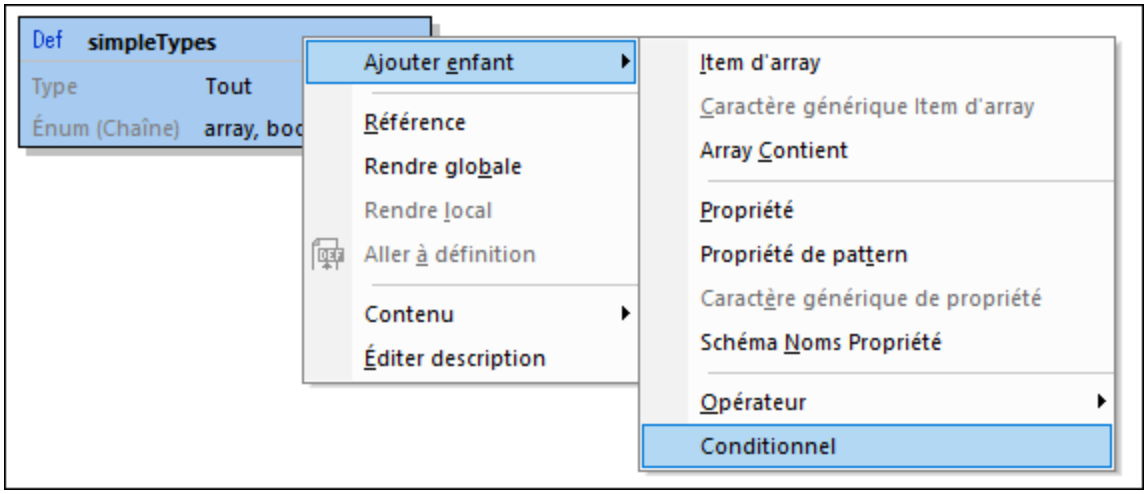


13.6.14 Conditionnels

Les conditionnels sont une nouvelle fonction dans draft-07. Ils permettent de modifier les différences entre les limitations de validation en raison de certains aspects de l'objet, comme son type et/ou des limitations spécifiques au type supplémentaires.

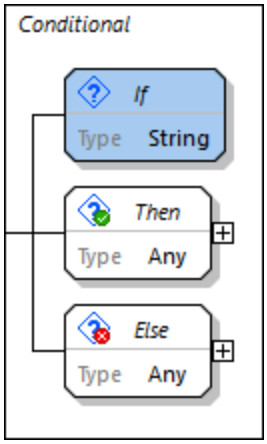
Ajouter un conditionnel

Vous pouvez ajouter un conditionnel à toute définition par le biais du menu contextuel de la définition *(voir capture d'écran ci-dessous)*. Pour accéder au sous-menu du conditionnel, cliquer avec la touche de droite sur la définition que vous souhaitez ajouter le conditionnel, puis sélectionner **Ajouter enfant | Conditionnel**.



Configurer la validation conditionnelle

Le conditionnel s'ajoute en tant que cadre avec trois éléments : **`If-Then-Else`** *(voir capture d'écran ci-dessous)*.



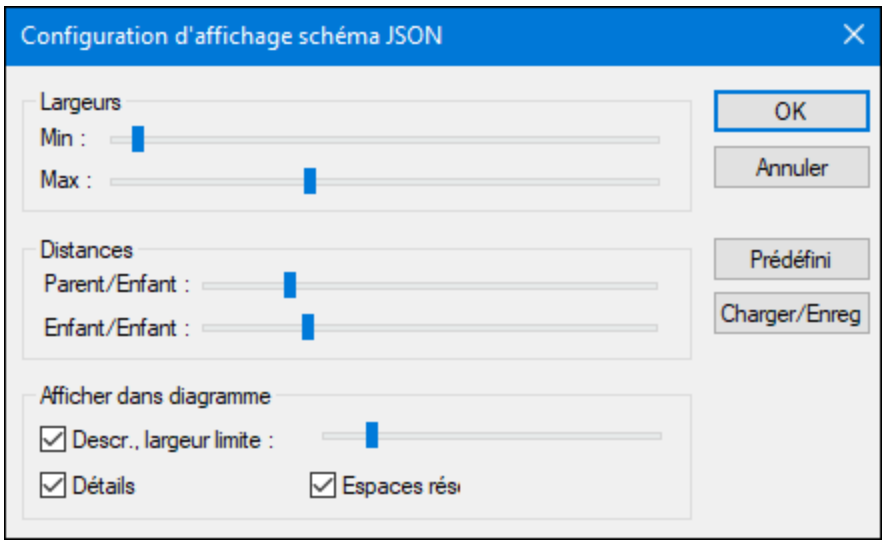
Pour configurer la validation, procéder comme suit :

1. Configurer la condition dans le cadre *If* en choisissant tout d'abord un type dans l'assistant à la saisie Détails puis une limite basée sur le type dans l'assistant à la saisie Contraintes.
2. Dans le cadre *Then*, configurer les exigences de validation dans le cas où la condition (spécifiée dans le cadre *If*) est remplie.
3. Dans le cadre *Else*, configurer les exigences de validation dans le cas où la condition (spécifiée dans le cadre *If*) n'est pas remplie.

© 2016-2022 Altova GmbH

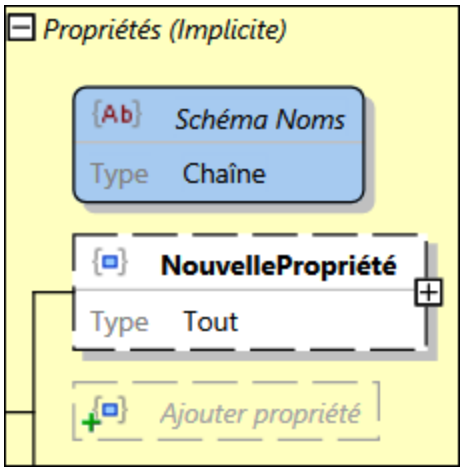
13.6.15 Configurer l'affichage Design

Lorsque la fenêtre principale se trouve dans le mode Affichage Design, vous pouvez accéder au dialogue Configuration d'affichage (*capture d'écran ci-dessous*) par le biais de la commande de menu **Conception de schéma | Configurer Mode**. Ici, vous pouvez configurer l'apparence du Mode Design.



Vous pouvez configurer les aspects suivants du Mode Configurer :

- *Largeurs :* deux coulisses déterminent, respectivement, les largeurs minimum et maximum des fenêtres dans le Mode Design. Ensemble, elles déterminent la largeur autorisée des fenêtres.
- *Distances parent/enfant :* Définit la distance horizontale entre chaque niveau dans la hiérarchie.
- *Distances enfant/enfant :* Définit la distance verticale entre les fenêtres.
- *Largeur des descriptions :* Définit la largeur des lignes de description. Si la longueur de texte dépasse cette largeur, le texte effectuera un retour à la ligne suivante.
- *Affichage des détails :* Les détails des définitions peuvent être activés pour s'afficher ou non dans les fenêtres de définition en cochant ou décochant cette option. Il existe une icône de barre d'outils correspondante.
- *Affichage des espaces réservés :* Les espaces réservés sont des items qui n'ont pas encore été définis ; ils représentent des items potentiels. Cette option définit si l'affichage des espaces réservés est activé ou pas. Il existe une icône de barre d'outils correspondante. Par exemple, l'item *Ajouter propriété* dans la capture d'écran est un espace réservé.



Note : La commande de menu **Configurer Mode** est uniquement activée dans le mode Affichage, qui montre la définition détaillée d'un objet. Elle n'est pas disponible dans Grille d'aperçu des définitions.

13.6.16 Générer une documentation de schéma JSON

Si un schéma JSON est le document actif, vous pouvez générer une documentation en cliquant sur la commande **Conception de Schéma | Générer une Documentation**. Vous pouvez générer la documentation en format HTML, MS Word, ou RTF et spécifier les composants que vous souhaitez inclure. Les composants JSON liés sont reliés par hyperliens dans la documentation générée, permettant une navigation simple.

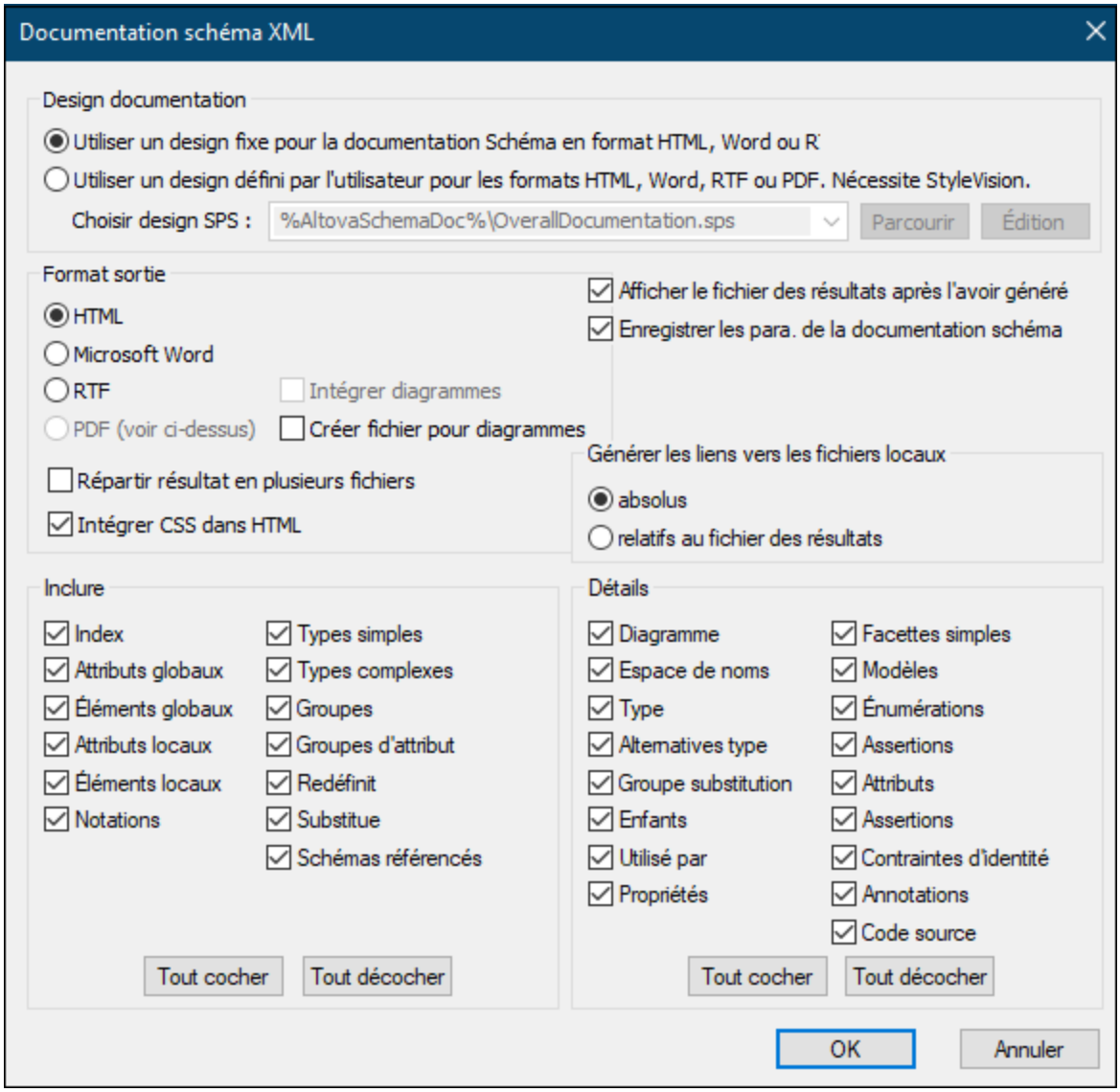
Note : Afin de générer une documentation en format MS Word, vous devez avoir installé MS Word (version 2000 ou plus).

Étapes pour générer une documentation de schéma JSON

Afin de générer une documentation pour un fichier de schéma JSON, procéder comme suit :

1. Faire du schéma JSON le document actif.
2. Passer au Mode Schéma.
3. Sélectionner la commande de menu **Conception de Schéma | Générer une Documentation**. Cela permet d'ouvrir le dialogue Documentation de Schéma JSON (*capture d'écran ci-dessous*).
4. Sélectionner le type de sortie que vous souhaitez générer, HTML, MS Word, ou RTF.
5. Sélectionner les composants spécifiques et les détails que vous souhaitez inclure à la documentation, et définir les autres options (*voir options de Documentation de Schéma JSON ci-dessous*).

© 2016-2022 Altova GmbH



6. Cliquer sur **OK** et saisir le nom du fichier de documentation de schéma JSON dans le dialogue Enregistrer sous qui apparaît.

Options de documentation de schéma JSON

Vous pouvez choisir parmi les options de documentation suivantes :

- Le modèle de design peut être un design XMLSpy intégré (fixe), ou bien il peut être un design défini par l'utilisateur qui est enregistré dans un fichier SPS. Pour une description de l'utilisation d'un design défini par l'utilisateur, voir la section Design défini par l'utilisateur.
- Le format requis est spécifié dans le volet de Format de sortie : HTML, Microsoft Word, ou RTF. La documentation peut être générée soit en un fichier unique, soit être partagé en plusieurs fichiers. Si vous générez plusieurs fichiers, chaque fichier correspond à un composant. Les composants inclus dans la sortie sont spécifiés à l'aide des cases à cocher dans le volet *Inclure*.

- L'option *Intégrer les diagrammes* est activée pour les options de sortie MS Word et RTF. Lorsque cette option est cochée, les diagrammes sont intégrés dans le fichier de résultat, dans un format PNG ou EMF. Sinon, les diagrammes sont créés en tant que fichiers PNG ou EMF, qui sont affichés dans le fichier de résultat par le biais des liens d'objet. Si la sortie est en HTML, tous les diagrammes sont créés en tant que fichiers PNG externes au document.
- Dans le volet *Inclure*, vous sélectionnez les items que vous souhaitez inclure dans la documentation. L'option *Aperçu* recense tous les composants organisés par type de composant, en haut du fichier. Si *Définitions de Schéma* n'est pas sélectionné, tous les composants enfants sont désactivés (c'est à dire, tout sauf *Schémas externes*).
- Le panneau *Détails* contient les détails qui peuvent être inclus pour chaque composant. Si If *Schema Definitions Définitions de Schéma* n'est pas sélectionné, tous les détails sont désactivés. Sélectionner les détails que vous souhaitez inclure dans la documentation.
- L'option *Afficher le fichier de résultat* est activée pour les trois options de sortie. Si cette option est cochée, les fichiers de résultat sont affichés dans le Mode Navigateur (sortie HTML), MS Word (sortie MS Word), et l'application par défaut pour les fichiers `.rtf` (sortie RTF).

© 2016-2022 Altova GmbH

13.7 Valider les documents JSON

XMLSpy contient un moteur de validation JSON qui peut être invoqué pour les tâches suivantes :

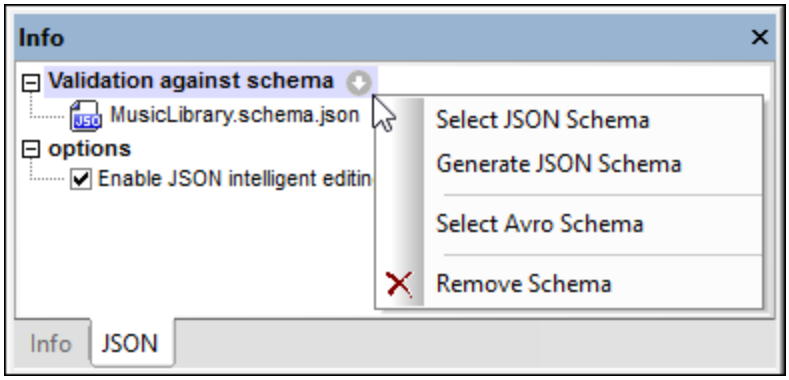
- *Si un schéma JSON est le document actif :* Valide le schéma JSON par rapport à la spécification de schéma JSON appropriée (pour laquelle aucune attribution de schéma supplémentaire n'est nécessaire) ; la version de schéma est indiquée par le mot-clé `$schema ;` la validation peut être effectuée dans un des trois modes (Texte, Grille, et Schéma JSON).
- *Si une instance JSON est le document actif :* Valide l'instance JSON par rapport à un schéma JSON. Le schéma est attribué à l'instance JSON comme décrit ci-dessous. La validation d'instance JSON peut être effectuée dans le Mode Texte et le Mode Grille.
- *Si une instance JSON5 est le document actif :* Valide l'instance JSON par rapport à un schéma JSON. Le schéma est attribué à l'instance JSON5 comme décrit ci-dessous. La validation d'instance JSON5 peut être effectuée dans le Mode Texte et le Mode Grille.

Validation Avro (uniquement Enterprise Edition)
Les données Avro et les documents de schéma Avro, comme des documents JSON, peuvent être validés dans le Mode Texte et le Mode Grille :

- *Si une instance de données Avro dans le format JSON est le document actif :* valide l'instance Avro par rapport à un schéma Avro. Le schéma est attribué à l'instance tel que décrit ci-dessous.
- *Si un schéma Avro est le document actif :* Valide le schéma Avro par rapport à la spécification de schéma Avro (aucune attribution de schéma n'est nécessaire) ; la validation peut être effectuée dans le Mode Texte ou le Mode Grille.

Attribuer un schéma JSON ou Avro à une instance JSON

Afin de valider une instance JSON par rapport à un schéma JSON ou un schéma Avro, le schéma doit être attribué au document d'instance actif. L'attribution est saisie dans la fenêtre Info (*capture d'écran ci-dessous*) du document d'instance actif JSON, ou par le biais du dialogue Propriétés de Projet (l'option *Valider avec*). Veuillez noter que les documents d'instance JSON5 sont validés par rapport aux schémas JSON.



Dans l'onglet JSON de la fenêtre Info, cliquer sur l'icône située à côté de *Validation par rapport au schéma*, et, dans le menu qui apparaît, cliquer sur **Sélectionner Schéma JSON** ou **Sélectionner Schéma Avro** (*voir capture d'écran ci-dessus*). Parcourir le schéma, et cliquer sur **OK**. Le schéma sera attribué au document d'instance JSON actif et le nom de fichier de schéma sera saisi dans la fenêtre Info. Si le document d'instance JSON est vide, l'attribution d'un schéma JSON ou Avro à l'instance remplira automatiquement l'instance JSON avec des données échantillon sur la base du schéma.

Pour supprimer l'attribution, sélectionner la commande **Supprimer Schéma** depuis le même menu (*voir capture d'écran ci-dessus*).

Pour plus d'informations concernant la génération de schéma JSON depuis l'instance JSON, voir la section Générer un schéma JSON depuis une Instance JSON.

Valider des documents d'instance et de schéma

Sélectionner la commande **XML | Valider XML (F8)** ou cliquer sur l'icône **Valider (F8)**  dans la barre d'outils pour valider le document JSON actif (instance ou schéma) ou un schéma Avro. Si un document d'instance est en cours de validation, un document de schéma doit être attribué à l'instance (*voir ci-dessus*). Les résultats de validation sont affichés dans la fenêtre Messages.
Les erreurs sont aussi marquées dans la marge de numérotation de la ligne. Si un smart fix est disponible pour une erreur, une ampoule s'affichera sur la ligne qui génère l'erreur. Lorsque vous placez la souris sur l'icône, une fenêtre popup apparaît qui liste les smart fixes disponibles. Choisir une réparation pour l'appliquer immédiatement.

Note : Les indicateurs et smart fixes d'erreur de validation décrits ci-dessus sont uniquement réinitialisés lorsque la commande **XML | Valider (F8)** est exécutée ; ils ne sont pas mis à jour en arrière-plan. Donc, après avoir corrigé une erreur, vous devez exécuter la commande **Valider (F8)** à nouveau pour vous assurer que l'erreur a bien été réparée.

Pour vous rendre au document de schéma document depuis le document d'instance, double-cliquer sur le schéma dans la fenêtre d'Info (*voir capture d'écran ci-dessus*), ou sélectionner la commande **DTD/Schéma | Aller au Schéma**. Pour vous rendre directement à la définition de schéma d'un mot-clé ou un objet JSON, sélectionner le mot-clé ou l'objet dans le document d'instance et sélectionner **DTD/Schéma | Aller à la définition**.

Vous pouvez aussi valider un dossier de projet contenant les fichiers JSON en utilisant la commande **Valider**.

Valider sur modification

Le mode *Valider sur Édition* est activé par défaut. Lorsqu'il est activé, des vérifications de bonne formation et de validation sont effectuées pendant que vous modifiez un document dans le Mode JSON Grid. Pour que la validation d'un document JSON soit effectuée (en plus des vérifications de bonne forme), un Schéma JSON doit être attribué au document JSON. Les erreurs sont indiquées en affichant du texte erroné en rouge et en marquant l'emplacement avec un point d'exclamation rouge. Voir Valider des documents JSON pour plus d'informations.

Le mode *Valider sur Édition* peut être activé ou désactivé soit (i) via la commande de menu **XML | Valider sur Édition**, (ii) via la touche de la barre d'outils **Valider sur Édition**, ou (iii) via l'option *Sur Édition* des Paramètres de validation du dialogue Options.

© 2016-2022 Altova GmbH

13.8 Inserting JSON Fragments

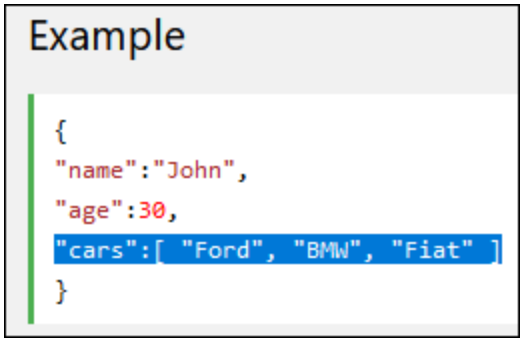
Vous pouvez insérer des fragments JSON d'autres applications et pages web. Ces fragments peuvent être insérés des deux manières suivantes :

- En utilisant glisser-déposer dans le Mode Texte ou le Mode Grille. Si vous glissez-déposez dans le Mode Grille, l'information intelligente disponible dans les superpositions pour glisser peut vous aider à décider où vous souhaitez déposer le fragment.
- En utilisant copier-coller dans le Mode Texte ou le Mode Grille.

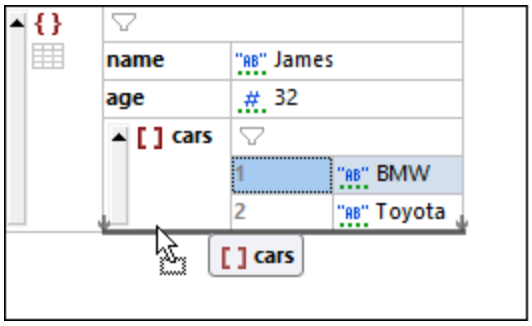
Exemple

L'exemple suivant montre comment un fragment peut être ajouté rapidement et dans le bon emplacement d'un document JSON.

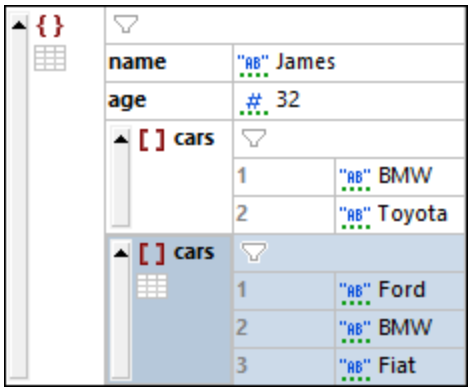
1. Le fragment en surbrillance jaune ci-dessous (du tutoriel JSON sur w3schools.com) est sélectionné. C'est un array nommé **cars**.



2. La capture d'écran ci-dessous montre le Mode Grille d'un document JSON qui contient un array **cars** similaire. Lorsque le fragment de la page web est glissé dans l'array déjà existant **cars**, une superposition pour glisser apparait contenant l'information comme quoi le fragment JSON glissé sera déposé en dessous de l'array existant en tant que nouvel array nommé **cars**.



3. Lorsque le fragment est déposé, il est placé exactement là où il doit être (*voir la capture d'écran ci-dessous*).



© 2016-2022 Altova GmbH

13.9 Transformations JSON avec XSLT/XQuery

Des mappages, des arrays et des objets JSON peuvent être ciblés avec des expressions **XPath/XQuery 3.1**. En résultat, les documents JSON peuvent être transformés avec des documents **XSLT 3.0, XQuery 3.1 et XQuery Update 3.0** en utilisant les moteurs intégrés de XMLSpy.

Les fonctionnalités suivantes sont disponibles :

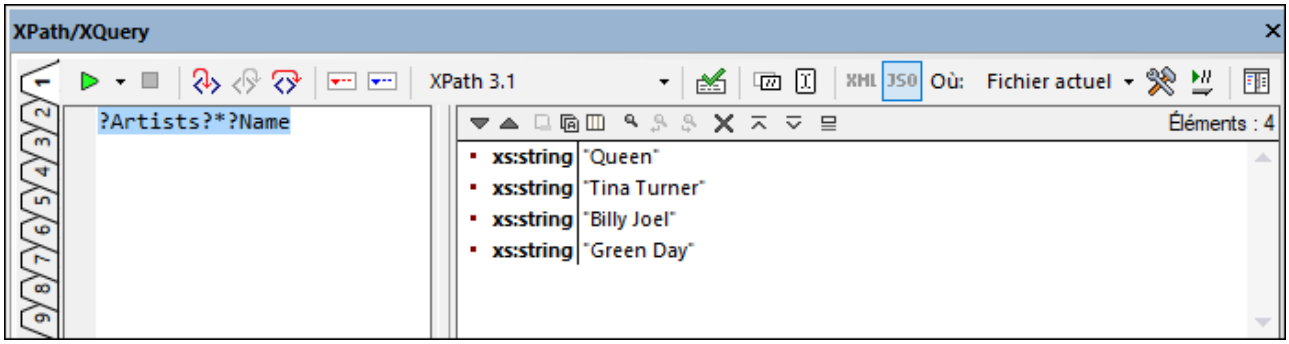
- Un document JSON active peut être interrogé avec des expressions XPath/XQuery 3.1 depuis la fenêtre de sortie XPath/XQuery
- Un document JSON actif peut être transformé avec un fichier XSLT ou XQuery sélectionné par l'utilisateur
- Un document XSLT ou XQuery actif peut être exécuté sur un fichier de source JSON sélectionné par l'utilisateur

Ces fonctions sont décrites ci-dessous plus en détail. For information about constructing XQuery expressions for JSON documents, see the section XQuery Expressions for JSON.

Note : Vous pouvez essayer les transformations JSON en utilisant les fichiers JSON, XSLT et XQuery dans le dossier *Exemples JSON* du projet **Examples** situé dans votre dossier d'application : `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\Examples.spp`.

Interroger un document JSON par le biais de la fenêtre XPath/XQuery

Les documents JSON peuvent être interrogés en saisissant une expression de requête XPath/XQuery 3.1 dans la fenêtre de sortie XPath/XQuery (*voir la capture d'écran ci-dessous*). Choisir soit l'icône **XPath 3.1** ou l'icône **XQuery 3.1**, et s'assurer que la fenêtre se trouve dans le mode d'évaluation JSON (*expliqué ci-dessous*).



L'information donnée ci-dessous se rapporte aux évaluations des documents JSON dans le mode d'évaluation JSON. (Pour un aperçu de la fenêtre XPath/XQuery et des informations détaillée concernant son utilisation, voir la section Fenêtre de sortie : XPath/XQuery.)

Mode d'évaluation JSON
Le mode d'évaluation JSON est décrit par ces points :

- La fenêtre XPath/XQuery sera soit dans le mode d'évaluation XML ou dans le mode d'évaluation JSON. Le mode actuellement **actif** est indiqué par la touche du mode actif illuminé. Voir les touches de mode d'évaluation XML/JSON dans la capture d'écran. Dans la capture d'écran, la fenêtre se trouve en mode d'évaluation JSON.

- Dans la capture d'écran ci-dessus, veuillez noter que les touches XML et JSON sont grisées, indiquant qu'elles sont **désactivées**. Lorsque les touches sont désactivées, leur statut ; qu'il soit activé ou désactivé ; ne peut pas être modifié. Inversement, si les touches sont **activées** (pas grisées), alors le mode d'évaluation de la fenêtre peut être modifié.
- L'état *activé/désactivé* des touches de mode d'évaluation XML/JSON dépend de l'étendue d'évaluation (la valeur du champ *Où* ; *voir capture d'écran ci-dessus*). Les valeurs d'étendue d'évaluation sont divisées en deux groupes pour la détermination de l'état *activé/désactivé* : (i) Fichier unique (*Fichier actuel*), et (ii) Fichiers multiples (*Fichiers ouverts, Projet, Dossier*).
- Si, pour l'étendue d'évaluation, un fichier unique (*Fichier actuel*) est sélectionné (*comme dans la capture d'écran ci-dessus*), alors le mode de la fenêtre (JSON ou XML) est déterminé sur la base de l'extension du fichier. Soit le fichier est conforme à JSON, et dans ce cas, le mode d'évaluation JSON est **activé** ; ou le fichier n'est pas conforme à JSON, et le mode d'évaluation XML est activé. Puisque le type de fichier du fichier unique est connu, le mode d'évaluation approprié est activé et les deux touches sont **désactivés** de manière à ce que le mode ne puisse pas être modifié.
- Si une option fiches multiple (*Fichiers ouverts, Projet, Dossier*) est sélectionnée, les deux touches de mode d'évaluation sont **activés**, et l'utilisateur peut sélectionner le mode à activer (JSON ou XML). Le mode d'évaluation par défaut pour une étendue pour fiches multiples est XML.
- Dans le mode d'évaluation XML, les fichiers conformes à XML seront traités et les fichiers JSON seront sautés.
- Dans le mode d'évaluation JSON, les fichiers conformes à JSON seront traités et les fichiers XML seront sautés.
- Les expressions JSON peuvent aussi être interrogées dans le Mode Débogage.

Transformer un document JSON avec XSLT/XQuery

Pour transformer un document JSON actif avec un document XSLT 3.0, XQuery 3.1 ou XQuery Update 3.0, procéder comme suit :

- *Transformation XSLT 3.0 :* cliquer sur **Transformation XSL/XQuery | XSL**, chercher le fichier XSLT 3.0, et cliquer sur **OK**.
- *Transformation XQuery 3.1 ou XQuery Update 3.0 :* cliquer sur **Exécution XSL/XQuery | XQuery/Update**, chercher le fichier XQuery 3.1 ou XQuery Update 3.0, et cliquer sur **OK**.

Les documents transformés seront générés et peuvent être consultés directement dans XMLSpy.

Note : Le Débogueur XSLT/XQuery peut être lancé depuis un document JSON, mais les points de rupture et les points de trace peuvent être définis uniquement dans le document XSLT ou XQuery.

Fournir une source JSON pour un document XSLT/XQuery

Pour exécuter un document XSLT ou XQuery actif sur un fichier source JSON, procéder comme suit :

- *Document XSLT 3.0 actif :* cliquer sur **Transformation XSL/XQuery | XSL**, chercher le fichier JSON, et cliquer sur **OK**.
- *Document XQuery 3.1 ou XQuery Update 3.0 actif :* cliquer sur **Exécution XSL/XQuery | XQuery/Update**, chercher le fichier JSON, et cliquer sur **OK**.

Les documents transformés seront générés et peuvent être consultés directement dans XMLSpy.

Note : Le Débogueur XSLT/XQuery peut être lancé depuis un document XSLT ou XQuery et un document JSON peut être attribué en tant qu'entrée pour la session de débogage. Néanmoins les points de rupture et les points de trace peuvent être définis uniquement dans le document XSLT ou XQuery.

© 2016-2022 Altova GmbH

13.10 XQuery Expressions for JSON

Expressions XQuery 3.1 pour JSON

Puisque les structures de données JSON utilisent généralement des objets et des tableaux, c'est l'opérateur de consultation XQuery 3.1 `?` qui est utilisé pour situer des nœuds dans des objets JSON (qui sont principalement des mappages depuis une perspective XQuery) et des tableaux JSON. Cette méthode pour situer un nœud diffère de la manière dont les expressions de chemin sont écrites pour situer des nœuds dans des documents XML. Dans ceux-ci, l'opérateur "barre oblique" `/` est utilisé pour connecter des étapes dans une expression de chemin (par exemple : `items/*`). Dans des expressions XQuery pour JSON, l'opérateur de barre oblique n'est pas utilisé pour situer des nœuds.

Exemples d'expressions XQuery pour JSON

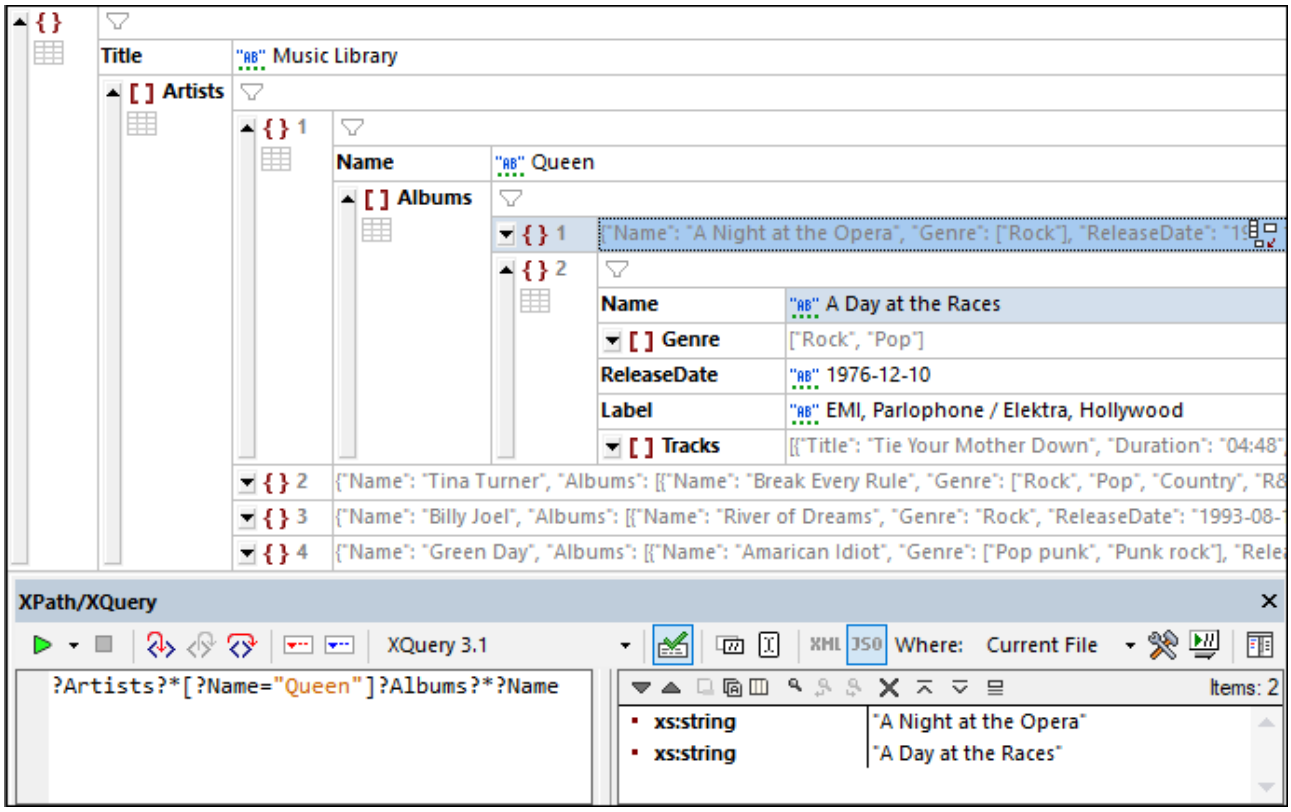
`?items?*`
Signification de l'expression : consultation du nœud enfant `items` puis consultation de tous ses nœuds enfants. Veuillez noter que `items` est censé être un nœud enfant du nœud contextuel.

`?Artists?1?Albums?2?Name`
Signification de l'expression : consultation du nœud enfant `Artists` puis consultation de son premier nœud enfant. Dans ce nœud, consulte le nœud enfant `Albums` puis consulte son second nœud enfant. Retourne maintenant le nœud `Name` de ce second nœud enfant.

`?Tracks?*[contains(?Writer, 'Brian')]`
Signification de l'expression : consultation du nœud enfant `Tracks` puis consultation de tous ses enfants. Lors de la consultation des enfants, consulte tous les enfants de nœud `Writer` de l'enfant, et ne sélecte que ceux qui contiennent le string `'Brian'`. Veuillez noter qu'il y a trois opérateurs de consultation dans cette expression. Chacun est utilisé dans une nouvelle étape, à l'endroit où un nodeset doit être consulté.

`?Artists?*[?Name="Queen"]?Albums?*?Name`
Signification de l'expression : Dans l'objet racine, consulte le nœud enfant `Artists` puis consulte tous ses enfant qui ont un nœud `Name` avec une valeur de `"Queen"`. Dans ces nœuds, consulte tous les nœuds de l'enfant `Albums`, puis leurs enfants. Dans ces enfants, consulte (et retourne) les nœuds respectifs `Name`. Dans la capture d'écran ci-dessous, cette expression est connue dans la fenêtre XPath/XQuery avec la représentation Mode JSON Grid du document cible JSON.

© 2016-2022 Altova GmbH


Title
Music Library
Artists
Name
Queen
Albums
"Name": "A Night at the Opera", "Genre": ["Rock"], "ReleaseDate":
Name
A Day at the Races
Genre
["Rock", "Pop"]
ReleaseDate
1976-12-10
Label
EMI, Parlophone / Elektra, Hollywood
Tracks
{"Name": "Tina Turner", "Albums": [{"Name": "Break Every Rule", "Genre": ["Rock", "Pop", "Country",
{"Name": "Billy Joel", "Albums": [{"Name": "River of Dreams", "Genre": "Rock", "ReleaseDate": "1993-08-
{"Name": "Green Day", "Albums": [{"Name": "Amarican Idiot", "Genre": ["Pop punk", "Punk rock"],
XPath/XQuery
XQuery 3.1
Where: Current File
?Artists?*[?Name="Queen"]?Albums?*?Name
Items: 2
xs:string "A Night at the Opera"
xs:string "A Day at the Races"

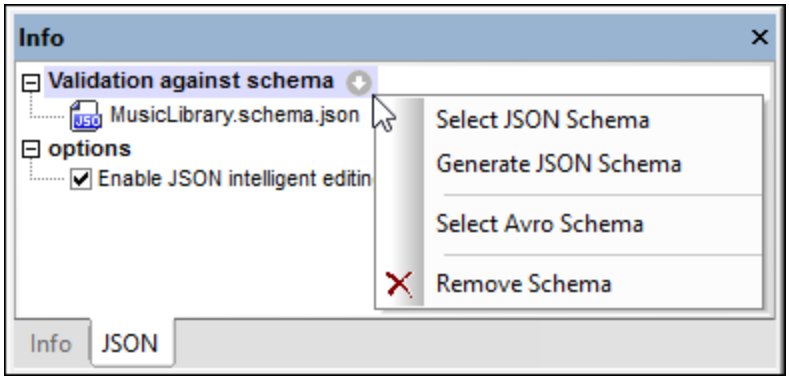
13.11 Générer un schéma JSON depuis une instance JSON

XMLSpy peut générer un schéma JSON depuis un document d'instance JSON (y compris des instances JSON5). Cette fonction est très utile du fait qu'elle vous fournit rapidement un schéma basé sur une instance JSON existante et vous épargne la création manuelle d'un schéma à partir de zéro. Vous pouvez ensuite modifier ou étendre le schéma généré selon vos besoins.

Générer le schéma

Pour générer un schéma JSON depuis une instance JSON, procédez comme suit :

1. Faire du document d'instance JSON le document actif.
2. Dans l'onglet JSON de la fenêtre Info, cliquez sur l'icône de la flèche située à côté de *Validation par rapport au schéma*, et dans le menu qui apparaît, cliquez **Générer Schéma JSON** (*voir la capture d'écran ci-dessous ; prise en charge Avro est uniquement disponible dans l'édition Enterprise*). En alternative, sélectionnez la commande de menu **DTD/Schéma | Générer DTD/Schéma**.

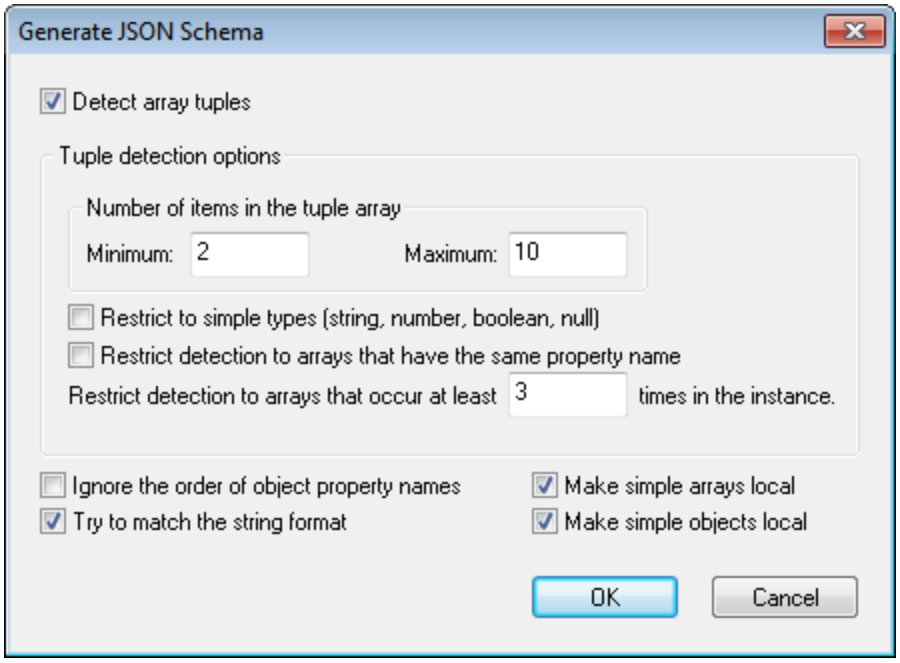


3. Dans le dialogue Générer schéma JSON qui apparaît à présent (*voir la section ci-dessous*), modifiez les paramètres comme vous le souhaitez (*voir ci-dessous pour plus de détails*). Cliquez sur **OK** une fois que vous avez terminé.
4. Vous serez invité à fournir un chemin et un nom de fichier pour le schéma JSON généré. En cliquant sur **Enregistrer**, le schéma est généré et devient le document actif. Dans le document d'instance, le fichier de schéma généré sera attribué en tant que le schéma à utiliser pour la validation ; toutes les attributions précédentes seront écrasées. Pour changer l'attribution, utilisez la commande **Sélectionner Schéma JSON** du menu contextuel (*voir capture d'écran ci-dessus*).

Paramètres pour la génération du schéma

Vous pouvez spécifiez des options pour la génération de schéma dans le dialogue Générer schéma JSON (*voir la capture d'écran ci-dessous*). Voir la section précédente pour plus d'informations concernant l'accès à ce dialogue.

© 2016-2022 Altova GmbH



Détecter des tuples d'array

Un tuple d'array est la séquence des items dans un array. Par exemple, l'array suivant présente un tuple de trois items : `[ 1, 2, "abc" ]`. Pour la validation des arrays, le schéma peut spécifier si l'ordre et le type de données des items (tuples) doivent être considérés ou pas. Si l'option *Détecter des tuples d'array* est cochée *(voir la capture d'écran ci-dessus)*, alors l'ordre et le type de données des items seront détectés. Sur la base de ce qui est détecté, une définition correspondante sera créée dans le schéma. Les options pour ce paramètre sont les suivantes :

- *Nombre d'items de tuple :* Un nombre minimum et maximum d'items de tuple peut être spécifié. Si un tuple dans l'instance a un décompte d'item se trouvant dans cette plage, cet array sera détecté et défini.
- *Types simples uniquement :* uniquement des tuples qui ont des items de type simple (les types atomiques `string`, `number`, `integer`, `boolean`, et `null`) seront considérés pour une détection.
- *Arrays nommés identiquement :* seuls les arrays qui sont définis en tant que valeurs des propriétés qui ont le même nom sont considérés pour une détection. Par exemple, dans le fragment de données JSON suivant, les arrays marqués par des crochets rouges sont tous des valeurs de propriétés nommées `a1` (colorées en bleu) : `{"object1": [ { "a1": [ 1, 2, "abc" ] }, { "a1": [ 3, 4, "def" ] }, { "a1": [ 5, 6, "ghi" ] } ] }`.
- *Nombre minimum d'arrays :* un nombre minimum d'arrays peut être spécifié pour activer la détection d'arrays.

Autres paramètres

- *Ignorer l'ordre des noms de propriété d'objet :* si non sélectionné, l'ordre des propriétés d'un objet est coché et recréé aussi précisément que possible. Sinon, l'ordre n'est pas coché.
- *Essayer de faire correspondre le format de chaîne :* le schéma peut spécifier que les types de données de chaîne doivent avoir un format spécifique. Si cette option est sélectionnée, XMLSpy tentera de détecter le format de chaîne et d'ajouter une définition de format pour les chaînes, lorsque cela est possible.

- *Faire des arrays simples localement :* un array simple est un array dans lequel tous les items sont du même type de données simple. Une fois sélectionnés, tous les arrays simples seront définis localement dans le schéma, au lieu d'utiliser des définitions globales qui sont référencées localement.
- *Faire des objets simples localement :* un objet simple est un objet dans lequel toutes les valeurs de propriété sont du même type de données simple. Une fois sélectionnés, tous les objets simples seront définis localement dans le schéma, au lieu d'utiliser des définitions globales qui sont référencées localement.

Note : une fois que le schéma JSON a été généré, vous pouvez faire des définitions locales d'objets individuels et d'arrays globaux et vice-versa. Pour plus d'informations, voir la section Définitions globales et locales.

© 2016-2022 Altova GmbH

13.12 Générer une instance JSON depuis un schéma JSON

Vous pouvez générer une instance JSON depuis un schéma JSON lorsque le schéma JSON est le fichier actif dans le Mode Texte, le Mode Grille ou le Mode Schéma. Cliquer sur **DTD/Schéma | Générer un fichier échantillon XML/JSON**.

13.13 Convertir entre JSON et XML

Les options de conversion suivantes sont disponibles :

- Convertir une instance XML en JSON : lorsqu'un document d'instance XML est le document actif, un document d'instance JSON est généré depuis l'instance XML en cliquant sur **Convertir | Convertir une instance XML de/vers JSON**.
- Convertir une instance JSON en XML : lorsqu'un document d'instance JSON est le document actif, un document d'instance XML est généré depuis l'instance JSON en cliquant sur **Convertir | Convertir une instance XML de/vers JSON**.
- Convertir un schéma XML en un schéma JSON : lorsqu'un document de schéma XML est le document actif, un document de schéma JSON est généré depuis le schéma XML en cliquant sur **Convertir | Convertir un schéma XML de/vers un schéma JSON**.
- Convertir un schéma JSON en un schéma XML : lorsqu'un document de schéma JSON est le document actif, un document de schéma XML est généré depuis le schéma JSON en cliquant sur **Convertir | Convertir un schéma XML de/vers un schéma JSON**.

Toutes ces conversions sont activées dans le Mode Texte et dans le Mode Grille. Cliquer sur les liens ci-dessus pour lire une description des fonctions respectives.

© 2016-2022 Altova GmbH

14 Avro, Schéma Avro

Apache Avro™ est un système pour la sérialisation des données dans un format binaire compact. Une structure de données Avro est définie dans un schéma Avro, qui est écrit en format JSON. Dans les scénarios de déploiement réels, un document Avro est généralement sérialisé en tant qu'un fichier binaire qui contient non seulement les structures de données Avro ais aussi le schéma Avro utilisé pour définir ces structures. Le binaire Avro contient donc les données et la définition de structure des données (le schéma Avro). Néanmoins, les données Avro peuvent aussi être sérialisées en tant que JSON ; dans ce cas, les données Avro (dans un fichier JSON) référencent un schéma Avro externe.

XMLSpy prend en charge Apache Avro™ 1.8.1.

Dans XMLSpy, la prise en charge Avro suivante est disponible :

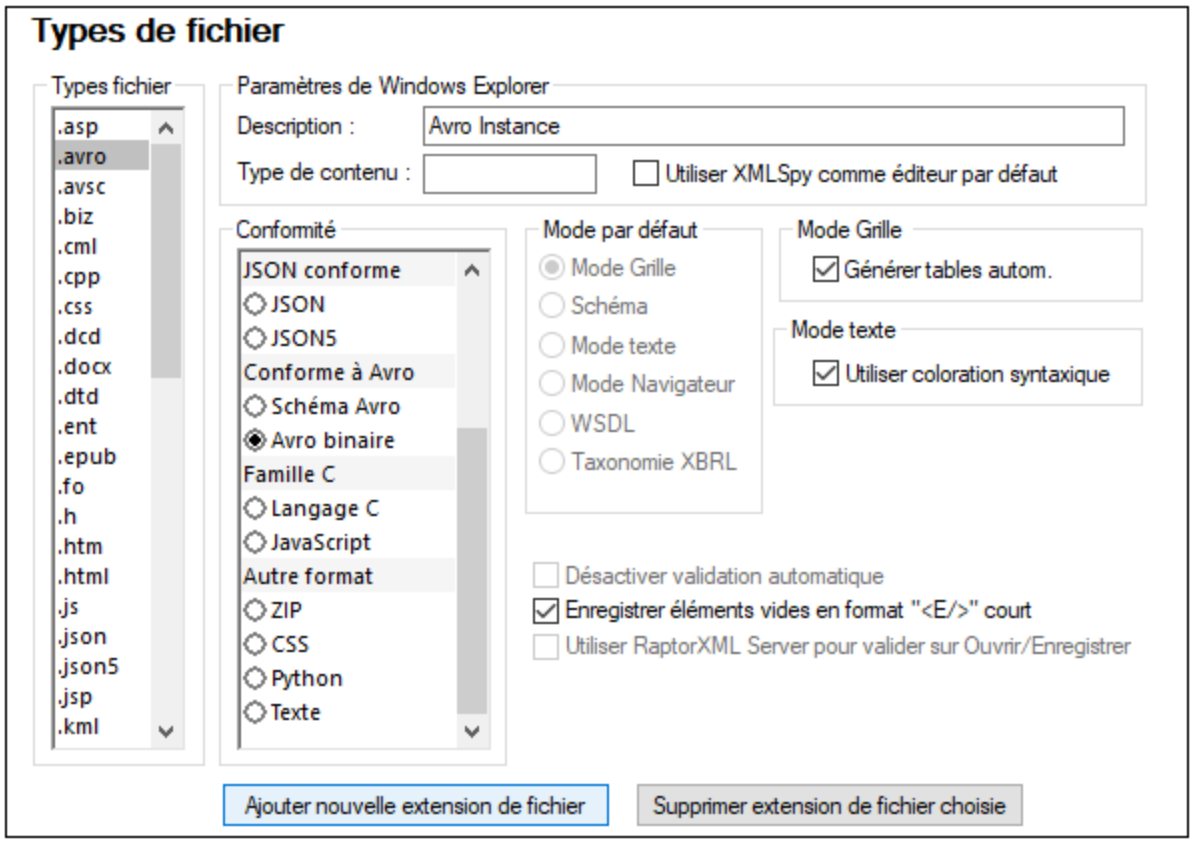
- Vous pouvez éditer des données Avro (en tant que documents JSON **.json**) dans le Mode Texte et le Mode Grille ; les deux modes proposent des fonctions d'édition intelligentes. Le document de données peut être attribué à un schéma Avro et être validé par rapport à ce schéma.
- Vous pouvez éditer les schémas Avro (en tant que documents de Schéma Avro **.avsc**) dans le Mode Texte et le Mode Grille. Les schémas Avro peuvent être validés par rapport à la spécification de schéma Avro, et les modes fournissent des fonctions d'édition intelligentes.
- Vous pouvez consulter les instances binaires Avro (fichiers **.avro**) dans le Mode Avro, qui affiche les blocs de données Avro dans une grille tabulaire.

Les éditions RaptorXML propose la prise en charge Avro suivante :

- Validation de données Avro (sérialisé JSON ; fichier `.json`) (par rapport à un schéma Avro)
- Validation de données Avro (sérialisé binaire ; fichier `.avro`)
- Validation de Schéma Avro (généralement fichier `.avsc`) (par rapport aux spécifications de schéma Avro)
- Extraction du schéma Avro du binaire Avro

Ouvrir les document Avro existants et création de nouveaux documents

Dans la section **Options | Types de fichier** (*capture d'écran ci-dessous*), vous pouvez définir le mode par défaut dans lequel les différent types de documents Avro (format de données JSON, schéma Avro, binaire Avro) s'ouvrent. Vous pouvez passer entre les modes disponibles à tout moment.



Type de document	Extension de fichier	Conformité	Modes disponibles
Données Avro dans le format JSON	`.json`	Conforme à JSON | JSON	Mode Texte, Mode Grille
Schéma Avro	`.avsc`	Conforme à JSON | Schéma Avro	Mode Texte, Mode Grille
Données Avro dans un fichier binaire	`.avro`	Conforme à Avro | Binaire Avro	Mode Avro

Veuillez noter les points suivants :

- Les documents existants et les nouveaux documents d'un type sélectionné s'ouvriront dans le mode par défaut que vous aurez choisi dans la section *Types de fichier*.
- Les binaires Avro peuvent uniquement être consultés dans le Mode Avro, qui est un mode en lecture seule. Lorsqu'un type de fichier est défini comme étant conforme à Avro, le seul mode disponible est le Mode Avro.
- Si vous souhaitez que XMLSpy lise des fichiers d'une certaine extension de fichier en tant qu'un des types de document Avro recensé ci-dessus, ajoutez cette nouvelle extension de fichier et attribuez-y la conformité pertinente.

© 2016-2022 Altova GmbH

- Pour créer un nouveau document, cliquer sur **Fichier | Nouveau**, et choisir le type de document que vous souhaitez. Les binaires Avro, étant des binaires ne peuvent bien sûr pas être créés de cette manière ; ils peuvent uniquement être lus dans le Mode Avro.

14.1 Schéma Avro

Un schéma Avro spécifie la structure d'un bloc de données Avro. Il spécifie quels champs de données sont attendus et comment les valeurs sont représentées. Vous trouverez des informations concernant le schéma Avro et ses spécifications ici.

Veuillez noter les points suivants concernant les schémas Avro :

- Un schéma Avro est créé dans un format JSON
- Un schéma Avro peut être : une chaîne JSON, un objet JSON ou un tableau JSON
- Un schéma Avro peut contenu quatre attributs : **`name`**, **`namespace`**, **`type`**, et **`fields`**
- Il existe huit types de données primitives : **`null`**, **`boolean`**, **`int`**, **`long`**, **`float`**, **`double`**, **`bytes`**, et **`string`**
- Il existe six types complexes : **`records`**, **`enums`**, **`arrays`**, **`maps`**, **`unions`**, et **`fixed`**
- Les types primitifs n'ont pas d'attributs; chaque type complexe a son propre jeu d'attributs

Pour plus de détails et d'informations concernant le schéma Avro, voir la spécification de schéma Avro.

Exemples

Ci-dessous, vous trouverez des exemples simples de schémas Avro, chacun contenant des fragments de données Avro correspondants dans un format JSON. Veuillez noter que le schéma définit une certaine structure. Dans certains cas, lorsque la structure définie est instanciée plusieurs fois, la sortie résultant peut ne pas être un JSON valide. Par exemple, un schéma peut définir la structure d'un objet JSON. Si l'objet JSON est instancié plusieurs fois, chaque objet (séparément) peut être valide par rapport au schéma Avro, mais le document entier ne serait pas un JSON valide, parce qu'il n'y a pas d'objet contenant. SI un JSON valide est requis, vous souhaiterez éventuellement réécrire le schéma Avro pour valider un tableau d'objets JSON. Comparer les exemples 4 et 5 ci-dessous pour consulter une illustration de ce point.

01: Schéma Avro en tant que chaîne JSON

Ce schéma est une chaîne unique, et il spécifie que le bloc de données doit contenir une valeur qui est du type de données primitif (int) Avro : **`"int"`**

Valid Avro: **`2016`**
Invalid Avro: **`"2016"`**

02: Schéma Avro en tant qu'objet JSON

Ce schéma spécifie exactement la même chose que le schéma précédent, mais il s'agit d'un objet JSON. Le bloc de données doit contenir un item qui est un valeur du type de données primitif (int) Avro:

```
{
   "type": "int"
}
```

Valid Avro: 2016
Invalid Avro: "2016"

03: Schéma Avro en tant qu'objet : tableau d'entiers

Ce schéma est un objet JSON qui spécifie un tableau d'entiers :

© 2016-2022 Altova GmbH

```
{
   "type": "array",
   "items": "int"
}
```

Valid Avro: `[2016, 2017]`
Valid Avro: `[2016]`
Valid Avro: `[2016]`
Invalid Avro: `2016, 2017`

- *04: Schéma Avro en tant qu'objet : enregistrements*

  Ce schéma est un objet JSON qui spécifie un seul enregistrement :

```
{
   "type": "record",
   "name": "ages",
   "fields" : [
   {"name": "name", "type": "string"},
   {"name": "age", "type": "int"}
   ]
}
```

  Valid Avro: `{"name":"John", "age":35}`

- *05: Schéma Avro en tant qu'objet : enregistrements multiples*

  Ce schéma est un objet JSON qui spécifie un tableau d'items d'enregistrement, chacun d'entre eux doit être un objet JSON :

```
{
   "type": "array",
   "items": {
   "type": "record",
   "name": "ages",
   "fields" : [
   {"name": "name", "type": "string"},
   {"name": "age", "type": "int"}
   ]
    }
}
```

  Valid Avro: `[{"name":"Mary", "age":34}, {"name":"John", "age":35}]`

Types de fichier de schéma Avro

Si vous souhaitez utiliser les fonctions de XMLSpy pour l'édition et la validation liée à Avro, XMLSpy doit être capable de reconnaître un fichier en tant qu'un schéma Avro. Un fichier est reconnu en tant que schéma Avro si l'extension du fichier est définie comme telle dans le dialogue Options de XMLSpy (**Outils | Options | Types de fichier**). Les paramètres par défaut de XMLSpy définissent une extension de fichier, l'extension `.avsc`, comme étant celle d'un fichier de schéma Avro. Si vous souhaitez créer d'autres extensions de fichier qui spécifient les documents de schéma Avro, ajoutez ces extensions de fichier en tant qu'extensions de schéma Avro à la liste dans le dialogue Options.

Créer et éditer des schémas Avro

Dans XMLSpy, vous pouvez créer un nouveau fichier en tant qu'un schéma Avro en spécifiant une extension de fichier de schéma Avro en tant que son type de fichier. XMLSpy fournit une aide à l'édition intelligente pendant la saisie. Cette aide comprend des suggestions de mots-clés sensibles au contexte, la saisie automatique de paires de crochets, de parenthèses et de guillemets, une coloration syntaxique et une saisie automatique des mots-clés. En outre, il existe trois assistants à la saisie : Propriétés JSON, Valeurs JSON et Entités JSON. Les entrées qui sont valables sont sensibles au contexte. Double-cliquer dans une entrée pour l'insérer à l'emplacement actuel du curseur. Vous pourrez ensuite valider le fichier par rapport à la spécification de schéma Avro avec la commande de menu **Valider | Valider XML (F8)**.

© 2016-2022 Altova GmbH

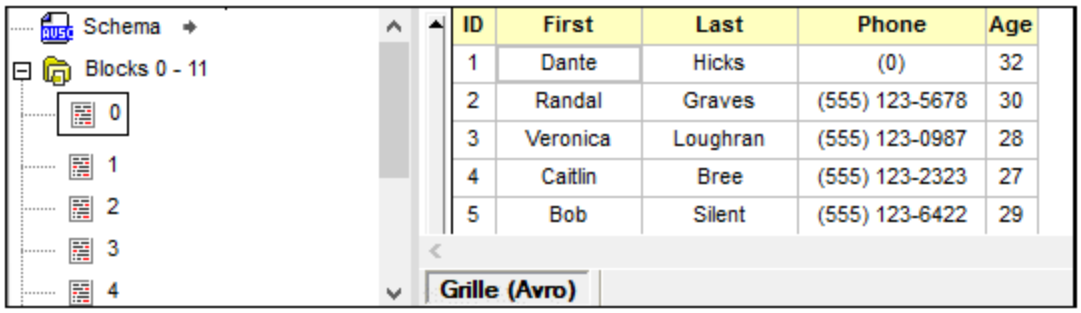
14.2 Données Avro en format JSON

Les données Avro peuvent être sérialisées dans un format binaire ou un format JSON. Les points suivants expliquent la prise en charge de XMLSpy pour ce format Avro.

- Les données Avro dans le format JSON sont généralement enregistrées en tant que fichier `.json`. Vous pouvez spécifier que XMLSpy devrait reconnaître des extensions de fichier supplémentaires comme conformes à JSON.
- Les fichiers Avro JSON peuvent être ouverts dans le Mode Texte et le Mode Grille et édités dans ces modes.
- Un fichier de schéma Avro peut être attribué au fichier Avro JSON, et le fichier de données peut ensuite être validé par rapport au schéma Avro.
- Les fonctions d'édition intelligentes pour les documents JSON sont disponibles aussi bien dans le Mode Texte que dans le Mode Grille. De plus, si un schéma Avro est attribué à un document de données Avro dans un format JSON, alors le saisie automatique des mots-clés définis par les schémas est disponible dans l'instance Avro.

14.3 Mode Avro : un Mode Grille des binaires Avro

Les données Avro peuvent être sérialisées dans un format binaire ou un format JSON. Le format binaire contient aussi bien les structures de données Avro que leur schéma et est généralement généré par le biais de procédures de traitement de données automatisées. Un binaire Avro peut être ouvert dans le Mode Avro qui est un mode grille qui affiche les structures de données Avro dans un format agréable à lire (*voir capture d'écran ci-dessous*). Le Mode Avro sert donc de visualisation binaire Avro conviviale.



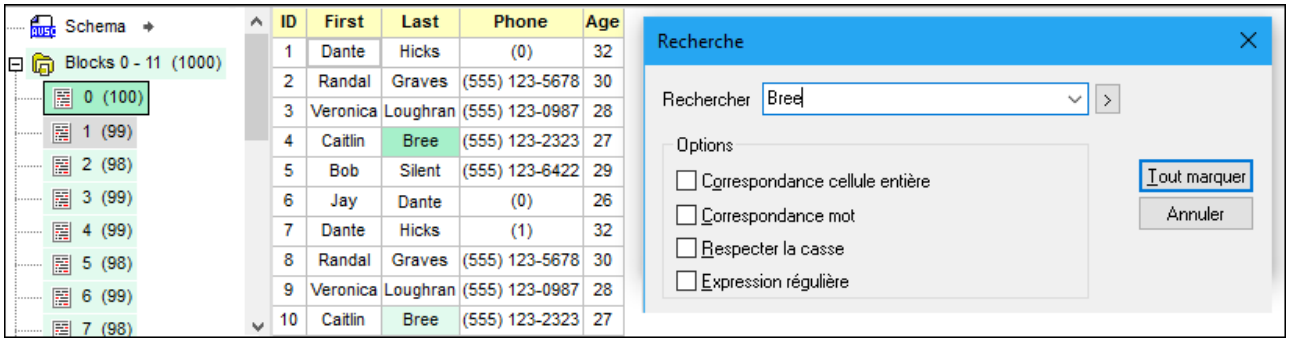
Veuillez noter les points suivants :

- Le binaire Avro doit être reconnaissable en tant que tel pour XMLSpy. Cela peut se régler dans la section **Options | Types de fichier** en configurant l'extension de fichier du binaire Avro pour être conforme à Avro. Par défaut, l'extension de fichier `.avro` a été réglée pour être conforme à Avro. Vous pouvez ajouter plus de types de fichier conformes au binaire Avro. Ces fichiers seront ouverts dans le Mode Avro.
- Le Mode Avro consiste en deux volets : (i) un volet *Blocs* pour parcourir, et (ii) un volet *Données*, qui affiche la structure des données que vous avez sélectionnées dans le volet *Blocs*.
- Le volet *Blocs* organise les blocs de données en groupes de `1000`. Chaque groupe peut être agrandi/réduit. Les blocs de données sont affichés par leur numéro d'index.
- Pour consulter un bloc de données particulier, il faut le localiser dans le volet *Blocs*, et le double-cliquer.
- Le volet *Blocs* contient aussi une entrée appelée *Schéma*. Si vous cliquez sur la touche située à droite de l'entrée, le schéma Avro sera extrait du binaire Avro et sera ouvert dans un nouvel onglet Mode Texte. Vous pouvez ensuite enregistrer le schéma Avro si vous le souhaitez.

Recherches de texte

Pour chercher une string de texte, choisir la commande de menu **Édition | Rechercher (Ctrl+F)**. Dans le dialogue qui apparaît, (*voir capture d'écran ci-dessous*), saisir le terme de recherche en tant que string de texte ou expression régulière. Choisir toute option applicable (décrit ici). Cliquer sur **Marquer tout**.

© 2016-2022 Altova GmbH



- Les correspondances sont marquées dans les panneaux *Blocs* et *Données* : la correspondance actuellement sélectionnée en vert foncé, les autres en vert clair.
- Dans le panneau *Blocs*, le nombre de correspondances dans chaque bloc est affiché à côté de son entrée.
- Vous pouvez parcourir les correspondances en allant de bloc en bloc, sélectionnant un champ dans le bloc, puis en utilisant **F3** (**Édition | Trouver suivant**) et **Shift+F3** (Trouver précédent) pour vous déplacer.
- Veuillez noter que le Mode Avro est un mode en lecture seule, vous ne pourrez pas éditer de données dans le binaire Avro.

15 WSDL et SOAP

Site Web Altova : Éditeur WSDL
Cette section décrit les fonctions WSDL et SOAP de XMLSpy.

WSDL

Un document WSDL est un document XML qui décrit un service web. XMLSpy prend en charge WSDL 1.1 et WSDL 2.0. Vous pouvez créer et éditer les documents WSDL 1.1 et WSDL 2.0 dans le Mode WSDL de XMLSpy, qui fournit automatiquement l'environnement d'édition correct pour la version WSDL en cours d'édition.

Dans le Mode WSDL de XMLSpy, un document peut être "construit" à l'aide de modules de construction graphiques, permettant ainsi une création particulièrement simple. Le Mode WSDL est décrit dans la section Modes d'édition. Pour une description détaillée de la création d'un document WSDL, voir le tutoriel WSDL dans cette documentation. Vous pouvez aussi consulter et éditer des documents WSDL dans le Mode Texte et le Mode Grille. Dans ces deux modes, les documents WSDL sont édités en tant que documents XML.

Les signatures XML pour les fichiers WSDL dans le Mode WSDL peuvent être créées en tant que fichiers externes et peuvent être "enveloppées" dans le fichier WSDL. Le travail avec des signatures est décrit dans la section signatures XML.

SOAP

SOAP est une spécification de message XML et est utilisé pour transmettre des messages entre les applications. Dans XMLSpy, vous pouvez non seulement créer et éditer un document SOAP dans le Mode Texte et le Mode Grille grâce aux fonctions intelligentes d'édition de XMLSpy pour les documents XML, mais vous pouvez générer un fichier de requête SOAP depuis un fichier WSDL. Le tutoriel WSDL décrit comment générer une requête SOAP depuis un fichier WSDL. XMLSpy peut aussi recevoir et envoyer des requêtes SOAP (à l'aide de commandes dans le menu SOAP). De plus, vous pouvez déboguer les requêtes SOAP avec le Débogueur SOAP de XMLSpy, qui est décrit dans une sous-section de cette section.

© 2016-2022 Altova GmbH

15.1 Tutoriel WSDL

Ce tutoriel est divisé en deux parties :

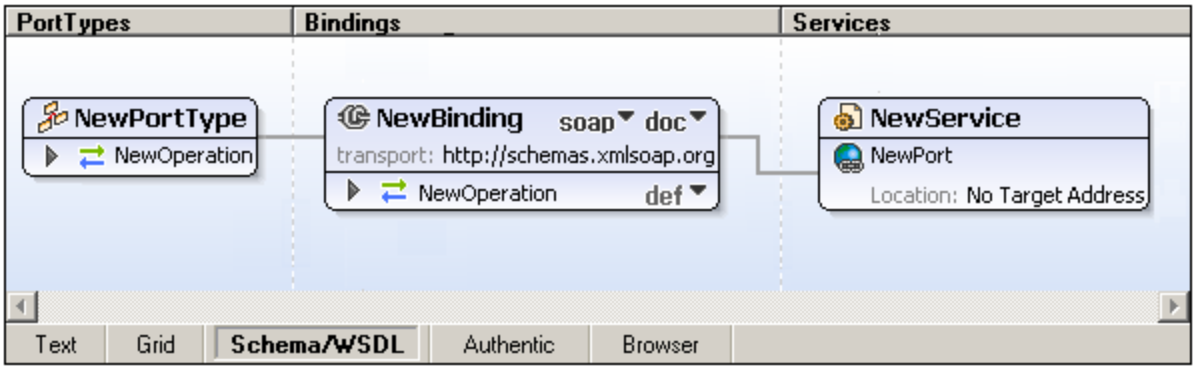
- Dans la première partie, nous vous montrons comment un document WSDL 1.1 est créé dans le Mode WSDL graphique de XMLSpy. Dans cette partie, vous allez : (i) créer un document WSDL 1.1 rudimentaire à l'aide de l'option de menu **Fichier | Nouveau** ; (ii) créer un PortType; (iii) créer une liaison ; (iv) créer un service et un port ; (v) valider le document et l'enregistrer.
- Dans la seconde partie, nous allons vous montrer comment se connecter à un service web, enregistrer le fichier WSDL localement, et envoyer une requête SOAP au service web.

Vous pouvez faire tout cela dans le Mode WSDL graphique sans avoir à utiliser le Mode Texte. Vous pouvez manipuler directement les composants WSDL par glisser/déposer et saisir des valeurs de propriété dans les assistants à la saisie du Mode WSDL.

Voir aussi : Vous trouverez plus d'informations concernant le travail avec les documents WSDL dans les sections Mode WSDL et Référence de l'utilisateur | Menu WSDL.

15.1.1 Créer un nouveau document

Pour créer un nouveau document WSDL, choisir la commande **Fichier | Nouveau**. Dans le dialogue Créer nouveau document qui apparaît, choisir `WSDL` (WSDL Web Service Description v1.1) en tant que le type de document que vous souhaitez créer et cliquer sur **OK**. Cela crée le squelette d'un nouveau document (*capture d'écran ci-dessous*) qui s'ouvre dans le Mode WSDL graphique (nommé Mode WSDL dans ce tutoriel).



Attribuer un espace de noms cible

Passer au Mode Texte. La balise de démarrage de l'élément `wsdl:definitions` ressemblera à :

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:tns="http://new.webservice.namespace"
    targetNamespace="http://new.webservice.namespace">

```

Changer l'attribut de l'espace de noms cible (`targetNamespace`) en `http://mywebservice.namespace` ou en tout ce que vous voulez (puisque le point fort de ce tutoriel est de montrer comment créer un document WSDL et qu'il ne fournit pas un service réel). Vous pouvez ensuite changer la valeur de l'espace de noms de l'attribut `tns` en `http:://mywebservice.namespace` (ou de l'espace de noms que vous avez choisi pour l'espace de noms cible).

Note : Dans le document squelette de démarrage, les éléments WSDL se trouvent dans l'espace de noms cible alors que des références aux éléments WSDL sont effectuées avec le préfixe `tns`. Par exemple, : `<wsdl:binding name="NewBinding" type="tns:NewPortType">`. Afin que le préfixe `tns` corresponde à l'espace de noms cible, sa valeur d'espace de noms doit être identique avec l'espace de noms cible.

15.1.2 Créer un PortType

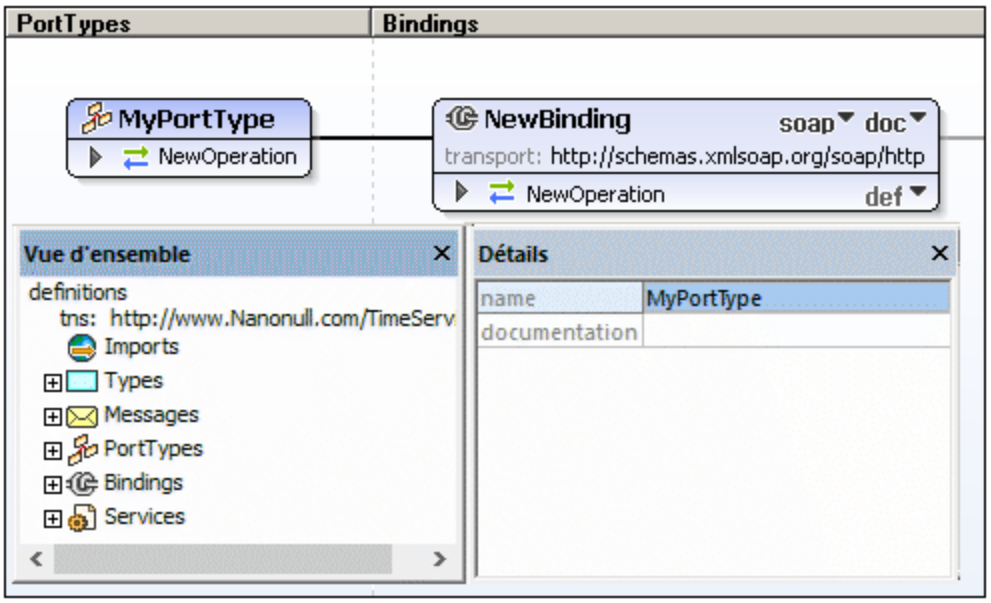
Pour créer un PortType il vous faudra :

- Nommer le PortType
- Insérer une opération
- Ajouter des messages d'entrée et de sortie
- Ajouter des paramètres aux messages

Nommer le PortType

Renommer le `NewPortType` en `MyPortType` en double-cliquant dans la barre de titre de la fenêtre `NewPortType` dans le design, puis en éditant le nom et en appuyant sur **Entrée**. Veuillez noter que le nom du PortType change aussi dans l'assistant à la saisie Aperçu et Détails (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Insérer une opération

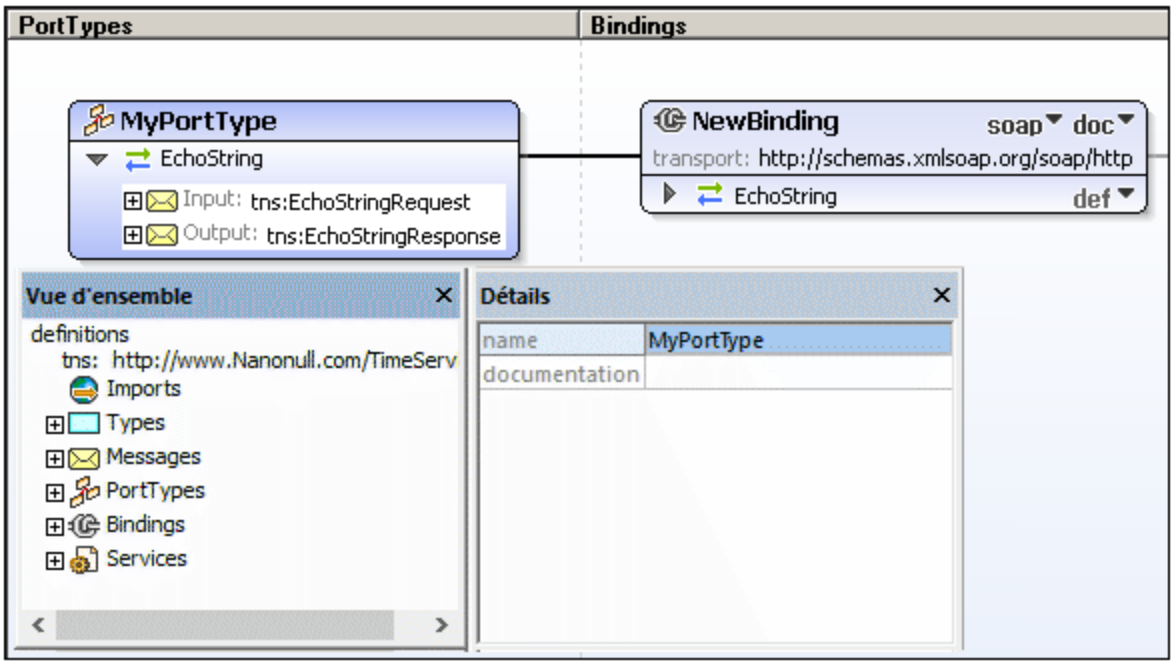
Dans le cas de `MyPortType`, une opération, `NewOperation`, est déjà présente, nous allons donc travailler avec cette opération. Commencer en renommant `NewOperation` par exemple en, `EchoString` (double-cliquer sur le nom, l'éditer et appuyer sur **Entrée**). (Pour insérer des opérations supplémentaires pour un PortType, cliquer avec la touche de droite dans la fenêtre PortType, sélectionner **Ajouter opération**, et puis cliquer sur le type requis d'opération.)

Ajouter des message d'entrée et de sortie

Lorsqu'une opération est apposée à un PortType, vous pouvez choisir si l'opération doit être un des cinq types :

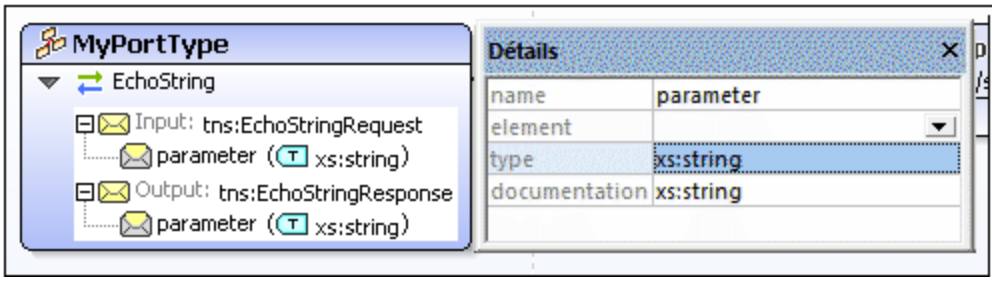
- demander réponse
- exiger réponse
- sens unique
- Notification
- Opération vide

Pour chaque type, des messages d'entrée et de sortie sont ajoutés automatiquement conformément au type d'opération. Lorsque **Opération vide** est sélectionné, cliquer avec la touche de droite sur l'opération pour vous permettre de sélectionner un type de message à insérer. Un message peut être supprimé en cliquant avec la touche de droite et en choisissant **Supprimer élément entrée/sortie**. Dans le cas de l'opération `EchoString`, renommer les messages d'entrée et de sortie en `EchoStringRequest` et `EchoStringResponse`, respectivement.



Ajouter des paramètres aux messages

Chaque message d'entrée ou de sortie est créé avec une partie de message par défaut (ou paramètre) de type `xs:string` (*voir capture d'écran ci-dessous*). Pour ajouter un autre paramètre, cliquer avec la touche de droite sur le message ou sur un de ses paramètres et sélectionner **Ajouter partie de message (paramètre)**.



Pour éditer un paramètre, procéder comme suit : (i) double-cliquer sur le texte pour l'éditer ; ou (ii) cliquer avec la touche de droite sur le paramètre et sélectionner **Éditer**, ou (iii) utiliser l'assistant à la saisie Détails (*voir capture d'écran ci-dessus*).

15.1.3 Créer une liaison

Une liaison est un protocole concret et une spécification de format de données pour un particulier. La création d'une liaison implique donc les éléments suivants :

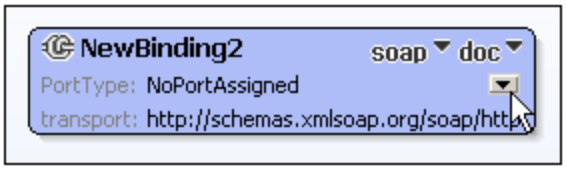
- Associer la liaison avec un PortType.

© 2016-2022 Altova GmbH

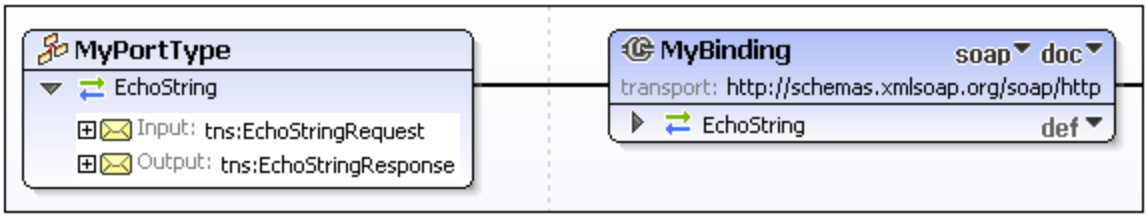
- Définir le protocole de la liaison et la spécification du format de données.
- Associer la liaison avec un port.

Associer une liaison avec un PortType

Lorsqu'une nouvelle liaison est créée (pour en créer une, cliquer avec la touche de droite n'importe où dans un endroit vide du design et choisir **Insérer liaison**), elle n'a aucun PortType qui lui est associé (*capture d'écran ci-dessous*). (Si vous avez créé un nouveau document WSDL, la liaison créée par défaut sera associée, par défaut, avec le PortType qui a été créé par défaut et l'association sera affichée par une ligne reliant les deux fenêtres).



Pour associer un PortType avec une liaison, dans la nouvelle fenêtre Liaison, cliquer sur la flèche Bas de l'entrée PortType (*voir capture d'écran ci-dessus*). Une liste des PortTypes définis dans le document s'affiche. Choisir le PortType avec lequel vous souhaitez associer la liaison. Lorsqu'un PortType a été associé avec la liaison, l'association est indiquée par une ligne reliant la fenêtre du PortType sélectionné à la fenêtre Liaison, comme ceci :



Choisir le protocole et le format de données

Le protocole de la liaison est sélectionné en cliquant sur la flèche Bas dans la barre de titre de la fenêtre Liaison (celle de l'entrée `soap/http`), et en sélectionnant un des quatre protocoles disponibles : `SOAP`, `SOAP 1.2`, `HTTP-GET`, et `HTTP-POST` (*capture d'écran ci-dessous*). Lorsque le protocole SOAP 1.1 ou 1.2 est choisi, vous pouvez sélectionne `document` ou `rpc` en tant que son format de données (en utilisant les options de la liste qui s'affiche en activant la flèche orientée vers le bas et située à droite de la sélection de protocole).



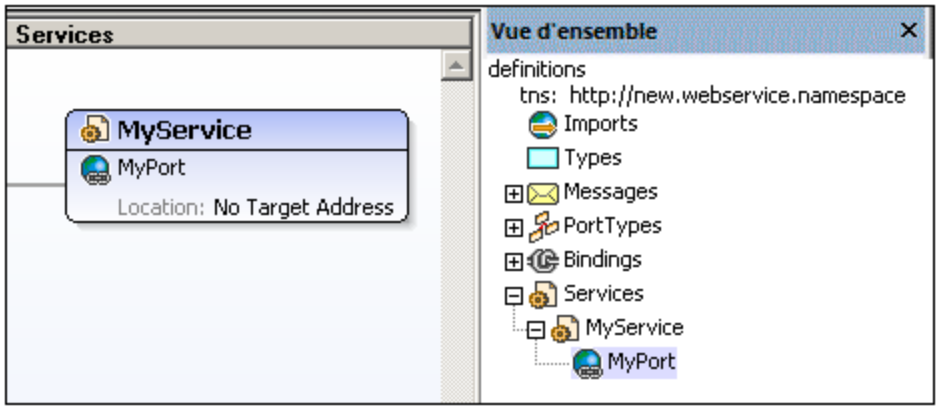
Le `soapAction` pour chaque opération dans la liaison peut être défini dans le design (*voir capture d'écran ci-dessus*) ou dans l'assistant à la saisie Détails lorsque cette opération est sélectionnée.

Associer une liaison avec un port

Pour associer une liaison avec un port, celui-ci doit tout d'abord être défini. Dans la section Créer un service et des ports, vous trouverez une description pour créer un port dans un service et associer un port avec une liaison.

15.1.4 Créer un service et des ports

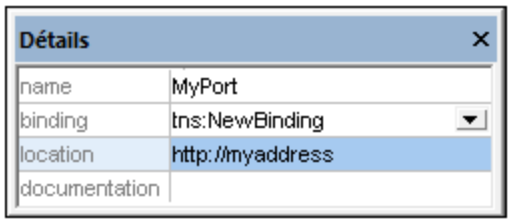
Pour ajouter un nouveau service, cliquer avec la touche de droite dans un endroit vide du design et choisir **Insérer Service** depuis le menu contextuel. Si vous avez créé un nouveau document WSDL, un service sera déjà présent dans le design. Vous pouvez renommer le service en double-cliquant dans son nom, en éditant le nom et en appuyant sur **Entrée**. Veuillez noter que le nom du service change aussi dans l'assistant à la saisie Aperçu (*capture d'écran ci-dessous*).



Dans l'assistant à la saisie Aperçu, double-cliquer dans l'entrée `NewPort`, la changer en `MyPort`., et appuyer sur **Entrée**. Veuillez noter que le nom du port change aussi dans la fenêtre `MyService` dans le design (*capture d'écran ci-dessus*). Pour ajouter des ports supplémentaires, cliquer avec la touche de droite soit sur le service soit sur le port, et, à partir du menu contextuel, sélectionner **Insérer port**.

Saisir l'adresse d'un port

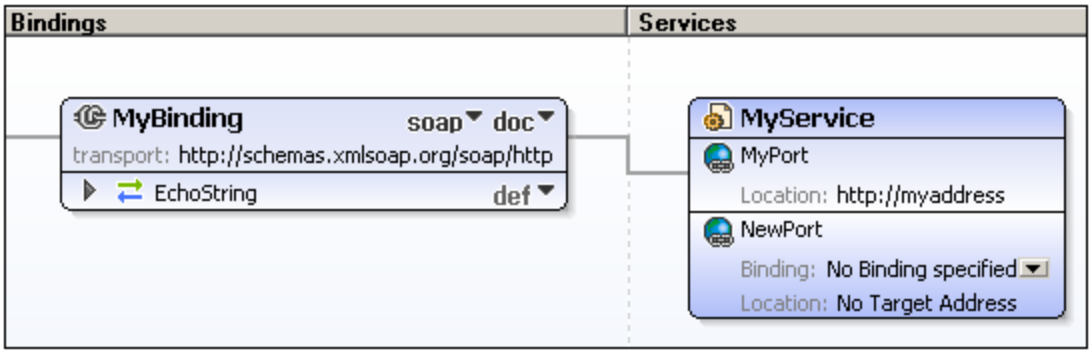
L'adresse d'un port peut être saisie soit : (i) directement dans le design, en tant que la valeur de l'item `Location` (*voir capture d'écran ci-dessus*), soit (ii) dans l'assistant à la saisie Détails (en double-cliquant dans le champ `Location` et en saisissant l'adresse (*capture d'écran ci-dessous*)).



© 2016-2022 Altova GmbH

Associer une liaison avec un port

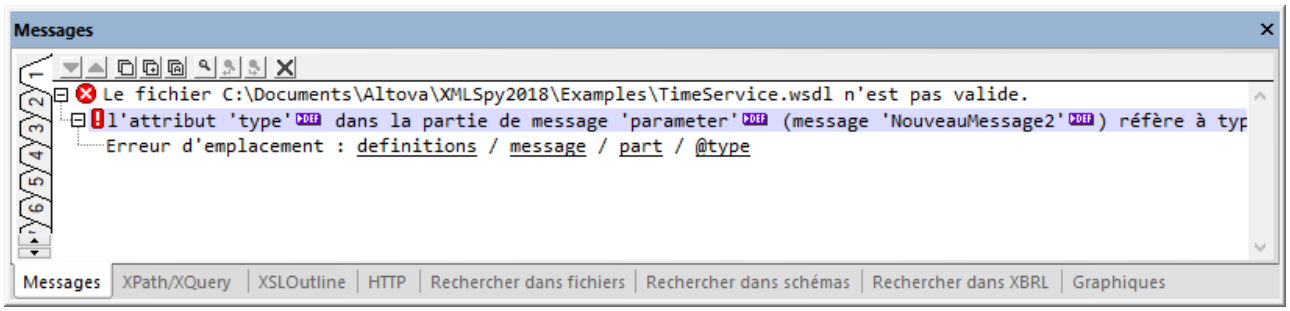
Un port est le point final qui combine une liaison avec une adresse de réseau. Une fois que l'adresse d'un port a été définie, tout ce qu'il vous reste à faire est d'associer une liaison avec le port. Pour associer une liaison avec un port, cliquer sur la flèche Bas de l'item de Liaison d'un port et effectuer votre choix dans la liste des liaisons définies dans le document.



Note : Si une liaison est déjà associée avec un port et que vous souhaitez associer une autre liaison, vous devrez supprimer la référence à la liaison (à l'aide de la touche de droite menu du port), puis en insérant la nouvelle référence à la liaison.

15.1.5 Valider le document WSDL

Une fois avoir terminé le document WSDL, il peut être validé en sélectionnant la commande **XML | Valider XML (F8)**. Les résultats de la validation sont affichés dans la fenêtre Messages (*capture d'écran ci-dessous*).



Des informations concernant toute erreur détectée sont affichées vous permettant ainsi de localiser rapidement l'erreur et de la réparer.

15.1.6 Se connecter à un service web et ouvrir des fichiers

Dans cette section, vous allez apprendre à :

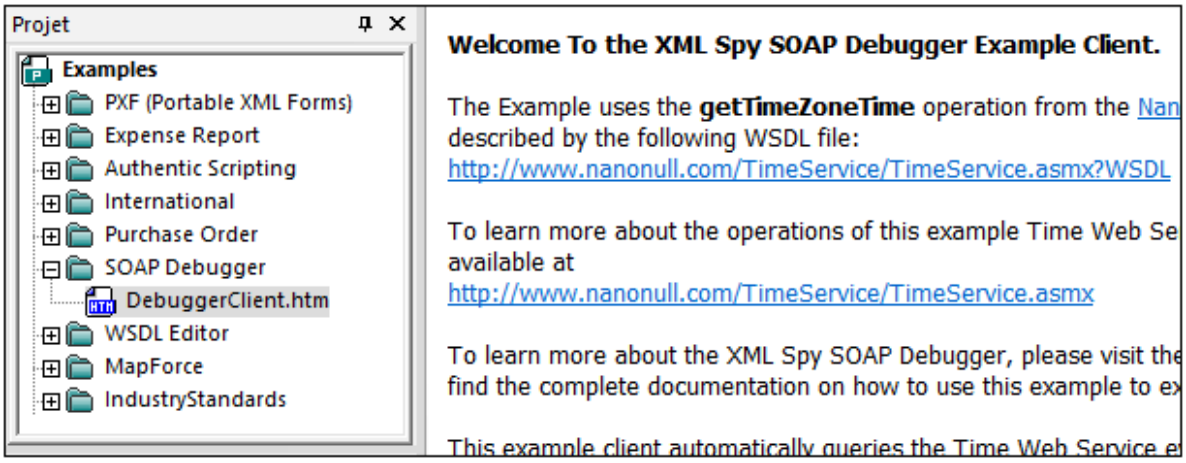
- Accéder à un service web à l'aide de XMLSpy

- Ouvrir dans XMLSpy un fichier WSDL sur le web
- Ouvrir dans XMLSpy un Schéma XML associé au document WSDL

Accéder à un service web

Un service web est généralement accédé par le biais d'une page HTML. Par exemple, `DebuggerClient.htm`, qui se trouve dans le dossier *Exemples* et dans le projet XMLSpy **`Examples/Soap Debugger`** (ouvrir **`Examples.spp`** pour travailler avec le projet). Pour accéder au service web affiché sur la page, procéder comme suit :

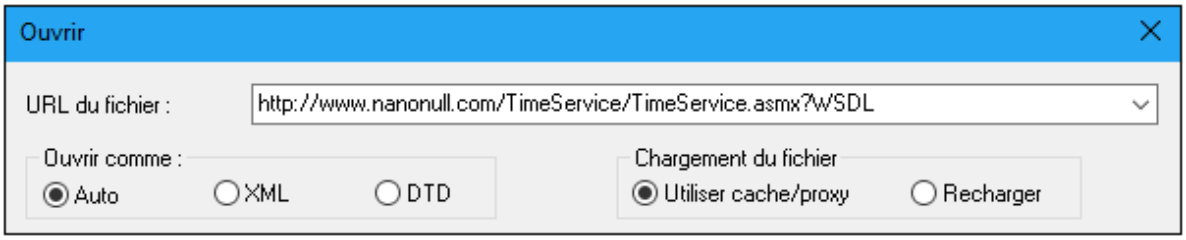
1. Activer la fenêtre Projet si celle-ci n'est pas visible (en utilisant l'option de menu **Fenêtre | Fenêtre de projet**).
2. Cliquer sur l'icône Agrandir située à côté du dossier Débogueur SOAP, et double-cliquer sur le fichier `DebuggerClient.htm`. Cela permet d'ouvrir le Client d'exemple de débogueur SOAP dans la Fenêtre principale.



Ouvrir un fichier WSDL dans XMLSpy

Pour ouvrir dans XMLSpy un fichier WSDL basé sur le web, procéder comme suit :

1. Choisir l'option de menu **Fichier | Ouvrir** et, dans le dialogue Ouvrir, cliquer sur **Passer à URL**. Ensuite, saisir ou copier l'adresse `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` dans le champ Fichier URL du dialogue.



2. Cliquer sur **Ouvrir** pour charger le fichier WSDL. Le fichier WSDL est affiché dans le Mode Texte.

3. Choisir l'option de menu **Fichier | Enregistrer sous**, et enregistrer le fichier localement, en le nommant, par exemple, `timeservice.wsdl`.
4. Cliquer sur l'onglet Mode WSDL pour afficher le fichier dans l'éditeur graphique WSDL.

© 2016-2022 Altova GmbH

Consulter le fichier de schéma associé avec le fichier WSDL actif

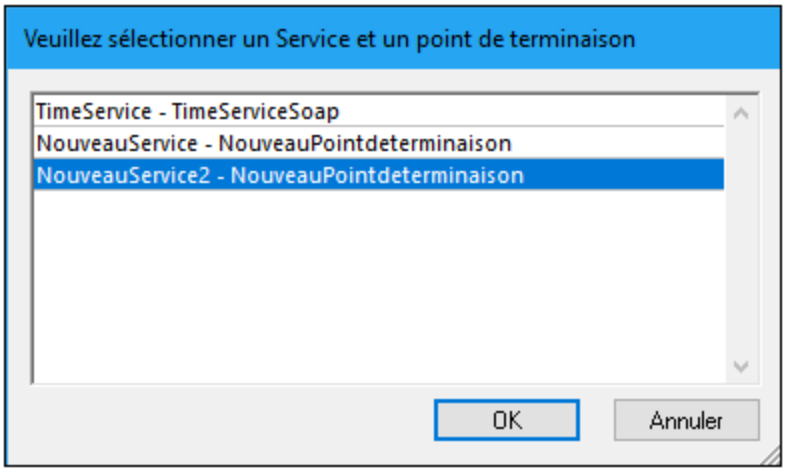
Avec le fichier `timeservice.wsdl` en guise de document actif dans le Mode WSDL, choisir l'option de menu **WSDL | Types | Éditer les schémas dans le Mode Schéma**. Cela permet d'ouvrir le fichier de schéma qui définit tous les types de données utilisés dans le fichier `timeservice.wsdl`. Vous pouvez modifier le schéma et enregistrer les changements. Ces changements prendront effet dès que le fichier WSDL sera reparsé.

Note : Il est recommandé d'accéder à WSDL en utilisant son nom de fichier (par exemple : **`timeservice.wsdl`**) au lieu d'utiliser la méthode de requête **`?wsdl`**. Cela est dû au fait que la méthode de requête peut retourner un fichier WSDL auquel il manquerait certaines fonctions du fichier WSDL original, ou un fichier ne fonctionnant pas correctement.

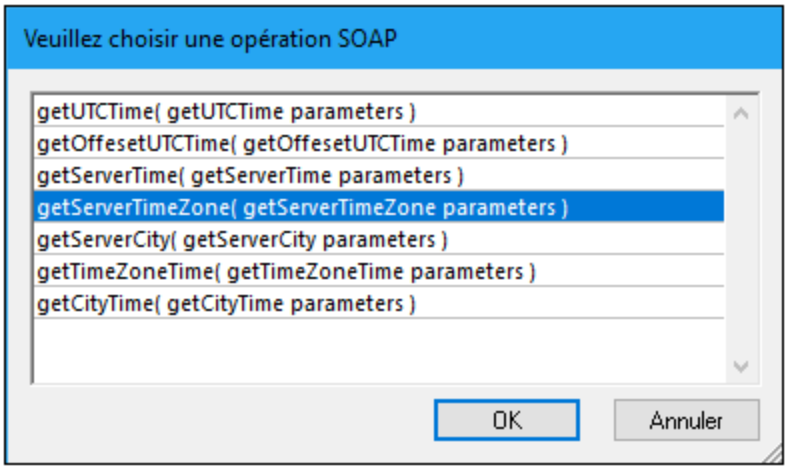
15.1.7 Envoyer une requête SOAP depuis le fichier WSDL

Pour envoyer une requête SOAP depuis le fichier `timeservice.wsdl`, procéder comme suit :

1. Faire du fichier `timeservice.wsdl` le fichier actif dans la Fenêtre principale.
2. Sélectionner l'option de menu **SOAP | Créer une Nouvelle requête SOAP**.
3. Chercher le fichier `timeservice.wsdl` et confirmer avec **OK**.
4. Si, parmi les différents services définis dans le document, il y a plus d'un port qui référence une liaison SOAP 1.1 ou 1.2, alors une fenêtre popup apparaîtra *(capture d'écran ci-dessous*) vous invitant à choisir le service et le port requis. Une fois avoir effectué la sélection, cliquer sur **OK**.



5. Dans le dialogue qui apparaît (*capture d'écran ci-dessous*), choisir une opération SOAP, par exemple `getServerTime`, et cliquer sur **OK**.



Cela permet de créer un document de requête SOAP contenant l'opération `getServerTime`. Vous pouvez l'enregistrer si vous le souhaitez.

6. Faire du document de requête le document actif et choisir l'option de menu **SOAP | Envoyer la requête au serveur**. Le document de réponse SOAP apparaît dans la Fenêtre principale, contenant l'élément `getServerTimeResult`, qui affiche l'heure de serveur actuelle du service horaire Nanonull.com.

15.1.8 Créer une Documentation WSDL

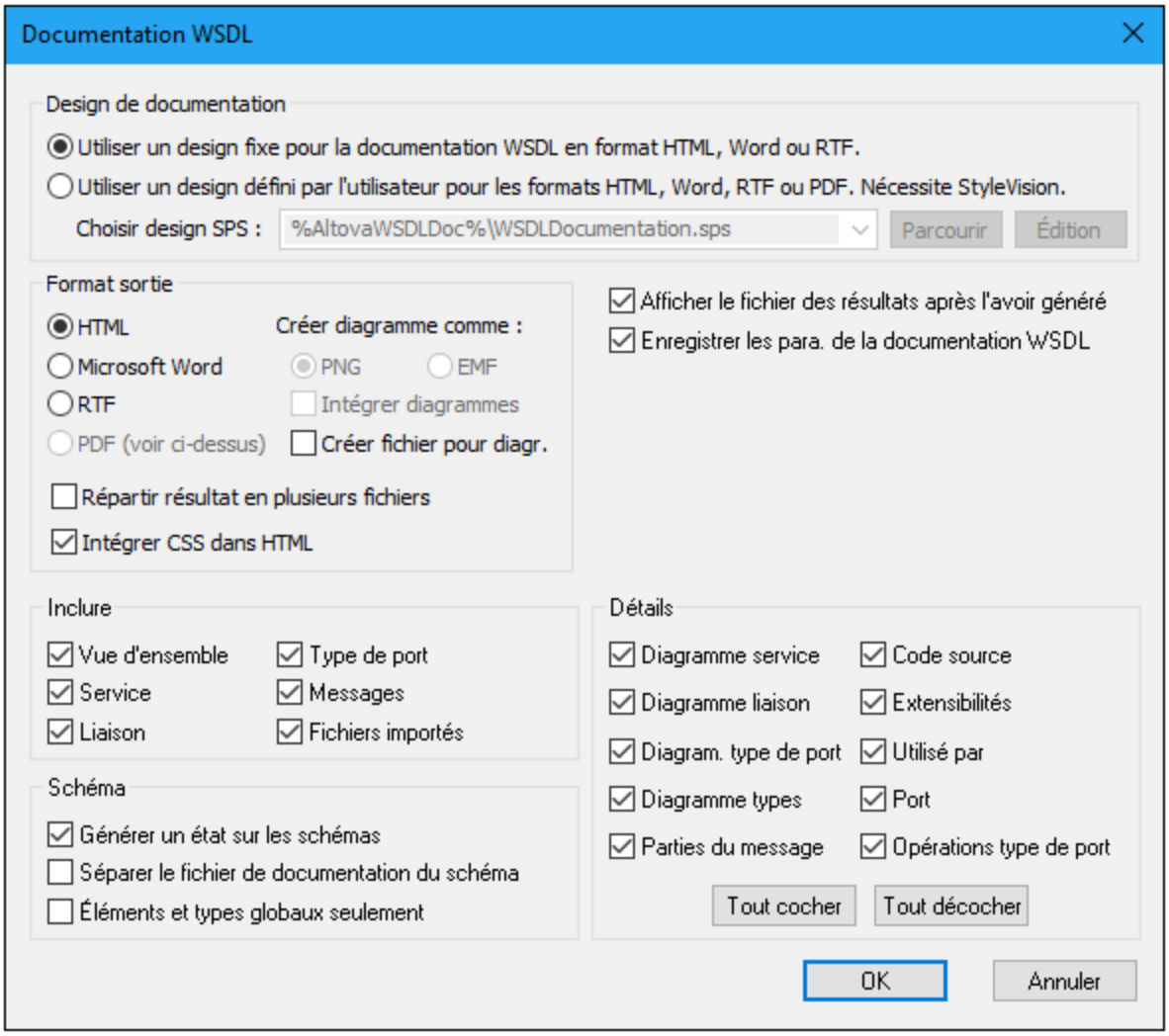
L'option **WSDL | Générer une documentation** vous permet de produire une documentation détaillée du document WSDL actuel. Vous pouvez réaliser une documentation sous format HTML, MS Word, ou RTF et spécifier les composants que vous souhaitez inclure. Les éléments WSDL liés sont reliés par hyperlien dans la documentation générée, permettant une navigation aisée.

Note : Pour générer une documentation dans un format MS Word, vous devez avoir installé MS Word (version 2000 ou plus).

Afin de générer une documentation pour le fichier WSDL, procéder comme suit :

1. Faire du fichier `timeservice.wsdl` le document actif.
2. Passer au mode **WSDL**.
3. Choisir l'option de menu **WSDL | Générer une documentation**.
   Le dialogue Documentation WSDL s'ouvre (*capture d'écran ci-dessous*).
4. Choisir le type de sortie que vous souhaitez générer : HTML, MS Word, ou RTF.
5. Choisir les composants WSDL spécifiques que vous souhaitez inclure dans la documentation, et définir d'autres options (voir Options de Documentation WSDL ci-dessous).

© 2016-2022 Altova GmbH



6. Cliquer sur **OK** et saisir le nom du fichier de documentation WSDL dans le dialogue Enregistrer sous.

Options de documentation WSDL

Vous pouvez choisir parmi les options de documentation suivantes :

- Le format requis est spécifié dans le panneau Format de sortie : soit HTML, Microsoft Word, ou RTF. La documentation peut être générée soit en tant que fichier unique, soit en fichiers multiples. Si des fichiers multiples sont générés, chaque fichier correspondra à un composant. Les composants inclus dans la sortie sont spécifiés à l'aide des cases à cocher dans le panneau Inclure.
- L'option *Intégrer les diagrammes* est activée pour les options de sortie MS Word et RTF. Lorsque cette option est cochée, les diagrammes sont intégrés dans le fichier de résultat, sous format PNG ou EMF. Dans le cas contraire, les diagrammes sont créés en tant que fichiers PNG ou EMF, qui sont affichés dans le fichier de résultat par le biais de liens d'objet. Si la sortie est un HTML, tous les diagrammes sont créés en tant que fichiers PNG externes au document.
- Dans le panneau Inclure, vous sélectionnez les items que vous souhaitez inclure dans la documentation. L'option *Aperçu* regroupe tous les composants, organisés par type de composant, en

haut du fichier. Si l'option *Fichiers importés* est cochée, les composants dans les fichiers importés sont inclus dans la documentation de schéma.

- Dans le volet Schéma, vous pouvez sélectionner si les schémas dans le fichier sont rapportés ou non. Si vous choisissez de reporter les schémas, vous pouvez aussi choisir : (i) si la documentation de schéma doit être rapportée dans un fichier séparé ou dans le fichier de documentation principal, et (ii) si le schéma complet doit être rapporté ou bien seulement les éléments globaux, les types simples, et les types complexes.
- Le panneau Détails recense les détails qui peuvent être inclus pour chaque composant. Choisir les détails que vous souhaitez inclure dans la documentation.
- L'option *Afficher le fichier de résultat* est activée pour les trois options de sortie. Lorsque cette option est cochée, les fichiers de résultat sont affichés dans le Mode Navigation (sortie HTML), MS Word (sortie MS Word), et l'application par défaut pour les fichiers `.rtf` (sortie RTF).

15.1.9 Convertir en WSDL 2.0

Dans XMLSpy vous pouvez facilement convertir un document WSDL 1.1 existant au format WSDL 2.0. Testez cette fonction avec l'exemple `TimeService.wsdl` :

1. Faire du fichier `TimeService.wsdl` le fichier actif. (Ce fichier se trouve dans le dossier `WSDL Editor` dans le projet `Examples` de XMLSpy.)
2. Cliquer sur la commande **WSDL | Convertir en WSDL 2.0**.
3. Dans le dialogue Enregistrer sous qui apparaît, saisir le nom avec lequel vous souhaitez enregistrer le fichier WSDL 2.0, par exemple, `TimeService20.wsdl`.
4. Le nouveau fichier est généré, validé automatiquement et affiché dans le Mode WSDL.

© 2016-2022 Altova GmbH

15.2 SOAP

Site Internet Altova : Débogueur SOAP
Dans cette section vous allez apprendre à :

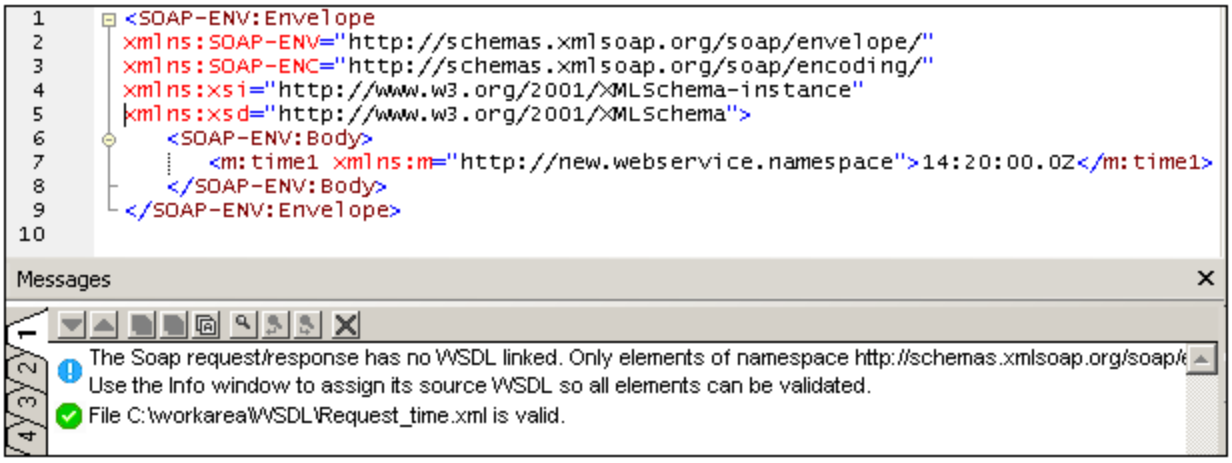
- Valider les messages SOAP par rapport à WSDL. La validité des messages SOAP peut être contrôlée non seulement par rapport à la spécification SOAP mais aussi par rapport à tout Schémas XML référencé dans la définition WSDL correspondante
- Envoyer et recevoir des requêtes SOAP à l'aide du Débogueur SOAP
- Définir des points d'arrêt pour l'envoi et la réception de requêtes SOAP
- Éditer une requête SOAP incorrecte avant de l'envoyer sur le service web

15.2.1 Validation SOAP

La validité des messages SOAP peut être contrôlée non seulement par rapport à la spécification SOAP, mais aussi par rapport au Schéma XML référencé dans la définition WSDL correspondante.

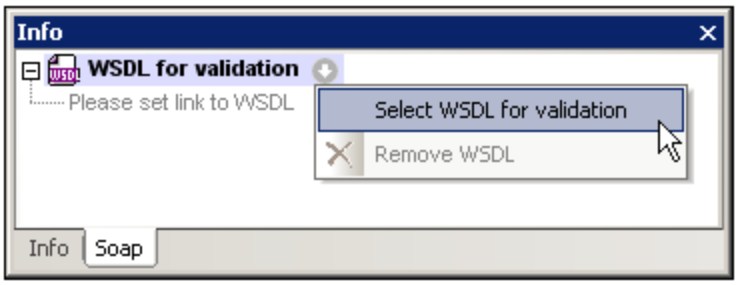
Valider par rapport aux règles SOAP uniquement

Pour valider un message SOAP, ouvrir le fichier de message SOAP (*capture d'écran ci-dessous*) et appuyer sur **F8** (ou sur la commande de menu **XML | Valider**). Puisqu'aucun fichier WSDL n'a été lié au fichier de message SOAP, celui-ci est validé conformément aux règles pour les messages SOAP. Le fichier est valide s'il est valide conformément à ces règles (*voir la fenêtre Messages dans la capture d'écran ci-dessous*).

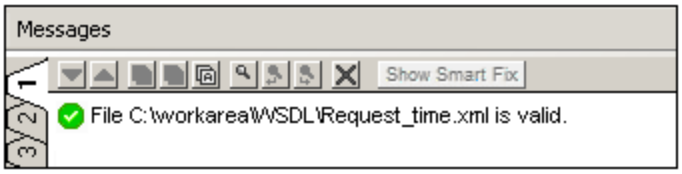


Valider par rapport aux règles SOAP et au WSDL lié

Pour valider un message SOAP supplémentaire conformément au WSDL lié, le fichier WSDL doit être lié au fichier SOAP. Cela est effectué dans l'onglet SOAP de la fenêtre Info (*capture d'écran ci-dessous*). Cliquer sur la touche située à gauche de l'item *WSDL pour validation* et sélectionner la commande **Sélectionner WSDL pour Validation**. Dans le dialogue qui s'affiche, chercher le fichier WSDL que vous souhaitez et cliquer sur **OK**. Le fichier WSDL sera saisi dans la fenêtre Info et le fichier de message SOAP y sera lié.



En pressant sur **F8** (ou sur la commande de menu **XML | Valider**) le message SOAP sera non seulement validé par rapport aux règles pour les messages SOAP mais aussi par rapport aux règles dans le fichier WSDL lié.



Le fichier est considéré valide s'il est valide par rapport aux deux ensembles de règles (*voir capture d'écran ci-dessus*).

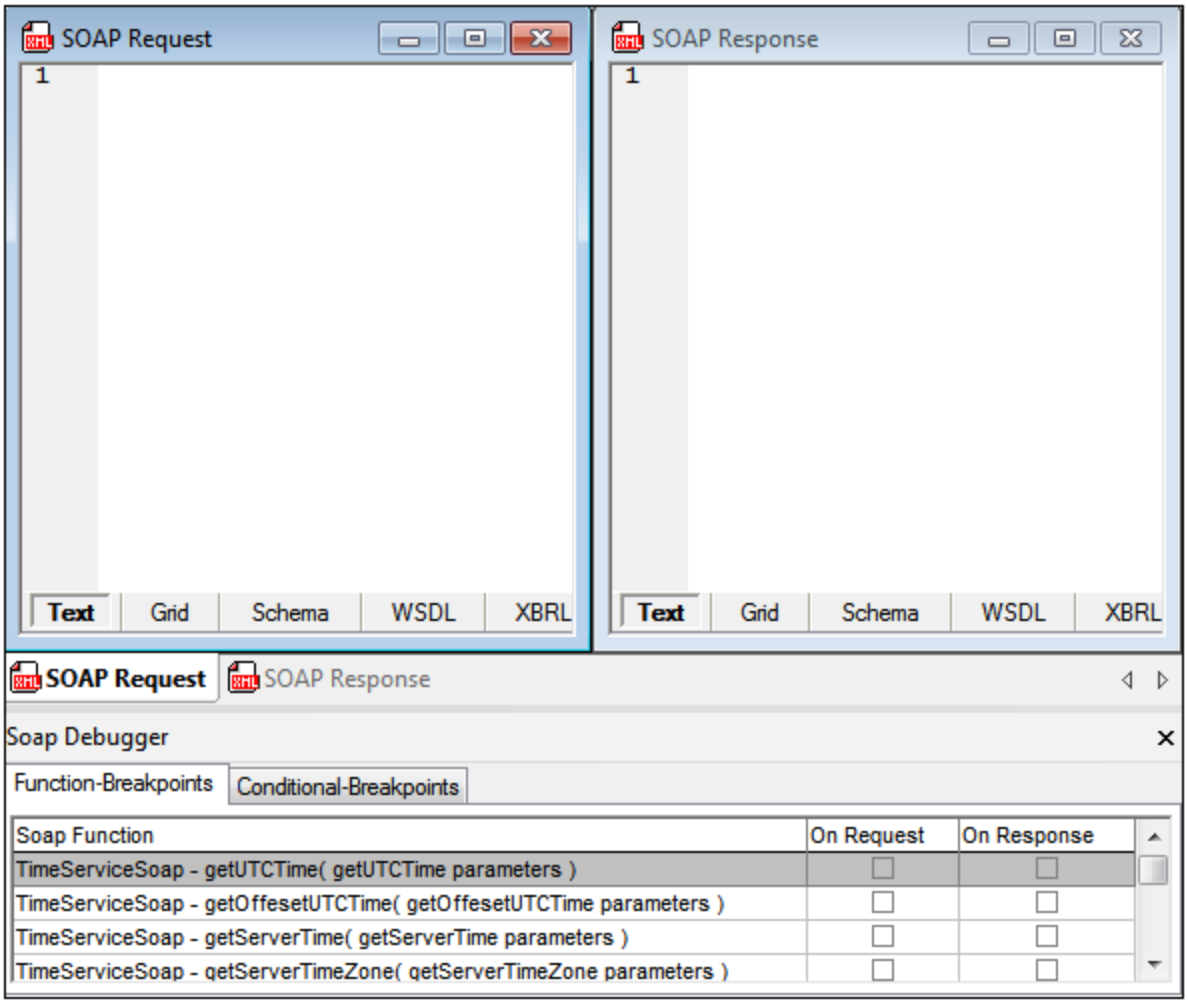
Note : L'onglet SOAP est visible dans la fenêtre Info si la requête SOAP a été créée avec la fonction de requête SOAP de XMLSpy à partir d'un fichier WSDL (**SOAP | Créer nouvelle requête SOAP**). Si l'onglet SOAP n'est pas visible dans la fenêtre Info (parce que la requête SOAP n'a pas été créée avec XMLSpy), alors l'enregistrement du fichier de requête SOAP permettra de rendre l'onglet SOAP visible.

15.2.2 Débogueur SOAP

Le débogueur SOAP (*capture d'écran ci-dessous*) peut être utilisé pour consulter et analyser les requêtes SOAP et les réponses. Il fonctionne en tant que **serveur proxy** entre votre client et le service web. Vous pouvez procéder aux étapes suivantes :

- Passer par les requêtes et réponses SOAP
- Modifier les requêtes et réponses SOAP
- Transmettre les requêtes modifiées au client ou au serveur
- Permettre les points d'arrêt pour chaque message de requête et de réponse, y compris les points d'arrêt conditionnels par le biais d'expressions XPath

© 2016-2022 Altova GmbH



Le débogueur SOAP fonctionne comme suit :

- Les options de débogueur SOAP peuvent être définies avant de lancer une session de débogage SOAP. Ces options incluent l'adresse IP de l'ordinateur et des options de mise en page et de temporisation pour le débogueur SOAP.
- Pour ouvrir le débogueur SOAP (lancer une session), sélectionner la commande à bascule **SOAP | Session de débogage SOAP**. À ce moment, vous devrez fournir (i) l'emplacement du fichier WSDL qui sera utilisé pour fournir les informations SOAP pertinentes et (ii) des informations concernant les ports de source et de cible.
- Dans la fenêtre Points d'arrêt du débogueur SOAP, définir les points d'arrêt requis.
- À présent, vous pouvez ouvrir le fichier qui effectue la requête SOAP et exécuter le débogueur SOAP.
- Vous pouvez ensuite analyse les résultats, et, en cas d'erreur, les réparer.
- Pour fermer le débogueur SOAP, choisir la commande à bascule **SOAP | Session de débogueur SOAP**.

Dans les sous-sections de cette section, nous allons décrire comment utiliser le débogueur SOAP.

Le fichier `DebuggerClient.htm`, qui se trouve dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples`, est utilisé en tant que fichier d'exemple. La fenêtre de navigation sert de l'application client pour ce fichier d'exemple, elle envoie et reçoit des messages SOAP. Le service

Nanonull Time Service est le serveur de service web et est situé sous :
`http://www.nanonull.com/TimeService/TimeService.asmx?WSDL`.

15.2.2.1 Processus de communications SOAP

Une fois que le serveur de proxy (débogueur SOAP) a été lancé, le processus de communication SOAP est le suivant :

Le serveur proxy écoute en continu sur une prise/port pour les requêtes client arrivantes

- L'application client envoie une requête au serveur proxy
- Les requêtes client peuvent être modifiées si/quand des points d'arrêt ont été déclenchés
- Les données de requête de serveur proxy sont transférées au serveur de service web

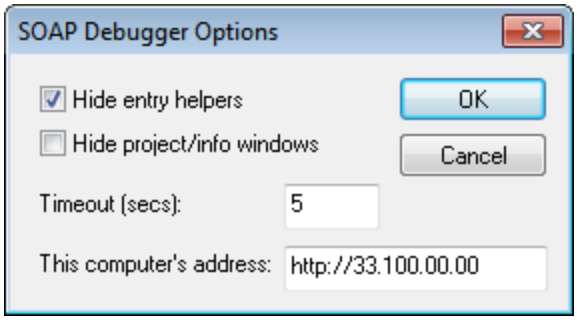
Le serveur de service web répond à la requête proxy, et envoie les données de réponse au serveur proxy

- Les réponses de serveur peuvent être modifiées si/quand des points d'arrêt ont été déclenchés
- Les données de réponse de serveur proxy sont transférées à l'application client
- L'application client reçoit les données de réponse depuis le serveur proxy

Paramètres de port
Le débogueur SOAP utilise le port 8080 pour contrôler les requêtes du client. Le port peut uniquement être modifié lors du lancement d'une nouvelle session de débogage SOAP. Si ce port est désactivé par des pare-feu personnels, vous devrez désactiver ces programmes ou choisir une autre adresse de port.

15.2.2.2 Options du Débogueur SOAP

Le dialogue Options du Débogueur SOAP (*capture d'écran ci-dessous*) vous permet de spécifier l'adresse IP de l'ordinateur et les autres options de débogage qui sont recensées ci-dessous. Accéder au dialogue avec la commande de menu **SOAP | Options du Débogueur SOAP**.



- *Adresse de l'ordinateur :* L'adresse du serveur proxy à partir de laquelle le débogueur est exécuté. Le débogueur sur le serveur proxy prend les requêtes depuis des appareils sur le réseau et les envoie au service web. Puisque le débogueur est exécuté dans XMLSpy, l'appareil sur lequel XMLSpy est installé sert aussi en tant que le serveur proxy. L'adresse IP de la machine est détectée automatiquement et saisie dans ce champ. Ce n'est que si vous ne pouvez pas détecter l'adresse IP automatiquement que vous devrez la saisir (en tant qu'adresse `http`) dans ce champ. Pour connaître l'adresse IP de votre

© 2016-2022 Altova GmbH

ordinateur, ouvrir une fenêtre d'invite de commande, saisir la commande `ipconfig /all`, et appuyer sur **Entrée**.
- *Temporisation :* cette valeur est la quantité de temps pendant laquelle le Débogueur SOAP demeure dans un point d'arrêt. Le défaut est de 5 secondes.
- *Dissimuler des assistants à la saisie ; Dissimuler des fenêtres de project/info :* ces options sont utiles pour fournir plus d'espace d'écran pour la fenêtre Débogueur SOAP.

15.2.2.3 Lancer une session de débogage

Une session de Débogage SOAP peut être lancée lorsqu'un fichier est actif dans XMLSpy; aucun fichier SOAP ou fichier de point d'entrée de requête SOAP ne doit nécessairement être actif lorsque le débogueur SOAP est lancé. Lors du démarrage d'une session de Débogage SOAP, vous serez invité à :

1. nommer l'emplacement du fichier WSDL qui sera utilisé pour fournir des informations SOAP, et
2. les paramètres de connexion.

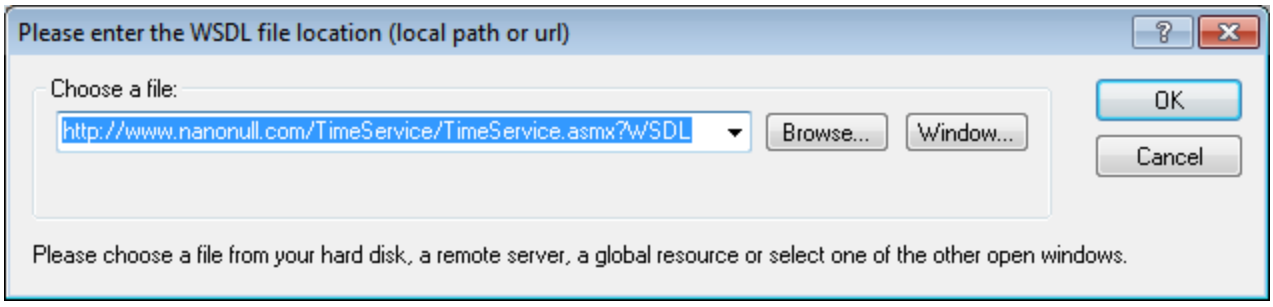
Ces étapes sont décrites ci-dessous.

Emplacement du fichier WSDL

Lorsqu'une session de Débogage est lancée, vous serez invité à saisir l'URL du fichier WSDL qui sera utilisé pour fournir les informations SOAP. Notre fichier d'exemple, `DebuggerClient.html`, utilise l'url de fichier WSDL suivante :

```
http://www.nanonull.com/TimeService/TimeService.asmx?WSDL
```

Lancer le Débogueur SOAP en choisissant la commande de menu **SOAP | Session de débogage SOAP**. Cela permet d'ouvrir le dialogue Emplacement du fichier WSDL (*capture d'écran ci-dessous*).



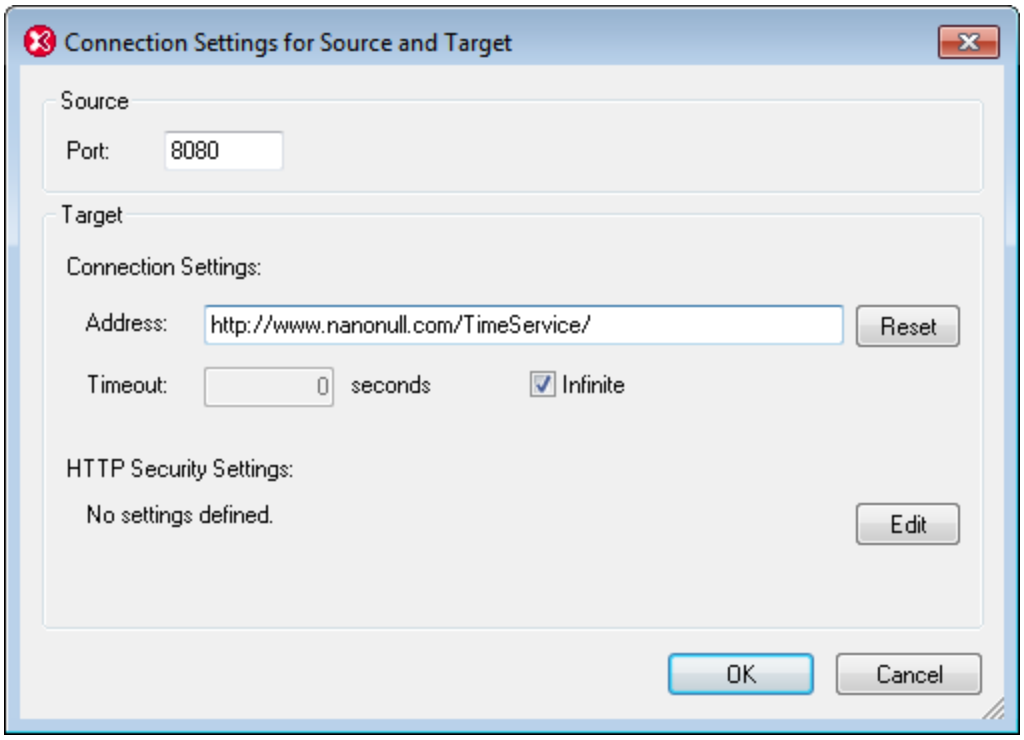
Saisir l'URL du fichier WSDL et cliquer sur **OK**. Le dialogue Ports source et cible (*capture d'écran ci-dessous*) s'affiche.

Paramètres de connexion

Le dialogue Paramètres de connexion pour la source et la cible (*capture d'écran ci-dessous*) propose les paramètres suivants :

- *Port source :* le port sur un serveur proxy (qui peut être voter ordinateur) qui sera utilisé pour la communication. Le réglage par défaut est `8080`. Ce paramètre peut être changé à chaque fois que le Débogueur SOAP est lancé.

- *Port et adresse cibles :* ces paramètres sont fournis par le fichier WSDL sélectionné dans l'étape précédente ; ils sont saisis automatiquement dans le dialogue. Le port par défaut est le port HTTP standard `80`. Vous pouvez définir une temporisation pour la connexion ou cocher l'option *Infini* pour désactiver la temporisation. Afin de définir les paramètres de sécurité HTTP, cliquer sur la touche **Éditer** du volet Paramètres de sécurité HTTP, et saisir les paramètres de sécurité. Pour plus d'informations concernant ces paramètres, voir la section Paramètres de requête SOAP.



Une fois avoir coché ces paramètres, et les avoir édités le cas échéant, cliquer sur **OK** pour lancer la session de Débogage SOAP. Le Débogueur SOAP démarre mais le serveur proxy n'est pas actif (indiqué par l'icône de serveur de proxy dans la barre outils du Débogueur SOAP grisée). Pour lancer le serveur proxy (le débogage), cliquer sur l'icône **Go** dans la barre outils XMLSpy, ou sélectionner la commande de menu **SOAP | Go**. Voir la section Débogage, pour plus d'informations concernant le débogage lui-même.

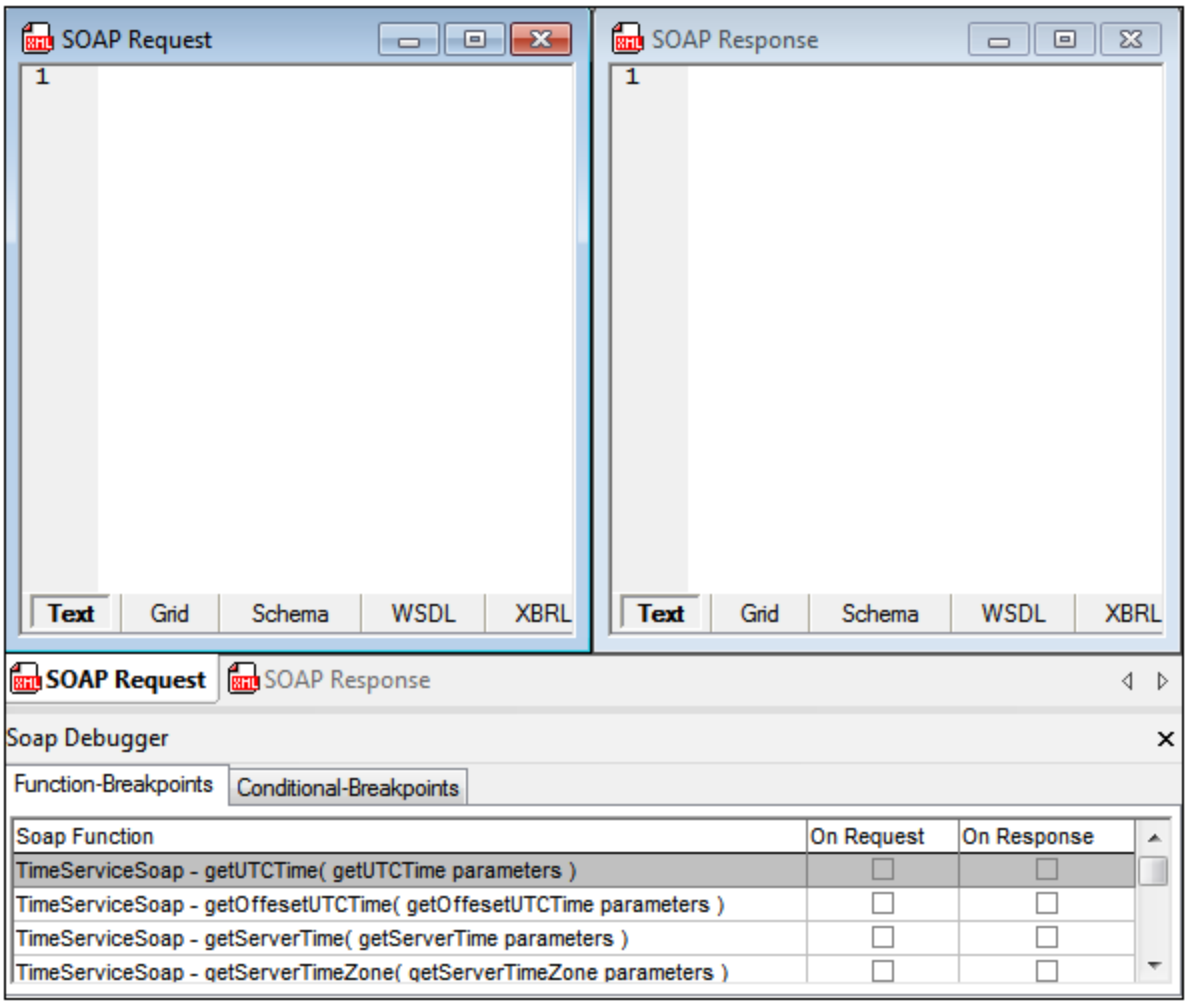
Mise en page du Débogueur SOAP

Le Débogueur SOAP a trois fenêtres (*voir capture d'écran ci-dessous*):

- une fenêtre Requête SOAP,
- une fenêtre Réponse SOAP et
- une fenêtre de configuration de points d'arrêt du Débogueur SOAP.

Par défaut, les fenêtres de Requête et de Réponse sont ouvertes dans la partie supérieure de l'interface XMLSpy et la fenêtre de configuration de points d'arrêt est alignée le long du bas. La capture d'écran ci-dessous montre la mise en page par défaut.

© 2016-2022 Altova GmbH



Les fenêtres du Débogueur SOAP peuvent se voir attribuer plus de place en dissimulant les fenêtres latérales de XMLSpy (fenêtres Projet, Info et assistant à la saisie). Les paramètres pour dissimuler/afficher ces fenêtres sont disponibles dans le dialogue Options du Débogueur SOAP (accessible via la commande de menu **SOAP | Options du Débogueur SOAP**).

À propos des certificats autorisés

Les produits Altova utilisent Internet Explorer (IE) pour accéder et gérer les certificats autorisés de serveurs web sûrs. L'installation du certificat d'un serveur web dans IE permet à IE d'accéder au serveur sans émettre d'avertissement ou d'interrompre le processus. La procédure de base pour installer le certificat d'un serveur web sûr est recensé ci-dessous. Les étapes peuvent devenir plus complexes selon la version du navigateur utilisée.

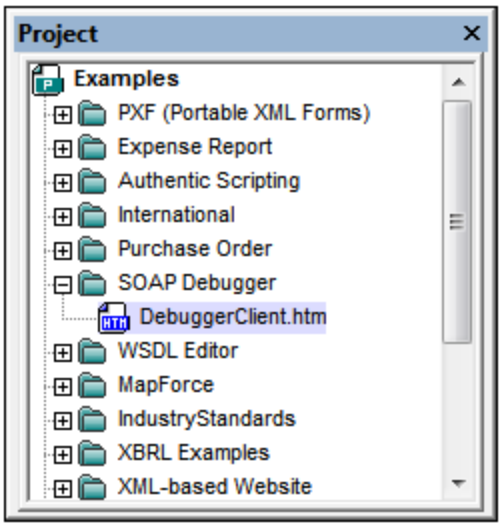
1. Dans Internet Explorer 9 (ou plus), ouvrir le site web sûr.
2. Choisir **Fichier | Propriétés**, et cliquer sur la touche Certificats.
3. Cliquer sur **Installer Certificat** et lancer l'Assistant d'importation du certificat. (Cet assistant peut aussi être accédé depuis **Outils | Options Internet | Contenu | Certificats | Importer**.)
4. Le certificat doit être placé dans le stockage Trusted Root Certification Authorities, que vous pouvez rechercher manuellement.

5. Terminer les étapes de l'assistant, fermer les dialogues Certificats et Propriétés en cliquant sur **OK**. Vous devrez redémarrer Internet Explorer.

15.2.2.4 Point d'entrée Requête SOAP

Vous trouverez dans le projet Examples (`DebuggerClient.htm`) un fichier HTML qui contient un script montrant comment utiliser le Débogueur SOAP. Ce fichier permet à l'utilisateur d'envoyer une requête SOAP à un service web et puis d'afficher la réponse depuis le service web. Suivez les instructions suivantes pour ouvrir ce fichier dans XMLSpy :

1. Choisir la commande de menu **Projet | Ouvrir projet**.
2. Aller dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples` et choisir le fichier `Examples.spp`. Cela permet de charger le projet Examples dans la fenêtre Projet (*capture d'écran ci-dessous*).



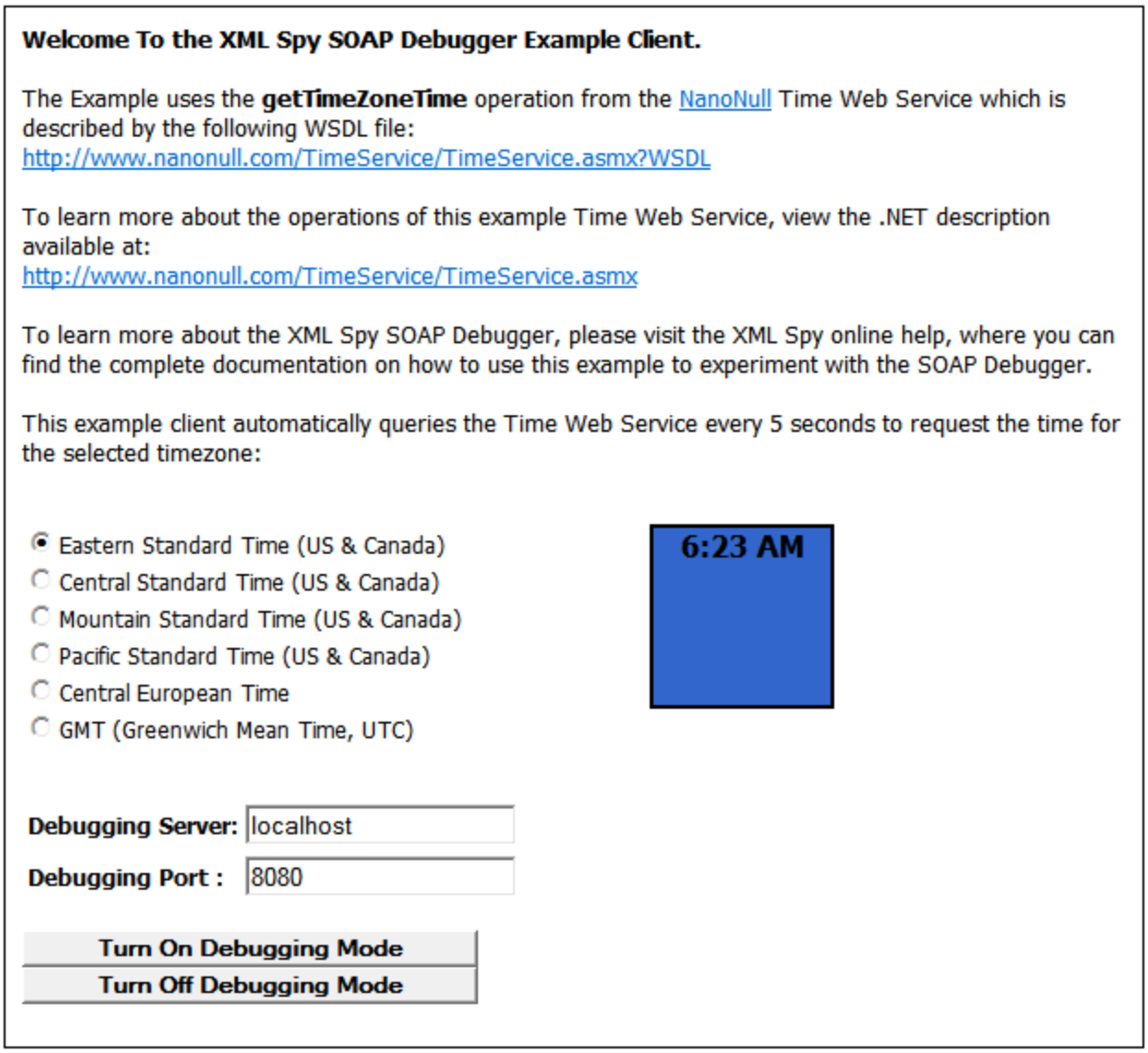
3. Cliquer sur le signe + du dossier SOAP Debugger pour consulter son contenu. Double-cliquer sur `DebuggerClient.htm` pour ouvrir le fichier dans XMLSpy.

Note : En alternative, vous pouvez ouvrir ce fichier (`DebuggerClient.htm`) par le biais de la commande de menu **Fichier | Ouvert**. Le fichier est situé dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples`.

Le fichier d'échantillon

Le fichier `DebuggerClient.htm` ressemblera à l'exemple suivant dans le Mode Navigateur de XMLSpy. Lorsqu'un des boutons radio est sélectionné, une requête SOAP est envoyée au Nanonull Time Web Service. La réponse du service web est affichée dans la fenêtre colorée située à droite des boutons radio.

© 2016-2022 Altova GmbH



Veuillez noter que la réponse à une requête Eastern Standard Time est affichée (`6:23`) dans un "cadre horaire" bleu dans la capture d'écran ci-dessus. À présent, sélectionner le bouton radio GMT. Au lieu d'afficher la valeur GMT dans le cadre horaire (le cadre de réponse du service Web), un message d'erreur s'affiche et le cadre horaire passe au rouge (*voir capture d'écran ci-dessous*).



Le Débogueur SOAP peut maintenant être utilisé pour analyser les messages SOAP afin de localiser l'erreur. Les trois sections suivantes discuteront, respectivement, (i) comment définir des points d'arrêt, (ii) comment exécuter le Débogueur SOAP avec `DebuggerClient.htm`, et (iii) comment analyser la sortie du Débogueur SOAP pour localiser les erreurs.

15.2.2.5 Configurer des points d'arrêt

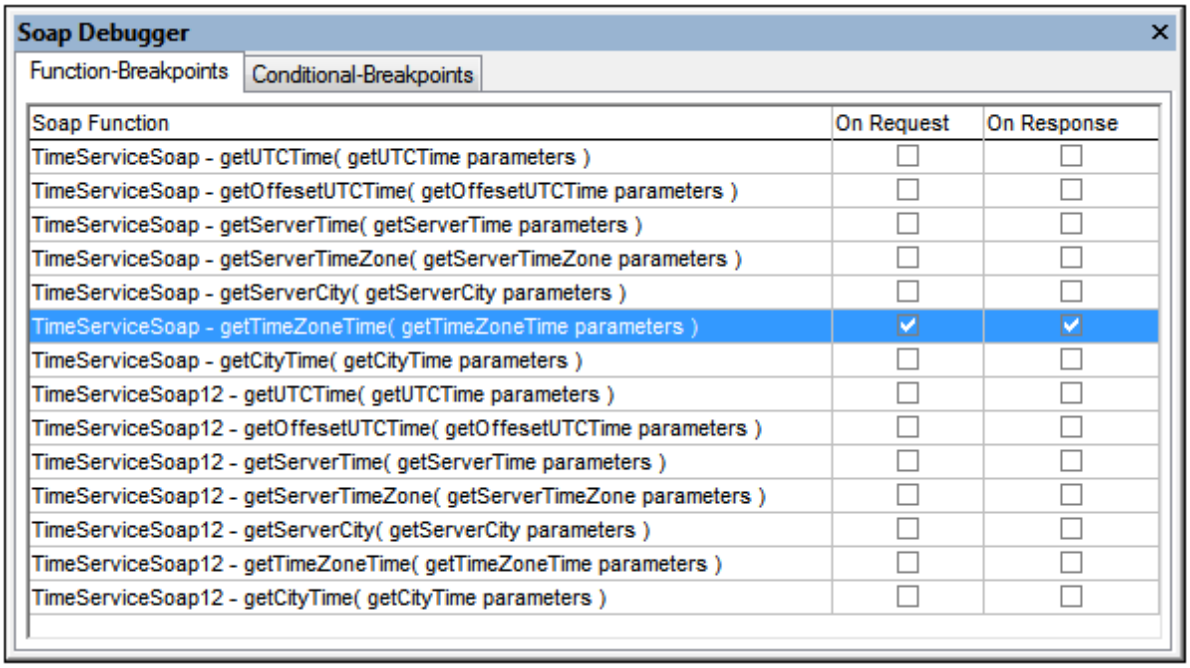
Avant de commencer le débogage, vous devez définir des points d'arrêt dans le Débogueur SOAP. Une fois que le débogage est lancé, le Débogueur SOAP montrera les requêtes et les réponses au niveau des points d'arrêt qu'il rencontre.

Le Débogueur SOAP recense les points d'arrêt (dans ses volets *Function-Breakpoints* et *Conditional-Breakpoints*) conformément à l'information obtenue depuis le fichier WSDL qui a été sélectionné au moment du lancement du Débogueur SOAP. Ces points d'arrêt sont liés aux requêtes SOAP qui peuvent être générées par le fichier WSDL. Pour chaque requête SOAP, un point d'arrêt sur requête et sur réponse peut être sélectionné en cochant les cases dans les colonnes correspondantes (*voir capture d'écran ci-dessous*).

Dans notre exemple, nous utilisons :

- `DebuggerClient.htm` en tant que le point d'entrée de requête SOAP, et
- le fichier WSDL `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` qui a été sélectionné lors du lancement du Débogueur SOAP.

 © 2016-2022 Altova GmbH



Le service Web demandé par `DebuggerClient.htm` utilise la méthode `getTimeZoneTime` pour trouver le temps dans le fuseau horaire sélectionné. Dans le Débogueur SOAP, les requêtes SOAP qui peuvent être générées à partir du fichier WSDL sélectionné sont regroupés en tant que points d'arrêt. Nous configurons des sous `getTimeZoneTime` aussi bien pour *Sur requête* que *Sur réponse* (*voir capture d'écran ci-dessous*). Cela vous permet d'analyser aussi bien les requêtes SOAP que les réponses de service Web en cas d'erreurs.

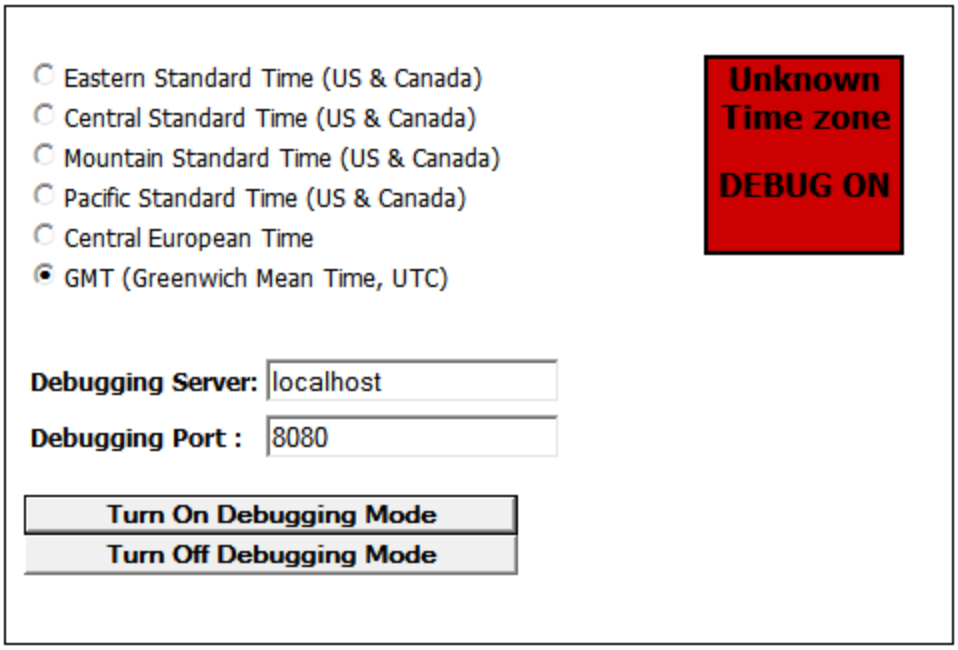
Pour des informations plus détaillées concernant la configuration des points d'arrêt, voir la section, En savoir plus sur les points d'arrêt.

15.2.2.6 Déboguer

Dans notre exemple, nous utilisons :

- `DebuggerClient.htm` en tant que le point d'entrée de requête SOAP, et
- le fichier WSDL `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` qui a été sélectionné lorsque le Débogueur SOAP a été lancé.

  Une fois avoir défini les points d'arrêt, cliquer sur l'icône **GO** (ou utiliser la commande de menu **SOAP | GO**). Puis cliquer sur l'onglet *DebuggerClient.htm* pour passer au fichier du point d'entrée SOAP. Veuillez vous assurer que l'option GMT a été sélectionnée et cliquer sur la touche **Turn On Debugging Mode** (*voir capture d'écran ci-dessous*). Un message *Debug On* s'affiche et envoie la requête SOAP au Débogueur SOAP. Les résultats du Débogueur sont affichés dans les fenêtres Requête SOAP et Réponse SOAP, et sont décrites dans la section suivante, Analyser les résultats du Débogueur en cas d'erreurs.



Commandes du Débogueur SOAP

La barre d'outils du Débogueur SOAP (*capture d'écran ci-dessous*) contient des icônes pour faire fonctionner le Débogueur SOAP.



Ces icônes sont, de gauche à droite :

- *Aller :* lance le débogage.
- *Étape unique :* passe par les étapes du processus de Requête-Réponse, s'arrête aux points d'arrêt.
- *Arrêter à la prochaine requête :* s'arrête à la Requête SOAP suivante.
- *Arrêter à la prochaine réponse :* s'arrête à la réponse suivante du service web.
- *Arrêter le serveur proxy :* arrête le débogage. Veuillez noter qu'il ne s'agit pas de la même chose que terminer la session du Débogueur SOAP. Pour terminer/lancer le Débogueur SOAP, sélectionner la commande de menu **SOAP | Session du Débogueur SOAP**.

15.2.2.7 Analyser les résultats et réparer les erreurs

Les résultats du Débogueur SOAP sont affichés dans deux fenêtres : Requête SOAP et la Réponse SOAP. Les points d'arrêt sont définis dans les volets de points d'arrêt du Débogueur SOAP qui sont situés par défaut en bas de la fenêtre du Débogueur SOAP. Conformément aux points d'arrêt qui ont été définis, le Débogueur SOAP affichera des résultats dans la fenêtre de résultats appropriée : Requête SOAP ou Réponse SOAP.

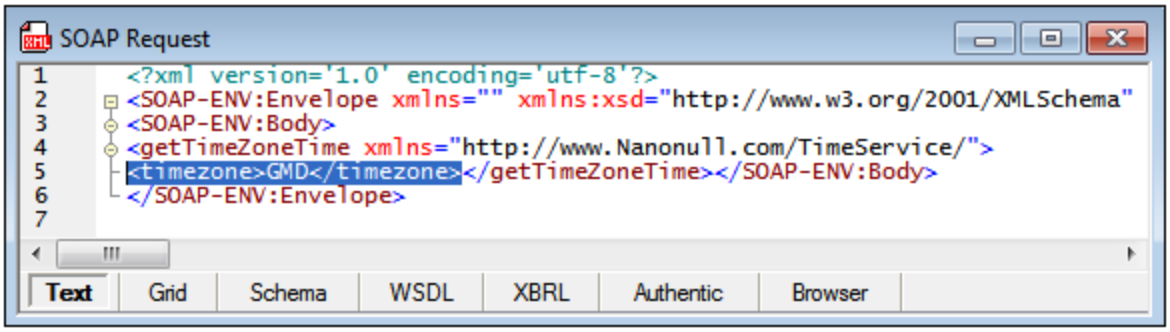
Dans notre exemple nous utilisons :

© 2016-2022 Altova GmbH

- `DebuggerClient.htm` en tant que le point d'entrée de Requête SOAP et
- le fichier WSDL `http://www.nanonull.com/TimeService/TimeService.asmx?WSDL` qui a été sélectionné lorsque le Débogueur SOAP a été lancé.

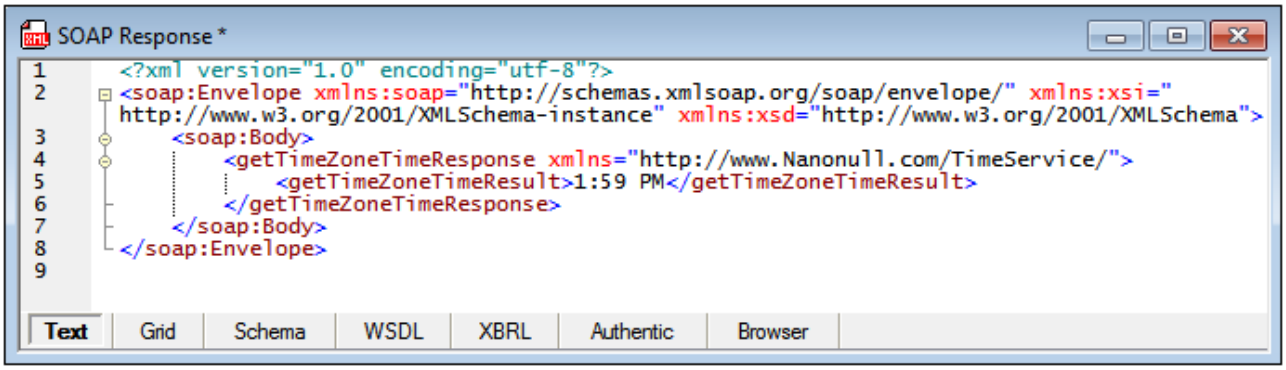
Détecter l'erreur et tester une réparation

Le débogage a été lancé comme décrit dans la section précédente, Déboguer. La demande SOAP pour la sélection GMT apparaît dans la fenêtre de demande SOAP du débogueur, dans le mode Texte. Examinons ensemble cette demande et éditer les erreurs qu'elle peut contenir.



En regardant l'élément `timezone`, nous remarquons que la valeur est `GMD`. Cela est incorrect, nous allons donc la changer en `GMT`. Pour ce faire, double-cliquer dans l'élément `timezone` et changer les contenus de l'élément en `GMT`.

Pour tester la réparation, cliquer sur l'icône **GO** dans la barre outils du Débogueur SOAP (ou utiliser la commande de menu **SOAP | GO**) pour envoyer la demande correcte au service web. Au bout de quelques secondes, la réponse du service web à la demande SOAP apparaît dans la fenêtre de réponse SOAP. Choisir **Mode | Retour à la ligne** pour voir toute la réponse SOAP (*capture d'écran ci-dessous*).



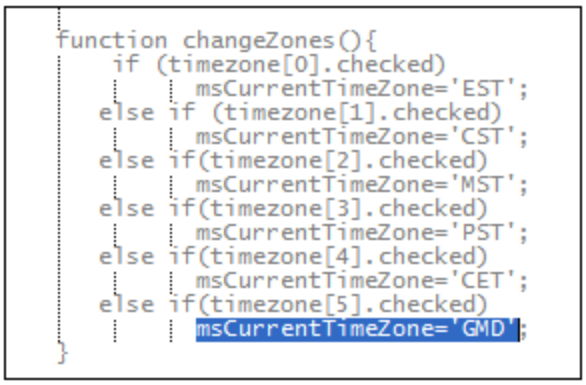
À présent, passer à l'onglet `DebuggerClient.htm`, et cliquer sur l'icône **GO** dans la barre d'outils Débogueur SOAP. Le message d'erreur disparaît et l'heure GMT correcte est affichée (*capture d'écran ci-dessous*).



Vous pouvez fermer la session du Débogueur SOAP en choisissant la commande de menu **SOAP | Session du Débogueur SOAP**.

Réparer l'erreur

Maintenant nous savons qu'une valeur invalide de `GMT` est générée dans la demande SOAP. Si nous examinons le fichier de point d'entrée de la requête SOAP et que nous exécutons une recherche de `GMD` (par le biais du dialogue Recherche, **Ctrl+F** ou **Édition | Recherche**), nous trouvons la coquille dans le fragment de code affiché dans la capture d'écran ci-dessous.



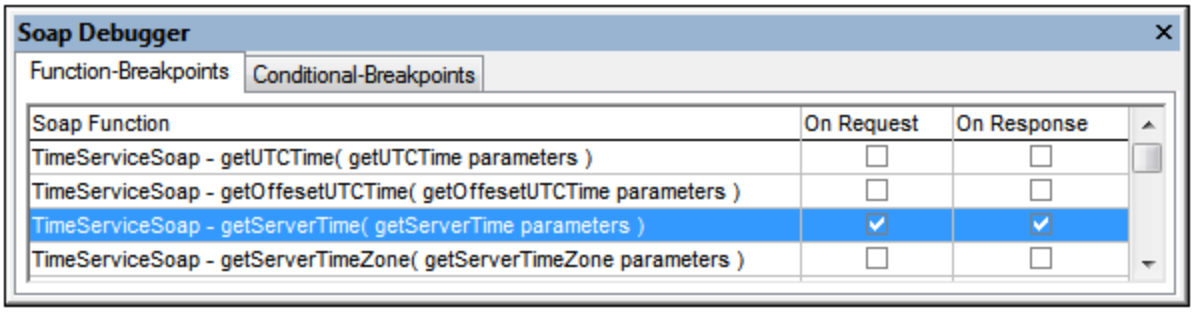
```
function changeZones(){
    if (timezone[0].checked)
        msCurrentTimeZone='EST';
    else if (timezone[1].checked)
        msCurrentTimeZone='CST';
    else if(timezone[2].checked)
        msCurrentTimeZone='MST';
    else if(timezone[3].checked)
        msCurrentTimeZone='PST';
    else if(timezone[4].checked)
        msCurrentTimeZone='CET';
    else if(timezone[5].checked)
        msCurrentTimeZone='GMD';
}
```

Une fois que cette erreur est corrigée et que le bouton radio GMT est sélectionné, l'erreur *Fuseau horaire inconnu* n'est plus affichée. L'heure GMT correcte est affichée à la place.

15.2.2.8 En savoir plus sur les points d'arrêt

La fenêtre du Débogueur SOAP est l'endroit où vous pouvez définir et supprimer les points d'arrêt. Elle est séparée en deux onglets (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



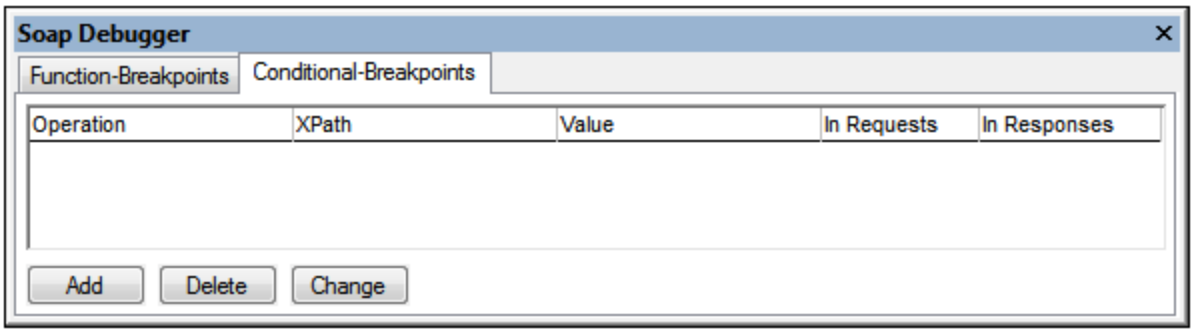
Onglet Fonction-Points d'arrêt

L'onglet Fonction-Points d'arrêt vous permet de définir un point d'arrêt sur les Requêtes et/ou les Réponses dans les méthodes SOAP. Le débogueur marque la fonction qui a déclenché le point d'arrêt. Les paquets de données envoyés de et vers le client sont analysées et mis en relation avec les fonctions correspondantes du fichier WSDL. Si un point d'arrêt est défini pour une méthode spécifique, c'est là que le Débogueur SOAP stoppe. Les boutons de la barre d'outils sont activés.

Les données sont affichées dans la fenêtre de document Requête SOAP ou Réponse SOAP. Les documents SOAP affichés dans les fenêtres SOAP peuvent être modifiés à ce moment. Les données sont envoyées au moment où vous cliquez sur une des icônes de la barre d'outils (sauf pour l'icône Arrêter Serveur).

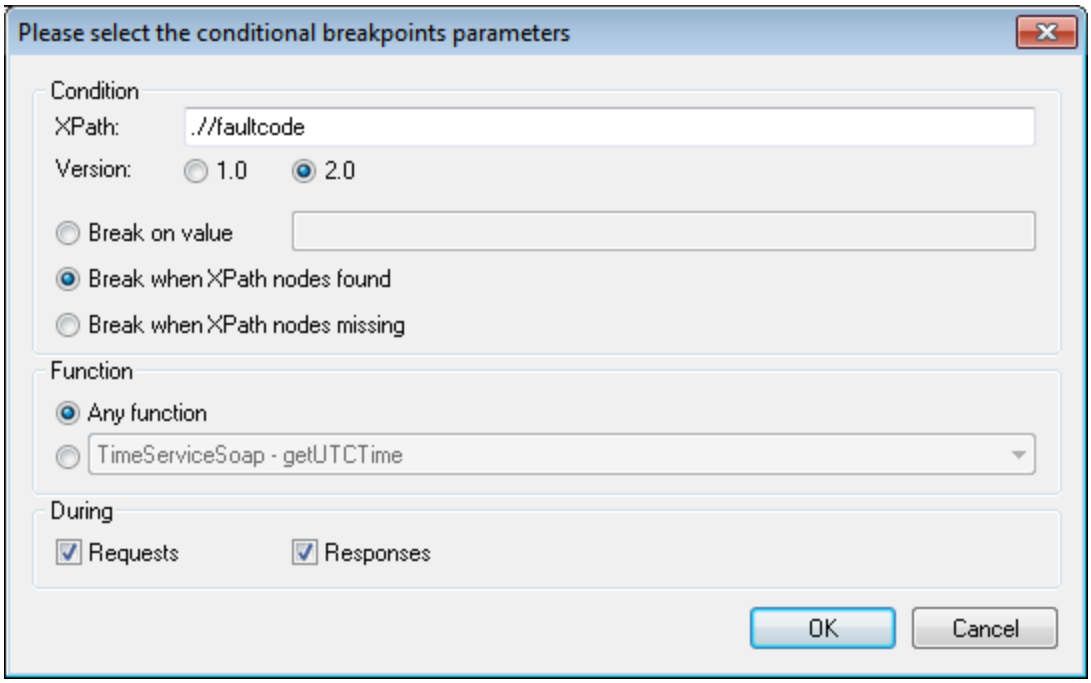
Points d'arrêt conditionnels

L'onglet Points d'arrêt conditionnels (*capture d'écran ci-dessous*) vous permet d'utiliser des expressions XPath pour définir des points d'arrêt. Si une requête SOAP entraîne une erreur, la réponse SOAP doit contenir un élément `faultcode`. C'est pourquoi nous aimerions déclencher un point d'arrêt à chaque fois qu'un élément `faultcode` apparaît.



Pour ajouter un point d'arrêt conditionnel, procéder comme suit :

1. Cliquer sur l'onglet Points d'arrêt conditionnels, puis sur la touche **Ajouter**. Le dialogue suivant apparaît.



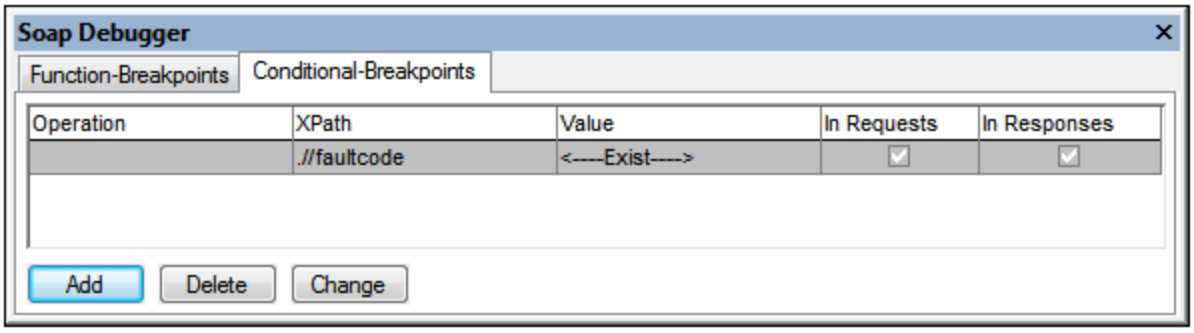
2. Saisir l'expression XPath (par exemple, `.//faultcode`) dans le champ .
3. Choisir la version de XPath requise (1.0 ou 2.0) et le bouton radio *Arrêt quand des nœuds XPath sont trouvés*.
4. Cliquer sur **OK** pour confirmer les paramètres. Le débogueur SOAP s'arrêtera à chaque fois qu'un élément `.//faultcode` apparaît dans une requête ou une réponse SOAP.

Les différentes options dans ce dialogue sont décrites ci-dessous :

- *Champ d'expression XPath :* saisir ici l'expression/le nœud XPath spécifique. Un XPath doit être saisi ici pour pouvoir utiliser une des options du bouton radio spécifique.
- *Version :* la version XPath que vous souhaitez utiliser pour l'expression XPath.
- *Boutons radio d'instruction Arrêt :* les débogueurs s'arrêtent lorsque l'option sélectionnée se produit. Les options disponibles sont : (i) Arrêt lorsque le nœud XPath ciblé correspond à la valeur saisie dans ce champ ; (ii) Arrêt lorsque le nœud XPath spécifié existe dans la requête ou la réponse SOAP ; et (iii) Arrêt lorsque le nœud spécifié n'existe pas dans la requête ou la réponse SOAP.
- *Requêtes et Réponses :* spécifie si les options dans le dialogue doivent aussi être appliquées dans les réponses et/ou les requêtes SOAP.
- *Fonctions :* soit toutes les méthodes/fonctions sont scannées pour la condition que vous définissez (bouton radio *Toute fonction*) soit vous saisissez une méthode/spécification spécifique à scanner.

Pour la condition définie dans le dialogue affiché ci-dessus, le point d'arrêt conditionnel suivant sera recensé dans l'onglet Points d'arrêt conditionnels.

© 2016-2022 Altova GmbH



Ci-dessous, vous trouverez une description des colonnes de cet onglet :

- La colonne *Opération* contient la méthode/fonction recherchée. Si vous sélectionnez le bouton radio *Toute fonction,* ce champ restera vide. Si vous sélectionnez une méthode/fonction spécifique, cette méthode/fonction sera affichée ici.
- La colonne *XPath* contient l'expression XPath que vous avez définie.
- La colonne *Value* contient la valeur XPath par rapport à laquelle les nœuds retournés sont cochés pour une correspondance. Si vous avez choisi *Arrêt sur valeur*, la chaîne spécifique que vous avez saisie est affiché ici. Si vous avez choisi *Arrêt quand des nœuds XPath sont trouvés,* alors `<--Exist-->` est affiché. Si vous avez choisi *Arrêt quand des nœuds XPath manquent*, alors `<--Missing-->` est affiché.
- Les cases à cocher *Dans Requêtes* et *Dans Réponses* indiquent où la condition est cochée. Vous pouvez changer les paramètres en cliquant directement sur la case à cocher dans la colonne.

Pour éditer un point d'arrêt conditionnel, double-cliquer sur sa ligne dans l'onglet ou cliquer sur le bouton **Changer** (*voir capture d'écran ci-dessus*). Pour supprimer un point d'arrêt conditionnel, sélectionner la ligne que vous souhaitez supprimer et cliquer sur **Supprimer**.

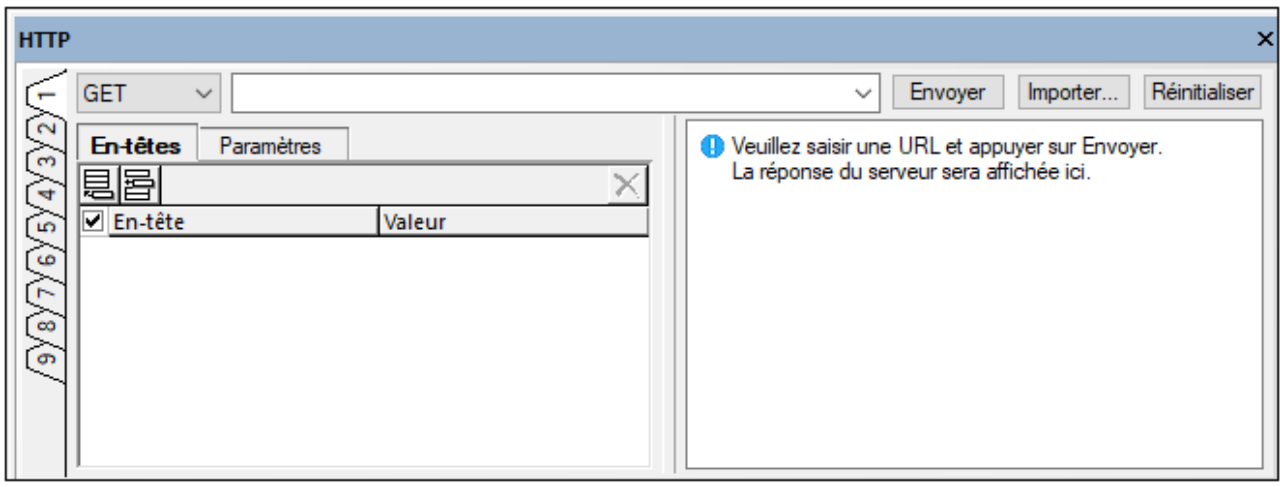
16 HTTP

Le protocole HTTP (Hypertext Transfer Protocol) est le protocole (ou l'ensemble de règles) qui définit comment les fichiers (texte, images, audio, vidéo, et autres fichiers multiculturels) sont transmis sur Internet. Chaque serveur web exécute un programme (aussi appelé daemon) qui attend en continu les requêtes HTTP et les gère lorsqu'ils arrivent. Par exemple, lorsque vous visitez la page d'accueil un site web, votre navigateur envoie une commande HTTP au serveur web de la page web demandant le téléchargement de la page d'accueil ; le daemon HTTP reçoit la demande et envoie la page requise. Une propriété significative de HTTP est remarquable : elle *n'a pas d'état*, ce qui signifie que chaque commande HTTP est effectuée indépendamment, sans aucune référence aux commandes précédentes ou suivantes.

Dans XMLSpy, vous pouvez tester les commandes HTTP dans la fenêtre de sortie HTTP (*capture d'écran ci-dessous*). Ici, vous pouvez créer et envoyer une requête HTTP sur un serveur web, et recevoir et cocher la réponse.

Parties de la fenêtre de sortie HTTP

La fenêtre de sortie HTTP a neuf onglets (*voir la capture d'écran ci-dessous*). Vous pouvez stocker une requête séparée dans chaque onglet et passer d'un onglet à l'autre. Après avoir créé une requête dans la fenêtre, vous pouvez envoyer la requête en cliquant sur le bouton **Envoyer**. La réponse est affichée directement dans la fenêtre.



La fenêtre consiste en les parties suivantes :

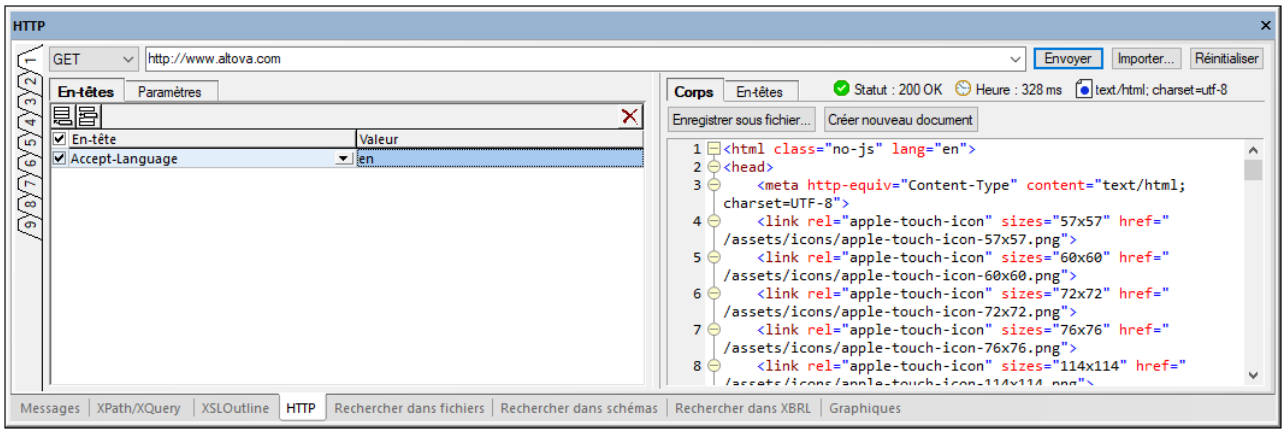
- En haut : (i) une zone de liste dans laquelle vous sélectionnez la méthode HTTP que vous souhaitez utiliser ; (ii) un champ de saisie pour l'URL sur le serveur web ; (iii) des touches liées à l'exécution de requêtes HTTP (**Envoyer**, **Importer** et **Réinitialiser**).
- Un volet de gauche pour créer la requête.
- Un volet de droite pour afficher la réponse.

Vous trouverez une description de la fenêtre de sortie HTTP dans les sous-sections de cette section.

© 2016-2022 Altova GmbH

16.1 Envoyer la requête

Vous pouvez envoyer la requête HTTP dans la fenêtre de sortie HTTP (*capture d'écran ci-dessous*). Une requête est définie dans le **volet de gauche de la fenêtre**. Vous pouvez définir une requête différente pour les neuf onglets de la fenêtre, chaque requête consistant en : (i) la méthode HTTP de la requête et l'URL cible ; (ii) les en-têtes HTTP de la requête (dans l'onglet *en-têtes*) ; (iii) les paramètres de connexion (dans les *Paramètres*) ; et (iv) dans le cas des méthodes POST et PUT, le corps de message HTTP (dans l'onglet *Corps* ; pas affiché dans la capture d'écran ci-dessus). Vous pouvez vider les onglets en cliquant sur **Réinitialiser** (situé en haut à droite de la fenêtre).



Pour envoyer une requête HTTP, procéder comme suit :

1. Dans la liste de choix en haut à gauche (*voir capture d'écran*) choisir une méthode HTTP (**GET**, **POST**, **PUT**, **DELETE**, **HEAD**, ou **OPTIONS**).
2. Saisissez l'URL de la page web cible (par exemple, **http://www.altova.com**. Vous pouvez aussi saisir uniquement **altova.com**; la partie **http://** de l'URL sera complétée pour vous).
3. Dans l'onglet *En-têtes*, vous pouvez préciser les valeurs d'en-tête HTTP (*voir la capture d'écran ci-dessus*). Vous pouvez sélectionner ou saisir une en-tête, puis saisir sa valeur. (Pour une liste d'en-têtes HTTP 1.1, voir ici.) Utilisez les icônes **Insérer**, **Ajouter**, et **Supprimer** de la barre d'outils de l'onglet pour ajouter ou supprimer les en-têtes. Au lieu de supprimer une en-tête, vous pouvez désactiver une en-tête en décochant l'option *Activer* à gauche du nom d'en-tête ; cela vous évitera de devoir ressaisir une en-tête supprimée si vous souhaitez la réutiliser plus tard. Voir aussi la section l'en-tête Accepter. (Si vous définissez une valeur pour toute en-tête qui serait ajoutée automatiquement lors de send-time, alors la valeur que vous saisissez sera utilisée à la place de la valeur qui aurait été ajoutée automatiquement.)
4. Si vous envoyez une requête **POST** ou **PUT**, un onglet *Corps* devient disponible en plus des onglets *En-têtes* et *Paramètres*. Vous trouverez une description de la création du corps **POST** ou **PUT** dans la section Le corps des requêtes POST et PUT ci-dessous.
5. Vous pouvez spécifier des délais de temporisation et des paramètres de sécurité dans l'onglet *Paramètres*. Pour une description de cet onglet, voir la section Paramètres pour la requête HTTP ci-dessous.
6. Cliquez sur **Envoyer** (*situé en haut à droite de la fenêtre*) pour envoyer la requête.
7. Si vous souhaitez rétablir l'état vide de l'onglet, cliquez sur **Réinitialiser**. Voilà ce qui se produit : (i) la méthode à utiliser est de réinitialiser à la première méthode dans la liste déroulante de la liste de choix (qui est **GET**); (ii) l'entrée d'URL actuelle est retirée ; (iii) toutes les en-têtes et les définitions de paramètres sont retirées.

Note : Vous pouvez également importer une requête d'un fichier WADL dans une fenêtre de sortie HTTP via le bouton **Importer**.

Note : La requête est envoyée dans un encodage UTF-8. Tout autre encodage est converti en UTF-8, et les données UTF-8 sont envoyées.

Méthodes HTTP

Les méthodes HTTP suivantes sont prises en charge :

GET
La méthode **GET** demande la ressource située dans l'URL spécifiée. Vous pouvez aussi ajouter une requête à l'URL ; par exemple : `http://www.altova.com?name1=value1&name2=value2`. La ressource est retournée dans un message qui contient une en-tête et un corps.

HEAD
La méthode **HEAD** est identique à la requête **GET**, mais ne retourne aucun corps de message, uniquement une tête de message contenant des méta-informations concernant la ressource localisée sous l'URL spécifiée.

POST
La méthode **POST** est utilisée pour mettre à jour une ressource existante située sous l'URL spécifiée, ou pour créer une nouvelle ressource sous l'URL spécifiée. Les données à soumettre à la ressource sont placées dans le corps de la requête HTTP ; voir *Le corps des requêtes POST et PUT* pour des informations concernant cette procédure.

PUT
La méthode **PUT** est utilisée pour créer une nouvelle ressource sous l'URL spécifiée. Les données à soumettre à la ressource sont placées dans le corps de la requête HTTP ; voir *Le corps des requêtes POST et PUT* pour des informations concernant cette procédure.

DELETE
La méthode **DELETE** supprimer la ressource située sous l'URL spécifiée.

OPTIONS
La méthode **OPTIONS** retourne une liste de méthodes HTTP que le serveur prend en charge.

Le corps des requêtes POST et PUT

Pour les requêtes **POST** et **PUT**, un onglet additionnel *Corps* devient disponible, dans lequel la requête **POST** ou **PUT** peut être spécifiée (*voir la capture d'écran ci-dessous*). L'onglet *Corps* a deux modes: **Mode Éditeur** et **Mode Fichier**. Vous pouvez basculer entre ces deux modes via les boutons de la barre d'outils en haut à gauche de l'onglet *Corps* (*voir la capture d'écran*). Dans le Mode Éditeur (*affiché sélectionné dans la capture d'écran ci-dessous*), vous pouvez éditer la requête HTTP directement dans le volet, alors que dans le mode fichier, vous pouvez sélectionner un fichier qui contient la requête HTTP.

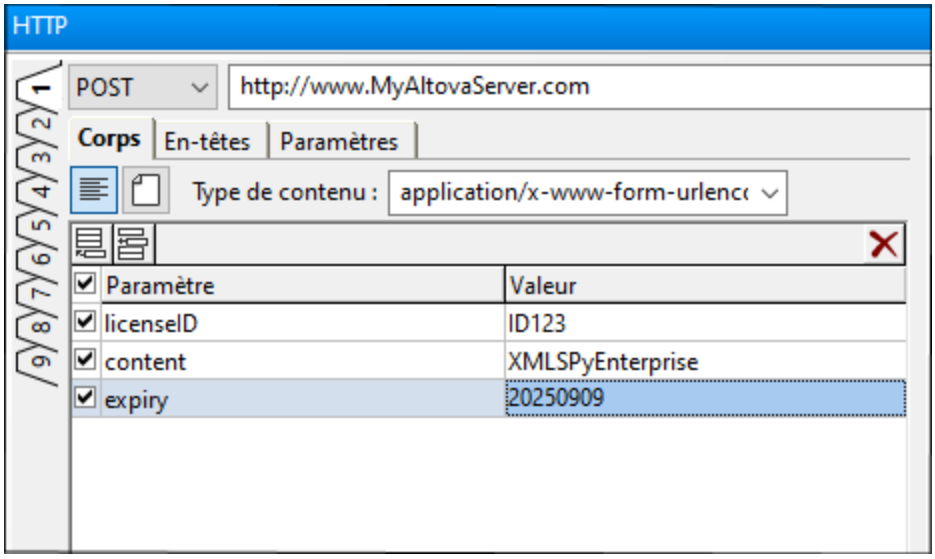
Le champ *Type de contenu* vous permet de spécifier l'en-tête **Content-Type** dans la requête. Les options de liste de choix de ce champ sont différentes pour chaque mode (Éditeur et Fichier). Vous pouvez sélectionner parmi les options disponibles dans la liste de choix ou saisir un type MIME. Veuillez noter que la valeur spécifiée dans ce champ ne peut pas être contournée par une en-tête **Content-Type** dans l'onglet *En-têtes* ou l'onglet *Corps*.

© 2016-2022 Altova GmbH

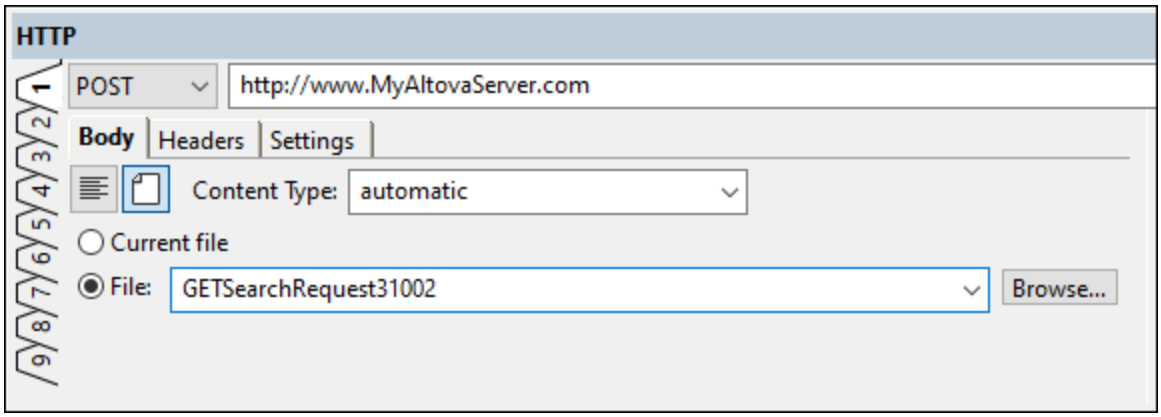


Mode Éditeur
Le style de l'éditeur dépend du type de contenu sélectionné :

- Pour les types de contenu **`text/plain`**, **`text/xml`**, **`application/xml`** et **`application/json`** : Un éditeur de texte qui propose des fonctions d'édition intelligentes telles que la coloration de la syntaxe pour des documents XML et JSON et line-numbering. La capture d'écran ci-dessus montre l'éditeur pour le type de contenu `text/xml`. Le corps de la requête est saisi dans l'éditeur. Le type-contenu de la requête est spécifié dans le champ *Type de contenu* et ne peut pas être contourné par des entrées ailleurs dans la requête.
- Pour le type de contenu **`application/x-www-form-urlencoded`** : L'éditeur est un mode Grille (*capture d'écran ci-dessous*) dans lequel chaque nouvelle ligne représente une paire name–value dans le corps de la requête.



Mode Fichier
Dans le mode Fichier (*capture d'écran ci-dessous*), le corps de la requête sera les contenus du fichier sélectionné. Ce fichier peut être soit le fichier actif actuellement dans la Fenêtre principale (option *Fichier actuel*) ou un fichier externe qui sera cherché.



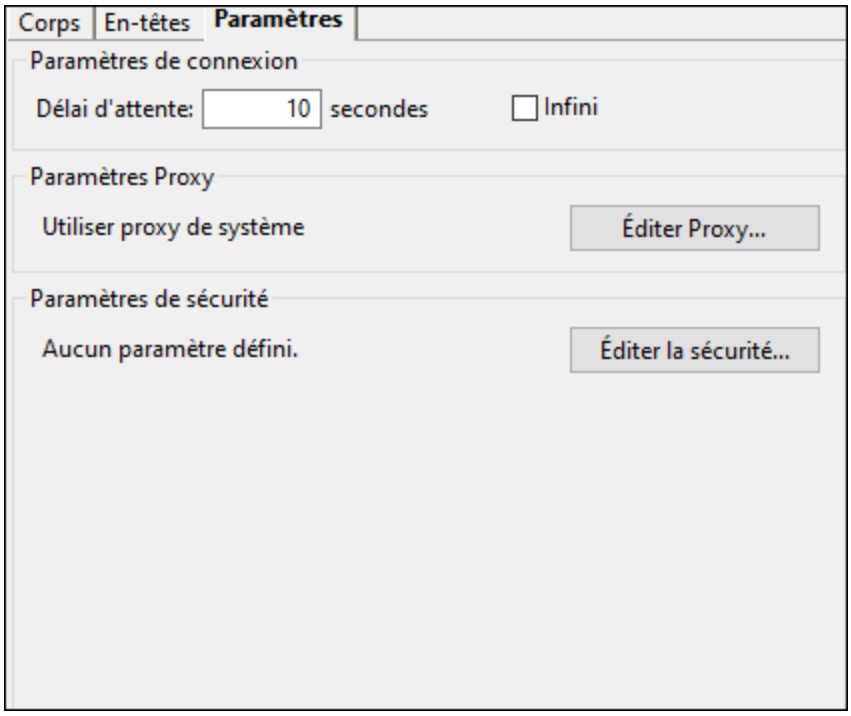
Pour passer au mode Fichier, sélectionner l'icône **Fichier** située près de la partie supérieure à gauche de l'onglet *Corps* (*voir capture d'écran ci-dessous*). Pour activer le type de contenu du corps à déterminer automatiquement, sélectionnez `automatic` dans le champ *Type de contenu*. La détermination automatique du type de contenu est basée sur l'extension du fichier. Si vous saisissez un type de contenu, la requête sera envoyée avec le type de contenu que vous saisissez ; dans ce cas, vous devez vous assurer que le type de contenu est le type correct.

Les en-têtes et les paramètres de la requête peuvent être spécifiés de la même manière que pour les autres requêtes (c'est-à -dire, dans l'onglet *En-têtes* et l'onglet *Paramètres*, respectivement).

Paramètres pour la requête HTTP

L'onglet *Paramètres* de la fenêtre de sortie HTTP (*capture d'écran ci-dessous*) vous permet de définir (i) un timeout de la connexion, et (ii) les paramètres de sécurité d'une requête. Notez que vous pouvez définir les paramètres séparément pour chacun des neuf onglets de la fenêtre. La capture d'écran ci-dessous montre les paramètres pour une URL HTTPS.

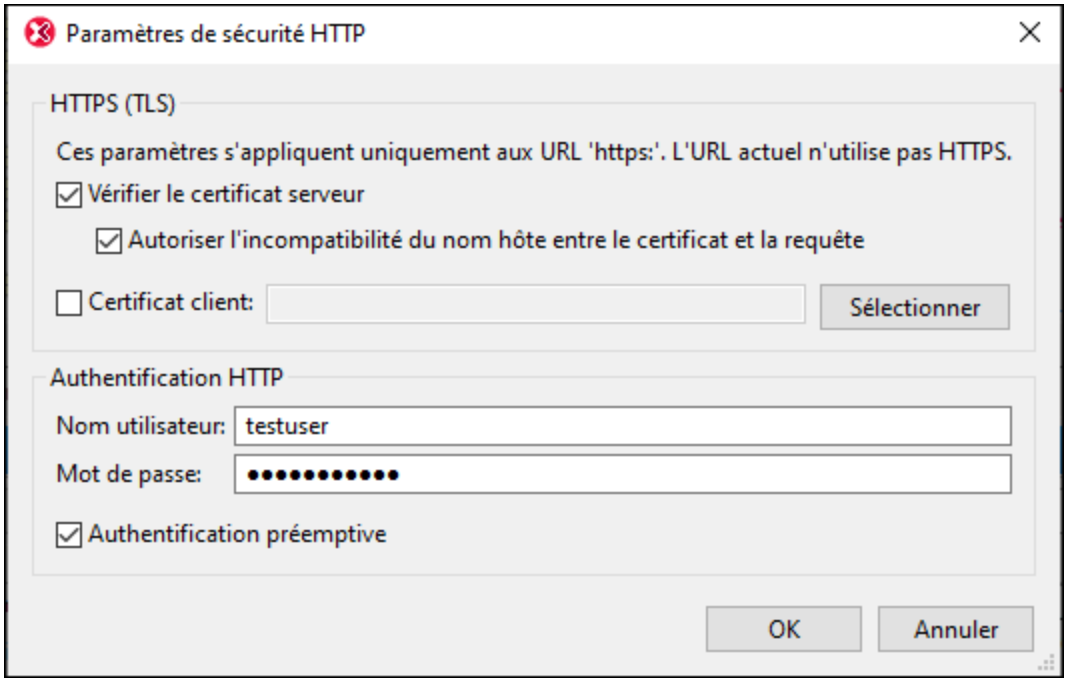
© 2016-2022 Altova GmbH



Paramètres de connexion
Vous pouvez spécifier la durée de temps en secondes pendant laquelle XMLSpy tentera d'établir une connexion avec le serveur web. Une fois que cette durée a été atteinte, sans qu'une connexion ait pu être établie, vous obtiendrez `I/O Error 28: Timeout was reached`. Si vous ne souhaitez pas spécifier une période de timeout, cochez la case *Infini*.

Paramètres Proxy
Fournit un résumé des paramètres de proxy actuels et un bouton pour ouvrir la section des paramètres Proxy du dilaogue des Options.

Paramètres de sécurité
Cliquez sur **Éditer** pour éditer les paramètres de sécurité d'une requête. Le dialogue Paramètres de sécurité HTTP (*capture d'écran ci-dessous*) sera affiché. Ici, vous pouvez spécifier les paramètres de sécurité HTTPS et définir les identifiants d'authentification HTTP pour que la requête soit effectuée par le biais de cet onglet. Si le serveur web de cible de la requête ne peut pas utiliser SSL, alors seuls les identifiants d'authentification HTTP seront utilisés. Si le serveur web de cible utilise SSL, alors les paramètres de sécurité HTTPS et les identifiants d'authentification HTTP seront utilisés.



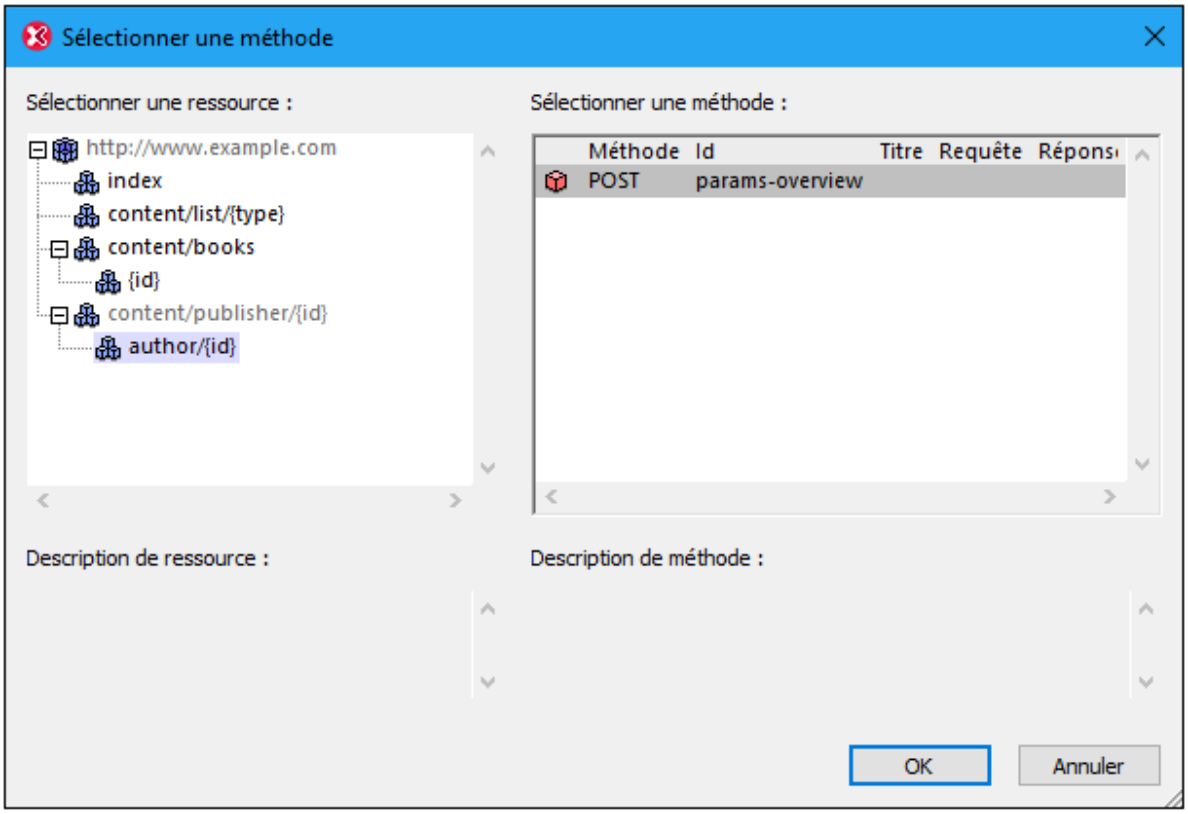
- *Paramètres de sécurité HTTPS :* Par défaut, l'option *Contrôler le certificat de serveur* sera cochée et vous pouvez spécifier si le nom d'hôte dans la requête peut être différent du nom d'hôte dans le certificat. Si vous ciblez une URL d'Intranet (par exemple, dans votre réseau d'entreprise), alors un certificat de client (généralement situé dans votre boutique de certificat local) peut être utilisé pour vérifier un certificat sur le serveur de l'Intranet.
- *Authentification HTTP:* Certaines requêtes vers un serveur pourraient requérir une authentification de l'utilisateur. Dans ces cas, vous pouvez saisir ici un nom d'utilisateur et un mot de passe. Lorsque l'authentification est requise par le serveur, elle sera fournie automatiquement. Sinon, vous serez invité à la saisir une fois que la connexion au serveur est établie. Lorsque la requête initiale au serveur contient les informations authentification, ce processus est appelé une authentification préemptive. Si cela est exigé par le serveur, sélectionner l'option *Authentification préemptive*.

© 2016-2022 Altova GmbH

16.2 Importer une requête à envoyer

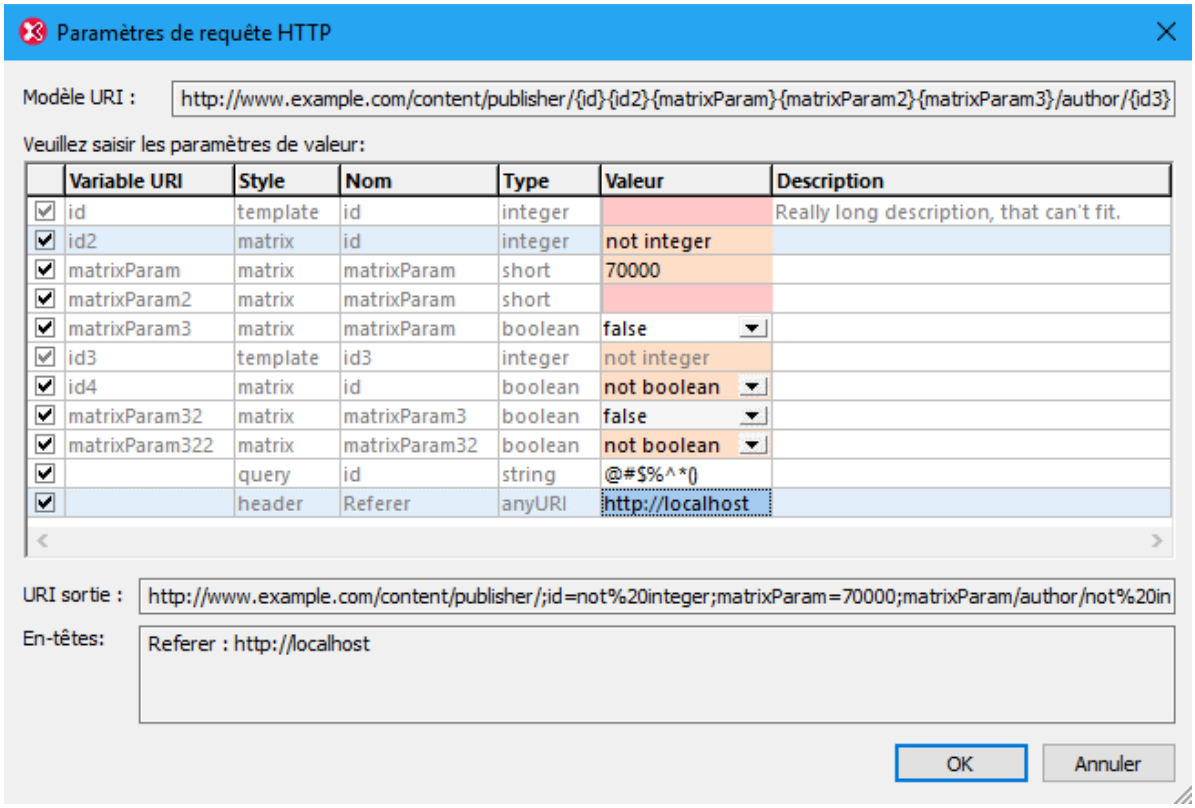
Dans la fenêtre de sortie HTTP, vous pouvez importer une requête provenant d'un fichier de Web Application Development Language (WADL), et puis l'envoyer. Pour ce faire, utiliser l'Assistant d'importation WADL de XMLSpy. L'Assistant ouvre un fichier WADL, sélectionne une requête depuis une des ressources du fichier, vous permet de modifier les paramètres éditables de la requête et puis de l'importer dans la fenêtre de sortie HTTP. Pour ce faire, procéder comme suit :

1. Dans la fenêtre de sortie HTTP, cliquer sur la touche **Importer** pour lancer l'Assistant d'importation WADL.
2. Dans le dialogue de sélection de fichiers qui apparaît, chercher le fichier WADL qui contient la requête que vous souhaitez importer, puis cliquer sur **OK**. Cela lance l'Assistant Importer WADL (*capture d'écran ci-dessous*).



3. Dans le panneau de gauche (*voir capture d'écran ci-dessus*), choisir la ressource pertinente (celle contenant la requête que vous souhaitez importer).
4. Dans le panneau de droite (*voir capture d'écran ci-dessus*), choisir la requête (l'opération WSDL ou la méthode WADL) que vous souhaitez importer. Veuillez noter que, pour l'importation via WSDL : (i) l'importation HTTP est fourni uniquement pour les extensibilités pour SOAP et HTTP et (ii) seules les liaisons prises en charge (SOAP et HTTP) sont affichés dans ce volet (de droite).
5. Cliquer sur **OK**. Si la requête contient un ou plusieurs paramètres, l'écran suivant de l'assistant (*capture d'écran ci-dessous*) affiche les paramètres de la requête que vous avez choisie. Sinon, la requête est importée dans la fenêtre de sortie HTTP et l'assistant se ferme ; *voir point 7 ci-dessous.* Les paramètres font partie de la requête. Par exemple, dans une requête de recherche, un paramètre peut être le terme de recherche. L'assistant valide une valeur de paramètre par rapport à son type de

données et indique un des trois états. Un arrière-plan rose indique une valeur invalide ; un arrière-plan beige indique une valeur incorrecte qui sera tout de même saisie dans la requête et envoyée, un arrière-plan blanc indique une valeur valide.

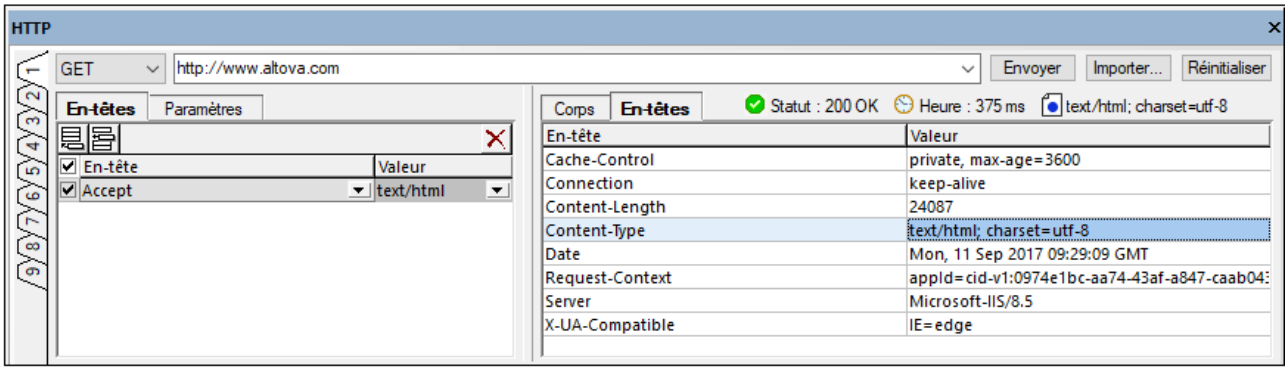


6. Dans l'écran Paramètres de requête HTTP (*capture d'écran ci-dessus*), saisir ou éditer les valeurs de paramètre selon vos besoins (dans la colonne *Valeurs*). Veuillez noter qu'il se peut que vous ne soyez pas autorisé à éditer certaines des valeurs de paramètres ; les cellules de ces valeurs sont désactivées pour l'édition. Si vous ne souhaitez pas utiliser un paramètre, le désactiver en décochant sa case *Activer* (dans la première colonne). Veuillez noter que certains paramètres sont obligatoires, c'est pourquoi la case à cocher *Activer* sera verrouillée et vous ne pourrez pas la décocher. Veuillez noter que, alors que vous éditez les valeurs de paramètre, la requête est construite dans le champ *URI de sortie*. Les en-têtes dans le champ *En-tête,* en bas de la fenêtre proviennent du fichier WADL ; ils peuvent être édités si cela est autorisé par les définitions dans le fichier WADL.
7. Cliquer sur **OK**. La requête est importée dans la fenêtre de sortie HTTP, et est affichée ici de la manière suivante : (i) dans la liste de choix des méthodes, la méthode HTTP de la requête sera affichée ; (ii) l'URL sera construite sur la base des paramètres de la requête ; (ii) les en-têtes HTTP de la requête seront saisies dans l'onglet *En-têtes* ; (iv) si la requête est une requête `POST` ou `PUT`, le corps de la requête ne sera pas saisi dans l'onglet *Corps*, il devra être ajouté manuellement.
8. Vérifier l'onglet *Paramètres* pour voir si vous devez modifier les paramètres.
9. Cliquer sur **Envoyer** pour envoyer la requête.

© 2016-2022 Altova GmbH

16.3 Recevoir la réponse

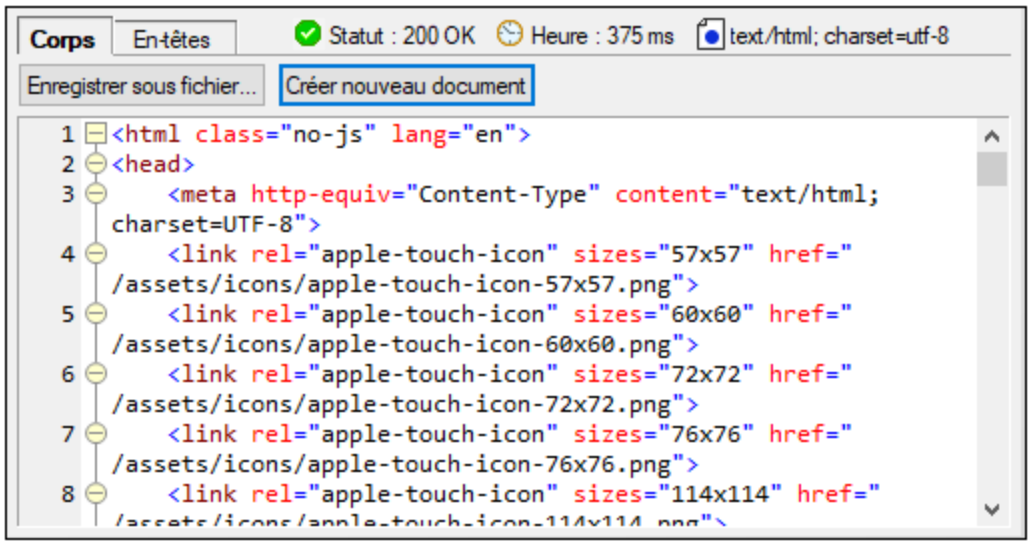
La réponse d'une requête HTTP est reçue dans le panneau de droite de la fenêtre de sortie HTTP (*voir capture d'écran ci-dessous*).

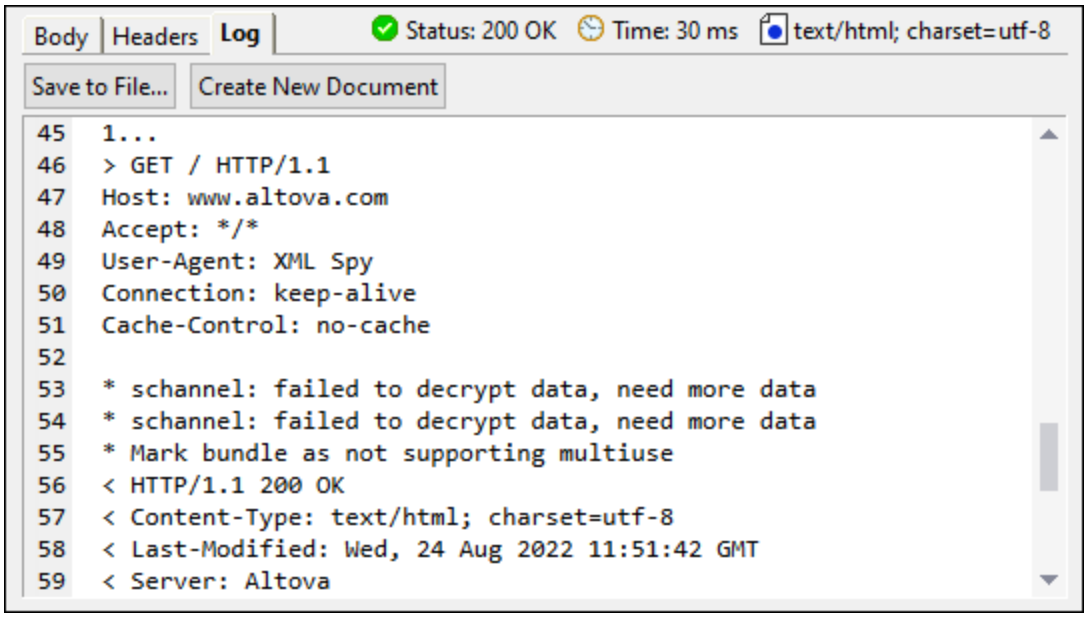


À la droite des onglets *Corps, En-têtes,* et *Log*, vous trouverez les détails suivants concernant la réponse :

- Le code de statut HTTP (expliqué ci-dessous)
- L'heure de la connexion établie jusqu'à la dernière partie de réponse reçue
- Le `Content-Type` de la réponse

Le volet Réponse a trois onglets : *Corps* (*capture d'écran ci-dessous gauche*), *En-têtes* (*capture d'écran ci-dessus*), et Journal (*capture d'écran ci-dessous droite*).





- *Onglet Corps :* Le corps de la réponse est affiché avec la coloration de la syntaxe si le document est HTML, XML ou JSON et avec line-numbering. L'onglet a deux boutons : (i) **Enregistrer sous fichier** pour enregistrer le corps sous un fichier et (ii) **Créer nouveau document** pour créer un nouveau document dans XMLSpy et afficher le document récemment créé dans la fenêtre principale de la GUI. Un document nouvellement créé peut être édité et enregistré de la manière usuelle. Si un nouveau document ne peut pas être créé depuis le corps de la réponse (par exemple si le corps est une image), alors le bouton **Créer nouveau document** sera désactivé (*voir la capture d'écran un peu plus bas*).
- *Onglet En-têtes:* Contient les en-têtes de la réponse. L'en-tête `Content-Type` est aussi affichée en haut du volet.
- *Onglet Journal :* Les événements et informations liés à la requête sont affichés dans l'onglet Journal qui a le line-numbering. Le journal peut être enregistré au fichier ou créé comme nouveau document dans la fenêtre principale. Un document nouvellement créé peut être édité et enregistré de la manière usuelle.

© 2016-2022 Altova GmbH

Codes de statut HTTP

Les codes de statut sont catégorisés comme suit :

- Les codes 2XX sont utilisés pour les requêtes réussies.
- Les codes 3XX sont utilisés pour les redirections.
- Les codes 4XX sont utilisés s'il y a eu un problème avec la requête.
- Les codes 5XX sont utilisés s'il y a eu un problème avec le serveur.

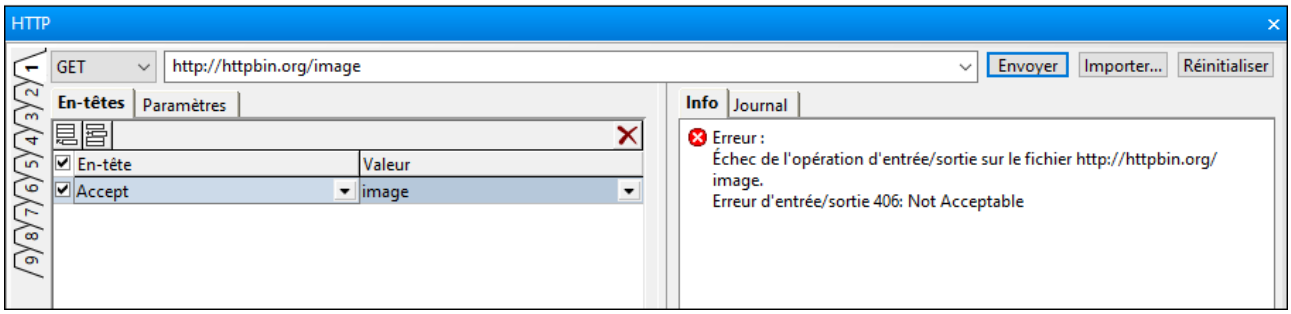
Certains des codes fréquemment rencontrés :

- *200 OK*: Envoyé en réponse à une requête réussie.
- *206 Contenu partial* : Le serveur envoie uniquement une partie de la ressource parce que seule une partie de la ressource a été requise.
- *301 Déplacé définitivement* : La requête doit être redirigée vers l'URL donnée.
- *401 Non-autorisé* : La ressource nécessite une authentification, et celle-ci a soit échoué soit n'a pas été fournie.
- *403 Interdit* : Requête valide, mais le serveur refuse l'action. Cela est peut-être dû au fait que l'utilisateur n'a pas les identifiants nécessaires.
- *404 Non trouvé* : La ressource n'a pas pu être trouvée.
- *500 Erreur de serveur interne* : Un message d'erreur générique, envoyé lorsqu'aucun message plus spécifique n'est disponible.

Voir : Liste complète de codes de statut HTTP.

L'en-tête Accepter

L'en-tête Accepter de la requête spécifie le type de contenu pour accepter dans la réponse. Par exemple, voir la différence entre les réponses quand **`Accept=image`** (*première capture d'écran ci-dessous*) et quand **`Accept=image/png`** (*deuxième capture d'écran ci-dessous*). Dans le premier cas, puisqu'il n'est pas précisé quel format d'image devrait être envoyé dans la réponse, un code de statut d'erreur et un message JSON contenant plus d'information sont envoyés.





© 2016-2022 Altova GmbH

17 XBRL

Le Mode XBRL de XMLSpy est un éditeur de taxonomie XBRL qui permet un aperçu graphique des taxonomies XBRL et offre des fonctions d'édition de taxonomie intelligentes. Dans cette section, nous décrivons les différentes fonctions du Mode XBRL, et comment créer et éditer des taxonomies dans le Mode XBRL.

Cette section est organisée comme suit :

- Gestionnaire de taxonomie, qui décrit comment utiliser l'outil Gestionnaire de taxonomie pour installer, mettre à niveau et gérer des taxonomies à utiliser avec XMLSpy.
- Procédures de base, qui décrit comment créer des taxonomies contenant les composants essentiels.
- Procédures supplémentaires, qui décrit les fonctions supplémentaires comme la manière de travailler avec des libellés préférés et des faits dupliqués.
- Éditeur de Formule XBRL, qui décrit comment utiliser le Mode XBRL pour travailler avec les formules XBRL.
- Éditeur de définitions de table XBRL, qui décrit la structure de table, comment utiliser l'éditeur pour définir les tables XBRL, et comment fonctionne l'Aperçu de mise en page de table. Cette section explique aussi comment utiliser les paramètres de table, y compris comment les paramètres de table sont utilisés avec les ensembles de table.
- Chercher dans XBRL, qui décrit les puissantes fonctions de recherche spécifiques à XBRL de XMLSpy.
- OIM, qui fournit un aperçu de fonctions OIM dans XMLSpy.
- Notes concernant la validation des instances et taxonomies XBRL.

Pour plus d'informations, voir les sections : Modes Édition | Mode XBRL et la description des commandes dans le menu XBRL. Par exemple, vous trouverez des informations concernant la génération de documentation pour la taxonomie (tel que vu dans le Mode XBRL) dans la section Commandes de menu | Menu XBRL | Générer documentation.

Les Signatures XML pour les fichiers XBRL dans le Mode XBRL peuvent être créées en tant que fichiers de signature externes. Vous trouverez les instructions pour travailler avec des signatures dans la section Signatures XML.

Prise en charge dans XMLSpy pour US-GAAP et les autres taxonomies
XMLSpy prend en charge les taxonomies suivantes :
- US-GAAP 1.0, 2005, 2008, 2009, 2011 to 2022
- IFRS

Les dernières versions de US-GAAP sont installées avec XMLSpy. Les taxonomies supplémentaires, y compris les taxonomies US-GAAP plus anciennes, sont disponibles pour une installation via un programme d'installation de taxonomie que vous pouvez télécharger gratuitement depuis le site web Altova.

Certification XBRL

XMLSpy a été certifié XBRL par XBRL International. Pour plus d'informations à propos de certification XBRL, voir Certification XBRL Software.

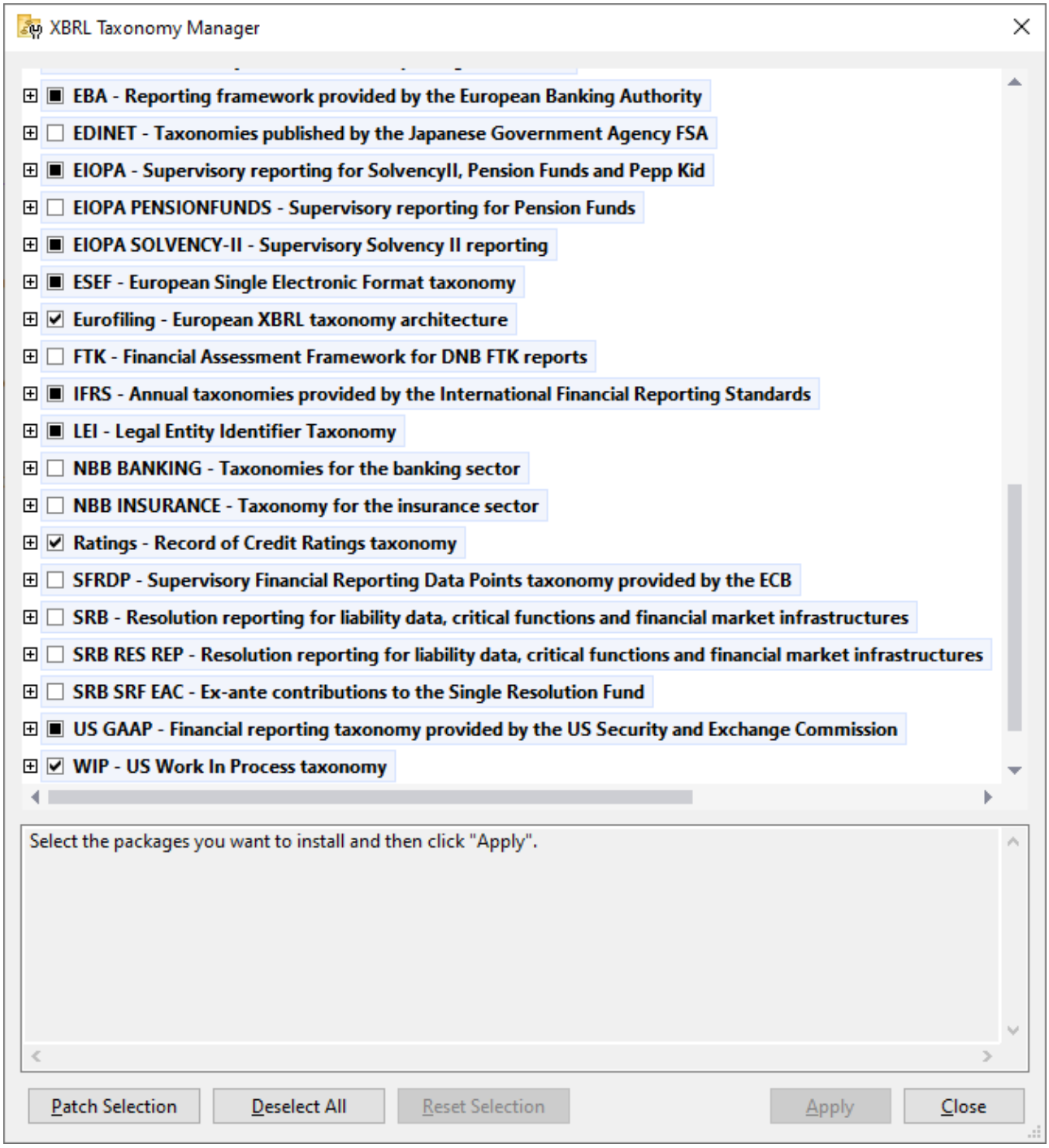
Site web d'Altova : Éditeur de taxonomie XBRL, Valideur XBRL

17.1 Gestionnaire de taxonomie

XBRL Gestionnaire de taxonomie est un outil qui propose un moyen centralisé d'installer et de gérer des taxonomies XBRL pour une utilisation sur toutes les applications XBRL Altova, y compris XMLSpy

- Sur Windows, Gestionnaire de taxonomie a une interface utilisateur graphique *(voir la capture d'écran ci-dessous*) et est aussi disponible dans la ligne de commande. (Les applications desktop d'Altova sont disponibles sur Windows uniquement ; *voir la liste ci-dessous*.)
- Sur Linux et Mac Gestionnaire de taxonomie, l'outil est disponible uniquement dans la ligne de commande. (Les applications serveur d'Altova sont disponibles sur Windows, Linux et macOS ; *voir la liste ci-dessous*.)

© 2016-2022 Altova GmbH



Les applications activées par XBRL d'Altova

Applications desktop (Windows uniquement)	Applications de serveur (Windows, Linux, macOS)
Altova XBRL Add-ins pour Excel (EBA, Solvency II)	MapForce Server (Standard et Advanced Edition).

MapForce Enterprise Edition	RaptorXML+XBRL Server
StyleVision Enterprise Edition	StyleVision Server
XMLSpy Enterprise Edition	

Installation et désinstallation de Gestionnaire de taxonomie

Gestionnaire de taxonomie est installé automatiquement quand vous installez d'abord une nouvelle version de l'Altova Mission Kit Enterprise Edition ou toute application activée par XBRL d'Altova (*voir la table ci-dessus*).

De même, il est supprimé automatiquement lorsque vous désinstallez la dernière application activée pour XBRL Altova depuis votre ordinateur.

Fonctions <% TAXON-MANAGER%>

Gestionnaire de taxonomie propose les fonctions suivantes :

- Consulter les taxonomies XBRL installées sur votre ordinateur, et contrôlez si de nouvelles versions sont disponibles pour le téléchargement.
- Télécharger des versions plus récentes des taxonomies XBRL indépendamment du cycle de release des produits Altova. (Altova stocke des taxonomies en ligne et vous pouvez les télécharger via Gestionnaire de taxonomie.)
- Installer ou désinstaller une des versions d'une taxonomie donnée (ou toutes les versions si nécessaire).
- Une taxonomie XBRL peut avoir des dépendances sur d'autres taxonomies. Lorsque vous installez ou désinstallez une taxonomie particulière, Gestionnaire de taxonomie vous informe sur d'autres taxonomies dépendantes et les installera ou désinstallera également automatiquement.
- Gestionnaire de taxonomie utilise le mécanisme du catalogue XML pour mapper les références de schéma aux fichiers locaux. Dans le cas de taxonomies XBRL larges, le traitement sera plus rapide que si les taxonomies étaient à un emplacement à distance.
- Toutes les taxonomies majeures sont disponibles via Gestionnaire de taxonomie et sont régulièrement mises à jour pour les dernières versions. Ceci vous fournit une ressource unique pour gérer toutes vos taxonomies et les mettre à disposition de toutes les applications activées par XBRL d'Altova.
- Les changements réalisés dans Gestionnaire de taxonomie prennent effet pour tous les produits d'Altova sur cet appareil.

Taxonomies XBRL personnalisées

Si vous devez travailler avec des taxonomies XBRL personnalisées qui ne sont pas incluses dans Gestionnaire de taxonomie, vous pouvez ajouter ces taxonomies à Gestionnaire de taxonomie pour la gestion comme suit :

- *Dans les applications desktop d'Altova :* Sélectionnez la commande de menu **Outils | Options** et allez à la section *XBRL | Packages de taxonomie*. Recherchez le package ZIP de votre taxonomie XBRL personnalisée. Pour plus d'information, voir la description de cette commande dans votre documentation de produit desktop.
- *Dans les applications de serveur Altova :* Lorsque vous exécutez des commandes de la ligne de commande qui appuie les taxonomies personnalisées, fournissez l'option **`--taxonomy-package`** or **`--taxonomy-package-config-file`**. Par exemple : Dans RaptorXML+XBRL Server, ces options sont appuyées par les commandes de validation XBRL telles que valxbrl ou valxbrltaxonomy ; dans MapForce, elles sont prises en charge par la commande d'exécution.

© 2016-2022 Altova GmbH

Comment cela fonctionne ?

Altova stocke toutes les taxonomies XBRL utilisées dans les produits Altova en ligne. Ce référentiel est mis à jour lorsque de nouvelles versions des taxonomies sont publiées. Gestionnaire de taxonomie affiche des informations sur les dernières taxonomies disponibles lorsqu'elles sont appelées dans son formulaire GUI de même que sur CLI. Vous pouvez ensuite installer, mettre à jour eou désinstaller les taxonomies via Gestionnaire de taxonomie.

Gestionnaire de taxonomie installe également les taxonomies d'une autre manière. Sur le site web d'Altova (https://www.altova.com/taxonomy-manager), vous pouvez sélectionner une taxonomie et ses taxonomies dépendantes que vous souhaitez installer. Le site web préparera un fichier de type `.altova_taxonomies` pour le téléchargement qui contient des informations sur la sélection de taxonomie. Lorsque vous double-cliquez sur ce fichier ou le passez à Gestionnaire de taxonomie via CLI comme argument de la commande **`installer`**, Gestionnaire de taxonomie installera les taxonomies que vous avez sélectionnés.

Cache local : suivre vos taxonomies

Toutes les informations sur les taxonomies installées sont suivies dans un répertoire cache centralisé sur votre ordinateur, situé ici :

Windows	`C:\ProgramData\Altova\pkgs\.cache`
Linux	`/var/opt/Altova/pkgs\.cache`
macOS	`/var/Altova/pkgs`

Ce répertoire cache est mis à jour régulièrement avec le dernier statut des taxonomies dans l'emplacement de stockage en ligne d'Altova. Ces mises à jour sont réalisées aux moments suivants :

- À chaque fois que vous lancez Gestionnaire de taxonomie.
- Lorsque vous exécutez XMLSpy pour la première fois dans un jour donné du calendrier.
- Si XMLSpy est ouvert plus de 24 heures, le cache est mis à jour toutes les 24 heures.
- Vous pouvez aussi mettre à jour le cache en exécutant la commande de `mise à jour` dans l'interface de ligne de commande.

Pour cette raison, le cache permet à Gestionnaire de taxonomie de suivre continuellement vos taxonomies installées par rapport aux taxonomies disponibles en ligne sur le site web d'Altova.

Ne modifiez pas le cache manuellement !

Le répertoire de cache local est entretenu automatiquement sur la base des taxonomies que vous installez ou désinstallez ; il ne doit pas être modifié ou supprimé manuellement. Il ne devrait pas être altéré ou supprimé manuellement. Si vous êtes amené à réinitialiser Gestionnaire de taxonomie à son état original "intact", alors, sur l'interface de la ligne de commande (CLI) : (i) exécutez la commande `reset`, et (ii) exécutez la commande `initialize`. (En alternative, exécutez la commande **`reset`** avec l'option **`--i`**.)

17.1.1 Exécuter le Gestionnaire de taxonomie

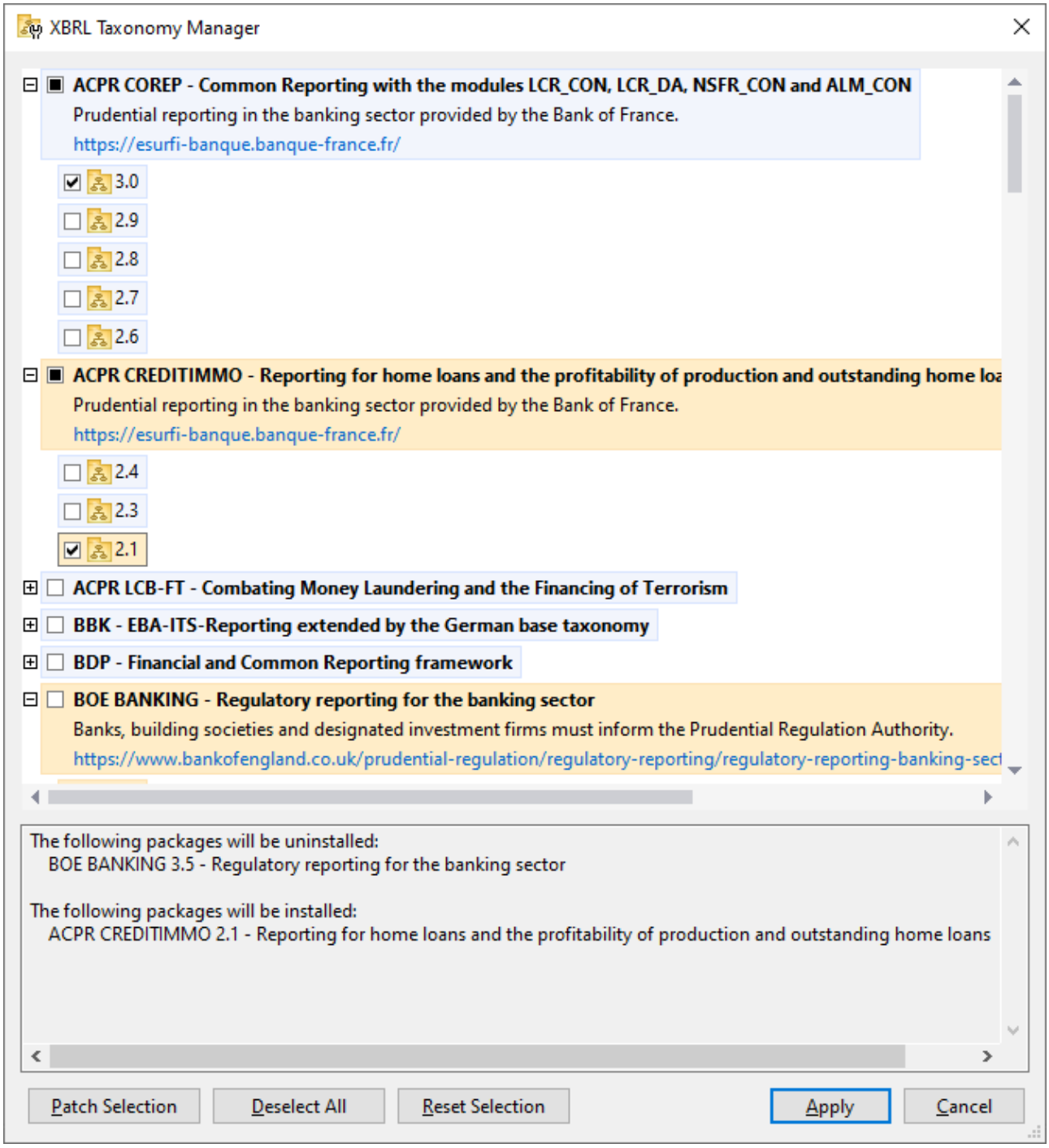
Interface utilisateur graphique

Vous pouvez accéder à la GUI de Gestionnaire de taxonomie des manières suivantes :

- *Durant l'installation de XMLSpy:* Vers la fin de la procédure d'installation, sélectionnez la case à cocher *Invoke Altova Taxonomy Manager* pour accéder au Gestionnaire de taxonomie XBRL directement. Ceci vous permettra d'installer les taxonomies au cours de la procédure d'installation de votre application Altova.
- *Après l'installation de XMLSpy:* Une fois que votre installation a été installée, vous pouvez accéder à la GUI de Gestionnaire de taxonomie à tout moment, via la commande de menu **Outils | Gestionnaire de taxonomie XBRL**.
- Via le fichier `.altova_taxonomies` téléchargé du Centre de téléchargement de la taxonomie XBRL d'Altova: Double-cliquez sur le fichier téléchargé pour exécuter la GUI de Gestionnaire de taxonomie, qui sera configurée pour installer les taxonomies que vous avez sélectionnées (le site web) pour installation.

Une fois que Gestionnaire de taxonomie GUI (*capture d'écran ci-dessous*) a été ouvert, les taxonomies déjà installées seront affichées telles sélectionnées. Si vous voulez installer une taxonomie supplémentaire, sélectionnez-la. Si vous voulez désinstaller une taxonomie déjà installée, désélectionnez-la. Une fois que vous avez faits vos sélections et/ou désélections, vous êtes prêts pour appliquer vos changements. Les taxonomies qui seront installées ou désinstallées seront mises en surbrillance et un message sur les modifications à venir sera posté dans le volet Messages au niveau inférieur de la fenêtre Gestionnaire de taxonomie (*voir la capture d'écran*).

© 2016-2022 Altova GmbH



Interface de ligne de commande

Vous pouvez exécuter Gestionnaire de taxonomie depuis une interface de ligne de commande en sélectionnant son fichier exécutable, `taxonomymanager.exe`.

Le fichier **`taxonomymanager.exe`** est situé dans le dossier suivant :

- *Sur Windows :* `C:\ProgramData\Altova\SharedBetweenVersions`
- *Sur Linux ou macOS (application serveur uniquement) :* `%INSTALLDIR%/bin`, où `%INSTALLDIR%` est le répertoire d'installation du programme.

Vous pouvez alors utiliser toute commande de la section de référence de la commande CLI.

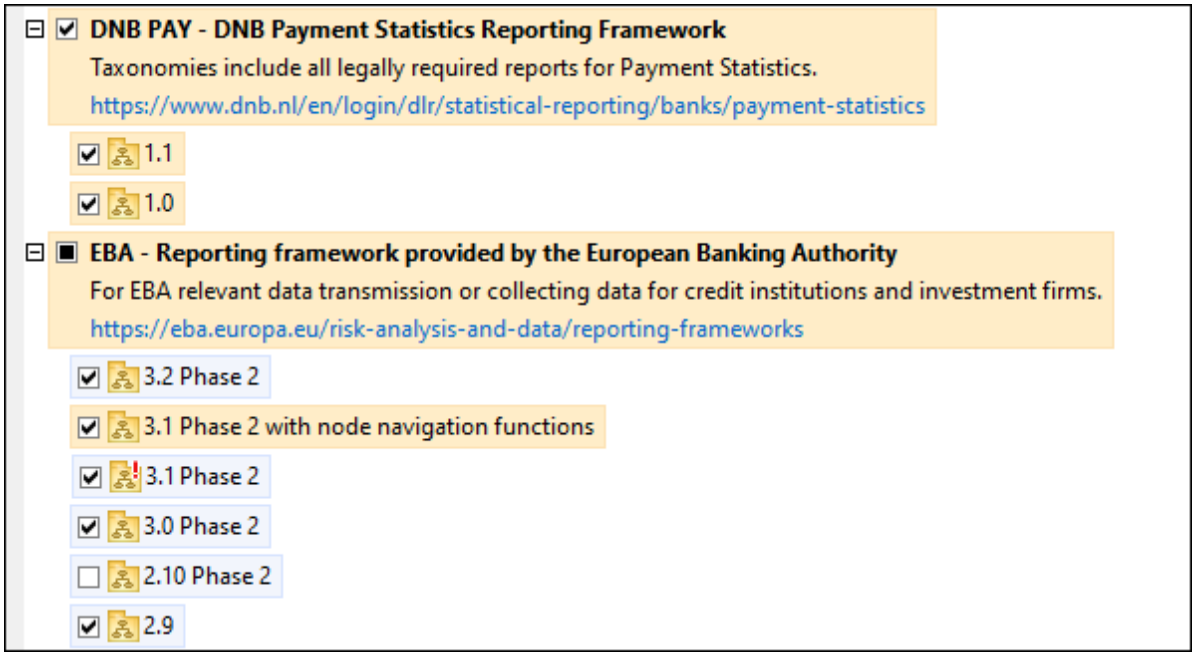
Pour afficher l'aide pour la commande, exécutez l'étape suivante :

- *Sur Windows :* `taxonomymanager.exe --help`
- *Sur Linux ou macOS (application serveur uniquement) :* `sudo ./taxonomymanager --help`

17.1.2 Catégories de statut

Gestionnaire de taxonomie catégorise les taxonomies sous sa gestion comme suit :

- *Installer taxonomies.* Celles-ci sont affichées dans la GUI avec leurs cases à cocher sélectionnées (*dans la capture d'écran ci-dessous, les versions cochées ci-de la taxonomie DNB et EBA sont des taxonomies installées*). Si toutes les versions de taxonomie sont sélectionnées, alors la marque de sélection est une coche. Si au moins une version est décochée, alors la coche de sélection est un carré coloré plein. Vous pouvez décocher une taxonomie installée pour la **désinstaller**.
- *Désinstaller taxonomies disponibles.* Elles sont affichées dans la GUI avec leurs cases à cocher non sélectionnées. Vous pouvez sélectionner les taxonomies que vous souhaitez **installer**.



- Les *taxonomies pouvant être mises à niveau* sont celles qui ont été revues par leurs émetteurs depuis qu'elles ont été installées. Elles sont indiquées par leur GUI par une icône  (*voir la capture d'écran ci-dessus*). Vous pouvez **retoucher** la taxonomie installée avec la révision disponible.

© 2016-2022 Altova GmbH

Points à noter

- Dans la capture d'écran ci-dessus, les taxonomies DNB et quelques unes des taxonomies EBA sont cochées. Celles avec un arrière-plan bleu sont déjà installées. Celles avec un arrière-plan jaune sont désinstallées et ont été sélectionnées pour l'installation. Notez que (i) la taxonomie EBA 2.10 Phase 2 n'est pas installée et n'a pas été sélectionnée pour l'installation, (ii) la taxonomie EBA 3.1 Phase 2 a été installée, mais elle a été retouchée par son émetteur depuis qu'elle a été installée et le correctif n'a pas encore été installé.
- Lorsque vous exécutez Gestionnaire de taxonomie depuis la ligne de commande, la commande `list` est utilisée avec différentes options pour pour recenser les différentes catégories de taxonomies :

`taxonomymanager.exe list`	Recense toutes les taxonomies installées et disponibles ; celles pouvant être mises à niveau sont également indiquées.
`taxonomymanager.exe list -i`	Recense les taxonomies installées uniquement ; celles pouvant être mises à niveau sont également indiquées
`taxonomymanager.exe list -u`	Recense les taxonomies pouvant être mises à niveau

Note : Sur Linux et macOS, **`use sudo ./taxonomymanager list`**

17.1.3 Retoucher ou Installer une taxonomie

Retoucher une taxonomie installée

Occasionnellement, des taxonomies XBRL peuvent recevoir des patchs (mises à niveau ou révisions) depuis leurs émetteurs. Lorsque Gestionnaire de taxonomie détecte que des patchs sont disponibles, ceux-ci sont indiqués dans les listes de taxonomie Gestionnaire de taxonomie et vous pouvez installer des patchs rapidement.

Dans la GUI

Les patchs sont indiqués par l'icône . (*Voir aussi la rubrique précédente sur les catégories de statut.*) Si les patchs sont disponibles, le bouton **Patch Selection** sera activé. Cliquez dessus pour sélectionner et préparer tous les patchs pour installation. Dans la GUI, l'icône de chaque taxonomie sera patchée de  à , et le volet des Messages en bas du dialogue recense les patchs qui doivent être appliqués. Lorsque vous êtes prêt pour installer des patchs sélectionnés, cliquez sur **Appliquer**. Tous les correctifs seront appliqués ensemble. Notez que si vous décochez une taxonomie marquée pour une correction, réellement vous désinstallerez cette taxonomie.

Sur le CLI

Pour appliquer un patch dans l'interface de ligne de commande :

1. Exécuter la commande `list -u` . Cela liste des taxonomies lorsque des mises à niveau de patch sont disponibles.
2. Exécuter la commande `upgrade` pour installer les patches.

Installer une taxonomie disponible

Vous pouvez installer des taxonomies en utilisant soit la GUI Gestionnaire de taxonomie ou en envoyant Gestionnaire de taxonomie les instructions d'installation via la ligne de commande.

Note : Si la taxonomie actuelle référence d'autres taxonomies, les taxonomies référencées sont aussi installées.

Dans la GUI

Pour installer une taxonomie utilisant la GUI Gestionnaire de taxonomie GUI, sélectionnez les taxonomies que vous voulez installer et cliquez sur **Appliquer**.

Vous pouvez aussi sélectionner les taxonomies que vous voulez installer sur le site web d'Altova et générer un fichier téléchargeable `.altova_taxonomies`. Lorsque vous double-cliquez sur ce fichier, il ouvrira Gestionnaire de taxonomie avec les taxonomies que vous vouliez présélectionner. La seule chose qui vous reste à faire, c'est cliquer sur **Appliquer**.

Sur le CLI

Pour installer des taxonomies via la ligne de commande, exécutez la commande `installer` :

```
taxonomymanager.exe install [options] Taxonomie+
```

où `Taxonomie` est la taxonomie (ou les taxonomies) que vous voulez installer ou un fichier `.altova_taxonomies`. Une taxonomie est référencée par un identifiant de format `<name>-<version>`. (Les identifiants de taxonomies sont affichés quand vous exécutez la commande `list`.) Vous pouvez saisir autant de taxonomies que vous le souhaitez. Pour plus de détails, voir la description de la commande `install`.

Note : sur Linux ou macOS, utilisez la commande `sudo ./taxonomymanager`.

Installer une taxonomie requise

Lorsque vous exécutez une commande activée par XBRL dans XMLSpy, et que XMLSpy découvre qu'une taxonomie dont elle a besoin pour exécuter la commande n'est pas présente ou est incomplète, Gestionnaire de taxonomie sera affiché l'information sur la taxonomie manquante. Vous pouvez ensuite installer directement toute taxonomie manquante via Gestionnaire de taxonomie.

Dans le Gestionnaire de taxonomie GUI, vous pouvez consulter toutes les taxonomies précédemment installées á tout moment en exécutant Gestionnaire de taxonomie de **Outils | Gestionnaire de taxonomie**.

17.1.4 Désinstaller une taxonomie, Réinitialiser

Désinstaller une taxonomie

Vous pouvez désinstaller des taxonomies en utilisant soit la GUI Gestionnaire de taxonomie ou en envoyant Gestionnaire de taxonomie les instructions d'installation via la ligne de commande.

© 2016-2022 Altova GmbH

Note : si la taxonomie que vous voulez désinstaller référence d'autres taxonomies, alors les taxonomies référencées sont également désinstallées.

Dans la GUI
Pour désinstaller les taxonomies utilisant la GUI Gestionnaire de taxonomie, effacez leurs cases à cocher et cliquez sur **Appliquer**. Les taxonomies sélectionnées et leurs taxonomies référencées seront désinstallées.

Pour désinstaller les taxonomies, cliquez sur **Désélectionner tout** et cliquez sur **Appliquer**.

Sur le CLI
Pour désinstaller des taxonomies via la ligne de commande, exécutez la commande **`désinstaller`** :

```
taxonomymanager.exe uninstall [options] Taxonomy+
```

où chaque argument **`Taxonomie`** est la taxonomie que vous voulez désinstaller ou un fichier **`.altova_taxonomies`**. Une taxonomie est spécifiée par un identifiant qui a un format **`<name>-<version>`**. (Les identifiants de taxonomies sont affichés quand vous exécutez la commande **`list`**.) Vous pouvez saisir autant de taxonomies que vous le souhaitez. Pour plus de détails, voir la description de la commande `désinstaller`.

Note : sur Linux ou macOS, utilisez la commande **`sudo ./taxonomymanager`**.

Réinitialiser Gestionnaire de taxonomie

Vous pouvez réinitilaiser Gestionnaire de taxonomie. Ceci supprime toutes les taxonomies installées et le répertoire de mise sous cache.

- Dans la GUI, cliquez sur **Reset Selection**.
- Dans la CLI, exécutez la commande `reset`.

Une fois avoir exécuté cette commande, vous devrez exécuter la commande `initialize`, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécutez la commande `reset` avec l'option `-i`.

Notez que `reset -i` restaure l'installation originale du produit, il est recommandé d'exécuter la commande `update` après avoir réalisé la réinitialisation. En alternative, exécutez la commande `reset` avec les options `-i` and `-u`.

17.1.5 Interface de ligne de commande (CLI)

Pour appeler Gestionnaire de taxonomie dans la ligne de commande, vous devez connaître le chemin de l'exécutable. Par défaut, l'exécutable Gestionnaire de taxonomie est installé dans le chemin suivant :

```
C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe
```

Note : sur les systèmes Linux et macOS, une fois que vous avez changé le répertoire à celui contenant l'exécutable, vous pouvez appeler l'exécutable avec **`sudo ./taxonomymanager`**. Le préfixe **`./`** indique

que l'exécutable est le répertoire actuel. Le préfixe **sudo** indique que la commande que la commande doit être exécutée avec des privilèges root.

Syntaxe de ligne de commande

La syntaxe générale pour utiliser la ligne de commande est la suivante :

```
<exec> -h | --help | --version | <command> [options] [arguments]
```

Dans l'extrait ci-dessus, la barre verticale **|** sépare un ensemble d'items mutuellement exclusifs. Les crochets **[]** indiquent des items optionnels. De manière générale, vous pouvez saisir le chemin d'exécutable suivi soit par les options `-h`, `--help`, ou `--version` ou par une commande. Chaque commande peut contenir des options et des arguments. La liste des commandes est décrite dans les sections suivantes.

17.1.5.1 help

Cette commande propose une aide contextuelle pour les commandes liées à l'exécutable Gestionnaire de taxonomie.

Syntaxe

```
<exec> help [command]
```

Où `[command]` est un argument optionnel qui spécifie un nom de commande valide.

Veuillez noter les points suivants :

- Vous pouvez invoquer de l'aide une commande en saisissant La section commande suivie par **-h** ou **--help**, par exemple : **<exec> list-h**
- Si vous tapez **-h** or **--help** directement après la commande exécutable et avant une commande, vous recevrez une aide générale (pas d'aide pour la commande), par exemple : **<exec> -h list**

Exemple

La commande suivante affiche une aide concernant la commande `list` :

```
./taxonomymanager help list
```

17.1.5.2 info

Cette commande affiche des informations détaillées pour chacune des taxonomies fournies en tant qu'argument de `Taxonomie`. Cette information inclut le titre, la version, description, l'éditeur et toute taxonomie soumise et toute taxonomie dépendante, et mentionne si la taxonomie a été installée ou non.

Syntaxe

```
<exec> info [options] Taxonomy+
```


© 2016-2022 Altova GmbH

- L'argument **Taxonomie** est le nom de la taxonomie ou une partie du nom de la taxonomie. (Pour afficher une ID de pack de taxonomie et des informations détaillées sur son statut d'installation, vous devriez utiliser la commande list.)
- Utiliser **<exec> info -h** pour afficher l'aide de la commande.

Exemple

La commande suivante affiche des informations détaillées concernant les taxonomies **eba-2.10** et **us-gaap-2020.0** :

```
taxonomymanager info eba-2.1.0 us-gaap-2020.0
```

17.1.5.3 initialize

Cette commande initialise l'environnement Gestionnaire de taxonomie. Elle crée un répertoire de cache où les informations concernant toutes les taxonomies sont stockées localement. L'initialisation est réalisée automatiquement la première fois qu'une application activée par XBRL est installée. Vous n'aurez pas besoin d'exécuter cette commande dans des circonstances normales, mais vous devrez l'exécuter généralement après la commande **reset**.

Syntaxe

```
<exec> initialize | init [options]
```

Options

La commande **initialize** accepte les options suivantes :

`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemple

La commande suivante initialise Gestionnaire de taxonomie:

```
taxonomymanager initialize
```

17.1.5.4 install

Cette commande installe une ou plusieurs taxonomies.

Syntaxe

```
<exec> install [options] Taxonomy+
```

Pour installer de multiples taxonomies, ajoutez l'argument **Taxonomy** de nombreuses fois.

L'argument **Taxonomy** est l'un des suivants :

- Un identifiant de taxonomie (avoir un format de **<name>-<version>**, par exemple : **eba-2.10**). Pour trouver les identifiants de taxonomie que vous voulez, exécutez la commande list. **Vous pouvez aussi utiliser des identifiants abrégés s'ils sont uniques, par exemple** eba. Si vous utilisez un identifiant abrégé, alors la dernière version de cette taxonomie sera installée.
- Le chemin vers un fichier **.altova_taxonomies** téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir *Introduction à TaxonomyManager : Comment cela fonctionne-t-il ?*.

Options
La commande **install** accepte les options suivantes :

--silent, --s	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
--verbose, --v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
--help, --h	Afficher l'aide pour la commande.

Exemple

La commande suivante installe les dernières taxonomies **eba** (European Banking Authority) et **us-gaap** (US Generally Accepted Accounting Principles) :

```
taxonomymanager install eba us-gaap
```

17.1.5.5 list

Cette commande recense les taxonomies sous la gestion de Gestionnaire de taxonomie. La liste affiche comme suit

- Toutes les taxonomies disponibles
- Les taxonomies contenant le string dans leur nom soumis comme argument **Taxonomy**
- Seules les taxonomies installées
- Seules les taxonomies qui peuvent être mises à niveau

Syntaxe

```
<exec> list | ls [options] Taxonomy?
```

Si aucun argument **Taxonomy** n'est soumis, alors toutes les taxonomies disponibles sont recensées. Autrrement, les taxonomies sont recensées par des options soumises (*voir l'exemple ci-dessous*). Notez que vous pouvez soumettre l'argument **Taxonomy** de nombreuses fois.

Options
La commande **list** accepte les options suivantes :

© 2016-2022 Altova GmbH

`--installed, --i`	Uniquement la liste des taxonomies installées. Le réglage par défaut est **faux**.
`--upgradeable, --u`	Recenser uniquement les taxonomies lorsque des mises à niveau (patches) sont disponibles. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemples

- Pour exécuter toutes les taxonomies disponibles, exécutez : **taxonomymanager list**
- Pour recenser les taxonomies installées, exécutez : **taxonomymanager list -i**
- Pour recenser tous les taxonomies qui contiennent soit "eba", soit "us-gaap" dans leur nom, exécutez **taxonomymanager list eba us-gaap** :

17.1.5.6 reset

Cette commande supprime toutes les taxonomies installées et le répertoire de mise sous cache. Vous réinitialiserez complètement votre environnement de taxonomie. Une fois avoir exécuté cette commande, vous devrez exécuter la commande `initialize`, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécuter la commande `reset` avec l'option `-i`. Pusique `reset -i` restaure l'installation originale du produit, nous vous recommandons que vous exécutiez la commande `update` après avoir réalisé la réinitialisation et l'initialisation. En alternative, exécutez la commande reset avec les options `-i` et `-u`.

Syntaxe

```
<exec> reset [options]
```

Options
La commande **reset** accepte les options suivantes :

`--init, --i`	Initialiser Gestionnaire de taxonomie après le reset. Le réglage par défaut est **faux**.
`--update, --u`	Mettre à jour la liste de taxonomies disponibles dans le cache. Le réglage par défaut est **faux**.
`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemples

- Pour réinitialiser Gestionnaire de taxonomie, exécuter : **taxonomymanager reset**
- Pour réinitialiser Gestionnaire de taxonomie et l'initialiser, exécutez : **taxonomymanager reset -i**

- Pour réinitialiser Gestionnaire de taxonomie, initialiser-la et mettez à jour sa liste de taxonomie, exécutez : **taxonomymanager reset -i-u**

17.1.5.7 uninstall

Cette commande désinstalle une ou plusieurs taxonomies. Par défaut, toute taxonomie référencée par la taxonomie actuelle sera également désinstallée. Pour désinstaller uniquement la taxonomie actuelle et garder les taxonomies référencées, définir l'option `--k`.

Syntaxe

```
<exec> désinstaller [options] Taxonomy+
```

Pour désinstaller de multiples taxonomies, ajoutez l'argument **Taxonomy** de nombreuses fois.

L'argument **Taxonomy** est l'un des suivants :

- Un identifiant de taxonomie (avoir un format de **<name>-<version>**, par exemple : **eba-2.10**). Pour trouver les identifiants de taxonomie qui sont installées, exécutez la commande `list -i` . **Vous pouvez aussi utiliser un nom de taxonomie abrégé s'il est unique, par exemple** eba. Si vous utilisez un nom abrégé, alors toutes les taxonomies qui contiennent une abréviation dans leur nom seront désinstallées.
- Le chemin vers un fichier **.altova_taxonomies** téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir *Introduction à TaxonomyManager : Comment cela fonctionne-t-il ?*.

Options
La commande **désinstaller** accepte les options suivantes :

`--keep-references, --k`	Définir cette option pour garder les taxonomies référencées. Le réglage par défaut est **faux**.
`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **faux**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **faux**.
`--help, --h`	Afficher l'aide pour la commande.

Exemple

La commande suivante désinstalle les taxonomies **eba-2.10** et **us-gaap-2020.0** et leurs dépendances :

```
taxonomymanager uninstall eba-2.10 us-gaap-2020.0
```

La commande suivante désinstalle la taxonomie **eba-2.10** mais pas les taxonomies qu'elle référence :

```
taxonomymanager uninstall --k eba-2.10
```


© 2016-2022 Altova GmbH

17.1.5.8 update

Cette commande requête la listes des taxonomies disponibles depuis le magasin en ligne et met à jour le répertoire de mise sous cache local. Vous devriez exécuter cette commande sauf si vous avez réalisé un `reset` et `initialize`.

Syntaxe

```
<exec> update [options]
```

Options
La commande **`mise à jour`** accepte les options suivantes :

`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **`faux`**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **`faux`**.
`--help, --h`	Afficher l'aide pour la commande.

Exemple

La commande suivante met à jour le cache local avec la liste des dernières taxonomies :

```
taxonomymanager update
```

17.1.5.9 upgrade

Cette commande met à niveau toutes les taxonomies éligibles qui peuvent être mises à niveau à la dernière version *patch* disponible. Vous pouvez identifier des taxonomies à mettre à niveau en exécutant la commande `list -u`.

Note : La commande **`mettre à niveau`** supprime une taxonomie dépréciée si aucune version plus récente n'est disponible.

Syntaxe

```
<exec> upgrade [options]
```

Options
La commande **`mise à niveau`** accepte les options suivantes :

`--silent, --s`	Afficher uniquement des messages d'erreur. Le réglage par défaut est **`faux`**.
`--verbose, --v`	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est **`faux`**.
`--help, --h`	Afficher l'aide pour la commande.

© 2016-2022 Altova GmbH

17.2 Procédures de base

La section Procédures de base décrit comment créer des taxonomies contenant les composants de base. Elle est structurée comme suit :

- Elle commence avec un bref aperçu de la distinction à effectuer entre les taxonomies nouvelles et existantes et la signification de cette distinction. Cette introduction est suivie par une explication des fichiers qui constituent une taxonomie XBRL et comment ceux-ci sont affichés dans le Mode XBRL.
- À partir de la section Créer une nouvelle taxonomie, nous décrivons les étapes pour construire une taxonomie dans le Mode XBRL. À la fin de chaque section, vous trouverez un ensemble d'instructions qui vous aideront à mettre en pratique ou à tester l'information donnée dans cette section et elle vous permet de poursuivre la création de la taxonomie jusqu'à ce point à l'aide des instructions contenues dans les sections précédentes.

17.2.1 Taxonomies : nouvelles et existantes

Dans le Mode XBRL de XMLSpy, vous pouvez éditer des taxonomies existantes et créer de nouvelles taxonomies.

- *Taxonomies existantes :* il y a deux types de taxonomies existantes : (i) les taxonomies standard qui ne doivent pas être éditées; et les (ii) taxonomies non-standard qui peuvent être éditées ; celles-ci peuvent avoir été créées par vous ou par quelqu'un d'autre.
- *Nouvelles taxonomies :* les nouvelles taxonomies peuvent être créées dans XMLSpy. Il existe deux types : les (i) taxonomies qui sont créées à partir de zéro ; et les (ii) taxonomies qui étendent une taxonomie standard.

Les deux types de taxonomies peuvent être consultées et éditées dans le Mode XBRL. Dans certains cas, comme par exemple lorsqu'une taxonomie standard est importée dans une taxonomie que vous créez (pour étendre la taxonomie importée), vous ne serez pas autorisé à éditer la taxonomie importée. Les éléments provenant des taxonomies importées qui ne sont pas autorisées à être éditées sont affichées en gris.

Packs de taxonomie

Un pack de taxonomie XBRL est un archive zippé qui contient une copie hors ligne d'une taxonomie. Le pack de taxonomie contient un fichier XML de catalogue qui remappe les URI aux emplacements du fichier de taxonomie hors ligne, et rend donc la taxonomie disponible hors ligne pour les applications. Les règles qui spécifient comment les packs de taxonomie doivent être structurés et créés sont définis dans Recommandation de Packages de Taxonomie de XBRL.org.

Si vous téléchargez un pack de taxonomie, vous pourrez l'enregistrer avec XMLSpy de manière à ce que XMLSpy puisse utiliser les ressources hors ligne du pack (comme des schémas) lors de la validation. L'enregistrement du pack s'effectue par le biais du volet **Outils | Options | Packs de taxonomie**, la procédure est décrite ici.

Étapes de création d'une nouvelle taxonomie

Une nouvelle taxonomie sera généralement créée sur une taxonomie déjà existante. Dans la nouvelle taxonomie, les nouveaux éléments seront ajoutés, et des relations entre ces nouveaux éléments et entre les

nouveaux éléments et les éléments importés seront créées. Les exigences générales d'une nouvelle taxonomie et de la procédure à suivre pour en créer une nouvelle sont décrites ci-dessous :

1. La nouvelle taxonomie doit être créée dans son propre espace de noms afin de la distinguer par rapport aux autres taxonomies. Si la nouvelle taxonomie s'étend à une taxonomie existante, la taxonomie existante doit être importée dans la nouvelle taxonomie.
2. Les nouveaux concepts (éléments) sont définis dans la nouvelle taxonomie.
3. Les fichiers de relation (ou base de liens) sont créés pour contenir la définition, la présentation, le calcul, le libellé, et les relations de référence de la nouvelle taxonomie .
4. Les relations pour la nouvelle taxonomie doivent être créées à partir de zéro.

Dans la description ci-dessus, nous avons utilisé le terme *taxonomie* pour dénoter toute la taxonomie, qui comprend plusieurs fichiers : les fichiers de définitions de concept et les fichiers de relation. (Voir la section fichiers de document de taxonomie pour une description des différents fichiers qui comprennent une taxonomie.)

Utiliser le Mode XBRL

Dans les sections suivantes, nous allons décrire comment utiliser les fonctions du Mode XBRL pour créer et éditer des taxonomies. Débutant avec la section Créer une nouvelle taxonomie, nous fournissons des instructions, à la fin de chaque section, pour créer votre propre taxonomie. Les instructions dans chaque section successive se basent sur le travail effectué dans la section précédente. Au moment où vous atteignez la section Créer des relations : Partie 1, vous vous serez familiarisé avec le Mode XBRL et pourrez l'utiliser avec aise.

La taxonomie que vous créerez mène, avec un travail supplémentaire, à la taxonomie fournie avec XMLSpy (`Nanonull.xsd`) et qui se trouve dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\XBRLExamples\Nanonull`. (Veuillez noter que le fichier de taxonomie principal présente toujours l'extension `.xsd`. L'extension de fichier `.xbrl` est utilisée pour les fichiers d'instance XBRL et non pas pour les fichiers de taxonomie.)

17.2.2 Aperçu des fichiers de taxonomie

Une taxonomie XBRL bien conçue stocke des concepts de taxonomie dans un fichier séparé des relations de taxonomie. Nous appellerons ce fichier le fichier de taxonomie principal ou bien le fichier de définitions de concept. De plus, étant donné qu'il existe plusieurs types de relations, celles-ci seront stockées dans des fichiers séparés par type de relation. La table ci-dessous recense les différents types de fichiers qui constituent normalement un document de taxonomie .

Fichier XBRL	Description	Type de fichier
Concepts	Chaque concept est défini dans un élément `element de` Schéma XML.	Fichier de schéma XML (`.xsd`) Fichier de définitions de concept
Relations de définition	Un élément `definitionLink` contient tous les localisateurs et arcs de définition pour les relations de concept.	fichier XML (`.xml`)

© 2016-2022 Altova GmbH

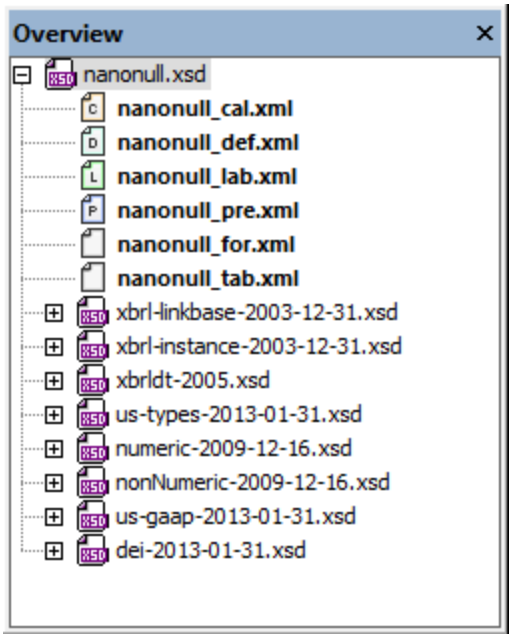
Relations de Calcul	Un élément `calculationLink` contient tous les localisateurs et arcs de calcul.	fichier XML (`.xml`)
Relations de Présentation	Un élément `presentationLink` contient tous les localisateurs et arcs de présentation.	fichier XML (`.xml`)
Libellés	Un élément `labelLink` contient tous les localisateurs, arcs de libellé et libellés.	fichier XML (`.xml`)
Références	Un élément `referenceLink` contient tous les localisateurs, arcs de référence et ressources de référence.	fichier XML (`.xml`)

Les localisations des fichiers de relation sont spécifiés dans le fichier de définitions du concept (le fichier `.xsd`) dans un élément `/schema/annotation/appinfo`, tel que la liste suivante :

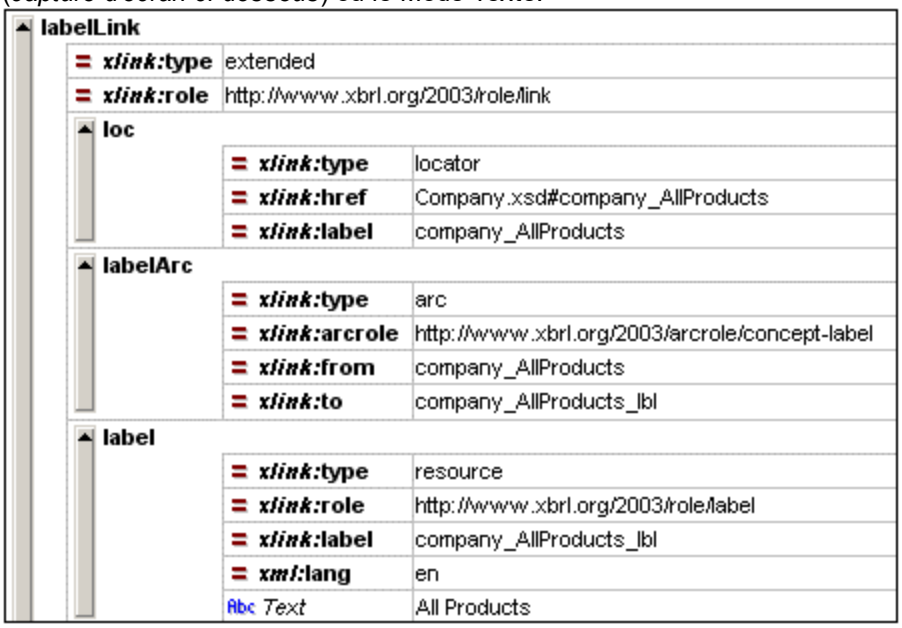
```
<xsd:annotation>
   <xsd:appinfo>
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullLabels.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/labelLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullDefinitions.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/definitionLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullPresentations.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/presentationLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullCalculations.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/calculationLinkbaseRef" />
      <link:linkbaseRef xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
         xlink:href="NanonullReferences.xml" xlink:type="simple"
         xlink:role="http://www.xbrl.org/2003/role/referenceLinkbaseRef" />

   </xsd:appinfo>
</xsd:annotation>
```

Lorsque le fichier de définitions de concept (le fichier `.xsd`) est ouvert dans le Mode XBRL, les différents fichiers de taxonomie sont affichés dans une structure d'arborescence dans l'assistant à la saisie Aperçu (*capture d'écran ci-dessous*).



Dans la capture d'écran ci-dessus, veuillez noter les icônes situés à la gauche des noms de fichier. Les fichiers de Schéma XML (`.xsd`) sont indiqués par une icône XSD. Les fichiers de relation présentent une icône de fichier colorée avec un caractère correspondant au caractère initial du type de relation. Par exemple, une icône indique un fichier de relations Définition, une icône indique un fichier de relations de Présentation, etc. Double-cliquez sur un de ces fichiers pour l'ouvrir dans XMLSpy, où il pourra être édité dans le Mode Grille (*capture d'écran ci-dessous*) ou le Mode Texte.



 © 2016-2022 Altova GmbH

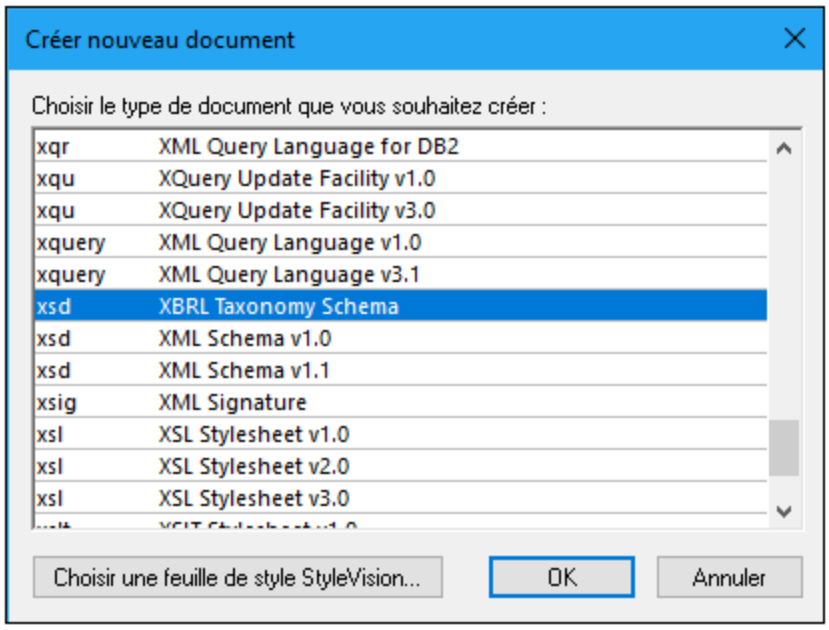
17.2.3 Créer une nouvelle taxonomie

Une nouvelle taxonomie est généralement créée pour étendre un ou plusieurs taxonomie standard. Si une nouvelle taxonomie est créée sur la base d'une taxonomie standard ou sur une taxonomie déjà existante, elle doit importer la taxonomie existante. En alternative, une nouvelle taxonomie peut être créée à partir de zéro. Dans le Mode XBRL de XMLSpy, vous pouvez facilement importer des taxonomies US-GAAP et IFRS à l'aide de l'Assistant dans votre taxonomie. La taxonomie importée peut ensuite être modifiée à l'aide de l'interface graphique du Mode XBRL.

La première étape de la création d'une nouvelle taxonomie est de créer son fichier de définitions de concept, qui est un fichier de Schéma XML (`.xsd`). Ce fichier, outre le fait de contenir les définitions de concept, définit et déclare l'espace de noms de la nouvelle taxonomie, localise les taxonomies à importer, localise les fichiers de relation de la taxonomie et déclare les espace de noms des taxonomies importées et les espace de noms utilisés.

Créer le fichier de définitions de concept

Pour créer une nouvelle taxonomie XBRL, choisir la commande de menu **Fichier | Nouveau**. Cela permet d'ouvrir le dialogue Créer un nouveau document (*capture d'écran ci-dessous*).



Choisir *xsd: XBRL Taxonomy Schema* puis cliquer sur **OK**. Une nouvelle taxonomie sera créée.

Le mieux sera d'enregistrer la taxonomie dans son propre dossier puisqu'il y aura d'autres composants de taxonomie pratiques à stocker dans le dossier commun.

Aperçu des étapes de création de la taxonomie

Les étapes pour construire une taxonomie sont expliquées ci-dessous.

1. Sélectionnez la taxonomie de base pour votre taxonomie via la commande de menu **XBRL | Importer/Référence**. Si vous voulez créer votre taxonomie de zéro, sautez cette étape.
2. Donnez à la taxonomie un espace de nom cible via la commande de menu **XBRL | Définir Espace de nom cible**.
3. Les espaces de nom de la taxonomie de base importée seront automatiquement déclarés dans votre taxonomie. Vous pouvez facilement ajouter tout autre espace de nom que vous souhaitez, tel que décrit dans la rubrique Configurer les fichiers de taxonomie.
4. Vous pouvez ensuite élargir la taxonomie de base avec vos propres éléments et relations.

Fichier d'exemple : Étape 1

Créer un nouveau document de taxonomie et enregistrez-le avec un nom quelconque à un endroit convenable. Ce fichier de taxonomie est le fichier principal, ou le fichier de définitions de concept. Il s'agit d'un fichier de Schéma XML et doit avoir une extension de fichier `.xsd`. Le fichier que nous sommes en train de créer est nommé `Nanonull.xsd`. Il s'agit du même nom que celui de l'exemple fourni dans `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\XBRLExamples\Nanonull`.

À l'étape suivante, nous allons importer la taxonomie de base dans notre taxonomie.

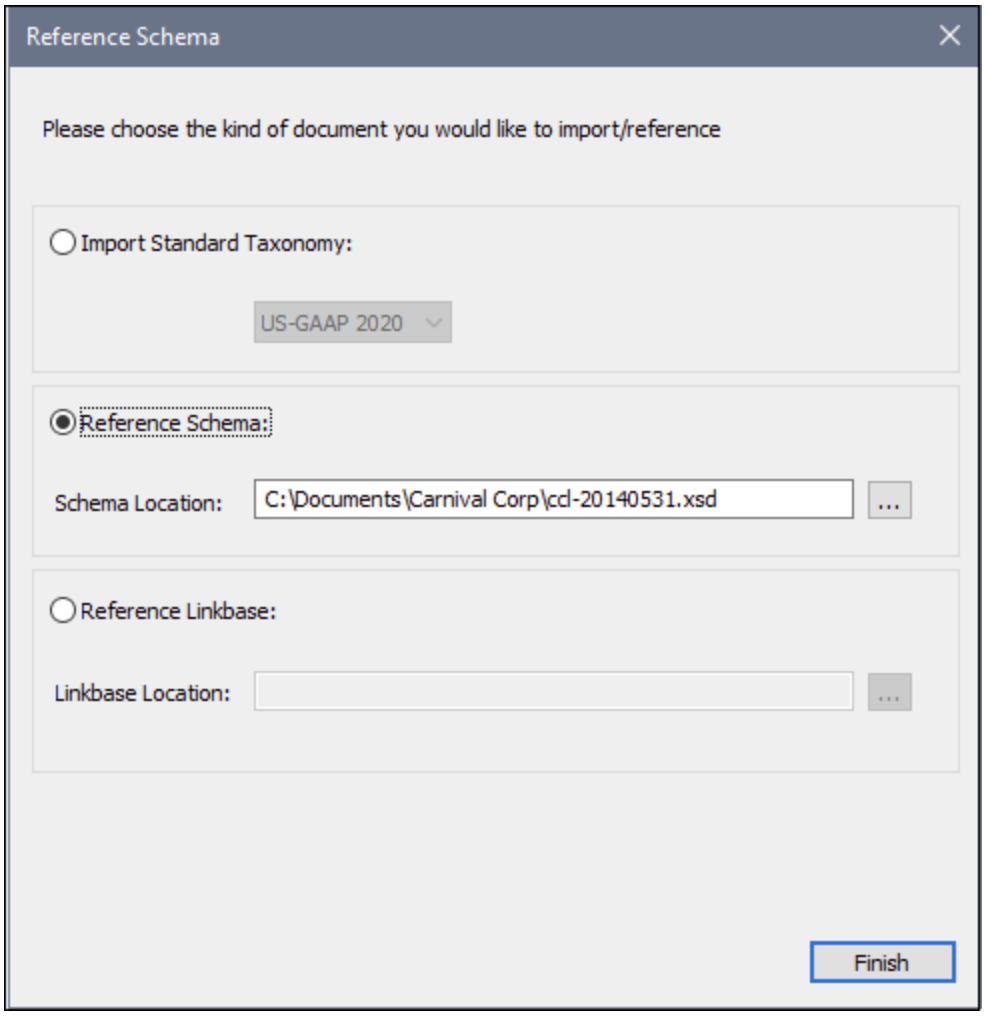
17.2.4 Importer une taxonomie

Si une nouvelle taxonomie doit être établie sur une taxonomie existante, celle-ci doit être importée dans la nouvelle taxonomie.

Pour importer une taxonomie, procéder comme suit :

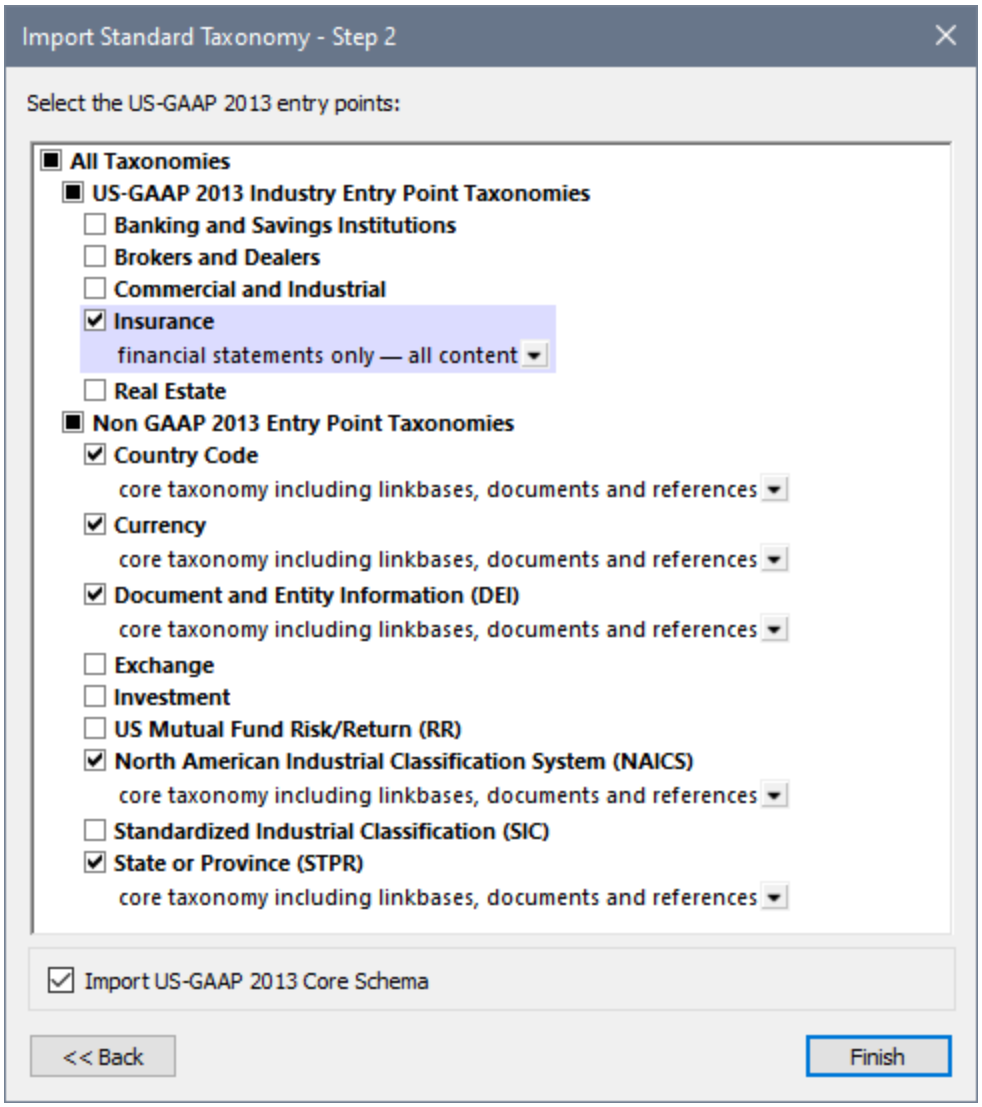
1. Cliquez avec la touche de droite dans l'Assistant à la saisie Aperçu dans le mode XBRL et sélectionnez **Importer/Référencer**. En alternative, sélectionnez la commande de menu **XBRL | Importer/Référencer**.
2. Dans le dialogue Importer la taxonomie standard qui s'ouvre (*capture d'écran ci-dessous*), sélectionnez une taxonomie pour importer ou une base de lien pour référencer. (Le nom du dialogue changera conformément à l'option que vous sélectionnez.)

© 2016-2022 Altova GmbH



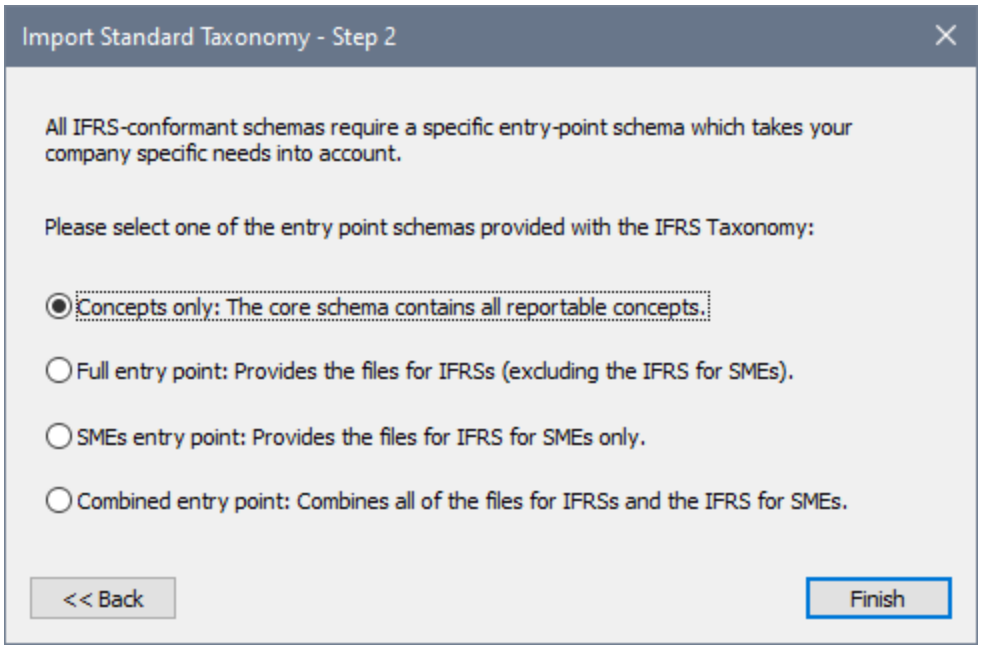
Il existe trois options pour importer/référence : (i) une taxonomie standard (US-GAAP ou IFRS) ; (ii) tout autre taxonomie (schéma de référence) ; et (iii) une base de lien. Si vous importez une taxonomie non-standard, sélectionnez le bouton radio de *Schéma de référence*, cliquez sur la touche **Parcourir** de la zone de texte de l'emplacement du schéma, et recherchez la taxonomie que vous souhaitez.

3. Une fois que vous avez terminé, cliquez sur **Terminer**. La taxonomie sélectionnée sera importée et ses éléments et relations seront affichées dans le Mode XBRL.
4. Si vous sélectionnez une taxonomie US-GAAP, alors un nouvel écran apparaît dans lequel vous pouvez (i) sélectionner les points d'entrée que vous souhaitez inclure dans la taxonomie, et (ii) spécifier si le US-GAAP Core Schema doit être importé (case à cocher en bas du dialogue).



Si vous avez séletionné IFRS comme base pour votre taxonomie, vous pouvez sélectionner un point d'entrée IFRS spécifique.

© 2016-2022 Altova GmbH



5. Cliquez sur **Terminer**, les schémas de point d'entrée sélectionnés sont importés et référencés par votre taxonomie. La taxonomie s'ouvre dans le Mode XBRL et est prête à être éditée.

Veuillez prendre note des points suivants :

- L'assistant à la saisie Aperçu recense également les taxonomies que la taxonomie importée importe elle-même, ainsi que les bases de liens que la taxonomie importée utilise.
- Dans l'assistant à la saisie Éléments globaux, les concepts définis dans la taxonomie importée sont énumérés.
- Dans la fenêtre Design et l'assistant à la saisie Détails, les concepts importés sont indiqués avec une couleur de police grise.
- Vous pouvez supprimer une taxonomie importée en cliquant avec la touche de droite sur dans l'assistant à la saisie Aperçu et en sélectionnant **Supprimer**.

Note : si vous trouvez qu'une taxonomie importante comme US-GAAP ralentit votre édition, utilisez le filtre dans la fenêtre principale pour limiter l'affichage vers des éléments créés dans la nouvelle taxonomie élargie. Cela permet d'accélérer considérablement l'édition.

Mécanisme d'importation

L'effet d'ajouter une importation, tel que décrit ci-dessus, est d'ajouter un élément `xs:import` au nouveau fichier de taxonomie. L'élément `xs:import` spécifie l'espace de nom et l'emplacement de la taxonomie importée(*liste ci-dessus*).

```
<xs:import namespace="http://fasb.org/us-gaap/2013-01-31"
           schemaLocation="http://xbrl.fasb.org/us-gaap/2013/elts/us-gaap-2013-01-31.xsd"/>
```

Dans la liste ci-dessus, l'attribut `schemaLocation` spécifie que la taxonomie est téléchargée via Internet. Mais cette URI mappe, par le biais du mécanisme de catalogue de XMLSpy, vers une copie locale de la taxonomie US-GAAP (qui est fournie avec votre paquet XMLSpy).

Afin de localiser une taxonomie enregistrée localement, une adresse locale peut être utilisée directement pour localiser la taxonomie. En alternative, une adresse web peut être utilisée qui est mappée vers une adresse locale par le biais d'un fichier de catalogue. L'accès à des taxonomie depuis des emplacements locaux vous permettra d'accélérer considérablement votre travail.

Fichier d'exemple : Étape 2

En suivant les étapes ci-dessus, importez la taxonomie US-GAAP 2013 comme taxonomie de base. Dans l'assistant à la saisie Aperçu, jetez un coup d'œil à toutes les taxonomies importées et bases de liens référencées. Passez au Mode Texte et recherchez les éléments `xs:import`. Dans la fenêtre principale du mode XBRL, notez que les concepts importés sont indiqués avec une police de couleur grise. Veuillez également noter que l'assistant à la saisie Aperçu énumère les bases de liens et les schémas importés de la taxonomie US-GAAP.

Dans la prochaine étape, nous définirons l'espace de nom cible et verrons comment éditer les espaces de noms de la taxonomie.

17.2.5 Espace de noms

L'espace de noms cible

L'espace de noms cible d'une taxonomie est **défini** dans l'attribut `xs:targetNamespace` de l'élément de la taxonomie `xs:schema` (*voir liste ci-dessous*). (L'élément `xs:schema` se trouve dans l'élément de document du fichier de définitions de concept.)

```
<xs:schema targetNamespace="http://www.altova.com/XBRL/Taxonomies">
      ...
</xs:schema>
```

Outre la définition de l'espace de noms cible (en fait, sa spécification), l'espace de noms cible doit aussi être **déclaré** sur l'élément `xs:schema` de manière à ce qu'il soit utilisable sur toute la longueur du document. La liste ci-dessous déclare l'espace de noms qui est l'espace de noms cible.

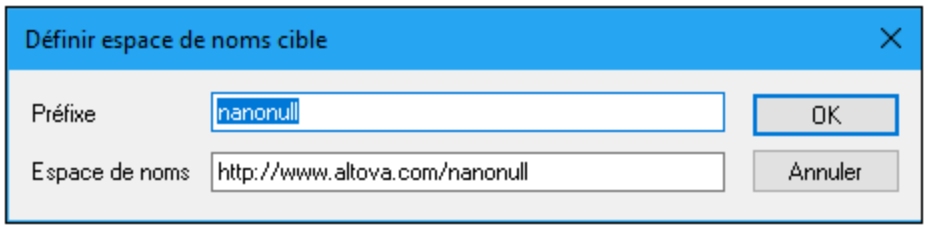
```
<xs:schema targetNamespace="http://www.altova.com/XBRL/Taxonomies"
xmlns:ns1="http://www.altova.com/XBRL/Taxonomies" >
    ...
</xs:schema>
```

Dans la liste ci-dessus, l'espace de noms est déclaré sur l'élément `xs:schema` et reçoit un préfixe de `ns1`.

Configurer l'espace de noms cible

Lorsqu'une nouvelle taxonomie est créée à l'aide de l'Assistant de la nouvelle taxonomie, un espace de noms cible par défaut et un préfixe sont automatiquement créés pour la taxonomie. L'espace de noms cible par défaut est basé sur les données que vous avez saisies dans l'Assistant de la nouvelle taxonomie. Le préfixe de l'espace de noms cible par défaut prendra la forme de `nX`, où `X` est un entier. La déclaration de l'espace de noms cible par défaut est du préfixe peuvent ensuite être édités en accédant au dialogue Définir espace de noms cible (par le biais de la commande **XBRL | Définir espace de noms cible**) et l'éditer à cet endroit (*capture d'écran ci-dessous*). Ces éditions modifieront non seulement la **définition** de l'espace de noms cible (la valeur de l'attribut `targetNamespace`) mais aussi la **déclaration** de l'espace de noms cible.

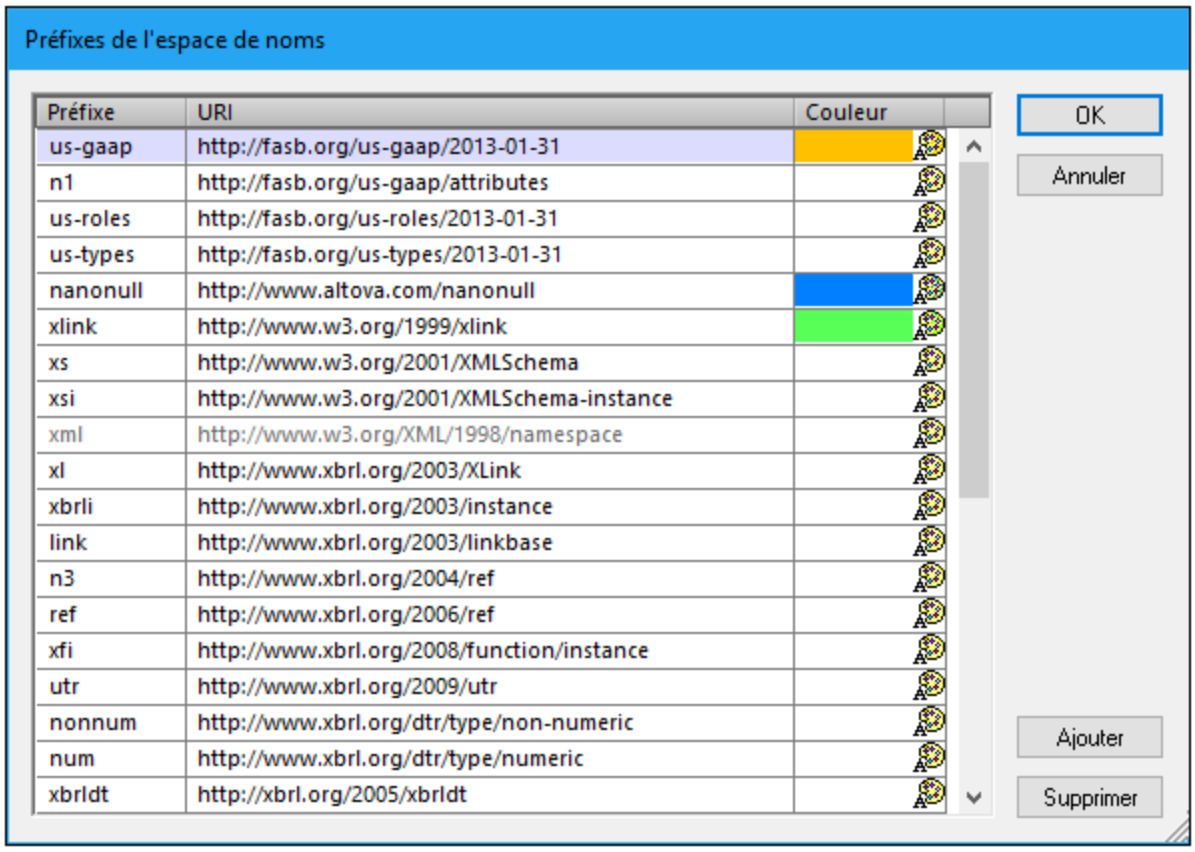
© 2016-2022 Altova GmbH



Pour modifier la déclaration de l'espace de noms cible (mais pas sa définition) ou de la déclaration de tout espace de noms, éditer le préfixe et la valeur de l'espace de noms dans le dialogue Préfixes de l'espace de noms (commande **XBRL | Préfixes espace de noms**).

Espaces de noms de la taxonomie

Les espaces de noms de la taxonomie peuvent être gérés dans le dialogue Préfixes d'espaces de noms (*capture d'écran ci-dessous*), qui est accédé dans le Mode XBRL depuis la commande de menu **XBRL | Préfixes Espace de noms**. Dans le dialogue, vous pouvez déclarer des espaces de noms et associer des préfixes et des couleurs d'arrière-plan pour chaque espace de noms. Les éditions effectuées dans ce dialogue modifient les déclarations des espaces de noms dans la taxonomie.



Le dialogue Préfixes d'espace de noms recense tous les espaces de noms de la taxonomie.

- Pour ajouter ou supprimer un espace de noms, utiliser les touches **Ajouter** ou **Supprimer**, respectivement. Une fois avoir ajouté un espace de noms, éditer le préfixe par défaut et l'URI par défaut en double-cliquant sur le champ respectif et en saisissant les modifications.
- Une couleur peut être attribuée à un espace de noms par le biais de la palette de couleurs disponible pour cet espace de noms. Si une couleur a été attribuée à un espace de noms, tous les composants dans cet espace de noms seront affichés avec cette couleur en arrière-plan dans la Fenêtre principale et les assistants à la saisie du Mode XBRL. Veuillez noter qu'un paramètre de coloris pour un espace de noms donné s'applique pour cet espace de noms sur tous les documents de taxonomie ouverts dans le Mode XBRL.

Lorsque vous avez fini d'éditer le dialogue Espaces de noms, cliquer sur **OK** pour que vos changements prennent effet.

Fichier d'exemple : Étape 3

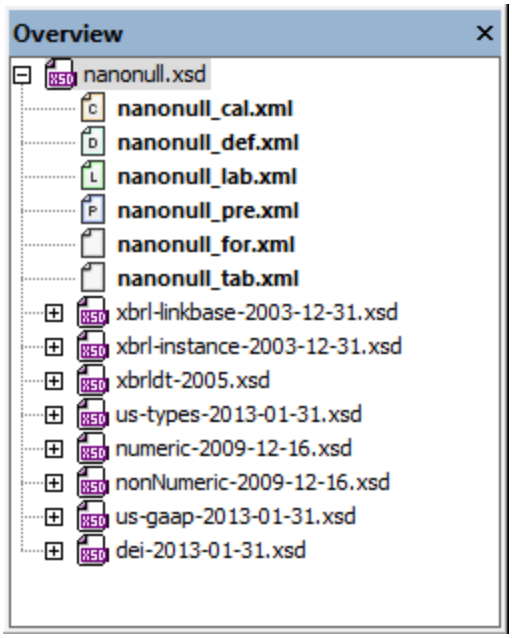
Ouvrir le dialogue Définir espace de noms cible avec la commande **XBRL | Définir espace de noms cible**. Double-cliquez les champs pour éditer. Nous avons utilisé l'espace de noms `http://www.altova.com/nanonull` et lui avons attribué un préfixe de `nanonull` (*voir capture d'écran ci-dessus*). En cliquant sur **OK** dans le dialogue, l'espace de noms cible sera attribué et l'espace de noms cible sera déclaré avec le préfixe que vous avez attribué. Dans notre cas, l'espace de noms cible et le préfixe sont, respectivement, `http://www.altova.com/nanonull` et `nanonull`.

Dans l'étape suivante, nous allons examiner de plus près les fichiers de base de lien et le mécanisme de référence.

17.2.6 Fichiers de taxonomie

L'assistant à la saisie Aperçu affiche dans une structure arborescente les fichiers qui constituent la taxonomie (*capture d'écran ci-dessous*). À la "racine" de l'arborescence se trouve le fichier de taxonomie principal (le fichier de définitions de concept) ; il s'agit du fichier actuellement actif. Les fichiers se trouvant au niveau suivant sont de deux types : (i) des fichiers base de lien qui spécifient les différentes relations dans la taxonomie ; ils sont indiqués par des icônes colorées; et (ii) des schémas importés (les fichiers `.xsd`).

© 2016-2022 Altova GmbH



Dans la section, Importer une taxonomie, vous avez vu comment une taxonomie peut être importée avec l'assistant à la saisie Aperçu. La taxonomie importée est contenue dans la liste des schémas importés dans l'assistant à la saisie Aperçu.

Dans cette section, nous allons vous montrer comment l'assistant à la saisie Aperçu peut être utilisé pour gérer les fichiers de base de lien. Les quatre opérations de gestion des bases de lien sont toutes accessibles par le menu contextuel de l'assistant à la saisie Aperçu. Il s'agit de :

- Ajouter de nouvelles bases de lien et les enregistrer avec la taxonomie.
- Configurer le type de base de lien. Dans les cas où le type de base de lien (calcul, définition, présentation, libellé ou référence) n'est pas affiché dans XMLSpy, le type de base de lien peut être spécifié explicitement.
- Configurer une base de lien en tant que la base de lien par défaut pour un type particulier de base de lien de relation. S'il y a plus d'une base de lien pour un type particulier de relation, par exemple, les relations de libellé, les nouveaux libellés que vous créez dans l'Éditeur de taxonomie seront créés dans la base de lien de libellé par défaut.
- Supprimer des bases de lien.

Note : Il existe cinq types de relations : (i) définition, (ii) calcul, (iii) présentation, (iv) libellé et (v) référence. Des fichiers base de lien séparés peuvent être créés pour chacun de ces types de relation.

Ajouter une nouvelle base de lien

Pour ajouter une nouvelle base de lien, procédez comme suit :

1. Cliquez avec la touche de droite dans l'assistant à la saisie Aperçu et choisir **Ajouter nouvelle base de lien | <relationship type>**. Un nouveau fichier de base de lien du type de relation sélectionné est créé dans l'assistant à la saisie Aperçu avec un nom par défaut. Veuillez noter que la nouvelle base de lien est créée en tant que la base de lien par défaut de ce type de relation (indiqué par le nom de fichier en police grasse).
2. Cliquer avec la touche de droite sur le nom par défaut, choisir **Renommer** et éditer le nom.

3. Un nouveau fichier de base de lien est enregistré physiquement à un endroit particulier uniquement lorsque le fichier principal de taxonomie est enregistré la prochaine fois. Voir ci-dessous pour plus de détails.

Enregistrer les fichiers de base de lien

Si un fichier de base de lien n'a pas été enregistré, cela est indiqué par un astérisque placé après le nom du fichier de base de lien. Lorsque vous enregistrez le fichier principal de taxonomie, les événements suivants se produiront :

1. Le dialogue Confirmer chemins de base de lien apparaît. Ce dialogue contient les noms et les emplacements (chemins) de toutes les bases de lien dans la taxonomie, y compris les fichiers de base de lien récemment créés. Tout fichier base de lien non enregistré aura un chemin par défaut vers le dossier dans lequel le fichier de taxonomie principal sera ou a été enregistré. Vous pouvez éditer le chemin des fichiers de base de lien individuels si vous souhaitez enregistrer un fichier de base de lien dans un autre emplacement. Vous pouvez aussi éditer le nom du fichier.
2. Cliquer sur **OK** une fois terminé. Les fichiers de base de lien seront enregistrés dans les emplacements spécifiques.

Configurer le genre de base de lien

Le type de base de lien d'un fichier (aussi appelé le type de base de lien d'un fichier) peut être configuré en utilisant cette commande. Cliquer avec la touche de droite sur le fichier pour lequel le type de base de lien doit être changé, et, depuis le menu contextuel, sélectionner la commande **Définir le genre de base de lien | <relationship type>**. L'option **Tous** vous permet de spécifier que le fichier de base de lien peut contenir plus d'un type de relation.

Configurer une base de lien par défaut

Un fichier de base de lien par défaut peut être configuré pour chaque type de relation. Lorsqu'une relation de ce type est définie dans l'éditeur de taxonomie, la relation est enregistrée dans le fichier de base de lien par défaut de ce type de relation. Pour définir un fichier de base de lien en tant que la base de lien par défaut, cliquer avec la touche de droite et choisir **Configurer base de lien par défaut**. Les noms des bases de lien par défaut sont affichés en gras.

Supprimer une base de lien

Une base de lien peut être supprimée de la taxonomie en la cliquant avec la touche de droite et en choisissant **Supprimer**.

Fichier d'exemple : Étape 4

Ajoutez les bases de lien en suivant la procédure décrite ci-dessus. Les fichier de base de lien sont :

- Base de lien de Calcul : `nanonull_cal.xml`
- Base de lien de Définition : `nanonull_def.xml`
- Base de lien de Libellé : `nanonull_lab.xml`
- Base de lien de Présentation : `nanonull_pre.xml`

Si vous ne voulez pas ajouter des fichiers de lien de base, vous pouvez travailler avec la taxonomie `Nanonull.xsd` (située dans `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\XBRLExamples\Nanonull`), qui contient déjà des fichiers de base de lien.

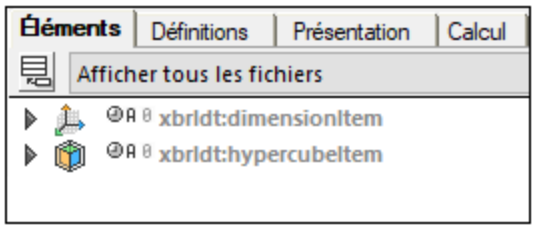
© 2016-2022 Altova GmbH

Afin de tester certaines des commandes introduites dans cette section, suivez les étapes suivantes : créer un fichier base de lien en utilisant la commande **Ajouter nouvelle base de lien** et en créant le type de base de lien que vous souhaitez. Le renommer comme décrit ci-dessus. Veuillez noter que la base de lien récemment créée devient la base de lien par défaut de ce type de relation (indiquée par le fait que son nom s'affiche en caractères gras). Le sélectionner et le configurer de manière à ce qu'il devienne un autre type de relation (avec la commande **Définir genre de base de lien**). Veuillez noter que le fichier n'est **pas** la base de lien par défaut de ce nouveau type de relation. À présent, supprimer la base de lien (avec la commande **Supprimer**). Puisqu'un des fichiers de base de lien d'origine n'est maintenant plus une base de lien par défaut, configurer un fichier de cette relation en tant que la base de lien par défaut de ce type de relation.

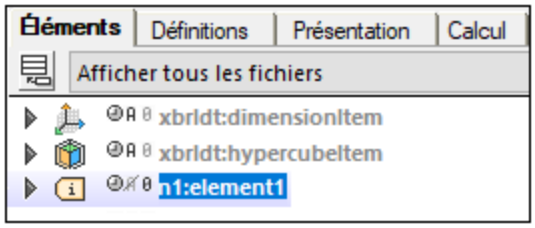
Dans l'étape suivante, nous allons ajouter de nouveaux éléments au fichier de taxonomie principal (ou fichier de définitions de concept).

17.2.7 Ajouter des éléments à une taxonomie

Pour ajouter un élément (concept) à la taxonomie, cliquer sur l'icône **Ajouter nouvel élément** dans la Fenêtre principale (*capture d'écran ci-dessous*).

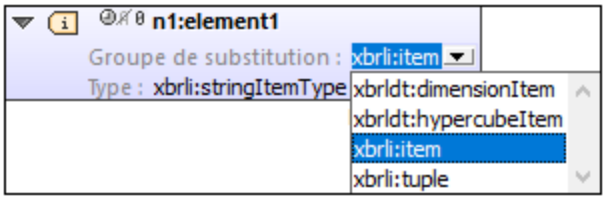


Le nouvel élément avec un groupe de substitution de `xbrli:item` et avec un nom par défaut, est ajouté à la liste des éléments dans l'écran (*capture d'écran ci-dessous*).



Pour une description de la fenêtre d'élément, voir Fenêtre principale : onglet Éléments. Vous pouvez maintenant éditer les propriétés de l'élément dans la fenêtre principale de plusieurs manières.

- Le nom de l'élément peut être modifié en double-cliquant sur le nom par défaut et en saisissant le nom correct. Veuillez noter que vous devez aussi saisir le préfixe d'espace de noms correct pour le nom.
- Le groupe de substitution de l'élément peut être changé en agrandissant la fenêtre d'élément (cliquer sur l'icône fléchée pour ce faire) et puis, sélectionner le groupe de substitution requis depuis la liste déroulante de ce champ (*capture d'écran ci-dessous*).



- Pour changer la propriété Balance, Period, Abstract, ou Nillable, cliquer sur l'icône correspondante située à gauche du nom d'élément et choisir une des options dans la fenêtre qui s'affiche.
- Pour ajouter un rôle de lien de libellé pour l'élément, cliquer avec la touche de droite n'importe où dans la fenêtre d'élément et choisir la commande **Ajouter rôle de lien de libellé**. Une ligne pour le rôle de lien de libellé est ajoutée ; dans cette ligne, vous pouvez saisir le rôle de lien de libellé ou choisir une option dans la liste de choix. Veuillez noter que si aucun fichier de base de lien de libellé n'est associé avec la taxonomie, un fichier sera créé maintenant et sera affiché dans l'assistant à la saisie Aperçu.
- Un libellé peut être ajouté pour un rôle de lien de libellé en cliquant avec la touche de droite sur le rôle de lien de libellé et en choisissant la commande **Ajouter libellé**. Pour saisir les détails du libellé, soit double-cliquer dans le champ que vous souhaitez éditer et saisir la nouvelle valeur, soit sélectionner la nouvelle valeur depuis les listes de choix respectives. Les changements que vous effectuez au libellés seront enregistrés dans la base de lien du libellé lorsque la taxonomie principale sera enregistrée.
- Les références sont ajoutées à la base de lien de référence de la même manière que les libellés sont ajoutés à la base de lien du libellé. Tout d'abord, un rôle de lien de référence est ajouté pour l'élément, puis une référence est ajoutée pour un rôle de lien de référence spécifique.

Les propriétés d'élément peuvent aussi être éditées dans l'assistant à la saisie Détails. Voir Assistants à la saisie dans le Mode XBRL pour une description de la procédure.

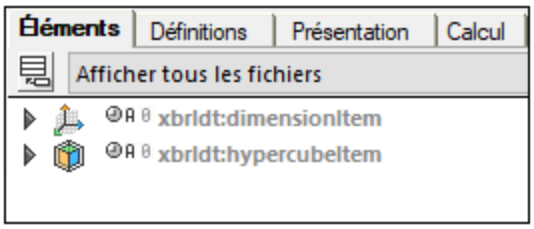
Fichier d'exemple : Étape 5

Dans cette section, nous allons étendre la taxonomie US-GAAP en créant de nouveaux éléments.

Le premier élément que nous allons créer est l'item `nanonull:OnboardAndOther`, qui représente les recettes provenant des ventes effectuées à bord des bateaux de croisière de Nanonull. Ce poste de recettes spécifique n'est pas disponible dans la taxonomie US-GAAP ; c'est pourquoi il doit être créé en tant qu'une extension d'US-GAAP. En tant qu'un nouvel élément créé spécialement pour la taxonomie Nanonull, il doit être créé dans l'espace de noms Nanonull (`http://www.altova.com/nanonull`) qui a été déclaré avec un préfixe de `nanonull`. La création de l'élément avec ce préfixe placera cet élément dans l'espace de noms Nanonull.

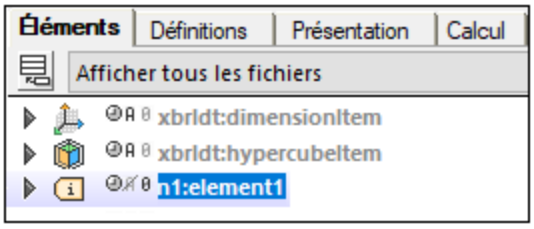
Pour créer l'élément, procéder comme suit :

1. Cliquer sur l'icône **Ajouter nouvel élément** dans la Fenêtre principale (*capture d'écran ci-dessous*).

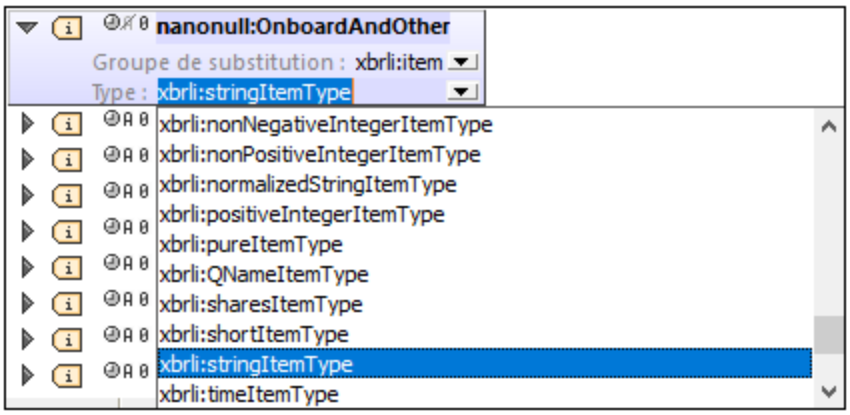


   Un nouvel élément avec un groupe de substitution de `xbrli:item` et avec un nom par défaut est ajouté à la liste des éléments dans l'écran (*capture d'écran ci-dessous*).

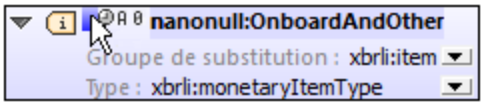
© 2016-2022 Altova GmbH



2. Double-cliquer sur le nom d'élément et saisir le nom `nanonull:OnboardAndOther` (*capture d'écran ci-dessous*). Cela permet de créer l'élément `OnboardAndOther` dans l'espace de noms Nanonull.
3. Étendre la fenêtre d'élément et, puisque l'élément contiendra une valeur monétaire, changer l'attribut `Type` dans `xbrli:monetaryItemType` (*capture d'écran ci-dessous*).



4. À présent, cliquer à gauche de l'icône de l'horloge et, depuis la fenêtre qui apparaît, choisir `credit` (*capture d'écran ci-dessous*).



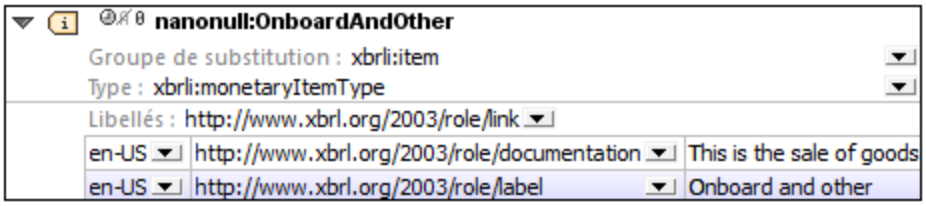
Cela permet de définir la valeur de l'attribut `xbrli:balance` à `credit`.

5. Cliquer sur les icônes de l'horloge, `A`, et `0`, et configurer les valeurs des attributs `xbrli:duration`, `xs:abstract`, et `xs:nillable` sur *duration, NON Abstrait, Nillable*, respectivement. (Dans le fichier `.xsd`, les valeurs d'attribut réelles seront : `credit`, `duration`, `false`, et `true`, respectivement.)
6. Cliquer avec la touche de droite sur la fenêtre élément et, depuis le menu qui s'affiche, choisir **Ajouter Rôle de lien de libellé**. Cela permet de créer un rôle de lien de libellé en bas de la fenêtre d'élément (*capture d'écran ci-dessous*).



7. Choisir le lien XBRL URI.
8. Cliquer avec la touche de droite la ligne de rôle de lien de libellé et depuis le menu qui s'affiche, sélectionner **Ajouter libellé**. Cela permet de créer une ligne de libellé dans le rôle de lien de libellé.
9. Double-cliquer dans le champ de langue de la ligne de libellé récemment créée (*capture d'écran ci-dessous*) et saisir `en-us` ; dans le champ suivant qui est le champ de rôle de lien, choisir le rôle

`documentation` depuis la liste déroulante ; dans le champ libellé, saisir le texte qui devrait apparaître dabs la documentation. Ensuite créer une nouvelle ligne de libellé pour le rôle de lien `label` en répétant l'étape 9. Une fois que l'affichage d'un élément a été étendu (en cliquant la tête de flèche à sa gauche), l'affichage du rôle `label` peut être activé ou désactivé en cliquant sur le symbole +/- que vous trouverez à droite du libellé (Afficher/Dissimuler Libellés).



L'élément `nanonull:OnboardAndOther` a été créé avec succès.

Veuillez noter que `OnboardAndOther` avait une valeur de crédit `xbrli:balance`. Cela est dû au fait qu'il s'agit d'un item de recette : de l'argent entre dans la caisse. Étant donné que les items vendus à bord sont soumis à des coûts (par ex. les coûts de production de l'entreprise), nous allons aussi créer un élément du côté du débit appelé `nanonull:CostOfOnboardAndOther`. Créer cet élément de la même manière que `nanonull:OnboardAndOther`, à une seule différence près : définir la valeur de `xbrli:balance` sur `debit` au lieu de `credit`.

Un autre coût à inclure concerne les commissions pour les agents. Cela doit être effectué par un élément de débit nommé `nanonull:CruiseCommissionsTransportationAndOther`. Créer cet élément exactement de la même manière que vous avez créé `nanonull:CostOfOnboardAndOther`.

Enfin, nous ajoutons trois éléments abstraits, `Asia`, `Europe`, et `US`, afin que les concepts puissent être regroupés par région. Puisque les éléments sont utilisés uniquement à des fins de regroupement et ne contiendront pas eux-même de valeur, ils sont connus en tant qu'éléments abstraits. C'est pourquoi le genre de type d'un tel élément est immatériel. Il vaut mieux donner à un élément abstrait un type qui correspond à sa sémantique. Par exemple, nous avons donné aux éléments abstraits `Asia`, `Europe`, et `US`, un type de `stringItemType`. Créer les éléments `nanonull:Asia`, `nanonull:Europe`, et `nanonull:USA` comme vous avez créé les éléments précédents. La seule différence cette fois est que la valeur de l'attribut Abstract doit être définie sur *Abstract* (la valeur d'attribut réelle dans le fichier XSD sera `true`) et il n'y aura pas d'attribut `xbrli:balance`.

Note : si un attribut `xbrli:balance` est présent sur un élément abstrait, celui-ci devra être du type `monetaryItemType`, sinon, la taxonomie sera invalide. Il vaut mieux omettre l'attribut optionnel `xbrli:balance` de tous les éléments abstraits.

Dans l'étape suivante, nous allons spécifier le rôle de liens pour la nouvelle taxonomie. Ces rôles de liens seront nécessaires lorsque nous créerons de nouvelles relations.

17.2.8 Relations et rôles de lien

Lorsqu'un ensemble de relations est créé, ces relations sont créées dans le cadre d'un élément contenant. Par exemple, lorsque des relations de définition sont créées, les éléments définissant les relations de définition (les arcs locateurs et de définition) sont tous créés dans le cadre d'un élément `definitionLink`, qui ressemble à :

```
<link:definitionLink xlink:type="extended"
```

 © 2016-2022 Altova GmbH

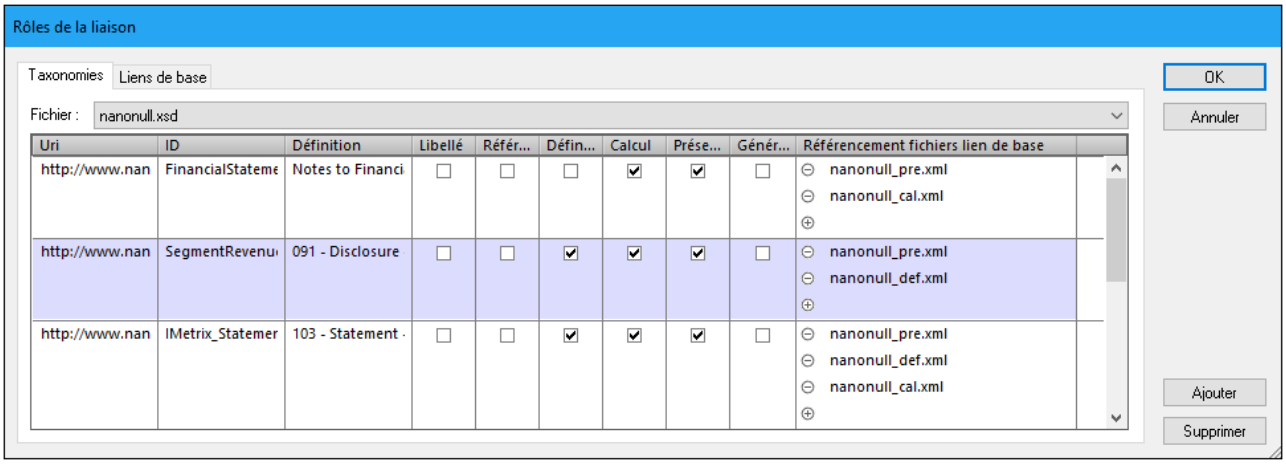
```
xlink:role="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome">
```

La valeur de l'attribut `xlink:role` dans le lien de définition (comme dans le lien de définition ci-dessus) doit avoir la valeur de l'attribut `roleURI` d'un des rôles de lien définis pour être utilisés sur des relations de définition (*voir liste ci-dessus*). Un rôle de lien (*comme dans la liste ci-dessous*) est contenu dans l'élément `appinfo` de la taxonomie.

```
<xs:appinfo>
   <link:roleType id="SegmentRevenueAndOperatingIncome"
   roleURI="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome">
      <link:definition>006091 - Disclosure - Segment Revenue and Operating
Income</link:definition>
      <link:usedOn>link:calculationLink</link:usedOn>
      <link:usedOn>link:definitionLink</link:usedOn>
      <link:usedOn>link:presentationLink</link:usedOn>
   </link:roleType>
</xs:appinfo>
```

Un rôle de lien peut être utilisé dans les éléments contenant d'autres types de relations en dehors des éléments `definitionLink` (par exemple, dans les éléments `calculationLink` et `presentationLink`). Dans la liste ci-dessus, veuillez noter qu'il existe des éléments `usedOn` qui spécifient dans quel type de relatons ce rôle de lien peut être utilisé.

Pour créer des liens de rôle dans un fichier de définitions de concept (fichier de taxonomie principal) dans le Mode XBRL, cliquez sur la commande de menu **XBRL | Rôles de lien**. Le dialogue Rôles de lien s'ouvre (*capture d'écran ci-dessous*).



Dans l'onglet taxonomies, choisissez le fichier de taxonomie depuis la liste déroulante dans la liste de choix *Fichier* et cliquez sur **Ajouter** pour ajouter un rôle de lien. Ensuite, spécifiez l'URI et l'ID du rôle de lien (*se référer à la liste ci-dessus*). À présent, spécifiez pour quels types de relations ce rôle de lien doit être disponible ; pour ce faire, cochez les cases des types de relation requis (*voir capture d'écran ci-dessus*).

Fichier d'exemple : Étape 6

Créer deux rôles de lien par le biais du dialogue Rôles de lien (**XBRL | Rôles de lien**) tel que décrit ci-dessus et affiché dans la capture d'écran ci-dessus :

1. `id="SegmentRevenueAndOperatingIncome"` `URI="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome"` (à utiliser dans les relations de définition, calcul et présentation)
2. `id="FinancialStatements"` `URI="http://www.nanonull.com/taxonomy/role/FinancialStatements"` (à utiliser dans les relations de calcul et présentation)

Dans l'étape suivante, nous allons créer des relations pour la nouvelle taxonomie.

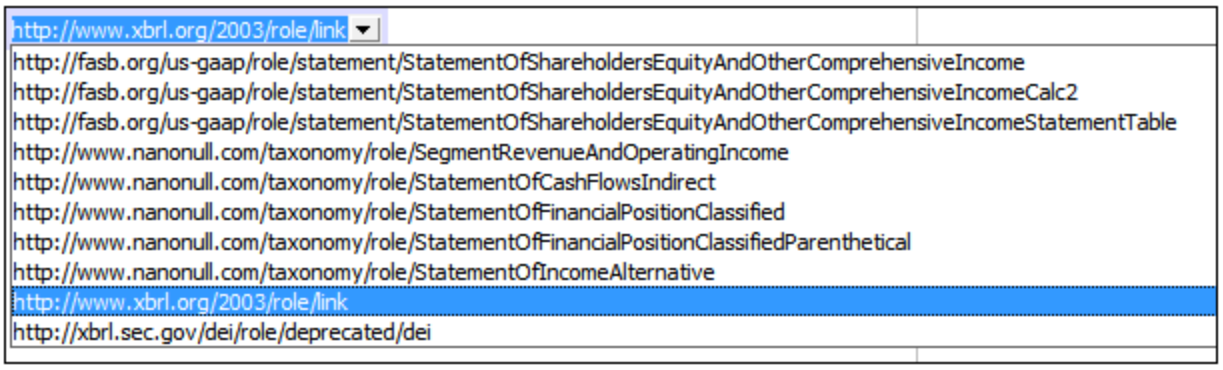
17.2.9 Créer des relations : partie 1

Les relations sont créées dans leurs onglets respectifs : Définitions, Présentation, Calcul. La façon dont les trois types de relation sont créés est semblable ; la principale différence étant que les relations de définition ont des rôles d'arc, alors que les relations de présentation et les relations de relation n'ont pas de rôles d'arc. Dans cette section, nous décrivons comment créer des relations en utilisant des relations de définition. Dans la section suivante, nous expliquons comment les relations de présentation et de calcul sont différents, et nous examinons les autres fonctions concernant les relations.

Avant de commencer à lire la description ci-dessous, nous vous recommandons d'ouvrir une taxonomie terminée dans le Mode XBRL. Vous trouverez la taxonomie Nanonull (`nanonull.xsd`) dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\XBRLExamples\Nanonull`.

Ajouter le rôle de lien

Cliquer sur l'onglet de relations requis dans la Fenêtre principale (Définitions, Présentation, Calcul). Ensuite, cliquer avec la touche de droite dans la Fenêtre principale et sélectionner la commande **Ajouter Rôle de lien étendu**. Cela permet d'ajouter une ligne contenant l'URI d'un rôle de lien par défaut (*capture d'écran ci-dessous*). Cliquer sur la flèche de déroulement située à droite de cette ligne pour afficher une liste des rôle de liens disponibles et sélectionner le rôle de lien requis.



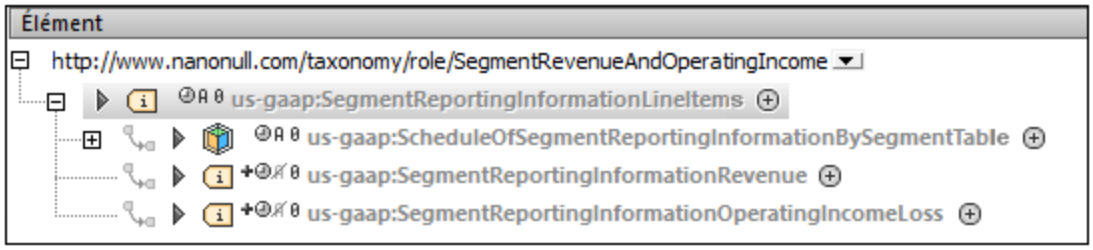
Si le rôle de lien requis n'est pas disponible, cela est dû au fait qu'il n'a pas été défini ni dans la taxonomie ni pour le type de relation actuel. Voir Relations et rôle de liens pour plus de détails concernant le rôle de liens et comment les créer.

N'importe quelle quantité de rôles de liens peut être ajoutée.

© 2016-2022 Altova GmbH

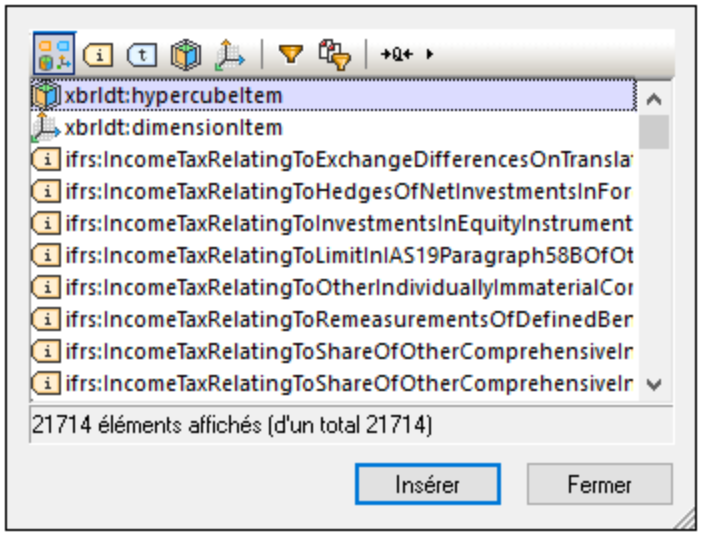
Insérer les références d'élément et les arcs dans le cadre d'un rôle de lien

Le premier élément à créer dans le cadre d'un rôle de lien est un élément **à partir duquel** une relation sera créée vers un autre élément (*voir capture d'écran ci-dessous*). Généralement, il s'agira d'un élément abstrait qui regroupe d'autres éléments (par exemple, un élément pour un compte de résultat). Cet élément n'aura pas d'entrées dans la colonne arcrole parce qu'il se trouvera du côté "de" d'un arc. Les arcroles sont regroupés sur les éléments du côté "vers" d'un arc.

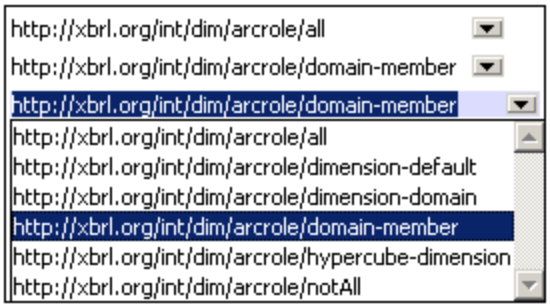


Dans la capture d'écran ci-dessus, l'élément marqué est la référence d'élément insérée. Il a trois arcs, un menant à un élément hypercube et deux menant vers des éléments d'item. Ces trois éléments se trouvent à l'extrémité "vers" de leurs arcs respectifs et la relation de-vers est définie par les arcroles correspondants, qui sont affichés dans la colonne Arcrole.

Pour insérer un arc sur une référence d'élément ou un élément, cliquer avec la touche de droite dans l'élément "de" et choisir **Insérer Arc** depuis le menu contextuel qui s'affiche. Cela entraîne l'affichage du dialogue Insérer arc (*capture d'écran ci-dessous*). Choisir l'élément à créer à l'extrémité "vers" de l'arc. Pour filtrer l'affichage dans ce dialogue, activer le filtre et spécifier une condition pour le filtre (voir Assistants à la saisie dans le Mode XBRL pour une description de la procédure).



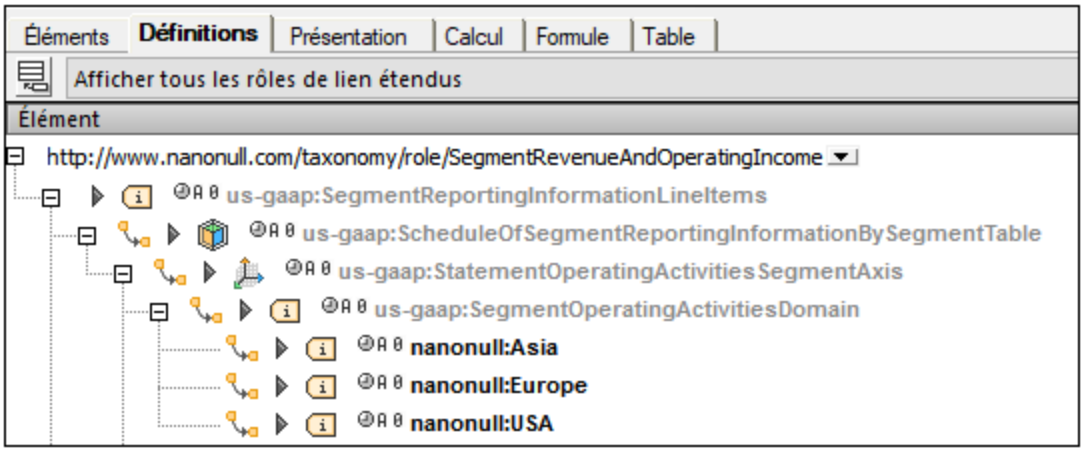
L'élément sera inséré avec un arcrole par défaut. Vous pouvez changer l'arcrole en choisissant une alternative depuis la liste déroulante de l'arcrole (*capture d'écran ci-dessous*).



Note : Les éléments, avec les arcs, peuvent aussi être ajoutés en les glissant depuis l'assistant à la saisie Éléments globaux.

Fichier d'exemple : Étape 7

Créer des relations de définition comme affiché dans les captures d'écran ci-dessous en utilisant la méthode décrite ci-dessus.



La capture d'écran ci-dessus montre les éléments à ajouter avec les arcs. La capture d'écran ci-dessous ci-dessous montre les arcroles des éléments récemment ajoutés.

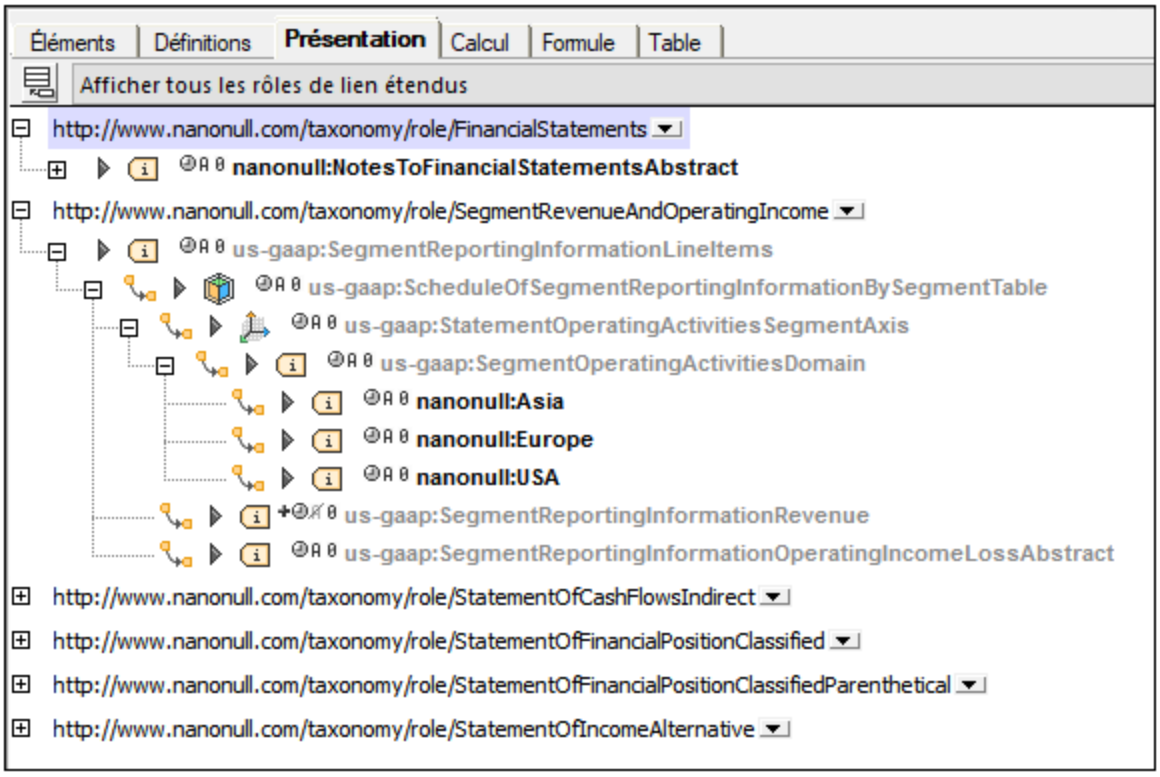


Vous pouvez comparer la taxonomie que vous avez créée avec celle fournie avec votre pack XMLSpy. La taxonomie fournie (`nanonull.xsd`) se trouve dans le dossier `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Examples\XBRLExamples\Nanonull`.

 © 2016-2022 Altova GmbH

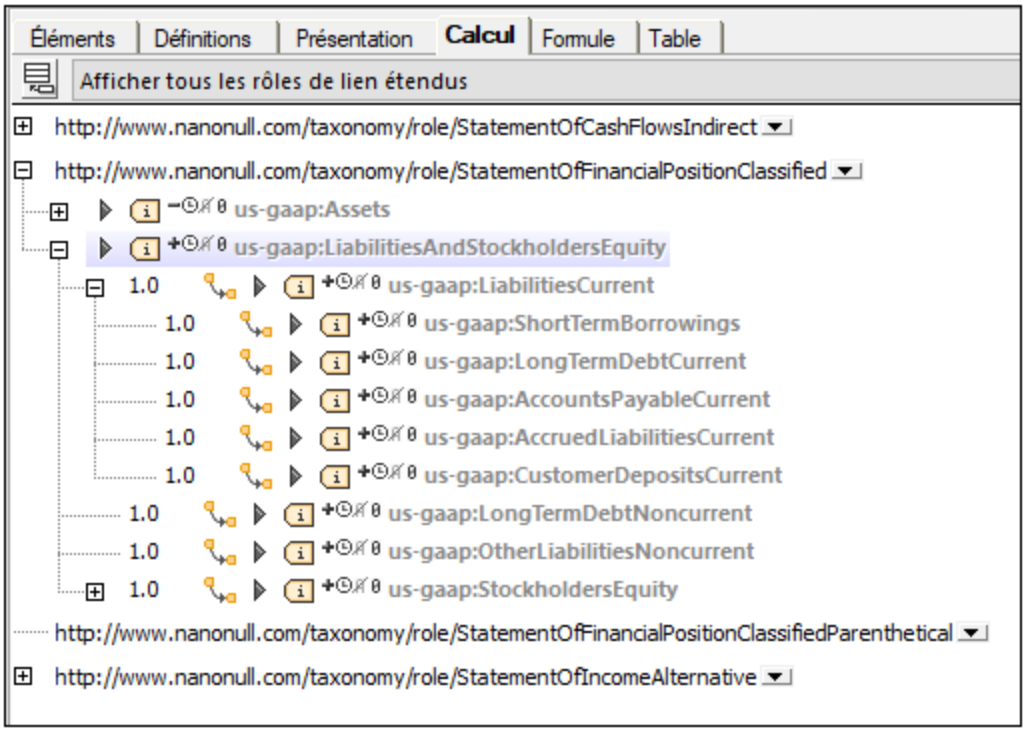
17.2.10 Créer des relations : partie 2

Dans la section précédente, Créer des relations : Partie 1, nous avons expliqué comment créer des relations en utilisant les relations de définition pour démontrer le mécanisme. Les relations de Présentation (*capture d'écran ci-dessous*) et les relations de calcul sont créés de manière semblable. La seule différence est qu'il n'y a pas de colonne Arcrole dans les relations de présentation et de calcul.



Veuillez noter les points suivants :

- Les relations de présentation et de calcul peuvent être considérées être un arc simple entre deux éléments à la manière de relations parent-enfant. Les icônes d'arc symbolisent cette relation. Donc le fait d'insérer un arc dans un élément revient à créer un élément enfant dans la représentation graphique. Donc, en utilisant des arcs, vous pouvez établir une hiérarchie.
- Il est également possible de glisser des éléments depuis les assistants à la saisie des éléments globaux dans l'arborescence. Ces éléments sont toujours déposés à la position "vers" d'un arc. Une flèche apparaît lorsque l'élément se trouve dans la position où elle doit être déposée.
- Les arcs de calcul ont des attributs `weight` qui indiquent comment la valeurs de l'élément "vers" dans l'arc doit être totalisé (*voir capture d'écran ci-dessous*). Par exemple, une valeur de poids de `+1.0` indique que 100% de la valeur des éléments doit être ajoutée vers la valeur de l'élément "de" (ou somme). Une valeur de `-1.0` indique que 100% de la valeur devrait être soustraite depuis la valeur de l'élément de somme. Double-cliquer sur la valeur d'attribut `weight` vous permet de saisir une valeur optionnelle.



L'attribut `weight` peut aussi être modifié dans l'assistant à la saisie Détails (*voir ci-dessous*).

Empêcher l'utilisation de l'arc

Tous les arcs, qu'il s'agisse de la définition, de la présentation ou du calcul, ont un attribut `use` qui peut prendre une valeur `optional` ou `prohibited`. Lorsque la valeur `prohibited` est utilisée, l'arc sera invalidé.

Couleur et menu contextuel

Lorsque des éléments ont été créés dans la taxonomie actuelle et peuvent être édités, ils sont affichés en noir. Sinon (lorsqu'ils proviennent de taxonomies importées, qui ne doivent pas être éditées), les éléments sont affichés en gris.

Les entrées suivantes apparaissent dans les menus contextuels dans la fenêtre principale des onglets de relation.

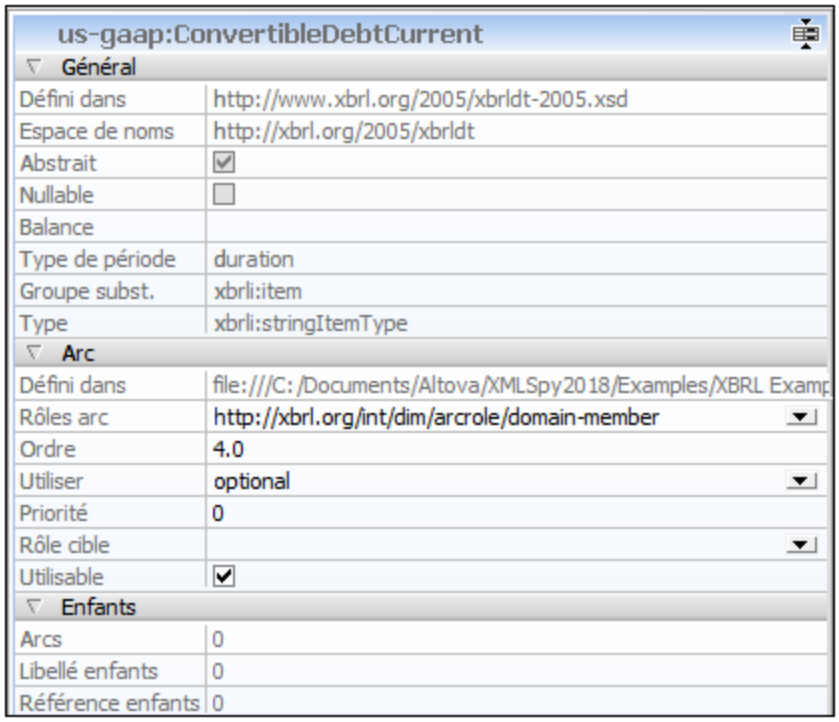
- *Insérer référence de l'élément :* disponible dans les rôles de lien étendus. Ajoute un élément sous le rôle de lien qui sera toujours à l'extrémité "de" des arcs.
- *Supprimer référence de l'élément :* disponible dans les références d'élément immédiatement sous un rôle de lien.
- *Insérer arc:* disponible dans les éléments. Insère un arc et affiche un dialogue dans lequel l'élément qui sera dans l'extrémité "vers" de l'arc peut être sélectionné.
- *Définir rôle cible :* définit un rôle cible dans l'élément sélectionné.
- *Ajouter rôle de lien de libellé :* ajoute un rôle de lien de libellé à l'élément sélectionné.
- *Ajouter rôle de lien de référence :* ajoute un rôle de lien de référence à l'élément sélectionné.
- *Substituer arc :* remplace la valeur (implicite) `optional` de l'attribut `use` de l'arc par la valeur `prohibited`, invalidant donc l'arc.
- *Supprimer arc :* supprime l'arc sélectionné.

© 2016-2022 Altova GmbH

- *Afficher dans éléments globaux :* marque l'élément sélectionné dans l'assistant à la saisie Éléments globaux.

Assistant à la saisie Détails

Lorsqu'un élément dans une relation est sélectionné, les attributs d'arc peuvent être édités dans l'assistant à la saisie Détails (*capture d'écran ci-dessous*).



Les attributs qui ne peuvent pas être édités dans l'affichage graphique de la fenêtre principale, comme `order` et `priority`, peuvent être édités dans l'assistant à la saisie Détails.

17.3 Procédures supplémentaires

La section Procédures supplémentaires rassemble uns série de fonctions diverses :

- Libellés préférés
- Domaines typés
- Détection des duplicata et dé-duplication
- Inline XBRL

17.3.1 Libellés préférés

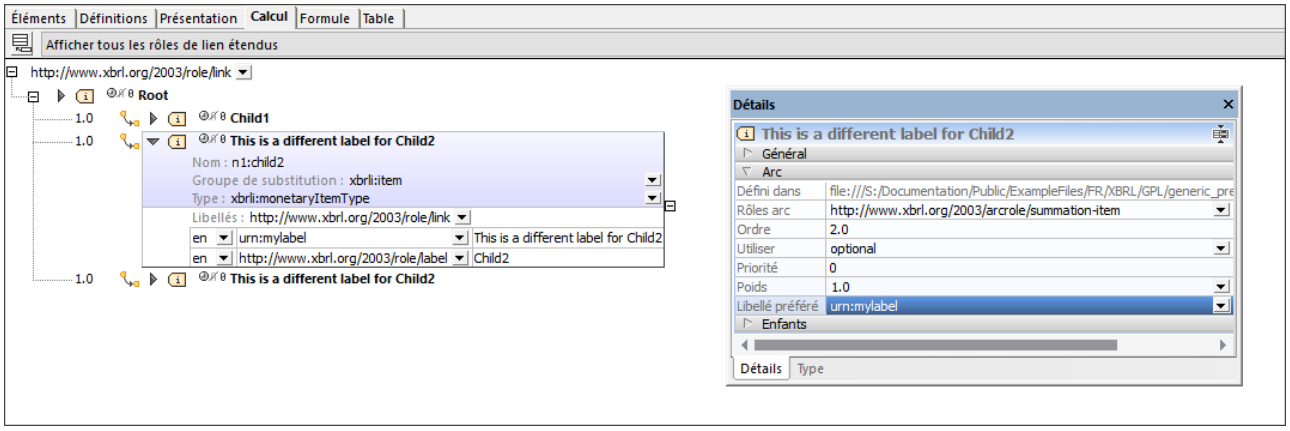
Plusieurs libellés peuvent être attribués à un concept ou une ressource générique (formules, tables, etc). Dans un arc de relation, le libellé souhaité du nœud cible/enfant est sélectionné par le biais de ce rôle de libellé du libellé en question. Le mécanisme utilisé est le suivant :

- Si un libellé préféré est défini sur l'arc de relation, ce libellé préféré sera utilisé
- Si aucun libellé préféré n'a été défini pour une relation, le libellé par défaut sera utilisé

La spécification add-on Generic Preferred Label 1.0 Recommendation permet à l'attribut **`gpl:preferredLabel`** d'être défini sur toute définition d'arc dans des définitions, et dans des calculs, des formules, et des liens de base de table. La valeur de l'attribut est le rôle de libellé qui doit être utilisé pour sélectionner le libellé du nœud de cible. Par conséquence, les libellés préférés sont pris en charge non seulement pour les relations de présentation (activés par les spécifications précédentes), mais également pour d'autres relations (comme des relations de calcul).

Libellés préférés
Les libellés préférés sont définis en choisissant la relation dans l'onglet principal, puis, dans l'assistant à la saisie des Détails, en choisissant le rôle de libellé désiré en tant que la valeur de la propriété *Libellé préféré* (*voir capture d'écran ci-dessous*). Dans la capture d'écran ci-dessous, veuillez noter que la deuxième relation de calcul comporte deux libellés qui ont été définis pour elle. Dans l'assistant à la saisie des Détails, le libellé préféré a été défini sur `urn:mylabel`. Il s'agit donc du libellé qui est utilisé en tant que le libellé de la relation. (Dans XMLSpy, vous pouvez spécifier, dans les Paramètres de Mode XBRL, que le libellé (au lieu des noms) est indiqué dans l'affichage des concepts et/ou des ressources.)

© 2016-2022 Altova GmbH



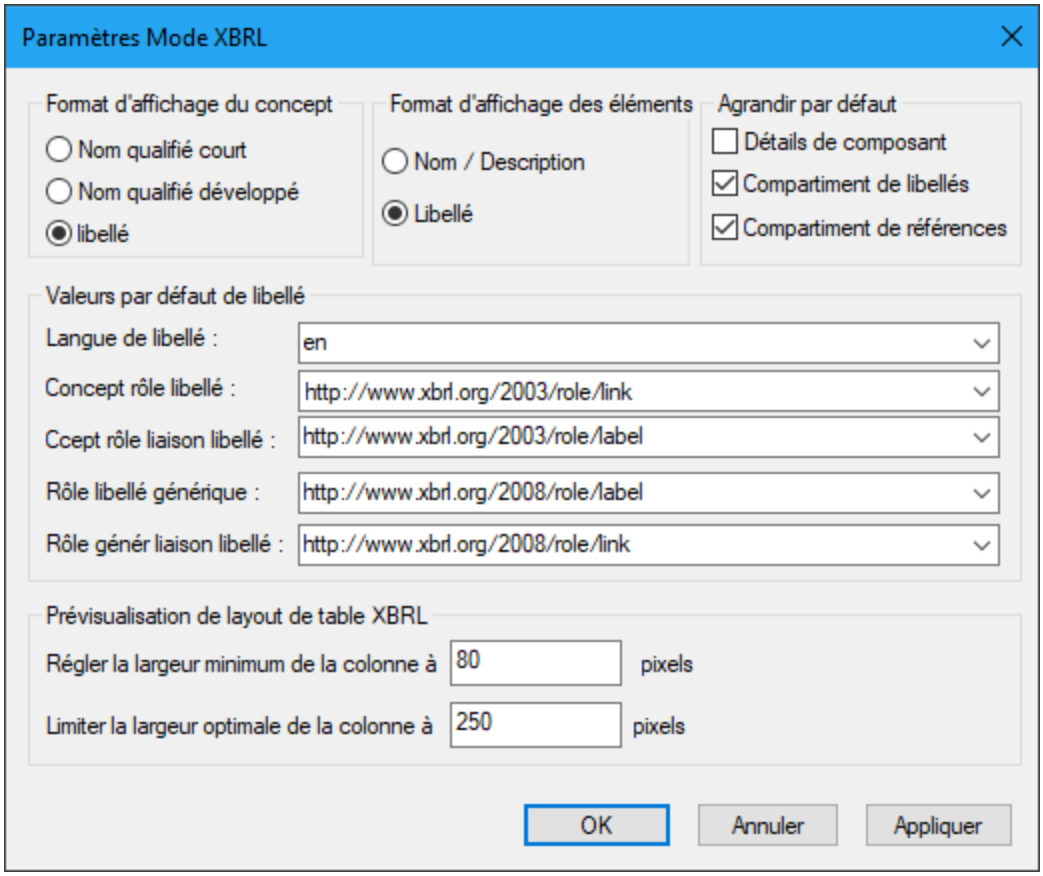
Note : Dans les bases de lien de Table, la propriété *Libellé préféré* ne sera pas affichée pour les nœuds de relation, les nœuds aspect, et les nœuds de règle de fusion puisqu'il n'est pas possible de spécifier des libellés pour ces nœuds.

Libellés par défaut

Si la propriété *Libellé préféré* d'un concept ou d'une ressource n'est pas définie, le libellé par défaut est utilisé. Les libellés par défaut des concepts et des ressources sont spécifiés dans le dialogue Paramètres de Mode XBRL.

Paramètres de libellé dans les paramètres de Mode XBRL

Le dialogue Paramètres de Mode XBRL (*voir capture d'écran ci-dessous*) est accessible par le biais de la commande de menu **XBRL | Paramètres Mode**.



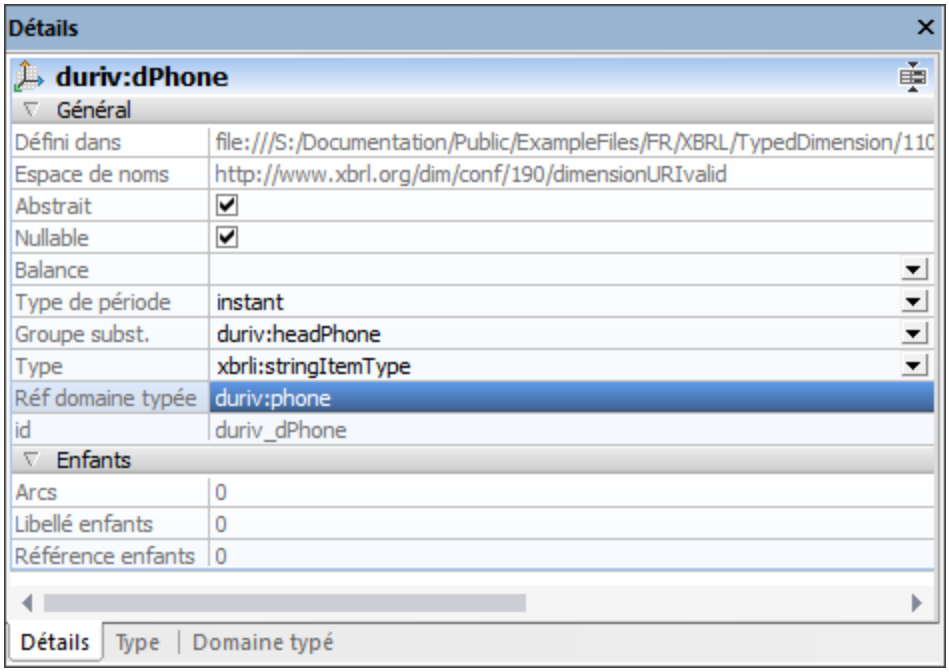
Il existe deux paramètres dans ce dialogue qui sont pertinents pour les libellés :

- Vous pouvez spécifier que les concepts et/ou les ressources sont affichés avec leurs libellés au lieu de leurs noms.
- Vous pouvez définir des défauts pour les libellés de concept et les libellés génériques (qui sont utilisés pour des ressources).

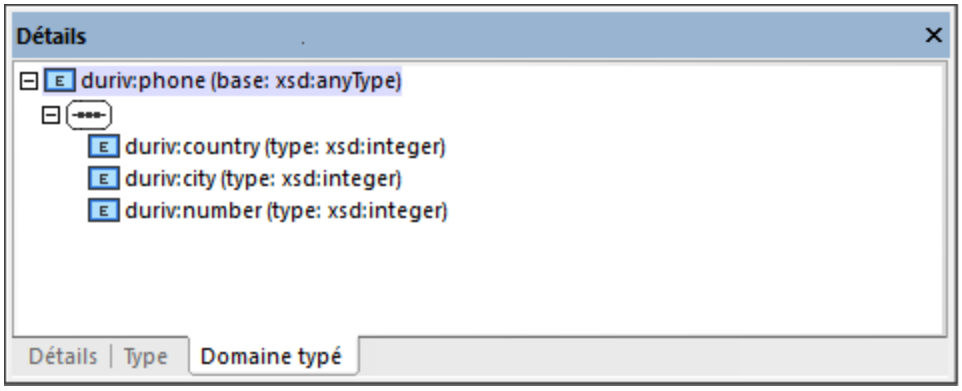
17.3.2 Typed Domains

Un domaine typé est la déclaration d'élément qui est référencée par une dimension typée. Dans la capture d'écran de l'assistant à la saisie des Détails ci-dessous, par exemple, vous pouvez constater que la dimension typée **`duriv:dPhone`** référence le domaine typé **`duriv:phone`**.

© 2016-2022 Altova GmbH



Pour consulter des informations concernant le domaine typé référencé, cliquer sur l'onglet Domaine typé (*voir capture d'écran ci-dessous*). Vous pouvez utiliser les commandes de menu contextuelles des items dans cet onglet pour ouvrir l'item sélectionné dans le Mode Schéma ou copier son emplacement dans le presse-papiers.



Note : L'onglet Domaine typé apparaît uniquement lorsqu'une dimension typée est sélectionnée qui référence le domaine typé.

17.3.3 Duplicate Detection and De-Duplication

La prise en charge suivantes de la gestion des faits dupliqués dans les documents d'instance XBRL est disponible :

- Dans les documents d'instance XBRL, une liste des faits dupliqués peut être détectée et recensée dans la fenêtre Messages. Exécuter la commande de menu **XBRL | Détecter faits dupliqués** ou

XBRL | Détecter faits dupliqués sur Serveur (haute performance). Voir la description des commandes pour plus d'informations.

- Lors de la validation d'un document d'instance XBRL, vous pouvez spécifier que les faits dupliqués qui touchent les calculs soient ignorés pour la validation. Les paramètres nécessaires sont disponibles en tant qu'option de validation XBRL.
- Lorsque les formules sont exécutées, les tables générées ou l'XBRL Inline transformé, les faits dupliqués existants peuvent être ignorés. Cela peut être spécifié dans le dialogue d'options de traitement XBRL.

Les faits dupliqués sont déterminés sur la base des règles définies dans la spécification Handling Duplicate Facts in XBRL and Inline XBRL 1.0.

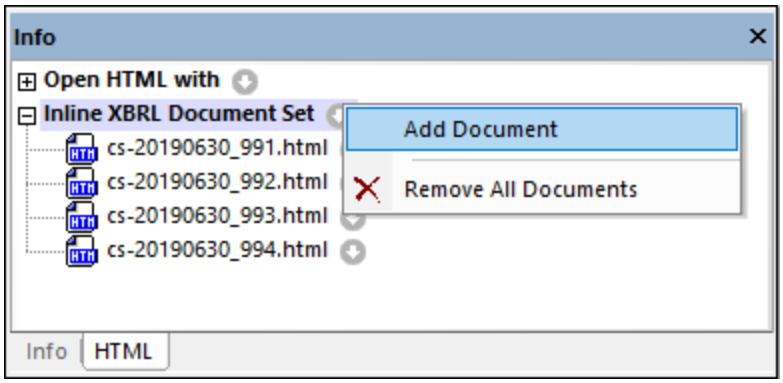
17.3.4 Inline XBRL

Les documents XBRL inline sont des documents HTML qui contiennent des données XBRL (qui sont marquées par des balises XBRL). Vous pouvez valider le document XBRL Inline actif (avec la commande **XML | Valider**) et le traiter (avec **XBRL | Transformer Inline XBRL**). Lorsque vous traitez les documents, vous extrayez les données XBRL depuis le document HTML.

Plusieurs documents XBRL Inline

Vous pouvez aussi valider/traiter plusieurs documents XBRL Inline. Procéder comme suit :

1. Ouvrir le document XBRL Inline principal pour qu'il soit le document actif. L'onglet HTML apparaît dans la fenêtre Info *(voir capture d'écran ci-dessous)*.



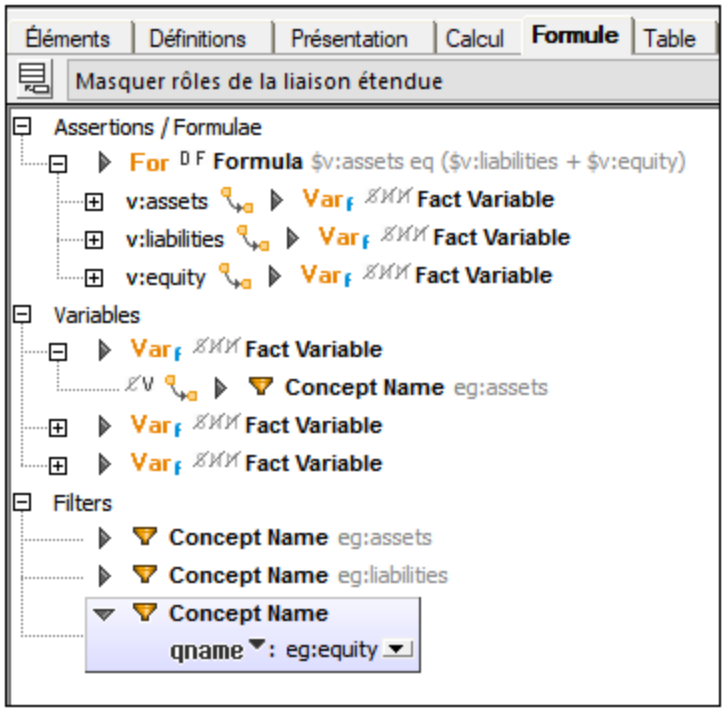
2. Dans l'onglet HTML, cliquer sur la touche de menu de *Ensemble de Document XBRL Inline*, puis cliquer sur **Ajouter Document** *(voir capture d'écran)* et chercher les fichiers XBRL Inline que vous souhaitez ajouter.
3. Exécuter la commande de validation ou de traitement.

© 2016-2022 Altova GmbH

17.4 Éditeur Formule XBRL

Les spécifications Formule XBRL, Variable et Filtre apportent une syntaxe pour l'expression de règles pouvant être utilisées pour dériver de nouvelles valeurs de fait depuis les données dans les rapports commerciaux XBRL. Les spécifications de libellé et de référence génériques prennent en charge le libellé de toute sorte de structures XBRL différentes. Dans le contexte de formule XBRL, ces libellés et ces références peuvent être utilisés pour associer la documentation humaine avec des formules, leurs variables et les filtres qui définissent quels faits dans un rapport de gestion XBRL sont sélectionnés par une variable pour être utilisé dans l'évaluation d'une formule. La validation et les trois spécifications d'assertion définissent une syntaxe pour exprimer des règles concernant le contenu attendu des rapports de gestion, en termes de variables, d'ensembles de variables et de formules. Une introduction à la syntaxe et aux règles de la sémantique de Formule XBRL peuvent être consultés sous Working Draft of XBRL Formula Overview 1.0

L'Éditeur Formule XBRL de XMLSpy est mis en place en tant que partie de l'Éditeur de taxonomie XBRL de l'application. Il est disponible dans l'onglet Formule du Mode XBRL (*voir capture d'écran ci-dessous*).



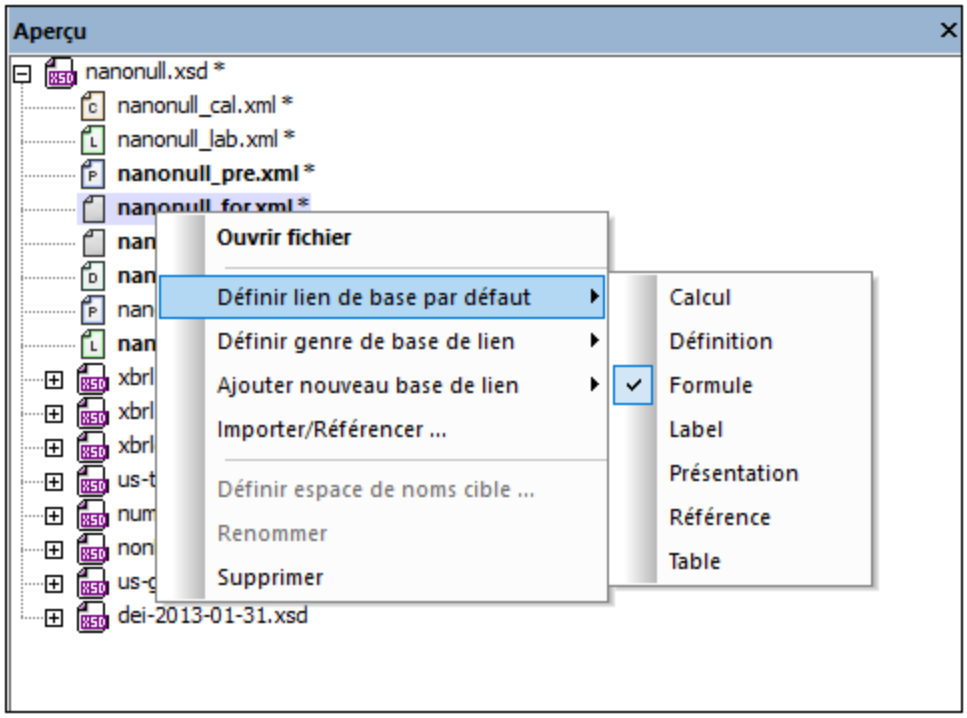
L'onglet Formule est utilisé avec les assistants à la saisie Aperçu et Détails pour créer et éditer des formules. L'assistant à la saisie Aperçu est utilisé pour définir la base de lien par défaut pour les formules XBRL (le fichier dans lequel les formules seront enregistrées par défaut), alors que l'assistants à la saisie Détails peut être utilisé pour éditer les propriétés et le contenu des composants de formule (bien qu'une telle édition puisse être effectuée directement dans l'onglet Formule).

17.4.1 Bases de lien et Rôles de lien de formule

Alors que les bases de lien XBRL standard (Définitions, Présentations, Calculs) définissent des relations entre des concepts par le biais de localisateurs et d'arcs standard dans des liens étendus standard, une base de lien de formule définit les composants de formule (formules, variables, filtres, assertions, etc.) et leurs relations. Ces définitions sont spécifiées par le biais de ressource et d'arcs génériques dans des liens étendus génériques.

Ajouter une base de lien de formule

Dans l'assistant à la saisie Aperçu (*capture d'écran ci-dessous*), cliquer avec la touche de droite sur le fichier de taxonomie ou sur une base de lien existante et choisir **Ajouter nouvelle base de lien | Formule**. La base de lien ajoutée deviendra le fichier de base de lien de formule par défaut. Le fichier de base de lien de formule par défaut est le fichier dans lequel les nouvelles formules de définitions seront enregistrées lorsque le fichier de taxonomie sera enregistré. Si vous souhaitez faire d'un autre fichier de base de lien de formule la base de lien de formule par défaut, cliquer avec la touche de droite et choisir **Définir la base de lien par défaut | Formule** (*voir capture d'écran ci-dessous*).



Veuillez noter que les bases de lien par défaut sont affichées en gras et que les bases de lien qui ont été modifiées mais pas encore enregistrées sont marquées avec un astérisque.

La base de lien de formule est affichée dans l'onglet Formule.

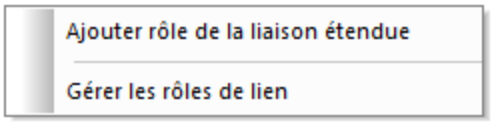
Note : Si un composant de formule est ajouté à la taxonomie à un moment où aucune base de lien de formule n'existe, une base de lien de formule est créée automatiquement.

© 2016-2022 Altova GmbH

Rôles de lien

Comme c'est le cas pour les liens étendus standard (pour Définitions, Présentations, Calculs), les liens génériques doivent définir une valeur de rôle de lien étendue, qui partitionne les relations du même type en des réseaux disjoints. Tous les liens étendus génériques possédant le même rôle de lien sont réunis sous un nœud de rôle de lien dans le diagramme contenu dans l'onglet Formule, *même s'ils se trouvent dans des fichiers de base de lien différents*.

Les rôles de lien génériques peuvent être créés dans le diagramme par le biais du menu contextuel de la surface d'arrière-plan (*capture d'écran ci-dessous*). Veuillez noter néanmoins que ce menu contextuel sera uniquement affiché si la liste de choix Option dans l'onglet Formule a été activé sur *Afficher tous les rôles de lien étendus*.

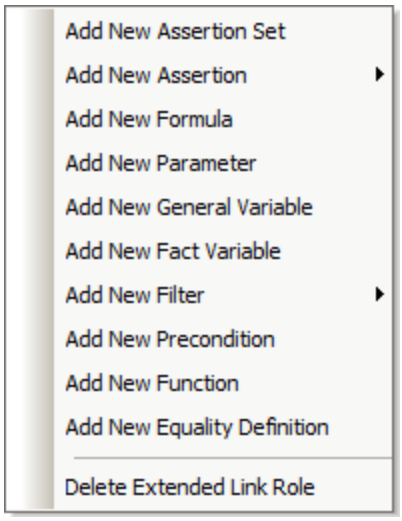


Ce menu est aussi disponible depuis l'icône de barre d'outils , *Ajouter lien étendu / Gérer les rôles de lien*. Puisque les réseaux de relation ne sont pas essentiels pour une base de lien de formule, l'affichage par défaut de l'onglet Formule est *Dissimuler les rôles de lien étendus*, qui dissimule les rôles de lien et, au lieu, affiche les composants de formule sans leurs rôles de lien.

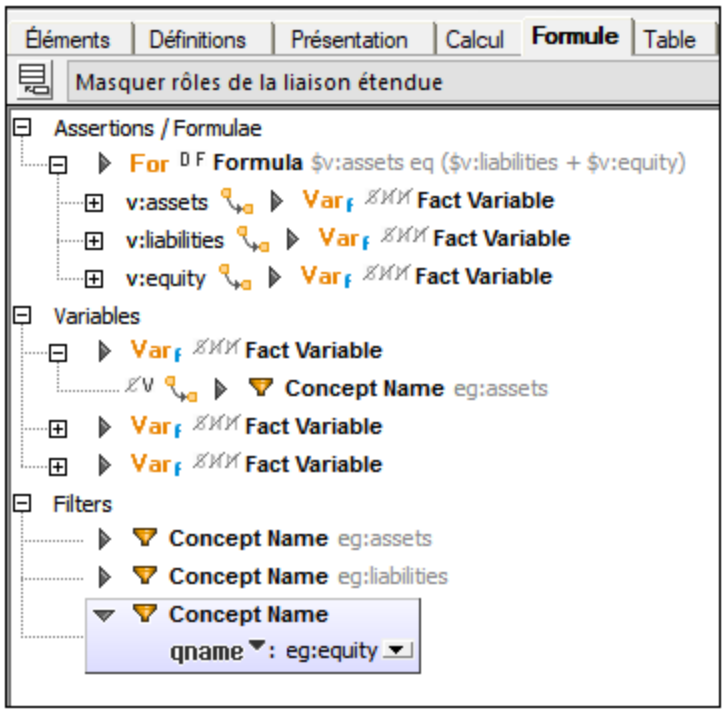
Si aucun fichier de base de lien de formule par défaut n'est disponible lorsque le rôle de lien étendu est créé, un fichier de base de lien de formule par défaut sera créé automatiquement. Et s'il n'y a pas de rôle de lien dans le fichier de base de lien par défaut au moment de la création d'un rôle de lien, alors un rôle de lien sera créé automatiquement dans le fichier de base de lien par défaut.

17.4.2 Composants de formule

De nouveaux composants de formule sont créés par le biais du menu contextuel d'un nœud de rôle de lien (*capture d'écran ci-dessous*) ; ou en configurant l'affichage sur *Masquer les rôles de lien étendus*, avec l'icône de barre d'outils  *Ajouter nouveau composant de formule*.



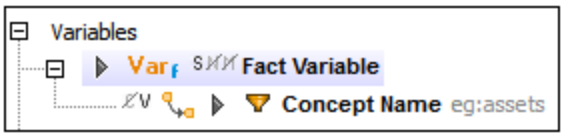
Les mécanismes impliqués dans l'ajout des différents composants sont décrits dans les sous-sections de cette section. Une fois qu'un composant de formule a été ajouté, il est affiché dans le diagramme de l'onglet de Formule (*voir capture d'écran ci-dessous*).



Pour un affichage plus clair, les composants de formule sont divisés en sections et les relations vers d'autres composants (les arcs) sont affichés dans une structure arborescente (*voir capture d'écran ci-dessus*).

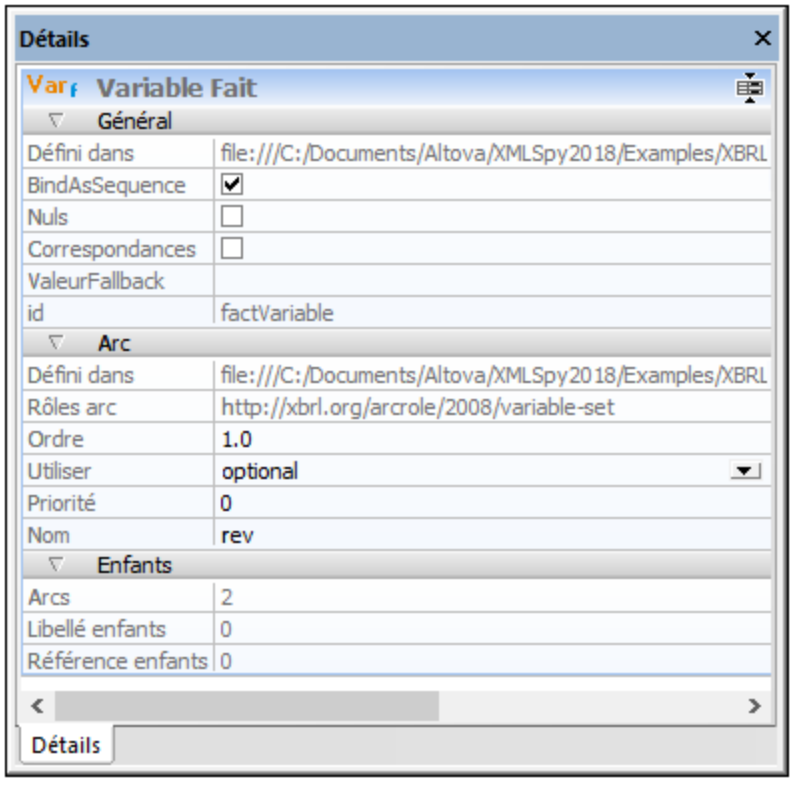
Les propriétés des composants et des relations (arcs) sont indiquées dans le diagramme sous la forme d'icônes situées à gauche du composant ou de l'arc, respectivement (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Par exemple, dans la capture d'écran ci-dessus, le composant Variable Fait comporte trois propriétés, `BindAsSequence` (indiqué par une icône `S`), `Nils` (icône `N`), et `Matches` (icône `M`). Ce sont toutes des propriétés booléennes. La première (`BindAsSequence`) a une valeur de `true`, indiqué dans le diagramme par le fait que l'icône n'est pas barré. Les deux autres propriétés ont une valeur de `false` (indiqué par des icônes barrés). L'arc (en-dessous de la variable) a deux propriétés, la première est la booléenne `false`, la deuxième la booléenne `true`.

Dans l'assistant à la saisie Détails, de la Variable Fait (*capture d'écran ci-dessous*), les propriétés de la variable sont recensées sous la section *Général*. Les valeurs des propriétés booléennes sont indiquées par une coche pour `true` et sans coche pour `false`.



Pour voir les propriétés d'un arc dans l'assistant à la saisie Détails, choisir le composant `to` (destination) dans le diagramme ; les propriétés de l'arc sont recensées dans la section *Arc*.

Les menus contextuels dans l'Éditeur de formule

Les menus contextuels des composants de formule varient selon le type de composant. Les items de menu sont organisés en sections :

- Modifications de contenu (pour les formules, certains filtres, des fonctions personnalisées) : par exemple, *Apposer/Insérer Règle d'aspect*
- Modifications de relation (uniquement pour les sous-items) : *Substituer/Supprimer Arc*
- *Ajouter Libellés/Références*
- Création de nouveaux composants enfants (y compris des relations) : par exemple, *Ajouter nouveau filtre*
- Suppression de composants (y compris des relations)
- *Trouver occurrence Suivante/Précédente* (de composants)

Note : Les items de contenu qui peuvent être créés ou supprimés par le biais du menu contextuel sont affichés dans l'assistant à la saisie Détails dans des sections supplémentaires, comme *Règle d'aspect de concept*.

17.4.2.1 Assertions et ensembles d'assertion

Il existe trois types d'assertions :

- Assertions de Valeur
- Assertions d'Existence
- Assertions de Cohérence

Assertions de valeur

Les assertions de valeur sont les fonctions de base de lien de formule les plus utilisées, elles proposent un moyen de contrôler les faits d'instance XBRL d'entrée par rapport à une expression XPath. Elle propose les propriétés *Modèle d'aspect* et *Filtre implicite* sous la forme d'icônes. La valeur de la propriété `test` est une expression XPath.

Assertions d'existence

Une assertion d'existence est utile pour les contrôles de l'existence statique, comme pour assurer que des faits descriptifs d'un document comme un type de forme, l'identification d'entreprise et l'identification de dépôt sont présents. Elle propose les propriétés *Modèle d'aspect* et *Filtre implicite* sous la forme d'icônes. La valeur de la propriété `test` est une expression XPath.

Assertions de cohérence

Une assertion de cohérence spécifie comment déterminer si un fait de sortie, produit par la formule associée est cohérent avec tous les faits correspondants d'aspect dans l'instance XBRL d'entrée. Elle propose la propriété Booléenne `strict` sous la forme d'une icône. Les valeurs des propriétés *Rayon acceptation absolue* et *Rayon acceptation proportionnelle* sont des expressions XPath.

Messages d'assertion satisfait/non satisfait

Ces sous-composants d'assertions permettent l'association de messages avec des évaluations d'assertion : des messages satisfaits avec des évaluations réussies, des messages non satisfaits avec des messages non réussis. Ces messages peuvent être ajoutés avec le menu contextuel des assertions individuelles.

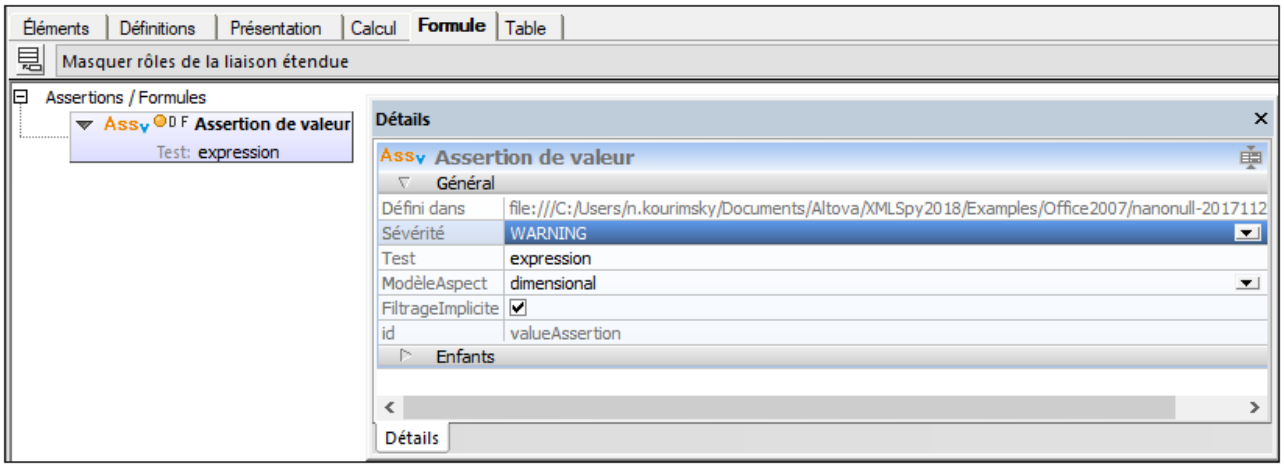
© 2016-2022 Altova GmbH

Relations Assertion-non satisfait-sévérité

Une assertion est soit satisfaite soit non satisfaite. Néanmoins, puisque les assertions ont des règles de niveau d'importance différents, les assertions non satisfaites sont classées selon la sévérité de cette non-satisfaction d'assertion particulière. Il existe trois niveaux de sévérités standard: `ERROR`, `WARNING`, et `OK`. La sévérité par défaut est `ERROR`. Elle est invoquée lorsqu'une assertion n'est pas associée avec une sévérité définie.

La relation assertion-non satisfait-sévérité se trouve entre une assertion et une des ressources de sévérité définie. Elle est exprimée par un arc XLink avec : (i) une valeur de rôle d'arc de `http://xbrl.org/arcrole/PR/2015-11-180/assertion-unsatisfied-severity`, (ii) une assertion en tant que sa ressource de démarrage ; et (iii) une ressource de sévérité en tant que sa ressource de fin.

Dans l'éditeur de taxonomie, la relation de sévérité peut être spécifiée en cliquant sur l'icône `Severity` du composant d'assertion dans le diagramme (*voir capture d'écran ci-dessous*), et puis en choisissant le niveau de sévérité depuis la fenêtre qui apparaît. En alternative, le niveau de sévérité peut être sélectionné dans l'assistant à la saisie Détails de l'Assertion (*capture d'écran*).



Ensembles d'assertion

Un ensemble d'assertions contient une ou plusieurs assertions. Le menu contextuel d'un ensemble d'assertion permet l'ajout d'assertions supplémentaires à l'ensemble d'assertions.

17.4.2.2 Formules

Une formule exprime un ensemble de règles pour construire un fait XBRL de sortie en transformant les valeurs auxquelles les variables ont été évaluées dans la variable de la formule. Les valeurs des variables sont obtenues à partir d'un instance XBRL d'entrée et de instance et de son DTS annexe ou depuis l'application traitant la formule.

La règle de valeur est une expression XPath qui permet d'obtenir la valeur à assigner au fait. Il peut s'agir d'une expression simple, comme une constante, ou elle peut contenir des termes qui réfèrent aux variables et aux

paramètres de l'ensemble de variable, de valeurs enchaînées provenant d'autres ensembles de variables et/ou de valeurs calculées à partir de fonctions personnalisées et intégrées.

Dans XBRL, les faits numériques non-fraction sont rapportés avec des informations concernant leur précision sous la forme d'un attribut de précision/décimales. C'est pourquoi les formules peuvent contenir des règles de précision réglementant la détermination de la précision à affirmer pour un fait de sortie.

De même qu'avec les règles permettant de déterminer les valeurs de fait de sortie et leur précision, les formules spécifient ou impliquent des règles d'aspect qui déterminent les valeurs pour tous les aspects de sortie nécessaires pour interpréter les valeurs de sortie. Les règles pour déterminer le concept de sortie, le contexte de sortie et les unités de sortie de la mesure (pour les faits numériques) sont tous des types différents des règles d'aspect.

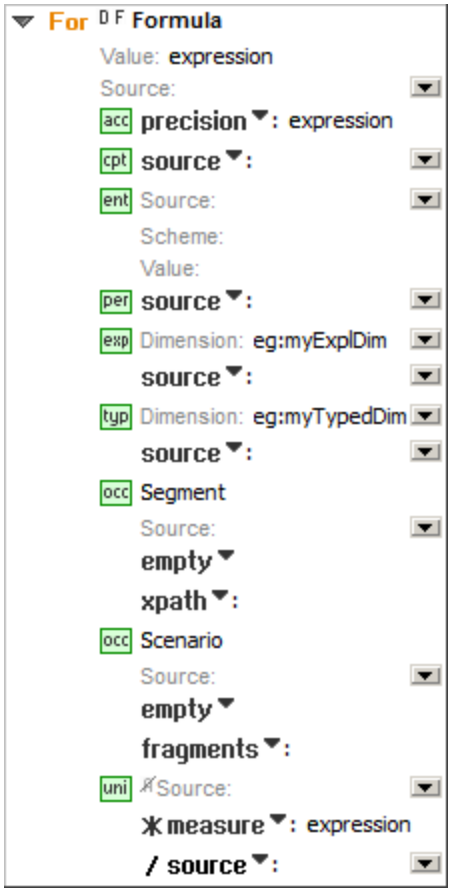
Un aspect peut être obtenu (partiellement ou entièrement) depuis une variable liée de l'évaluation en spécifiant une source. La source peut être spécifiée sur la base d'une règle ou peut être héritée depuis une source sur la base de l'élément de la formule (ou tuple). Lorsqu'il y a plusieurs sources, la source la plus proche d'une règle d'aspect domine.

Lorsqu'une formule est insérée, elle n'a pas de règle de précision ou d'aspect (*capture d'écran ci-dessous*).



Les règles de précision et d'aspect sont définies dans le cadre du contenu de la formule et sont ajoutées (ou supprimées) par le biais du menu contextuel. La capture d'écran ci-dessous montre une formule avec toutes les règles de précision et d'aspect.

© 2016-2022 Altova GmbH



Dans l'assistant à la saisie Détails, les règles de précision et d'aspect sont affichées dans des sections supplémentaires.

Règle de précision

Type : précision ou décimales
Valeur : expression XPath

Règles d'aspect

Les règles d'aspect sont regroupées par types.

Règles Concept
Type : qname, expr ou source
Valeur : QName du concept, expression XPath, ou variable de source (ou le QName non couvert)

Règles Identifiant de l'entité
Source: variable de source (ou le QName non couvert)
Scheme/valeur : expressions XPath

Règles Période
Type : instant, durée, pour toujours ou source

Valeur : expression XPath de la valeur, démarrage/fin/source, aucune valeur ou variable de source (ou le QName non couvert)

Règles Dimension explicite
Dimension: QName de la dimension, touché par la règle de dimension explicite.
Type : qname, exp, omettre ou source
Valeur : QName du membre, expression XPath du membre, aucune valeur ou variable de source (ou le QName non couvert)

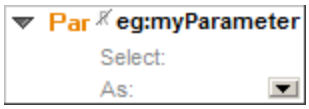
Règles Dimension typée
Dimension: QName de la dimension, touché par la règle de dimension typée.
Type : xpath, value, omettre ou source
Valeur : expression XPath, élément XML, aucune valeur ou variable de source (ou le QName non couvert)

Règles Composant de contexte ouvert
Les règles OCC sont regroupées par types, donc par les règles OCC de segment et les règles OCC de scénario.
Source: Variable de source définie dans la première règle OCC.
Pour chaque règle OCC :
Type : vide, fragments ou xpath
Valeur : aucune valeur, éléments XML ou expression XPath

Règles d'unité
Le flag booléen *Augmente* spécifie si les valeurs d'aspect de source doivent être utilisées ou pas.
Chaque règle de multiplication/division d'unité doit contenir :
Type : *mesure, /mesure ou *source, /source
Valeur : expression XPath de la mesure ou variable de source (ou le QName non couvert)

17.4.2.3 Paramètres

Un paramètre peut être référencé dans des expressions XPath. Il propose un flag *Required*. Une fois configuré, le paramètre est obligatoire, c'est à dire que sa valeur doit être fournie par l'application de traitement. Si le paramètre n'est pas obligatoire et qu'aucune valeur n'est fournie par l'application de traitement, la valeur fournie peut être calculée à l'aide d'une expression XPath donnée dans la propriété *Select*. La propriété *As* optionnelle spécifie le type de données requis par le paramètre.



17.4.2.4 Variables

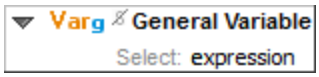
Les variables déclarent une manière de lier des données d'entrée, généralement des items de faits à un nom qui peut être référencé par un nom de variable comme depuis une assertion ou expression de formule. Les variables qui lient vers des items de faits d'entrée sont des variables de faits et utilisent des filtres pour déclarer

© 2016-2022 Altova GmbH

ce qu'elles peuvent lier dans l'entrée. Les variables générales sont utilisées pour des résultats d'expression intermédiaires et d'autres types de traitement.

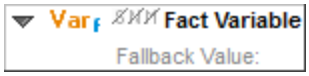
Variable Générale

Une variable générale propose la propriété booléenne Bind As Sequence sous la forme d'une icône. La valeur de la propriété Select est une expression XPath.



Variable Fait

Une variable fait propose les propriétés booléennes Bind As Sequence, Nils et Matches sous la forme d'icônes. La valeur de la propriété Fallback Value est une expression XPath.



17.4.2.5 Filtres

Un filtre définit les critères de sélection pour les faits dans l'instance XBRL d'entrée, c'est à dire, l'instance XBRL par rapport à laquelle les variables sont évaluées. Les filtres expriment des critères qui peuvent être appliqués à des faits d'entrée. Certains filtres peuvent avoir un contenu XML affiché dans les sous-lignes.

Aspect cover
Ces filtres n'effectuent pas de "filtrage", et ne possèdent donc pas d'expression XPath implicite. Ils sont traités ou appliqués après d'autres filtres (comme concept et dimension) et surmontent l'état "cover" des aspects résultant de l'application des autres filtres.

Un ou plusieurs items d'aspect
Type : aspect, dim-qname/excl-dim-qname ou dim-exp/excl-dim-exp
Valeur : type d'aspect (enum), QName de la dimension ou expression XPath
Les items sont affichés dans l'assistant à la saisie Détails dans d'autres sections.

Filtres booléens
Les filtres booléens sont liés à des sous-filtres.
Le filtre "and" fait correspondre les faits sur la base des critères exprimés par chacun de ses sous-filtres.



Le filtre "or" fait correspondre les faits sur la base des critères exprimés par un de ses sous-filtres.



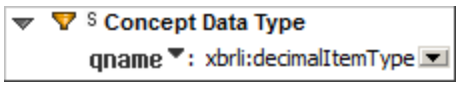
dans

Concept nom
Le filtre concept nom fait correspondre les faits sur la base des noms de leurs concepts.



Un ou plusieurs concepts :
Type : qname ou exp
Valeur : QName du concept ou expression XPath
Les concepts sont affichés dans l'assistant à la saisie Détails dans d'autres sections.

Concept type de données
Le filtre concept type de données peut être utilisé pour faire correspondre des faits sur la base de son type de données de Schéma XML.



Flag booléen : “strict” spécifie si le type de données du fait doit être non-dérivé ou pas.
Type : qname ou exp
Valeur : QName du type de données ou expression XPath

Concept groupe de substitution
Le filtre concept groupe de substitution peut être utilisé pour faire correspondre des faits sur la base de son groupe de substitution de Schéma XML.

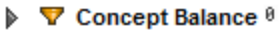


Flag booléen : “strict” spécifie si le concept du fait doit spécifier l'élément dans son attribut @substitutionGroup directement ou pas.
Type : qname ou exp
Valeur : QName du groupe de substitution or XPath expression

Concept type de période
Le filtre concept type de période peut être utilisé pour faire correspondre des faits selon qu'ils rapportent des valeurs pour les concepts de type de durée ou type instantané, tel que déterminé par l'attribut @xbrli:periodType.



Concept solde
Le filtre concept solde peut être utilisé pour faire correspondre des faits selon qu'ils ont un attribut @xbrli:balance et s'il a une valeur de débit ou de crédit.



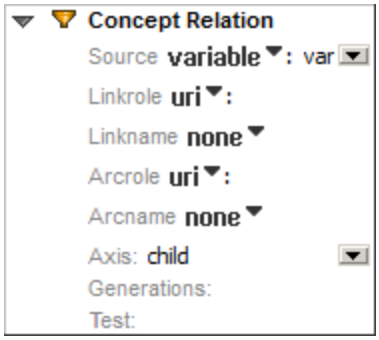
© 2016-2022 Altova GmbH

Concept attribut personnalisé
Le filtre concept attribut personnalisé peut être utilisé pour faire correspondre des faits sur la base de son existence ou sur la valeur d'un attribut personnalisé dans chaque déclaration du concept.



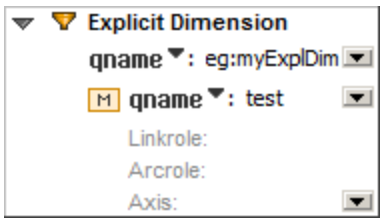
Type : qname ou exp
Valeur : QName de l'attribut ou expression XPath

Concept relation
Le filtre concept relation peut être utilisé pour faire correspondre des faits sur la base des relations effectives de leurs concepts par rapport au concept de source, dans un réseau URI de rôle de lien spécifié des relations effectives, d'un URI de rôle d'arc spécifié, sur un axe spécifié, en comprenant les générations spécifiées et conforme à une expression de test optionnelle.



Source: Type = variable, qname ou exp
Rôle de lien : Type = uri ou exp
Nom de lien : Type = aucun, qname ou exp
Rôle d'arc : Type = uri ou exp
Nom d'arc : Type = aucun, qname ou exp

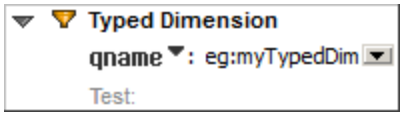
Dimension explicite
Un domaine de dimension explicite est défini dans le contexte d'un DTS donné en tant que l'ensemble de tous les membres de domaine dans l'union de tous les domaines des membres valides de la dimension du filtre. Le filtre de dimension explicite peut être utilisé pour faire correspondre des faits avec tout autre membre de domaine dans un domaine de dimension explicite en tant que la valeur pour cette dimension explicite.



Type de dimension : qname ou exp
Un ou plusieurs membres :
Type : variable, qname ou exp

Les membres sont affichés dans l'assistant à la saisie Détails dans des sections supplémentaires.

Dimension typée
Le filtre dimension typée peut être utilisé pour faire correspondre des faits sur la base de la valeur pour une dimension typée.



Type de dimension : qname ou exp

Filtre Identifiant d'entité
Le filtre identifiant d'entité peut être utilisé pour faire correspondre des faits sur la base des caractéristiques du schéma d'identification d'entité et/ou la valeur d'identification d'entité.

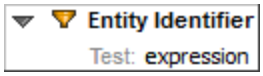
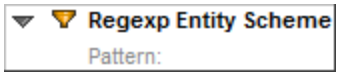


Schéma d'entité spécifique
Le filtre schéma d'entité spécifique peut être utilisé pour faire correspondre des faits selon qu'ils rapportent des valeurs pour le schéma identifié par le filtre.



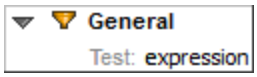
Schéma d'entité d'expression régulière
Le filtre Schéma d'entité d'expression régulière peut être utilisé pour faire correspondre des faits sur la base de motifs réguliers dans le texte du schéma d'entité.



Identifiant d'entité spécifique
Le filtre Identifiant d'entité spécifique peut être utilisé pour faire correspondre des faits selon qu'ils rapportent des valeurs utilisant la valeur d'identifiant d'entité indiquée par le filtre.

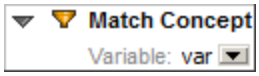
Identifiant d'entité d'expression régulière
Le filtre Identifiant d'entité spécifique peut être utilisé pour faire correspondre des faits sur la base de motifs réguliers dans le texte de la valeur d'identifiant de l'entité.

Général
Le filtre général ne couvre pas d'aspect.

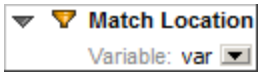


Match concept
Le filtre correspondance de concept peut être utilisé pour sélectionner des faits qui rapportent des valeurs pour le même concept.

© 2016-2022 Altova GmbH



Match location
Le filtre correspondance d'emplacement peut être utilisé pour sélectionner des faits qui ont le même élément parent.



Match unit
Le filtre correspondance d'unité peut être utilisé pour sélectionner des faits qui ont la même unité.

Match entity identifier
Le filtre correspondance identifiant d'entité peut être utilisé pour sélectionner des faits avec le même identifiant d'entité.

Match period
Le filtre correspondance de période peut être utilisé pour sélectionner des faits qui ont la même période.

Match dimension
Le filtre correspondance de dimension peut être utilisé pour sélectionner des faits qui ont la même valeur pour une Dimension XBRL spécifiée.



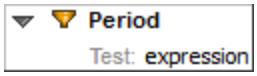
Match complete segment
Le filtre correspondance de segment complet peut être utilisé pour sélectionner des faits qui ont le même segment, et où le contenu du segment n'est pas interprété sur la base de la Spécification des dimensions XBRL.

Match non-XDT segment
Le filtre correspondance de segment non-XDT peut être utilisé pour sélectionner des faits qui ont le même segment, après avoir exclu tout contenu Spécification des dimensions XBRL de la comparaison.

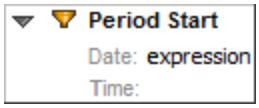
Match complete scenario
Le filtre correspondance scénario complet peut être utilisé pour sélectionner des faits qui ont le même scénario, où le contenu du scénario n'est pas interprété sur la base de la XBRL Dimensions Specification.

Match non-XDT scenario
Le filtre correspondance de scénario non-XDT peut être utilisé pour sélectionner des faits qui ont le même scénario, après avoir exclu tout contenu de XBRL Dimensions Specification content de la comparaison.

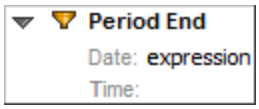
Période
Le filtre période peut être utilisé pour faire correspondre des faits sur la base d'une gamme étendue de critères liés à la période pendant laquelle ou à laquelle ils ont été mesurés.



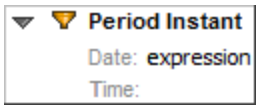
Début de période
Le filtre début de période peut être utilisé pour faire correspondre des faits basés sur le démarrage des durées de leurs mesures.



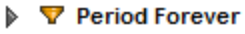
Fin de période
Le filtre fin de période peut être utilisé pour faire correspondre des faits basés sur la fin des durées de leurs mesures.



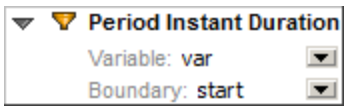
Période instant
Le filtre période instant peut être utilisé pour faire correspondre des faits basés sur l'instant des durées de leurs mesures.



Période pour toujours
Le filtre pour toujours peut être utilisé pour faire correspondre des faits qui sont rapportés avec une période pour toujours.



Période durée de l'instant
Le filtre durée de l'instant peut être utilisé pour faire correspondre des faits qui sont rapportés à un instant où l'instant correspond au démarrage ou à la fin de la durée pour laquelle un autre fait a été rapporté.

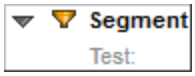


Relatif
Le filtre relatif peut être utilisé pour sélectionner des faits pour lesquels les aspects qui sont couverts par le filtre relatif ont des valeurs qui correspondent aux aspects correspondants d'un autre fait. Le fait qui est rendu correspondant par le filtre relatif doit être le résultat d'évaluation d'une autre variable de fait dans l'ensemble de variable en cours d'évaluation.

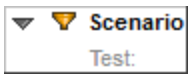
© 2016-2022 Altova GmbH



Segment
Le filtre segment peut être utilisé pour faire correspondre des faits qui ont un contenu non-XDT satisfaisant à des contraintes spécifiées. Le contenu non-XDT réfère au contenu de segment qui n'est pas basé sur les dimensions explicites ou typées définies dans la Spécification des Dimensions XBRL.



Scénario
Le filtre scénario peut être utilisé pour faire correspondre des faits qui ont un contenu non-XDT satisfaisant à des contraintes spécifiées. Le contenu non-XDT réfère au contenu de scénario qui n'est pas basé sur les dimensions explicites ou typées définies dans la Spécification des Dimensions XBRL.

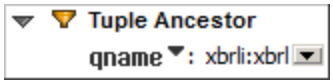


Parent tuple
Le filtre parent peut être utilisé pour sélectionner des faits qui ont un élément de parent spécifié.



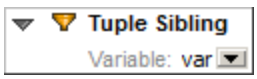
Type : qname ou exp

Ancêtre tuple
Le filtre ancestor peut être utilisé pour sélectionner des faits qui ont un élément ancêtre spécifié.

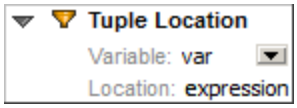


Type : qname ou exp

Enfant tuple
Le filtre enfant peut être utilisé pour sélectionner des faits qui sont enfants d'un autre fait.



Emplacement tuple
Le filtre emplacement peut être utilisé pour sélectionner des faits qui ont un emplacement spécifié relatif à l'emplacement d'un autre fait.

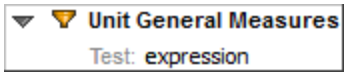


Unité de mesure simple
Le filtre unité de mesure simple peut être utilisé pour faire correspondre des faits qui sont rapportés avec une unité spécifiée par une mesure simple.



Type : qname ou exp

Unité de mesures générales
Le filtre unité générale peut être utilisé pour sélectionner des faits sur la base de critères impliquant un certain nombre de mesures d'unités.



Valeur nil
Le filtre nil peut être utilisé pour faire correspondre des faits qui sont rapportés en tant que nil.

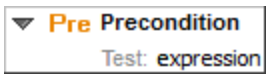


Valeur Précision
Le filtre précision peut être utilisé pour faire correspondre des faits en se basant sur le fait qu'ils possèdent une précision réelle ou présumée minimum, tout en notant que cette précision peut être déduite de la valeur de l'attribut @decimal. Veuillez noter que le filtre de précision ne sélectionnera pas de faits si le filtre suggère une précision requise minimum infinie. Le filtre ne sélectionnera pas de faits non-numériques ou de faits qui sont rapportés avec une valeur nil.



17.4.2.6 Préconditions

Les préconditions permettent de déterminer si un ensemble de variables liées peut activer une valeur de formule et un fait de sortie ou un test de valeur d'assertion ou un décompte d'existence.



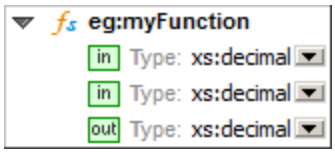
© 2016-2022 Altova GmbH

17.4.2.7 Fonctions

Une fonction personnalisée est une fonction XPath qui n'est pas définie dans la spécification des Fonctions XPath et XQuery et qui n'est pas non plus définie dans le registre des Fonctions XBRL. Les fonctions personnalisées peuvent être utilisées dans le cadre des expressions XPath.

Signature de fonction

La signature de fonction est comme indiquée dans la capture d'écran ci-dessous.



Les éléments enfants, le cas échéant, d'une signature de fonction personnalisée, spécifient les types de données des paramètres d'entrée de la fonction personnalisée. Le tri des paramètres d'entrée de la fonction personnalisée correspond à l'ordre du document des éléments enfant de la signature de fonction personnalisée.

Les entrées sont affichées dans l'assistant à la saisie Détails dans des sections supplémentaires.

L'implémentation de la fonction

L'implémentation de la fonction est comme indiquée dans la capture d'écran ci-dessous.



Une implémentation de la fonction personnalisée (CFI) contient une séquence des éléments enfant qui servent à définir les noms pour les entrées de la fonction, afin d'exprimer les expressions XPath qui contiennent l'implémentation de la fonction personnalisée et pour définir la sortie de fonction personnalisée.

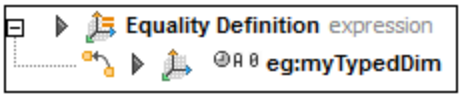
Une relation Fonction-Implémentation est une relation entre une signature de fonction personnalisée et une implémentation de fonction personnalisée. Puisqu'une implémentation de fonction doit être la cible d'une relation fonction-implémentation, elle est toujours affichée sous la signature de fonction correspondante. Si la relation manque (ou si la signature est définie sous un rôle de lien différent), l'implémentation est affichée directement sous la section *Fonctions*.

Les entrées et les étapes sont affichés dans l'assistant à la saisie Détails dans des sections supplémentaires.

17.4.2.8 Définitions d'égalité

Une définition d'égalité est une définition d'égalité entre deux valeurs quelconques dans une définition de domaine de définition à type. Il s'agit là d'un élément dans un Schéma XML qui définit le modèle de contenu pour une dimension à type et qui est identifié en tant que tel par un attribut `@xbrldt:typedDomainRef` de

l'élément de Schéma XML déclarant une dimension typée. Une relation de définition d'égalité, qui est la relation entre une définition de domaine de dimension à type et une définition d'égalité est affichée en tant que relation inverse entre la définition d'égalité et la dimension à type correspondante.



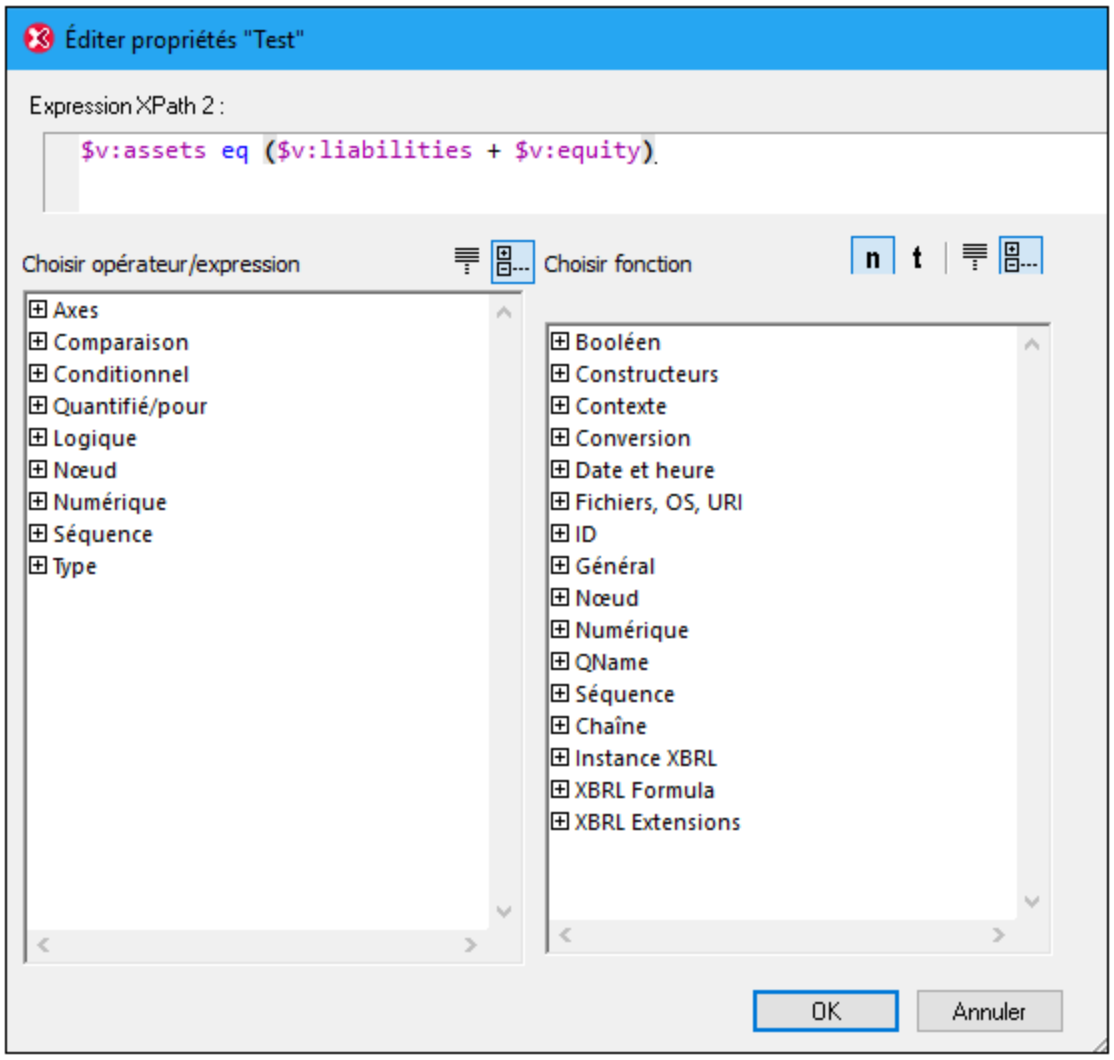
Une relation de définition d'égalité peut être établie en glissant une dimension à type depuis l'assistant à la saisie Éléments globaux dans un composant de définition d'égalité. Veuillez noter que ni la définition d'égalité ni la dimension à type ne peut encore être impliquée dans une relation de définition d'égalité existante.

17.4.3 Éditer les propriétés et contenus de composant

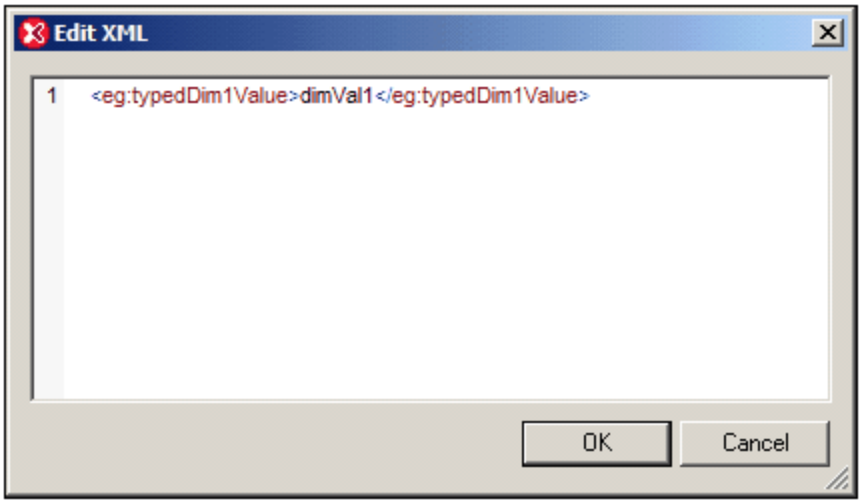
Les propriété de composants de formule peuvent être édités directement dans le diagramme ou dans l'assistant à la saisie Détails.

Dans le diagramme, lorsqu'un composant est réduit, soit son nom (s'il en a un), soit la valeur de la propriété de défaut appropriée est affichée en gris à côté du texte de description du composant. Double-cliquer sur le composant pour l'agrandir. Double-cliquer sur une propriété pour la mettre en mode d'édition. Si une propriété ou un contenu comprend une expression XPath, le dialogue Éditer l'expression XPath s'affiche.

© 2016-2022 Altova GmbH



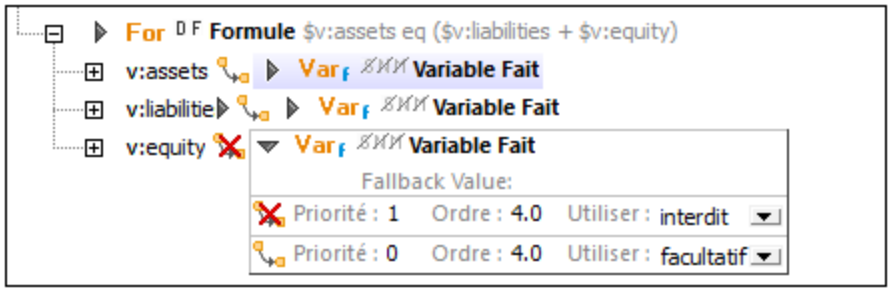
Lors de l'édition du contenu XML, comme le contenu de *Valeur* dans une Règle d'aspect de dimension typée ou le contenu des *Fragments* dans une Règle d'aspect OCC d'une formule, le dialogue Éditer XML (*capture d'écran ci-dessous*) s'affiche. (Cliquer avec la touche de droite sur une formule pour ajouter une règle d'aspect.)



Une fois avoir saisi le texte XML et cliquer sur **OK**, le texte XML sera saisi en tant que le contenu dans la propriété. Si le texte XML n'est pas bien formé, un message s'affichera et le texte ne sera pas autorisé.

17.4.4 Relations de composant de formule

Une relation entre deux composants de formule peut être créé en reliant un composant de formule à un autre par glisser/déposer. Les relations sont montrées par des arcs dans le diagramme (*voir capture d'écran ci-dessous*).



Les possibilités d'affichage et d'édition suivantes existent :

- L'ordre des enfants d'un composant dépend des valeurs de la propriété-arc *Ordre*, qui peut être modifié en déplaçant des enfants (*voir capture d'écran ci-dessus*).
- Un composant enfant peut être glissé sur ou sous un composant de parent différent pour copier ou déplacer la relation (et ses propriétés).
- Lors de la création d'un nouveau composant par le biais du menu contextuel d'un composant existant (parent), la relation (arc) est aussi générée automatiquement.
- Les commandes **Surmonter Arc** et **Supprimer Arc** dans le menu contextuel d'un composant enfant sert à, respectivement, surmonter et supprimer les relations entre le composant et son parent.
- Comme c'est le cas avec les relations de concept, les arcs multiples et les relations surmontées sont affichées dans des sous-lignes (*voir capture d'écran ci-dessus*).

© 2016-2022 Altova GmbH

Note : Le rôle d'arc des relations de composant de la formule ne peut pas être modifié.

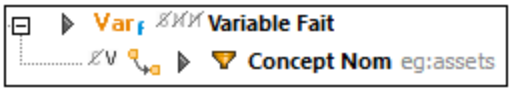
Relations Variable-set

Une relation variable-set est une relation entre (i) une ressource variable-set (une assertion de valeur, une assertion d'existence, ou une formule) et (ii) une variable (variable de fait ou variable générale) ou un paramètre. Le *Nom* d'une variable de paramètre est affiché devant l'icône d'arc (capture d'écran ci-dessous).



Relations Variable-filtre

Une relation variable-filtre est une relation entre une variable de fait et un filtre. Si le flag Booléen *Complement* (icône a C dans le diagramme) est défini, la relation sera une relation variable-filtre complétée. Si le flag booléen *Cover* (icône a V dans le diagramme) est défini, la relation sera une relation variable-filter couvrante (*affiché dans la capture d'écran ci-dessus*). Dans ce cas, le filtre couvre les aspects des faits qui passent par le filtre.



Relations Variable-set-filtre

Une relation variable-set-filtre (*voir capture d'écran ci-dessous*) est une relation entre une ressource variable-set et un filtre. Un filtre participant dans une relation variable-set-filtre est, par définition, associée avec chacune des variables de fait dans l'ensemble de variable défini par la ressource à laquelle elle est liée. Le flag booléen *Complement* spécifie si les variables utilisent le complément de filtre. Tous les filtres qui sont associés avec les variables de fait par les relations variable-set-filtre, par définition, ne couvrent pas d'aspects.

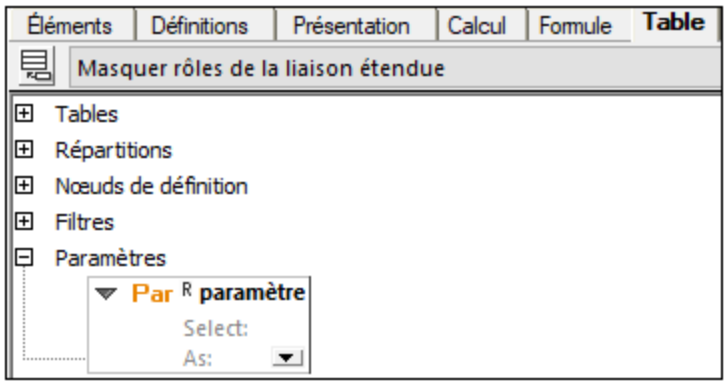


Construire des formules visuellement dans l'Aperçu de mise en page de table

Les développeurs de taxonomie XBRL peuvent également profiter de l'Aperçu de mise en page de table XBRL pour une approche pointer et cliquer pour construire des formules XBRL. Cette fonction est expliquée dans la section, Élaborer des formules dans l'Aperçu de mise en page de table.

17.4.5 Paramètres de formule

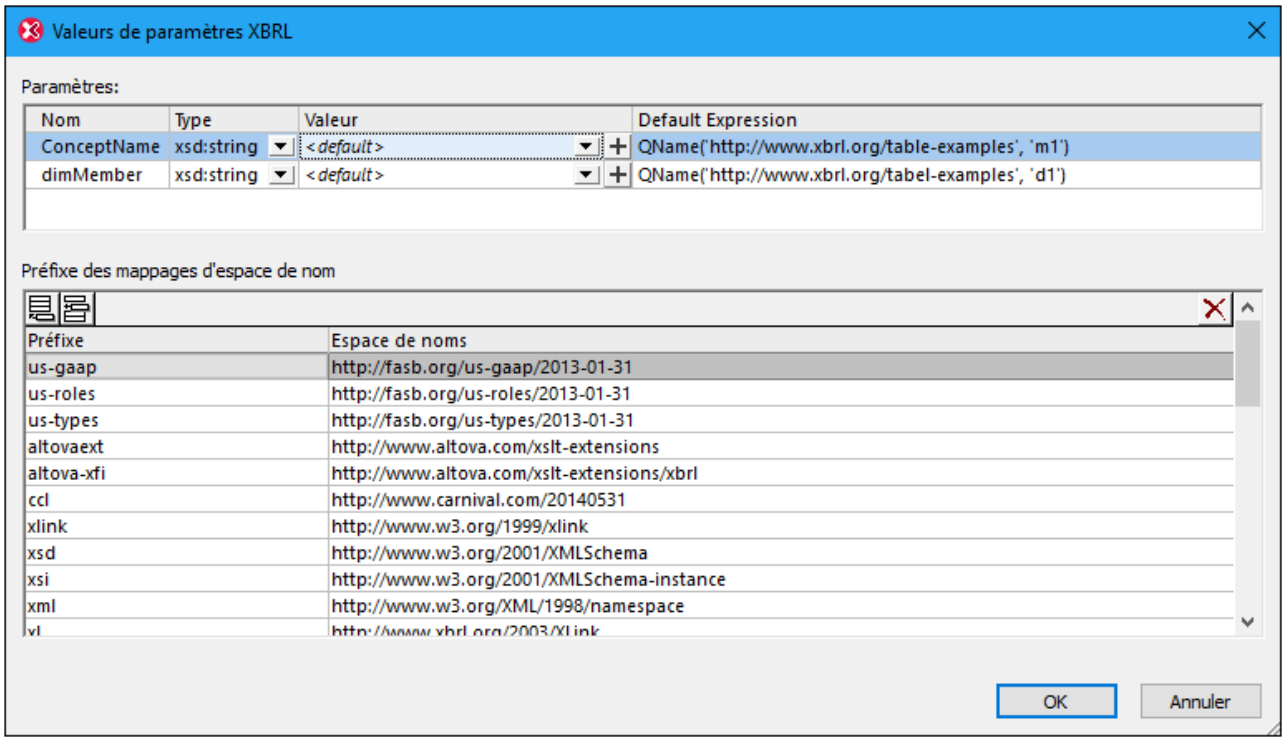
Les paramètres XBRL peuvent être utilisés dans des expressions XPath dans des formules et dans des définitions de table. Les paramètres qui seront utilisés en tant que paramètres de formule (résidant dans la base de lien de la formule) sont créés dans l'onglet Formule, alors que les paramètres de table (résidant dans la base de lien de table) sont créés dans l'onglet Table. Les deux paramètres de formule et les paramètres de table peuvent être locaux ou globaux. Les paramètres locaux sont principalement des paramètres globaux qui sont liés au composant respectif (formule ou table) au moment de sa création. Les paramètres locaux sont créés en cliquant avec la touche de droite sur le composant (formule ou table) et en sélectionnant **Ajouter nouveau paramètre**, alors que des paramètres globaux sont créés en cliquant avec la touche de droite dans un espace vide de l'onglet respectif et en sélectionnant **Ajouter nouveau paramètre**. Vous ajoutez ainsi in nouveau paramètre nommé `parameter` dans le diagramme (*la capture d'écran ci-dessous montre un paramètre global*). Pour changer le nom du paramètre, double-cliquer le nom puis l'éditer.



Chaque paramètre a un flag *Requis*. Une fois défini, le paramètre est obligatoire, c'est à dire que sa valeur doit être fournie par l'application de traitement. Si le paramètre n'est pas obligatoire et qu'aucune valeur n'est fournie par l'application de traitement, la valeur fournie peut être calculée à l'aide d l'expression XPath donnée dans la propriété *Sélectionner*. Double-cliquer dans le champ *Sélect* pour saisir une expression XPath. Cette valeur sera la valeur par défaut du paramètre. La propriété optionnelle *Tel que* spécifie le type de données requis par le paramètre. Choisir un type de données depuis la liste déroulante de la liste de choix.

Dans le cas des paramètres qui seront utilisés en tant que paramètres de table, vous pouvez éditer le type de données du paramètre et fournir une valeur de paramètre qui surmonte la valeur par défaut. Pour ce faire, cliquer sur **XBRL | Valeurs de paramètres**. Ensuite, dans le dialogue qui apparaît (*capture d'écran ci-dessous*), saisir une valeur de paramètre. Cette valeur surmontera la valeur par défaut. Puisque les paramètres qui sont utilisés en tant que paramètres de table peuvent prendre des valeurs multiples, vous pouvez ajouter des valeurs de paramètre supplémentaires pour un paramètre en cliquant sur l'icône **+** dans la colonne *Valeur*.

© 2016-2022 Altova GmbH



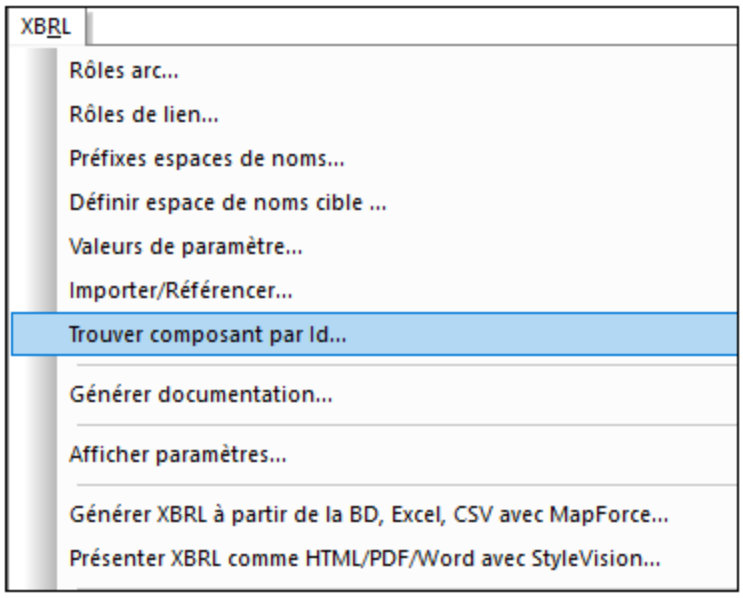
Les valeurs des paramètres globaux telles qu'attribuées dans ce dialogue sont uniquement évaluées pour les paramètres de table. Les valeurs des paramètres utilisés dans les formules ne sont pas éditables dans ce dialogue.

17.4.6 Trouver des composants de formule

Les composants de formule peuvent être trouvés à l'aide de leurs ID et en cherchant les occurrences du composant dans le document.

Trouver le composant de formule avec l'id

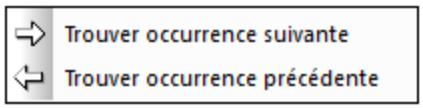
Dans les taxonomies à bases de lien de formule volumineuses contenant plusieurs composants du même type, il peut être utile de chercher un composant à l'aide de son ID. La commande de menu **XBRL | Trouver composant par Id** permet une recherche par ID.



Cliquer sur la commande pour ouvrir un dialogue vous invitant à saisir l'ID que vous cherchez.

Trouver les occurrences de composant

La plupart des composants de formule sont affichés dans le diagramme de base de lien de la formule plusieurs fois : (i) la définition, qui se situe directement sous le nœud de section approprié, et (ii) toutes les références au composant (via les relations). Les commandes **Trouver occurrence suivante** et **Trouver occurrence précédente** dans le menu contextuel du composant (*capture d'écran ci-dessous*) permettent de naviguer dans tous les endroits où le composant de formule est référencé.



Ces commandes peuvent aussi être accédées par le biais de leurs icônes de barre d'outils (*capture d'écran ci-dessous*).



Une fois que la définition de composant est atteinte, un message à cet effet est affiché.

© 2016-2022 Altova GmbH

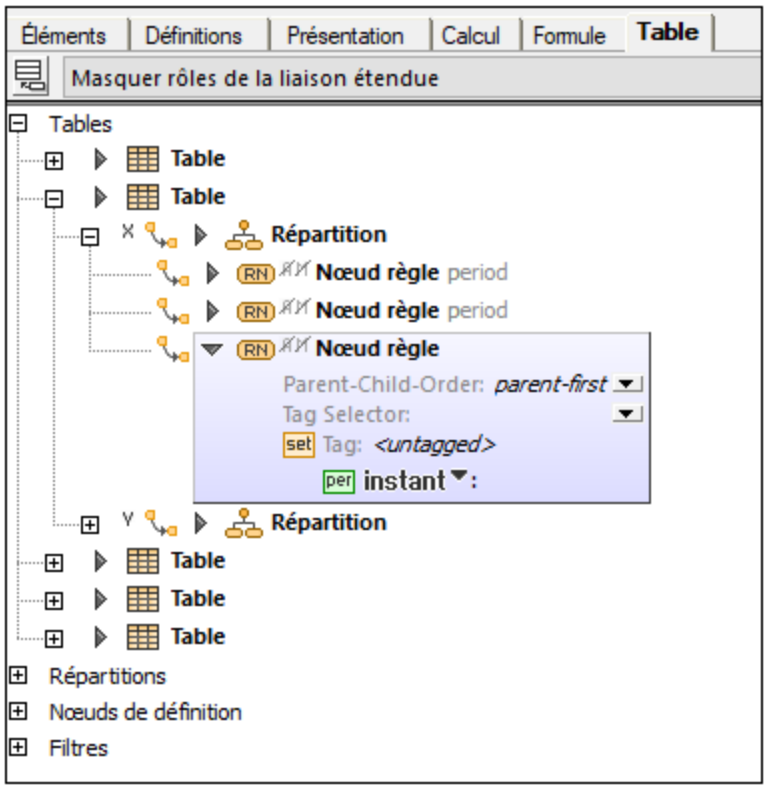
17.5 Éditeur de définitions de table XBRL

Les spécifications XBRL fournissent une base de lien de table qui complète la base de lien de présentation. Les tables fournissent un autre moyen de définir les consultations des concepts définis dans les taxonomie XBRL. Au lieu de montrer des concepts en tant que hiérarchie, comme c'est le cas pour la base de lien de présentation, elle permet aux tables d'être définies avec plusieurs axes. Il n'est pas nécessaire de limiter les composants des axes aux items individuels, les composants peuvent être définis en termes d'une combinaison de dimensions, de référence à une période de temps, d'unités, d'entités ou de tout autre propriété qui peut être utilisée pour identifier les faits financiers représentés par les taxonomies. Vous trouverez une introduction à la syntaxe et aux sémantiques des base de lien de table XBRL sous XBRL Table Linkbase Overview 1.0 et Table Linkbase 1.0 Recommendation du 18 mars 2014.

XMLSpy suit la Table Linkbase 1.0 Recommendation of 18 March 2014, et utilise l'espace de noms http://xbrl.org/2014/table.

Alors que les base de lien XBRL standard (présentation, calcul, définition) définissent les relations entre les concepts par le biais de locateurs et d'arcs standard dans des liens étendus standard, une base de lien de table contient des composants (tables, répartitions, nœuds de définition, etc.) et leurs relations par le biais d'arcs de ressource et génériques dans des liens étendus génériques. La spécification de base de lien de table définit une séquence de trois modèles et les traite pour transformer chaque modèle dans le suivant. Les trois modèles sont les suivants : le modèle de définition, le modèle structurel et le modèle de mise en page (ou de rendering). Le modèle de définition est un modèle du contenu sémantique de la base de lien de la table. Les tables sont définies par leurs axes et les définitions d'axe sont, quant à elles composées d'arborescences de nœuds de définition.

L'Éditeur de définitions de la table XBRL de XMLSpy est mis en place en tant que partie intégrante de l'Éditeur de taxonomie XBRL de l'application. Il est disponible dans l'**onglet Table** du Mode XBRL (*voir capture d'écran ci-dessous*).

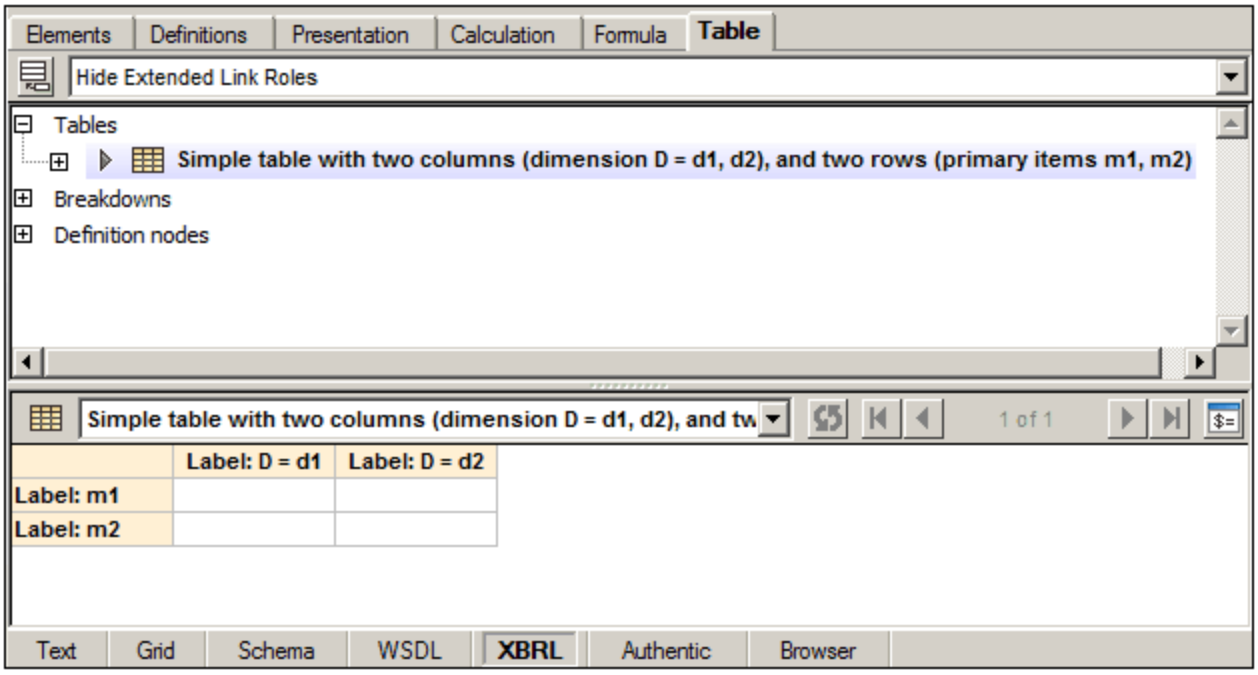


L'onglet Table est utilisé avec l'assistant à la saisie Aperçu et l'assistant à la saisie Détails pour créer et éditer les définitions de table. L'assistant à la saisie Aperçu est utilisé pour définir la base de lien par défaut pour les tables XBRL (le fichier dans lequel les définitions de table seront enregistrées par défaut), alors que l'assistant à la saisie Détails peut être utilisé pour éditer les propriétés et le contenu des composants de table. L'onglet Table lui-même permet aussi l'édition directe des définitions de table.

Aperçu de mise en page de la table XBRL

Afin de consulter au préalable la mise en page d'une définition de table, l'Éditeur de taxonomie XBRL propose un volet d'Aperçu de mise en page de table XBRL dans l'onglet Table du mode XBRL (*voir capture d'écran ci-dessous*). Lorsqu'une table ou un composant de table est choisi dans le diagramme, un aperçu de la table est affiché dans le volet Aperçu de mise en page de la table en-dessous du diagramme (*voir capture d'écran ci-dessous*). En alternative, vous pouvez sélectionner une table depuis la liste déroulante de la liste de choix du volet d'aperçu. Voici une liste des tables dans la base de lien de la table.

© 2016-2022 Altova GmbH



Pour plus d'informations concernant les fonctions d'aperçu, voir les sections suivantes :

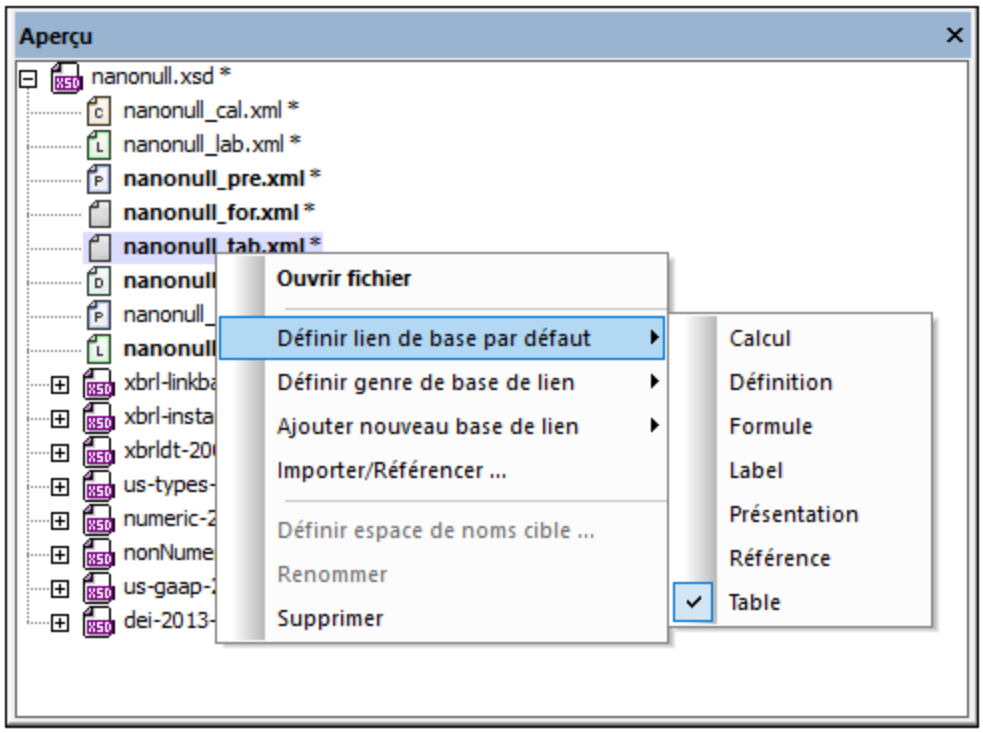
- Structure de table
- Paramètres de table
- Aperçu de mise en page de la table

17.5.1 Bases de lien et Rôles de lien de table

Alors que les base de lien XBRL standard (Définitions, Présentations, Calculs) définissent les relations entre les concepts par le biais de localisateurs et d'arcs de standard dans des liens étendus standard, une base de lien de table définit les composants de table (tables, répartitions, nœuds définition, etc.) et leurs relations. Ces définitions sont spécifiées par le biais d'arcs de ressource et génériques dans des liens étendus génériques.

Ajouter une base de lien de table

Dans l'assistant à la saisie Aperçu (*capture d'écran ci-dessous*), cliquer avec la touche de droite sur le fichier de taxonomie ou sur une base de lien existante et choisir **Ajouter nouvelle base de lien | Table**. La base de lien ajoutée deviendra la base de lien de la table par défaut. Le fichier de base de lien de table par défaut est le fichier dans lequel les nouvelles définitions de table seront enregistrées lorsque le fichier de taxonomie est enregistré. Si vous souhaitez faire d'un autre fichier de base de lien de table la base de lien par défaut, cliquer avec la touche de droite et choisir **Définir la base de lien par défaut | Table** (*voir capture d'écran ci-dessous*).



Veuillez noter que les bases de lien par défaut sont affichées en gras et que les bases de lien qui ont été modifiées mais pas encore enregistrées sont marquées par un astérisque.

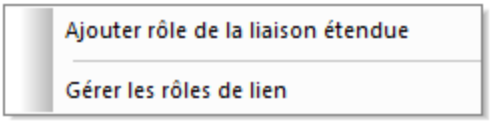
La base de lien de table est affichée dans l'onglet Table.

Note : Si un composant de table est ajouté à la taxonomie à un moment où aucune base de lien n'existe, une base de lien de table sera créée automatiquement.

Rôles de lien

Comme c'est le cas avec les liens étendus standard (pour Définitions, Présentations, Calculs), les liens génériques doivent définir une valeur de rôle de lien étendu, qui partitionne les relations du même type dans des réseaux disjoints. Tous les liens étendus génériques possédant le même rôle de lien sont combinés sous un seul nœud de rôle de lien dans le diagramme de l'onglet Table, *même s'ils résident dans des fichiers de base de lien différents*.

Les rôles de lien génériques peuvent être créés dans le diagramme par le biais du menu contextuel de l'espace d'arrière-plan (*capture d'écran ci-dessous*). Veuillez noter, néanmoins que ce menu contextuel sera affiché uniquement si la liste de choix de l'option Affichage a été réglée sur *Afficher tous les rôles de liens étendus*.



 © 2016-2022 Altova GmbH

Ce menu est aussi disponible par l'icône de barre d'outils , *Ajouter lien étendu / Gérer les rôles de lien*. Puisque des réseaux de relation ne sont pas essentiels pour une base de lien de table, l'affichage par défaut de l'onglet Table est *Dissimuler Rôles de lien*, qui dissimule les rôles de lien et, à la place, montre les composants de table sans leurs rôles de lien.

S'il n'y a pas de fichier de base de lien par défaut au moment de la création d'un rôle de lien, un fichier de base de lien par défaut sera créé automatiquement. Et s'il n'existe pas de rôle de lien dans le fichier de base de lien par défaut au moment de la création du rôle de lien, un rôle de lien sera créé automatiquement dans le fichier de base de lien par défaut.

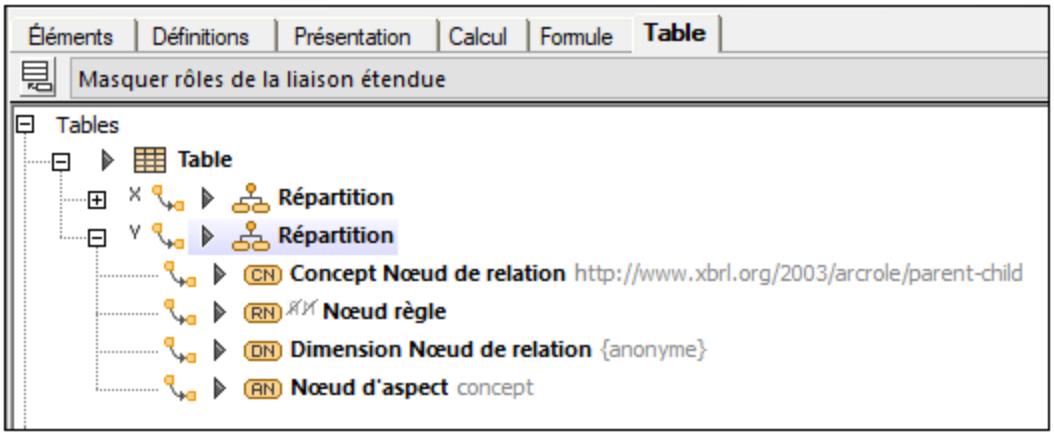
17.5.2 Structure de table

La structure d'une table dans la définition de table est définie par les axes de la table (X,Y,Z) dont chacune d'entre elle correspond à un ou plusieurs composants de répartition (*voir capture d'écran ci-dessous*).

- Les axes X et Y correspondent, respectivement, aux colonnes et aux lignes de la table générée. Elles sont décrites dans la section, Axes X et Y.
- Si un axe Z est défini, il est présenté en tant que table séparée. Voir la section Axe Z.
- Chaque composant de répartition peut contenir plusieurs nœuds de définition de table (*voir capture d'écran ci-dessous*). Il existe des types différents de nœuds de définition :

  – nœuds de règle (icône `RN` dans la capture d'écran ci-dessous)
  – nœuds de relation de concept (`CN`)
  – nœuds de relation de dimension (`DN`) et
  – nœuds d'aspect (`AN`).

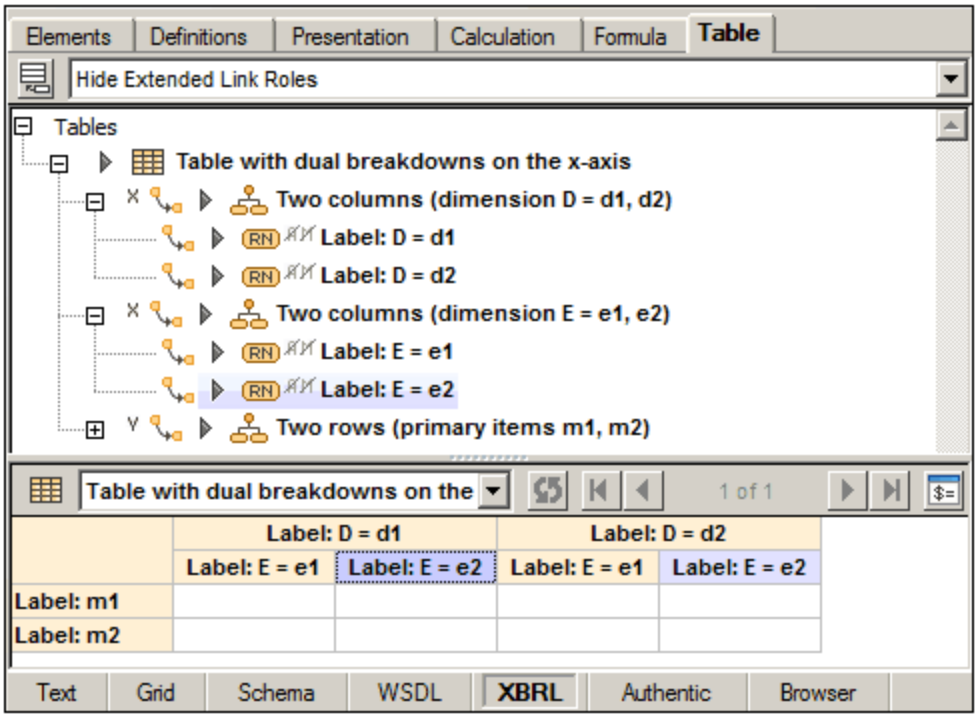
  Voir la section nœuds de définition pour une description des propriétés structurelles de ces nœuds de définition.



Projections pour les répartitions multiples

Des répartitions indépendantes multiples peuvent être associées avec une seule axe de table. Le mécanisme pour résoudre comment les répartitions multiples sont combinées dans une seule répartition “effective” s'appelle une projection. La priorité relative des répartitions multiples pour un axe unique est déterminée par l'attribut `@order` de chaque répartition. Les répartitions sont visualisées en tant qu'arborescences. Pour chaque feuille de la première répartition, toute la deuxième répartition est attachée, etc., récursivement.

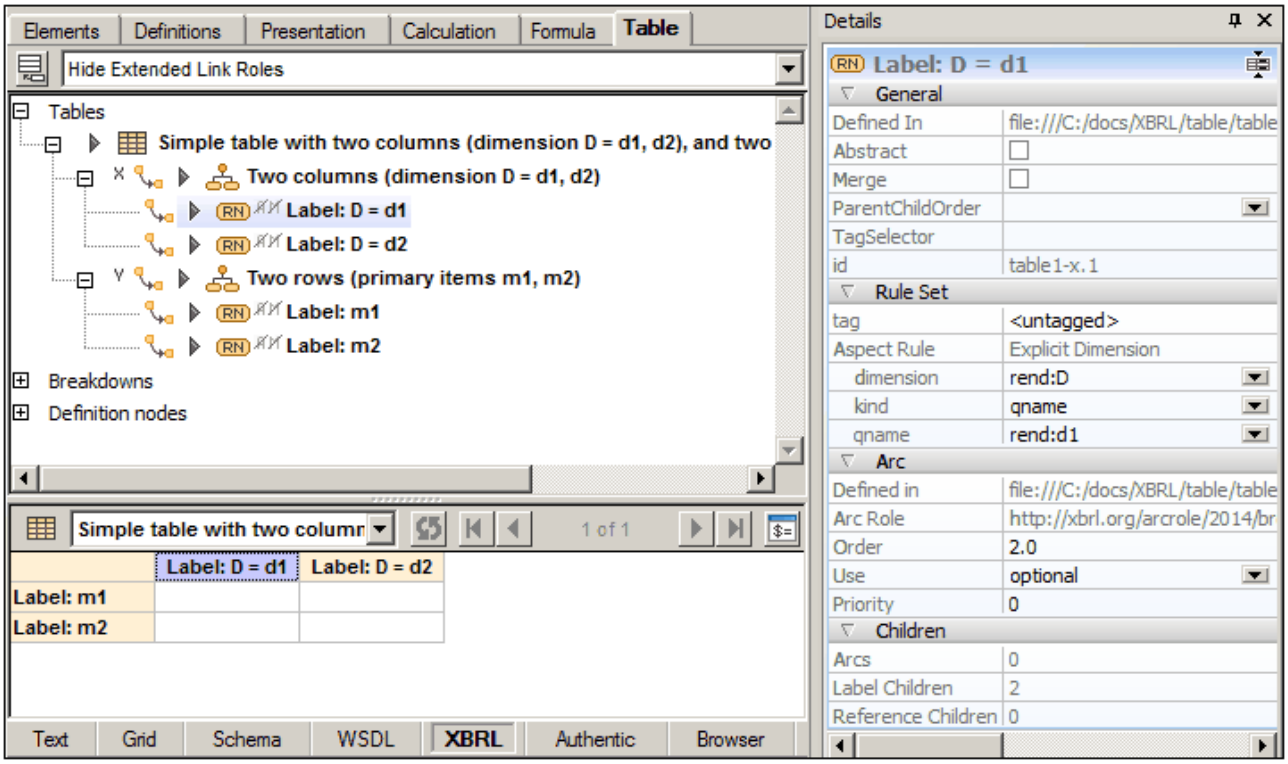
Dans la capture d'écran ci-dessous, par exemple, on peut voir deux répartitions pour les axes X : **`dimension D`** est classée à une propriété plus élevée que **`dimension E`**. Donc, pour chaque feuille de **`dimension D`** (**`d1`** et **`d2`**) toute l'arborescence de **`dimension E`** est attachée. Puisque l'axe X génère des colonnes, ces répartitions créent une projection pour la structure en colonne de la table. Voir l'aperçu de mise en page de la table dans la capture d'écran ci-dessous.



17.5.2.1 Axes X et Y

Les axes X et Y déterminent, respectivement, les colonnes et les lignes d'une table. Pour chaque axe, une ou plusieurs répartitions hiérarchiques sont définies (*voir capture d'écran ci-dessous*). Les répartitions correspondant à un seule axe sont résolues dans une seule répartition “effective. S'il n'y a qu'une répartition pour un axe, cette répartition sera la répartition effective. Si plusieurs répartitions sont définies pour un axe, la méthode de résolution est celle décrite dans la section Projections pour les répartitions multiples.

© 2016-2022 Altova GmbH



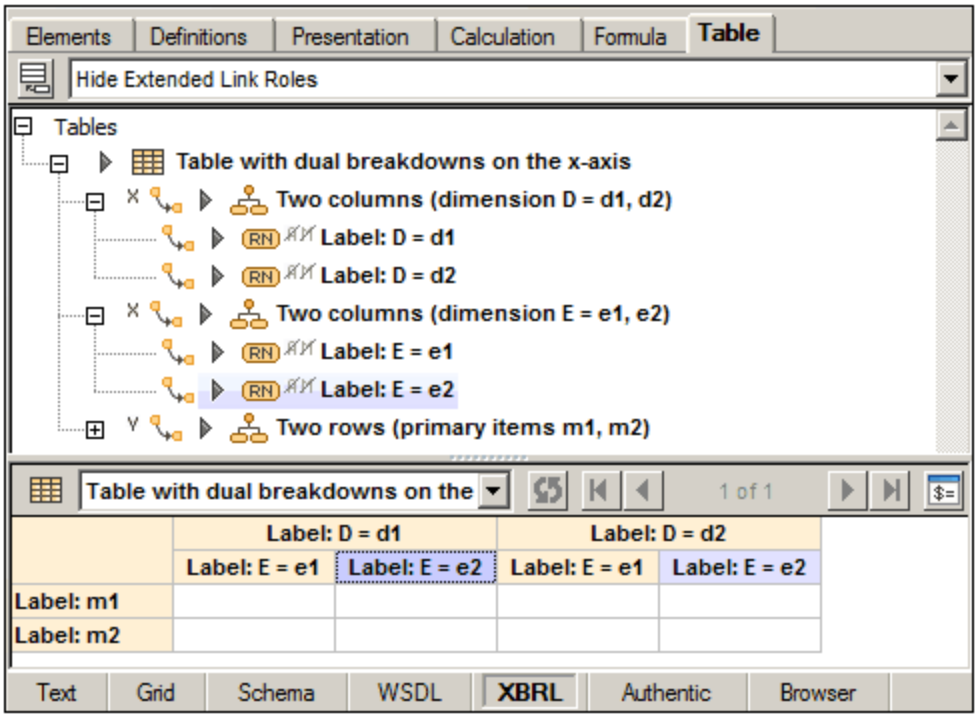
Veuillez noter les propriétés concernant les axes et les fonctions d'édition :

- Dans les définitions de table, l'axe X correspond aux colonnes de la table générée, alors que l'axe X correspond aux lignes de la table (*capture d'écran de l'aperçu de la mise en page de la table ci-dessus*).
- Chaque axe peut avoir une ou plusieurs répartition (voir Projections pour les répartitions multiples ci-dessous).
- Chaque cellule dans la table générée présente une couleur de fond orange pâle. Dans la définition de table, une cellule correspond à un nœud de définition dans une répartition de l'axe.
- Lorsqu'une cellule est sélectionnée, son nœud de définition correspondant est aussi sélectionné, et vice-versa. La couleur de fond des cellules des composants sélectionnés est mauve.
- Lorsqu'un composant est sélectionné, ses propriétés sont affichées dans l'assistant à la saisie Détails et peuvent y être éditées (*voir capture d'écran ci-dessus*).
- Les cellules de données n'ont pas de couleur de fond. Elles sont toujours vides parce que la taxonomie elle-même ne contient pas de faits.
- Les contraintes de cellule sont calculées depuis les axes (avec des sélecteurs de balise, le cas échéant) et sont affichées dans l'onglet Contraintes de l'assistant à la saisie Détails. Voir la capture d'écran dans la section, Axe Z.

Projections pour les répartitions multiples

Des répartitions indépendantes multiples peuvent être associées avec une seule axe de table. Le mécanisme pour résoudre comment les répartitions multiples sont combinées dans une seule répartition “effective” s'appelle une projection. La priorité relative des répartitions multiples pour un axe unique est déterminée par l'attribut `@order` de chaque répartition. Les répartitions sont visualisées en tant qu'arborescences. Pour chaque feuille de la première répartition, toute la deuxième répartition est attachée, etc., récursivement.

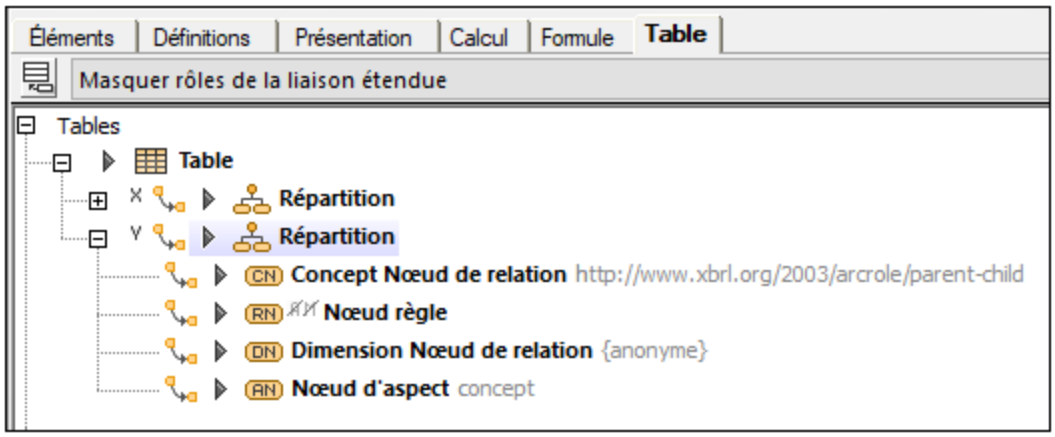
Dans la capture d'écran ci-dessous, par exemple, on peut voir deux répartitions pour les axes X : **dimension D** est classée à une propriété plus élevée que **dimension E**. Donc, pour chaque feuille de **dimension D** (**d1** et **d2**) toute l'arborescence de **dimension E** est attachée. Puisque l'axe X génère des colonnes, ces répartitions créent une projection pour la structure en colonne de la table. Voir l'aperçu de mise en page de la table dans la capture d'écran ci-dessous.



17.5.2.2 Nœuds de définition

Chaque composant de répartition peut contenir plusieurs nœuds de définition de table (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH

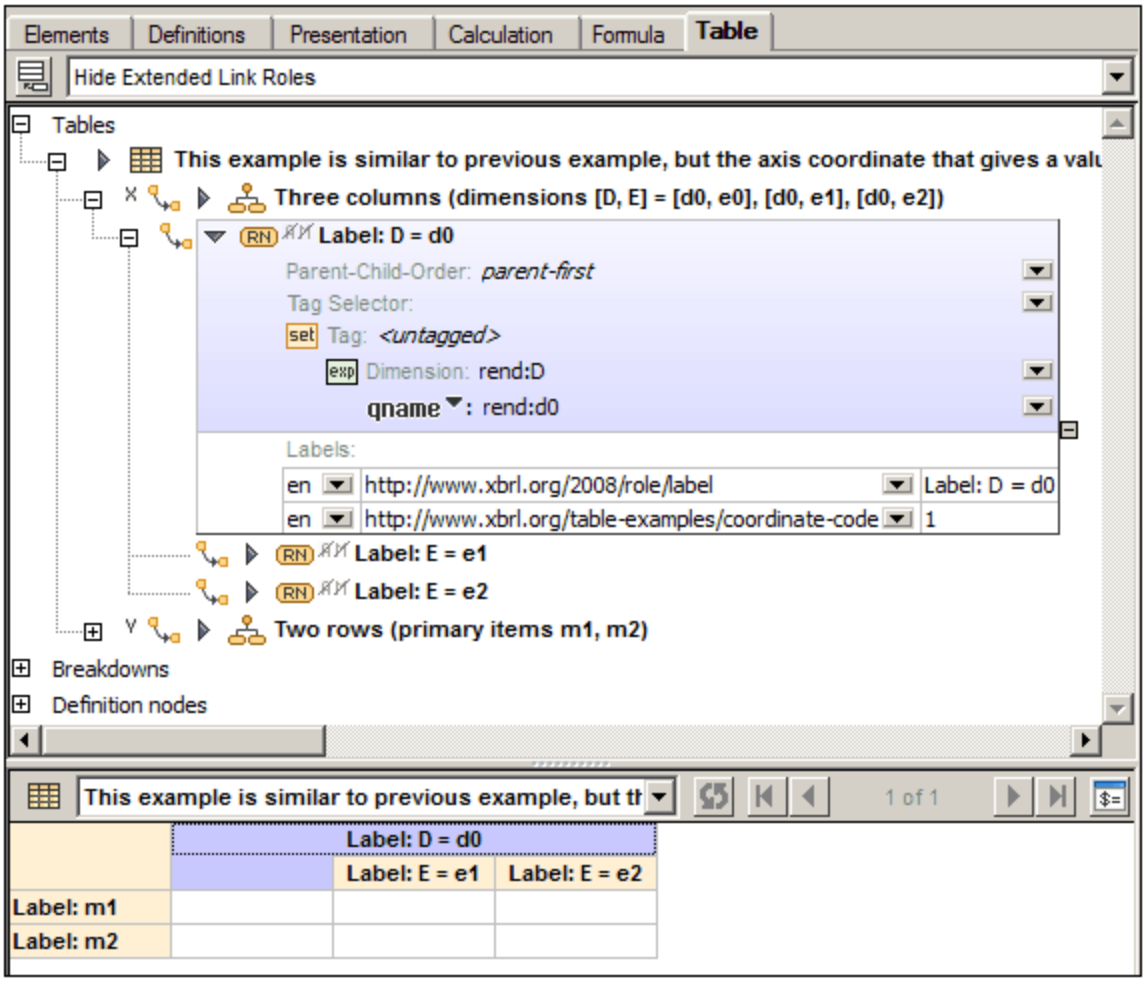


Il existe trois différents types de nœuds de définition :

- Nœuds de règle (**RN** icône dans la capture d'écran ci-dessus)
- Nœuds de relation de concept (**CN**)
- Nœuds de relation de dimension (**DN**)
- Nœuds d'aspect (**AN**)

17.5.2.2.1 Nœuds de règle

Un nœud de règle définit des règles d'aspect pour un ou plusieurs aspects : concept, période, unité, identifiant d'entité, dimension, ou ouvrir aspect de contenu. Le composant dans l'arborescence de définition correspond à exactement une cellule dans la mise en page si le nœud de règle est abstrait ou n'a pas d'enfants. Sinon, la mise en page contient une cellule roll-up supplémentaire dont le placement est déterminé par la valeur effective de la propriété du nœud de règle `parentChildOrder` :



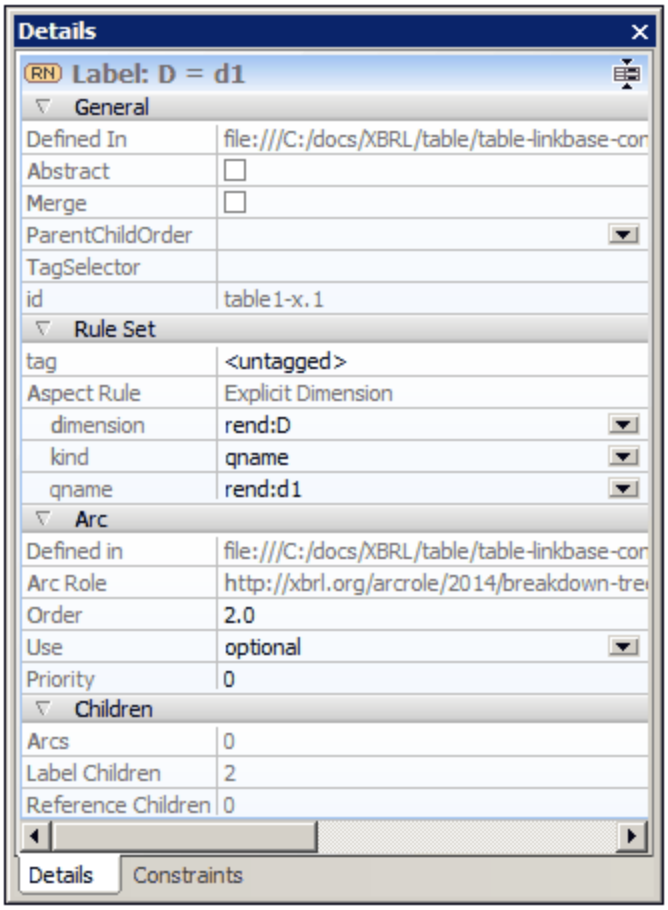
L'en-tête de la cellule de mise en page est calculée depuis le nœud de règle comme suit :

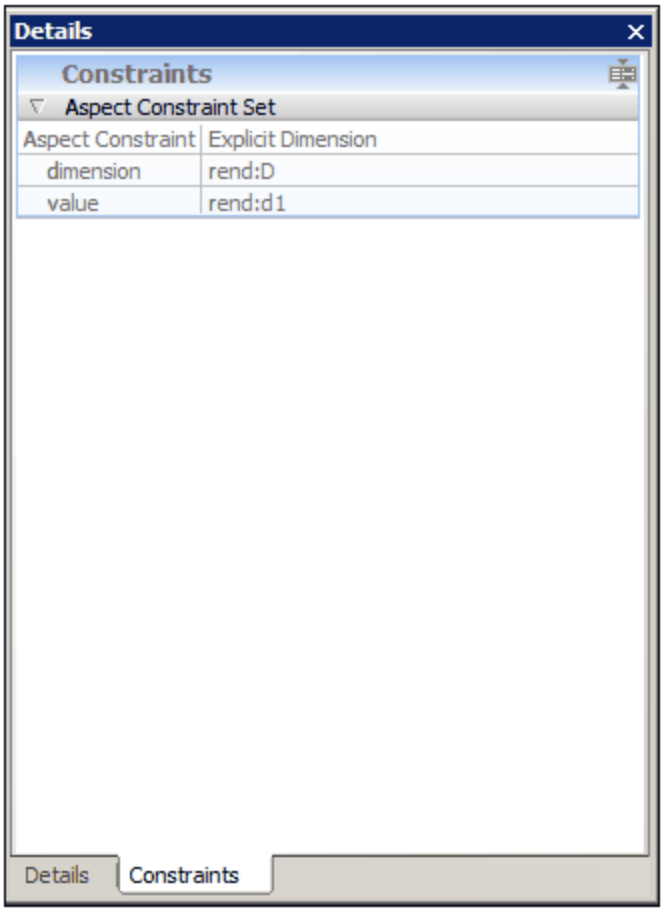
- Si le nœud est associé avec un libellé défini par l'utilisateur, le texte de ce libellé est affiché.
- S'il n'y a pas de libellé, mais que le nœud définit une contrainte d'aspect unique (concept, dimension, unité, identifiant d'entité ou période), sa valeur est affichée (par exemple, le nom qualifié du concept).
- Dans le cas contraire, le texte statique `Rule node` est utilisé.

Assistant à la saisie Détails

Les propriétés du nœud de définition sont affichées dans l'onglet Détails de l'assistant à la saisie (*capture d'écran ci-dessous à gauche*). L'onglet Contraintes (*capture d'écran ci-dessous droite*) permet une consultation en lecture seule des ensembles de contrainte d'aspect qui sont calculés à partir des règles d'aspect du nœud.

© 2016-2022 Altova GmbH

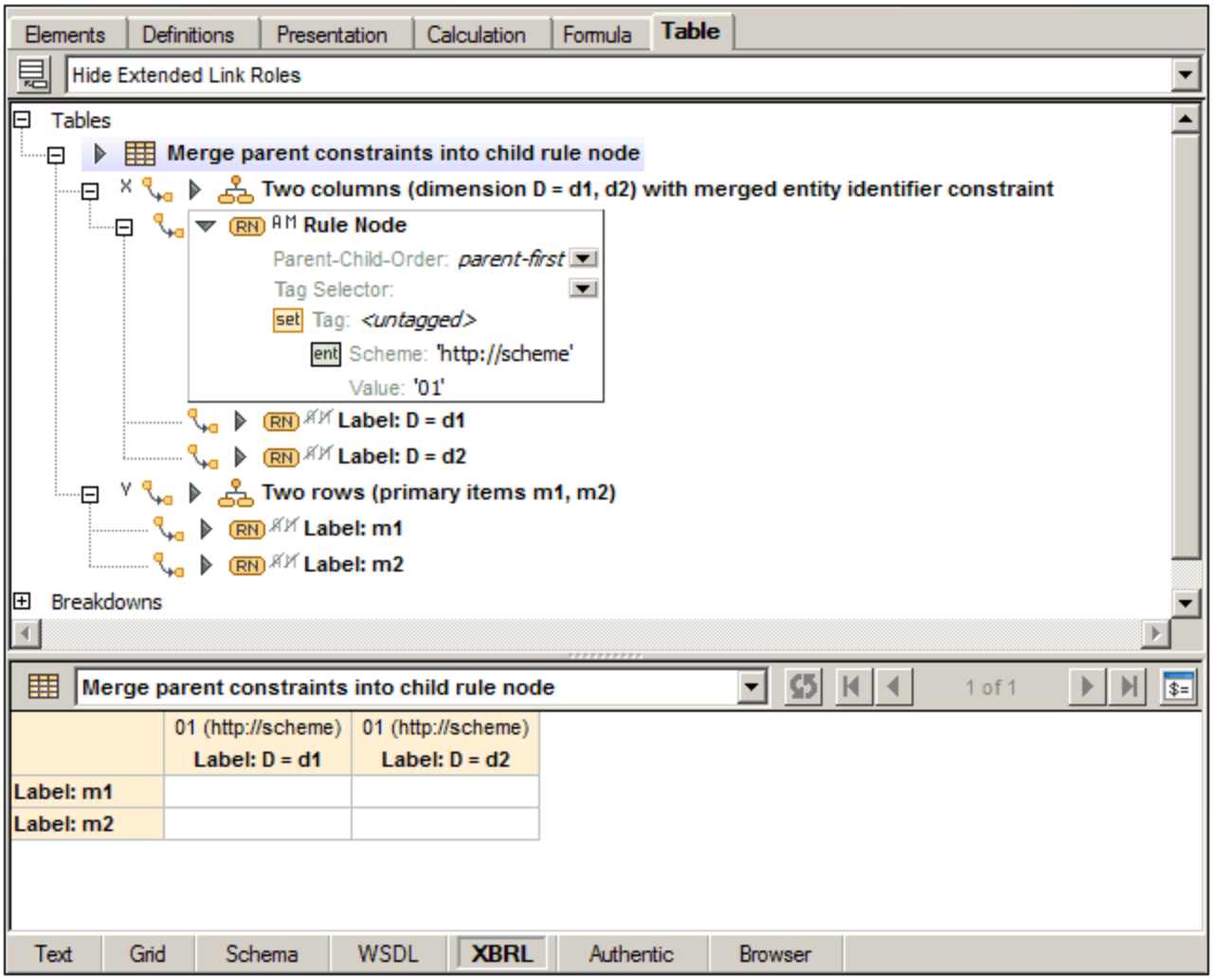

Details
Label: D = d1
General
Defined In
file:///C:/docs/XBRL/table/table-linkbase-cor
Abstract
Merge
ParentChildOrder
TagSelector
id
table1-x.1
Rule Set
tag
<untagged>
Aspect Rule
Explicit Dimension
dimension
rend:D
kind
qname
qname
rend:d1
Arc
Defined in
file:///C:/docs/XBRL/table/table-linkbase-cor
Arc Role
http://xbrl.org/arcrole/2014/breakdown-tre
Order
2.0
Use
optional
Priority
0
Children
Arcs
0
Label Children
2
Reference Children
0
Details
Constraints



Nœuds de règle fusionnés

Un nœud de règle fusionné indique des propriétés supplémentaires qui s'appliquent à tous ses enfants, c'est à dire qu'il contribue toutes ses contraintes dans chaque ensemble de contrainte produit par ses enfants (*voir capture d'écran ci-dessous*).

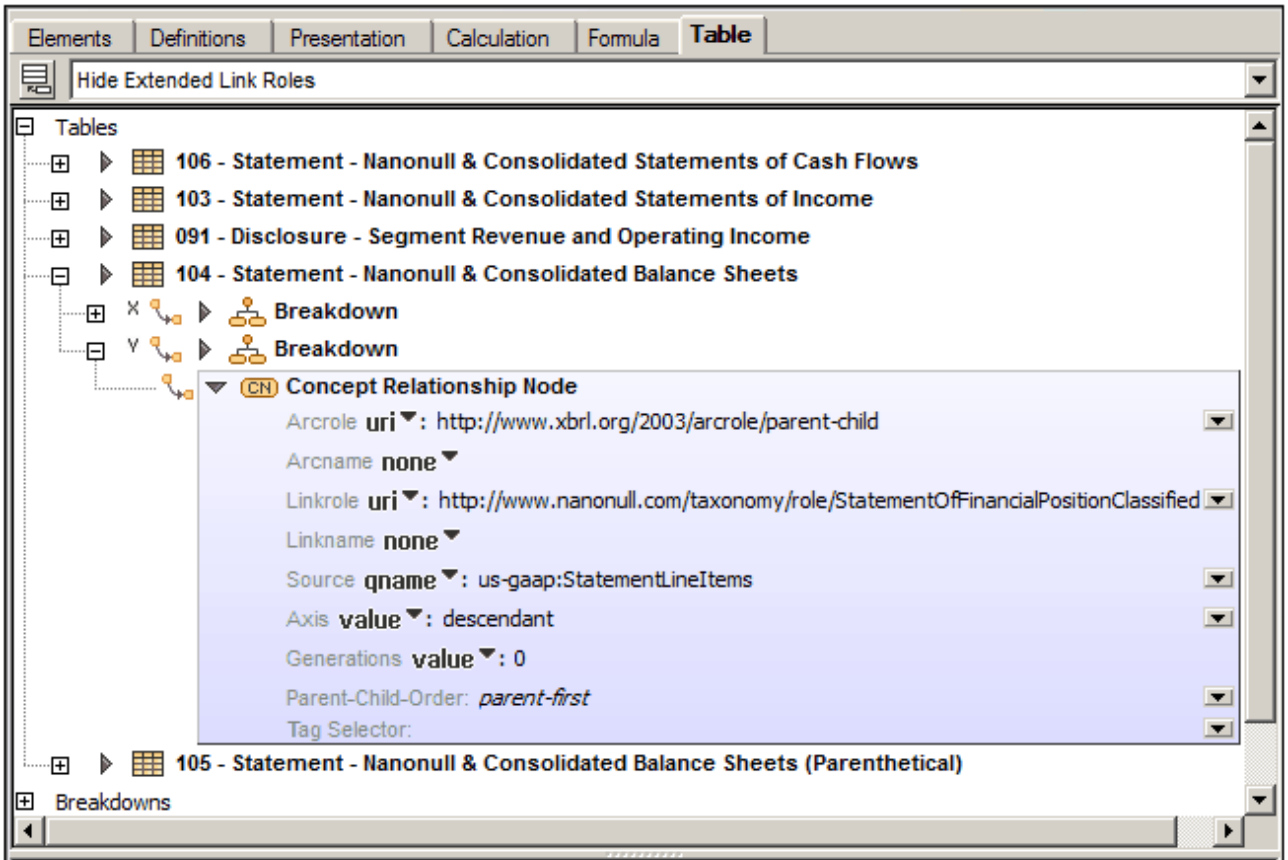
© 2016-2022 Altova GmbH



17.5.2.2.2 Nœuds de relation

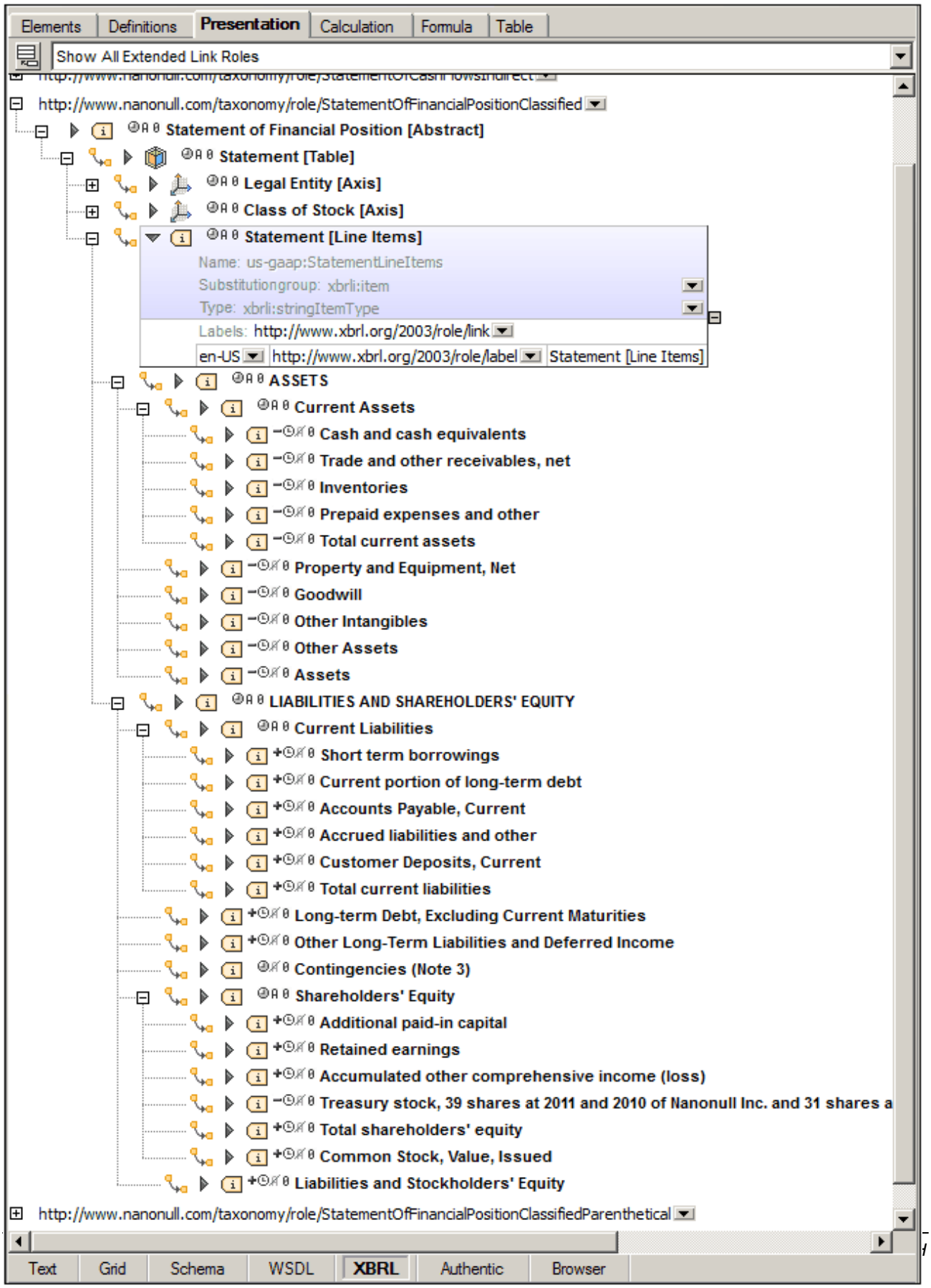
Un nœud de relation de concept ou de dimension résout en une arborescence de nœuds structurels, définis par des réseaux de concepts ou de membres de dimension explicite dans le DTS. C'est pourquoi le composant dans l'arborescence de définition correspond à un bloc de cellules dans la mise en page.

La capture d'écran à gauche montre une définition de table contenant un nœud de relation de concept. La capture d'écran à droite montre le réseau correspondant de concepts, dans ce cas, défini dans la base de lien de la présentation.



104 - Statement - Nanonull & Consolidated Balance Sheets — 1 of 1

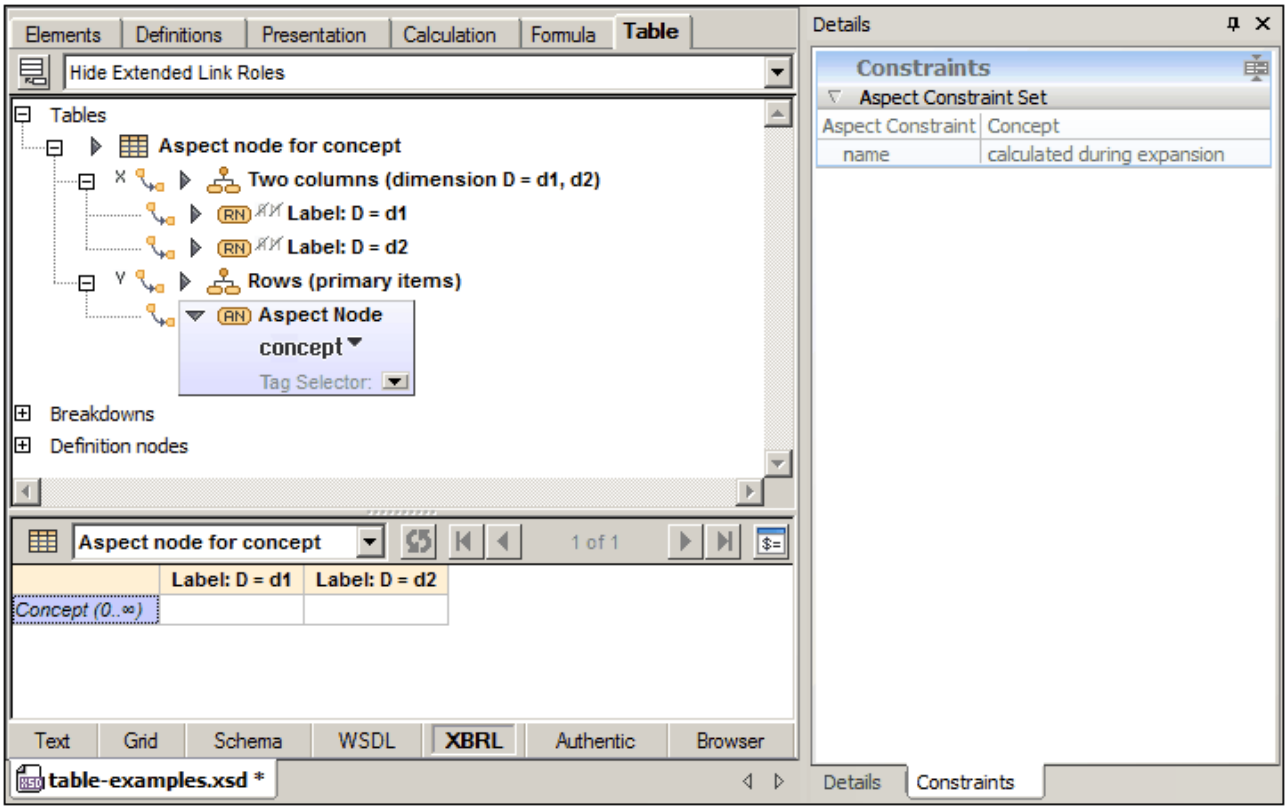
			2010-08-31	2009-11-30	2
ASSETS	Current Assets	Cash and cash equivalents			
		Trade and other receivables, net			
		Inventories			
		Prepaid expenses and other			
		Total current assets			
	Property and Equipment, Net				
	Goodwill				
	Other Intangibles				
	Other Assets				
	Assets				
LIABILITIES AND SHARE	Current Liabilities	Short term borrowings			
		Current portion of long-term debt			
		Accounts Payable, Current			
		Accrued liabilities and other			
		Customer Deposits, Current			
		Total current liabilities			
	Long-term Debt, Excluding Current Maturities				
	Other Long-Term Liabilities and Deferred Income				
	Contingencies (Note 3)				
	Shareholders' Equity	Additional paid-in capital			
		Retained earnings			
		Accumulated other comprehensive			
		Treasury stock, 39 shares at 2011 an			
		Total shareholders' equity			
		Common Stock, Value, Issued			
	Liabilities and Stockholders' Equity				


Elements
Definitions
Presentation
Calculation
Formula
Table
Show All Extended Link Roles
http://www.nanonull.com/taxonomy/role/StatementOfFinancialPositionClassified
Statement of Financial Position [Abstract]
Statement [Table]
Legal Entity [Axis]
Class of Stock [Axis]
Statement [Line Items]
Name: us-gaap:StatementLineItems
Substitutiongroup: xbrli:item
Type: xbrli:stringItemType
Labels: http://www.xbrl.org/2003/role/link
en-US
http://www.xbrl.org/2003/role/label
Statement [Line Items]
ASSETS
Current Assets
Cash and cash equivalents
Trade and other receivables, net
Inventories
Prepaid expenses and other
Total current assets
Property and Equipment, Net
Goodwill
Other Intangibles
Other Assets
Assets
LIABILITIES AND SHAREHOLDERS' EQUITY
Current Liabilities
Short term borrowings
Current portion of long-term debt
Accounts Payable, Current
Accrued liabilities and other
Customer Deposits, Current
Total current liabilities
Long-term Debt, Excluding Current Maturities
Other Long-Term Liabilities and Deferred Income
Contingencies (Note 3)
Shareholders' Equity
Additional paid-in capital
Retained earnings
Accumulated other comprehensive income (loss)
Treasury stock, 39 shares at 2011 and 2010 of Nanonull Inc. and 31 shares a
Total shareholders' equity
Common Stock, Value, Issued
Liabilities and Stockholders' Equity
http://www.nanonull.com/taxonomy/role/StatementOfFinancialPositionClassifiedParenthetical
Text
Grid
Schema
WSDL
XBRL
Authentic
Browser

Un nœud de relation a exactement une contrainte d'aspect : concept ou dimension explicite. C'est pourquoi l'en-tête de chaque cellule de mise en page est le libellé de concept (le cas échéant) ou son nom qualifié. L'onglet Détails de l'assistant à la saisie Détails montre les propriétés du nœud de relation, alors que l'onglet Contraintes fournit la contrainte d'aspect (ensemble) qui est définie par la cellule ciblée dans la mise en page.

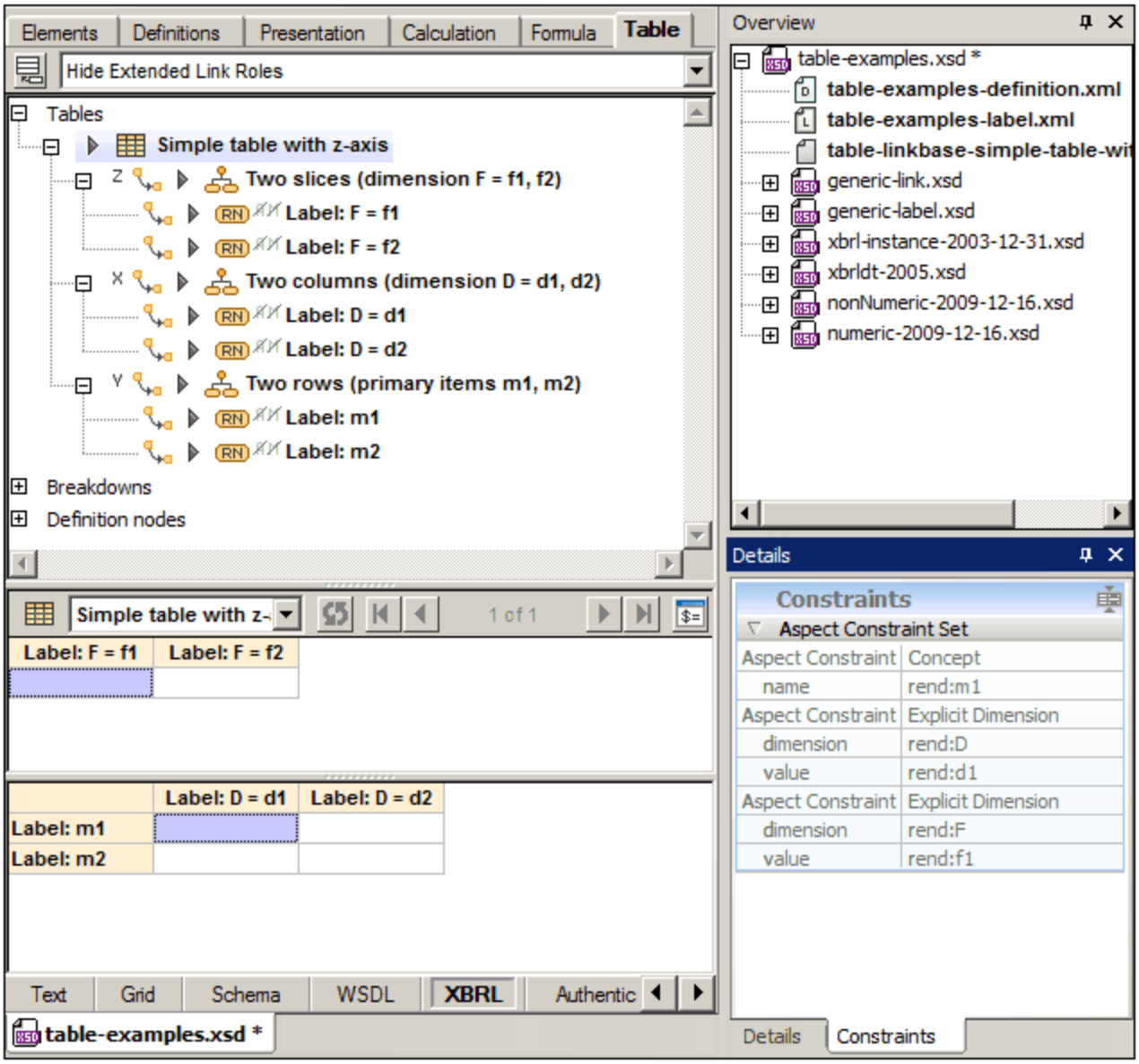
17.5.2.2.3 Nœuds d'aspect

Un nœud d'aspect est un nœud de définition ouvert qui spécifie directement un aspect de participation unique. Au cours du process de mise en page, un nœud d'aspect s'étend à une cellule pour chaque valeur distincte de son aspect de participation qui est présent parmi les faits d'un fichier d'instance XBRL. Puisque la contrainte de valeur d'aspect n'est pas entièrement déterminée par la définition de nœud et le DTS, l'aperçu de mise en page affiche un espace réservé (*voir capture d'écran ci-dessous*).



17.5.2.3 Axe Z

Si une définition de table contient un axe Z, cet axe sera interprété en tant que table bidimensionnelle multiple. Dans l'Aperçu de mise en page de la table, l'axe Z est affiché en tant que table séparée au-dessus de la table XY (*voir capture d'écran ci-dessous*).



Si une définition de table contient un axe Z, alors, à tout moment dans l'aperçu de table, le focus sera toujours porté sur **deux** cellules de données (voir capture d'écran ci-dessus). Les coordonnées pour les trois axes sont spécifiées de la manière suivante : les coordonnées X et Y dans la table XY, et les coordonnées Z dans la table Z. Vous pouvez le constater dans la capture d'écran ci-dessus.

Les propriétés et fonctions d'édition liées à l'axe sont les mêmes que pour les axes X et Y. Les contraintes de cellules sont calculées depuis les axes (en utilisant des sélecteurs de balise, le cas échéant) et sont affichées dans l'onglet Contraintes de l'assistant à la saisie Détails (*voir capture d'écran ci-dessus*).

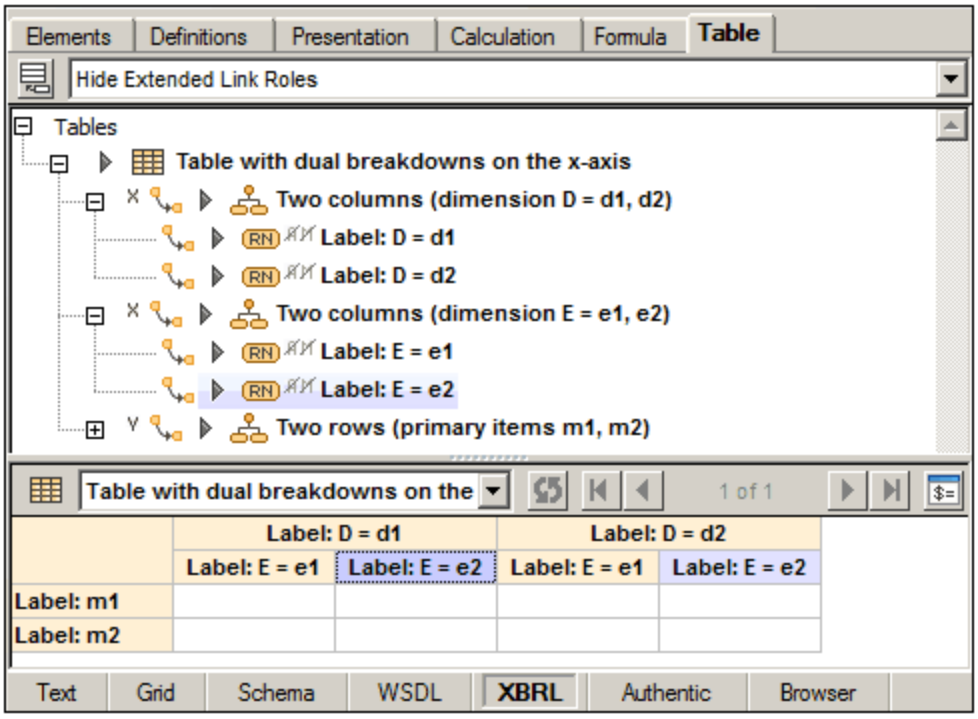
Projections pour les répartitions multiples

Des répartitions indépendantes multiples peuvent être associées avec une seule axe de table. Le mécanisme pour résoudre comment les répartitions multiples sont combinées dans une seule répartition “effective”

© 2016-2022 Altova GmbH

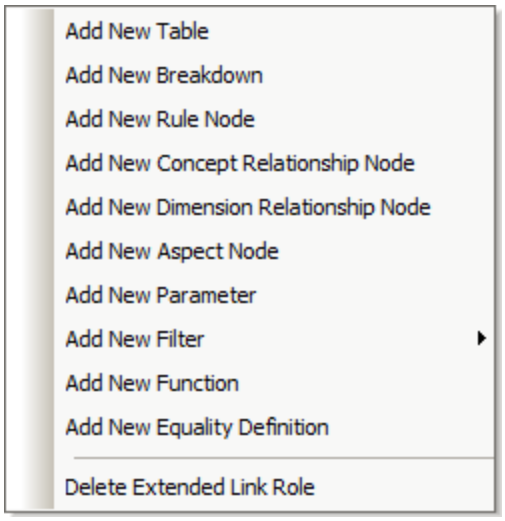
s'appelle une projection. La priorité relative des répartitions multiples pour un axe unique est déterminée par l'attribut `@order` de chaque répartition. Les répartitions sont visualisées en tant qu'arborescences. Pour chaque feuille de la première répartition, toute la deuxième répartition est attachée, etc., récursivement.

Dans la capture d'écran ci-dessous, par exemple, on peut voir deux répartitions pour les axes X : **`dimension D`** est classée à une propriété plus élevée que **`dimension E`**. Donc, pour chaque feuille de **`dimension D`** (**`d1`** et **`d2`**) toute l'arborescence de **`dimension E`** est attachée. Puisque l'axe X génère des colonnes, ces répartitions créent une projection pour la structure en colonne de la table. Voir l'aperçu de mise en page de la table dans la capture d'écran ci-dessous.

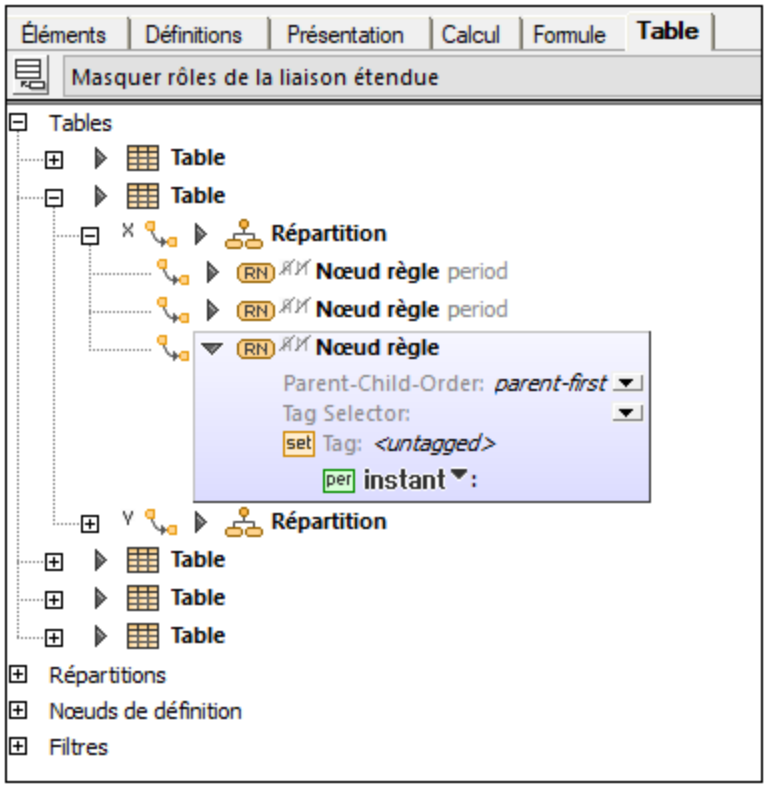


17.5.3 Composants de table

Les nouveaux composants de table sont créés par le biais du menu contextuel d'un nœud de rôle de lien (*capture d'écran ci-dessous*) ; ou, avec le mode défini sur *dissimuler les rôles de lien étendus*, par le biais de l'icône de barre d'outils,  *Ajouter nouveau composant de table*.



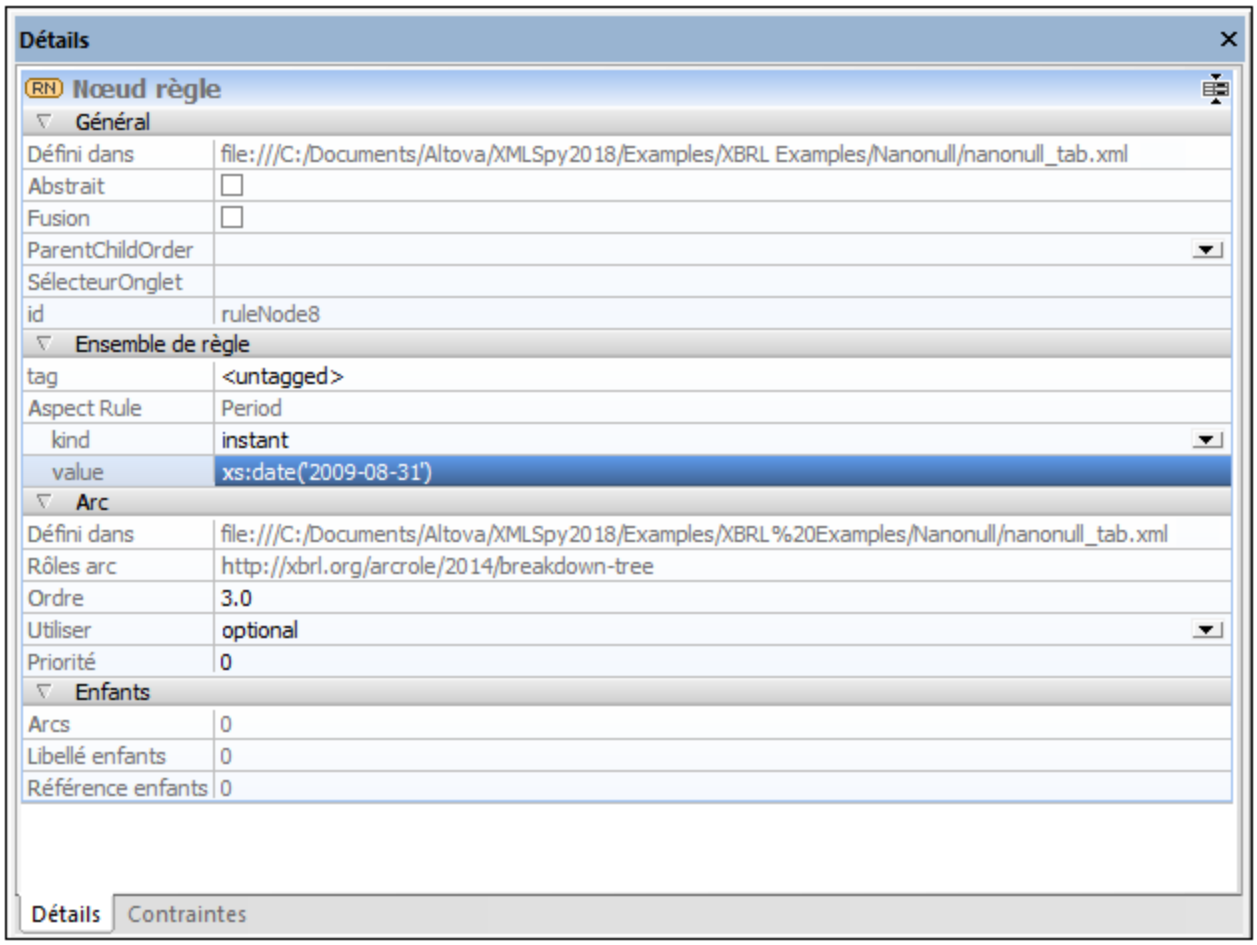
Les mécanismes impliqués dans l'ajout des différents composants sont décrits dans les sous-sections de cette section. Une fois qu'un composant de table a été ajouté, il est affiché dans le diagramme de l'onglet Table (*voir capture d'écran ci-dessous*).



Pour une consultation plus claire, les composants de table sont divisés en sections et les relations vers d'autres composants (les arcs) sont affichés dans le cadre d'une structure arborescente (*voir capture d'écran ci-dessus*). Les propriétés des composants et des relations (arcs) sont affichées dans le diagramme sous la forme d'icônes situées à gauche du composant ou de l'arc respectivement (*voir capture d'écran ci-dessus*).

© 2016-2022 Altova GmbH

L'assistant à la saisie Détails du Rule Node marqué dans la capture d'écran ci-dessus est affiché ci-dessous. Les propriétés du nœud sont recensées dans la section *Général*. Les valeurs des propriétés booléennes sont indiquées par une coche pour `true` et sans coche pour `false`. Des sections supplémentaires recensent d'autres détails liés au nœud.



Pour voir les propriétés d'un arc dans l'assistant à la saisie Détails sélectionner le composant `to` (destination) dans le diagramme ; les propriétés d'arc seront affichées dans la section *Arc*.

Menus contextuels dans l'Éditeur Table

Les menus contextuels des composants de table varient conformément au type du composant. Les items de menu sont organisés dans des sections, comme suit :

- Modification de relation (uniquement pour les sous-items) : *Surmonter/Supprimer Arc*
- Modification de contenu (rule node, nœuds de relation) : par exemple, *Ajouter/Insérer règle aspect*
- *Ajouter Libellés/Références*
- Création de nouveaux composants enfants (y compris les relations) : par exemple, *Ajouter nouvelle répartition*
- Suppression de composants (y compris les relations)
- *Trouver occurrence suivante/précédente* (de composant)

Note : les items de contenu qui peuvent être créés ou supprimés par le biais du menu contextuel sont affichés dans l'assistant à la saisie Détails dans des sections supplémentaires, comme *Ensemble de règle*.

17.5.3.1 Table

Une table fournit la propriété `parent-child-order` (parent en premier/enfants en premier). Elle définit le placement par défaut des nœuds roll-up contribués par tous les nœuds de définition fermés dans la table pour laquelle elle n'est pas contournée.



17.5.3.2 Répartition

Une répartition fournit la propriété `parent-child-order` (parent en premier/enfants en premier). Elle définit le placement par défaut des nœuds roll-up contribués par tous les nœuds de définition fermés dans la répartition et contourne la valeur héritée par la table.



17.5.3.3 Nœud de définition : Règle

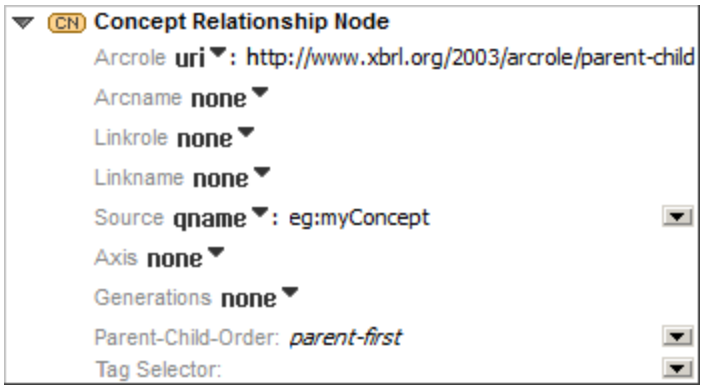
Un nœud de règle est un nœud de définition fermé qui définit des contraintes par le biais des règles d'aspect (voit composant de formule). Un nœud de règle définit zéro ou plusieurs ensembles de règle (c'est à dire des ensembles de règles d'aspect). Chaque règle peut spécifier une balise. Un seul de ces ensembles de règle, au maximum, peut omettre la balise. Cet ensemble de règle non balisé est toujours affiché avant tous les ensembles de règle balisés. Un ensemble de règle non balisé n'est pas affiché si au moins un ensemble de règle balisé est présent. Un nœud de règle fournit deux propriétés booléennes, `abstract` et `merge`, en tant qu'icônes. La capture d'écran ci-dessous montre un nœud de règle sans règles d'aspect.



© 2016-2022 Altova GmbH

17.5.3.4 Nœud de définition : Relation de concept

Un nœud de relation de concept découvre des concepts en effectuant une analyse d'arborescence d'un réseau XBRL 2.1. L'analyse d'arborescence est identifiée de manière unique par le réseau et par une ou plusieurs sources de relations. Un nœud de relation de concept doit identifier un réseau simple. Dans la plupart des cas, la combinaison de rôle de lien et de rôle d'arc est suffisant pour identifier de manière univoque le réseau, mais il peut être nécessaire de spécifier des informations supplémentaires comme le nom d'arc ou le nom du lien étendu.

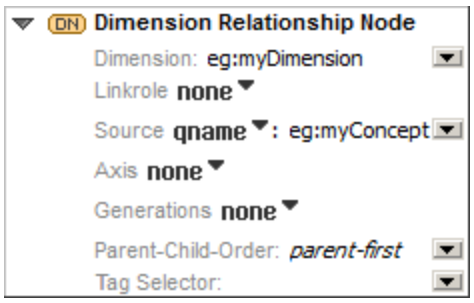



```
Arcrole:     Kind = uri | exp
Arcname:     Kind = none | qname | exp
Linkrole:    Kind = none | uri | exp
Linkname:    Kind = none | qname | exp
Source:      Kind = qname | exp
Axis:        Kind = none | value | exp
Generations: Kind = none | value | exp
```

Les nœuds de relation de concept ne peuvent pas avoir de sous-arborescences.

17.5.3.5 Nœud de définition : Relation de dimension

Un nœud de relation de dimension décrit une arborescence de membres de dimension explicite en termes d'une analyse arborescente d'en ensemble de relations dimensionnelles (DRS). Cette analyse arborescente est identifiée de manière unique par une ou plusieurs sources de relations.




```
Linkrole:     Kind = none | uri | exp
```

```
Source:       Kind = qname | exp
Axis:         Kind = none | value | exp
Generations:  Kind = none | value | exp
```

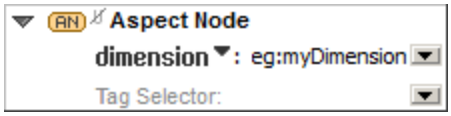
Les nœuds de relation de dimension ne peuvent pas avoir de sous-arborescences.

17.5.3.6 Nœud de définition : Aspect

Un nœud d'aspect spécifie exactement un aspect.



Les spécifications d'aspect dimensionnel fournissent une propriété booléenne supplémentaire `include unreported value` en tant qu'une icône.

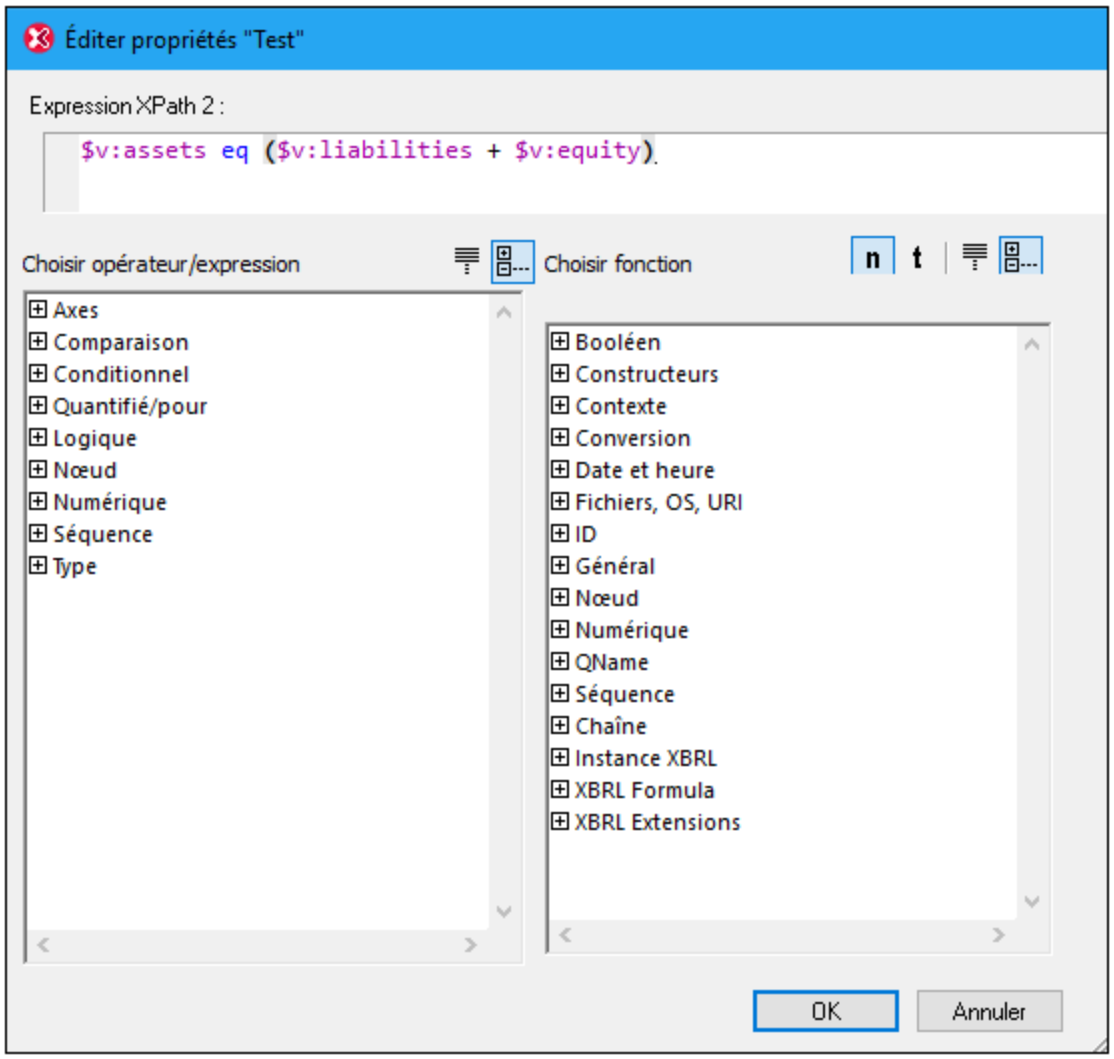


17.5.4 Éditer les propriétés et le contenu de composant

Les propriétés des composants de table peuvent être éditées directement dans le diagramme ou dans l'assistant à la saisie Détails.

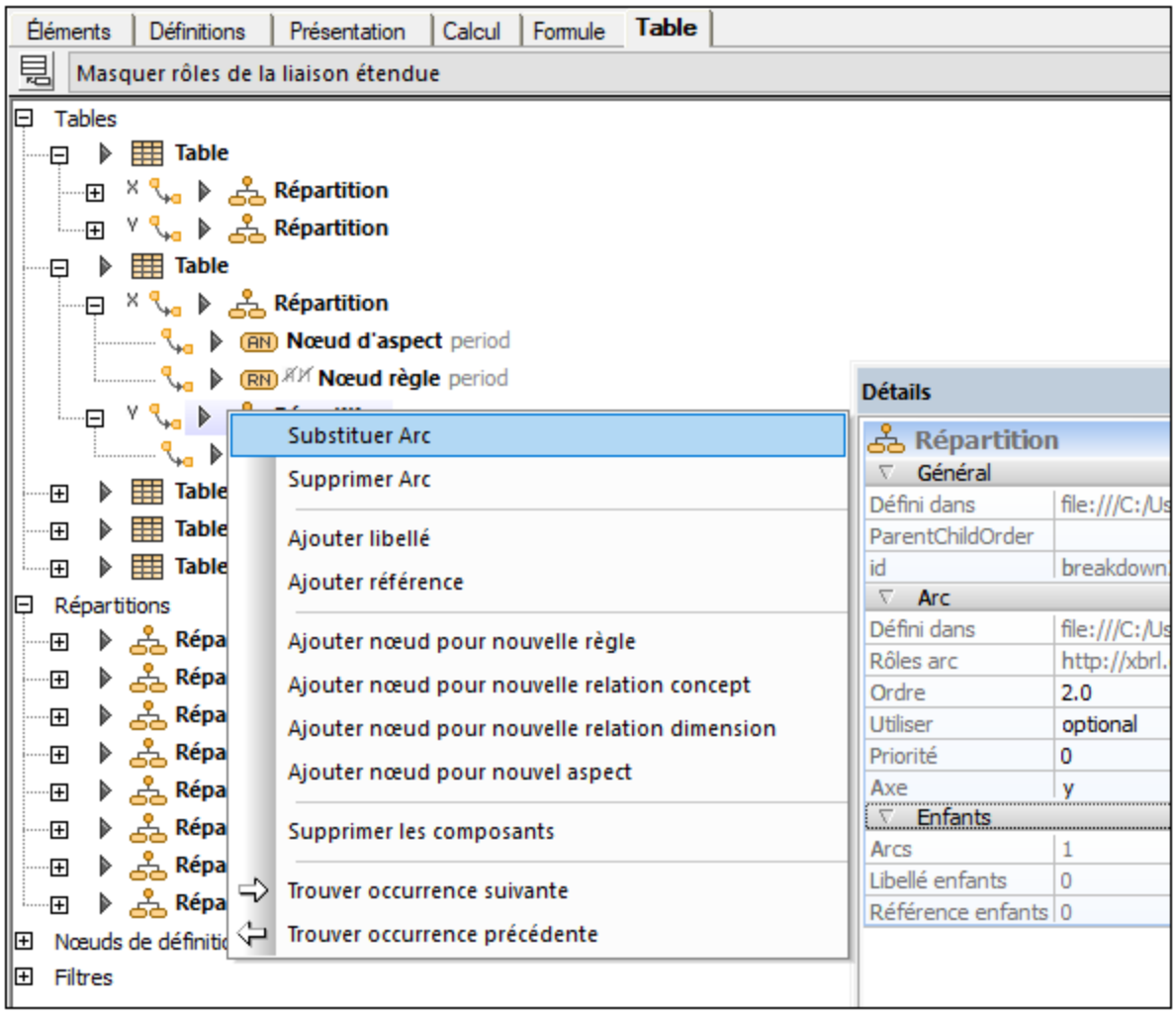
Dans le diagramme, lorsqu'un composant est réduit, soit son nom (s'il en a un), soit la valeur de la propriété par défaut appropriée est affichées en gris à côté du texte de description du composant. Double-cliquer le composant pour l'agrandir. Double-cliquer sur une propriété pour la mettre en mode d'édition. Si une propriété ou un contenu contient une expression XPath, l'expression Éditer XPath s'affiche.

© 2016-2022 Altova GmbH



17.5.5 Relations de composant de table

Une relation entre les composants de table peut être créée en reliant un composant de table à un autre par glisser/déposer. L'ordre des enfants de composant d'un parent dépend des valeurs de l'arc-propriété `order`. Cet ordre peut être modifié en déplaçant les enfants par glisser/déposer. Un composant d'enfant peut aussi être glissé dans ou sous un composant de parent différent afin de copier ou de déplacer la relation (y compris ses propriétés).

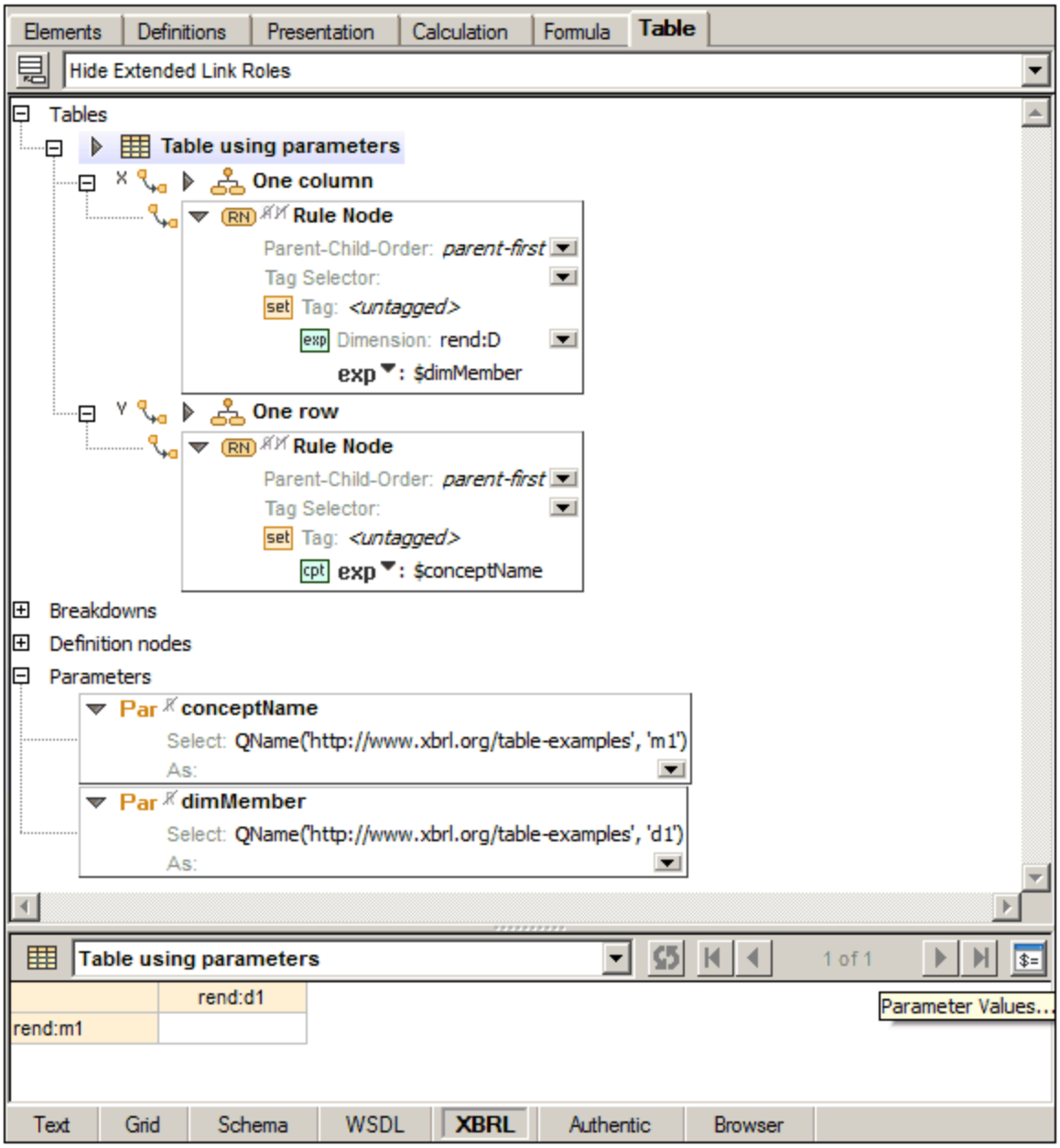


Lors de la création d'un nouveau composant par le biais du menu contextuel d'un composant de parent (existant), la relation (c'est à dire, l'arc) est générée automatiquement. Les commandes **Contourner Arc** et **Supprimer Arc** dans le menu contextuel de composant d'un enfant servent à contourner ou supprimer la relation entre le composant et son parent. Les arcs multiples des relations contournées sont affichés dans des sous-lignes. Le rôle d'arc des relations de composant de table ne peut pas être modifié.

17.5.6 Paramètres de table

Les paramètres de table peuvent être utilisés pour définir les axes d'une table. Par exemple, dans la capture d'écran ci-dessous, l'axe X de la table sélectionnée est définie par le paramètre `$dimMember` ; l'axe Y est défini par le paramètre `$conceptName`. Les définitions des deux paramètres, quant à eux, sont affichés dans la liste des paramètres (globaux) en dessous des définitions de la table. L'Aperçu de mise en page de la table dans le panneau inférieur montre la table qui sera générée. Les axes sont créés en tant que ligne et colonne de la table.

© 2016-2022 Altova GmbH

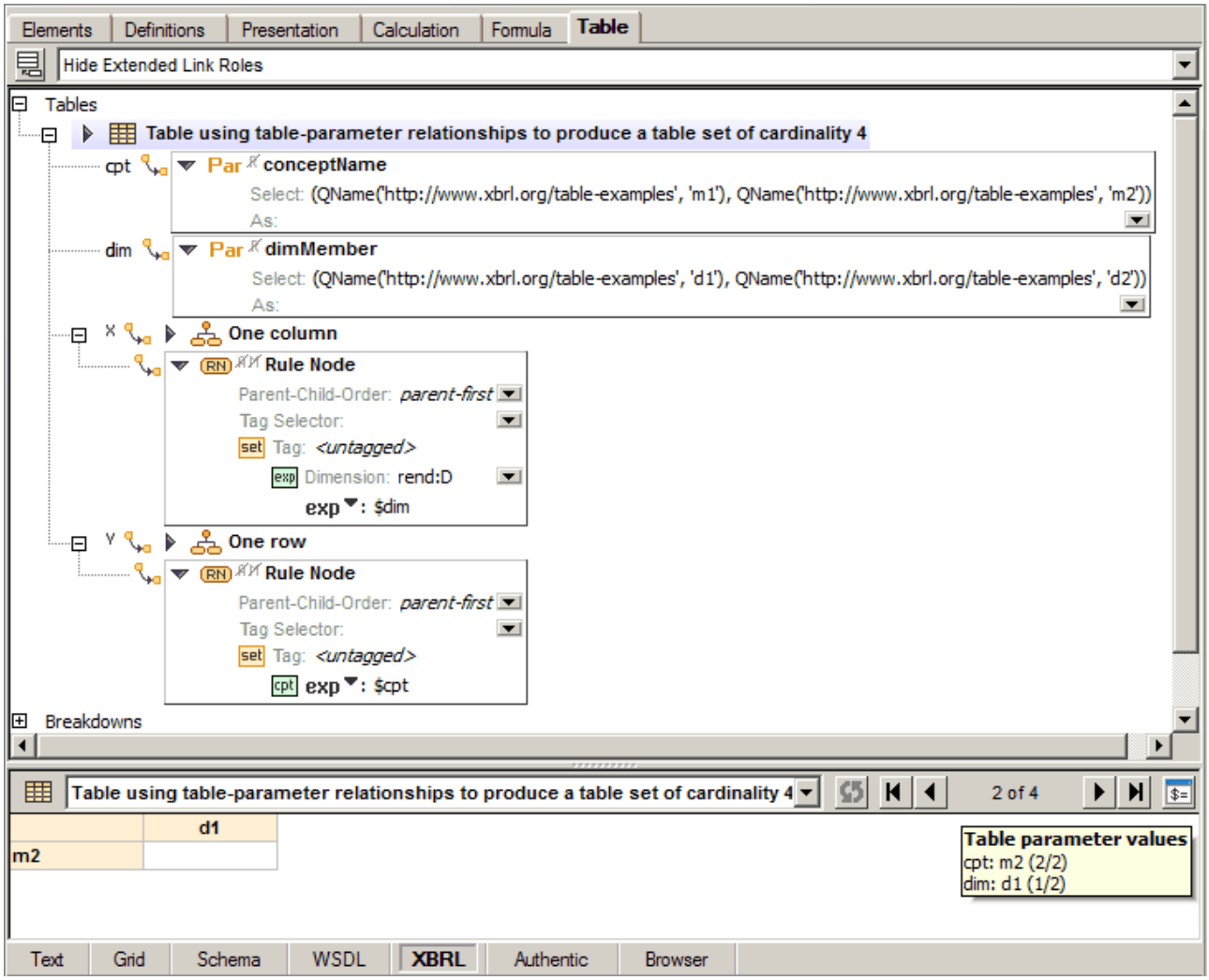


Les paramètres de table permettent de produire des tables liées plusieurs fois depuis une seule définition de table, formant un ensemble de table.

- Si un seul paramètre évalue à une séquence de valeurs, alors l'ensemble de table contient une table pour chaque item dans la séquence de résultat.
- Si la définition de table a des paramètres multiples, l'ensemble de table correspond à un produit cartésien ordonné des séquences obtenu en évaluant les paramètres. Un produit cartésien ordonné est affiché dans les exemples suivants :

```
A × B = {1,2} × {3,4} = {(1,3), (1,4), (2,3), (2,4)}
B × A = {3,4} × {1,2} = {(3,1), (3,2), (4,1), (4,2)}
```

Définition de table avec deux paramètres de table (`conceptName` et `dimMember`), dont chacun évalue à une séquence de deux QNames (*voir les expressions XPath de la propriété Select*).



Veuillez noter les points suivants :

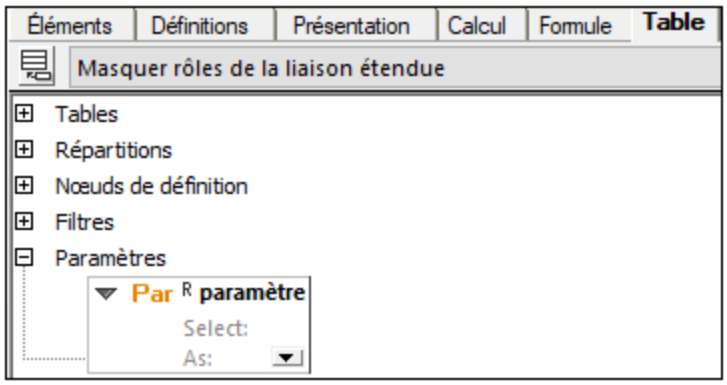
- Les paramètres sont des paramètres locaux, créés pour cette table spécifique en cliquant avec la touche de droite sur le composant de table et en sélectionnant µ. Ce ne sont pas des paramètres globaux comme c'était le cas dans la première capture d'écran ci-dessus.
- Le produit cartésien ordonné des deux séquences de deux QNames produit quatre tables : `dimMember × conceptName = {d1,d2} × {m1,m2} = {(d1,m1), (d1,m2), (d2,m1), (d2,m2)}`
- Lorsqu'une définition de table décrivant un ensemble de table est sélectionnée dans le diagramme, les icônes de navigation dans l'Aperçu de mise en page de la table sont activées et vous pouvez parcourir un aperçu des tables dans l'ensemble de table. La table actuellement consultée en aperçu est indiquée par son index dans l'ensemble de table ordonné dans la barre d'outils. Dans la capture d'écran ci-dessus, la table actuelle est `2 of 4`. Les valeurs de paramètres de cette table sont aussi affichées dans une fenêtre popup (*voir capture d'écran*).

© 2016-2022 Altova GmbH

- La barre outils Réinitialisée de l'Aperçu de la mise en page de table est activée lorsque l'aperçu n'est pas synchronisé avec les définitions actuelles, par exemple, après qu'un nouveau concept a été ajouté.
- Le bouton de la barre outils des Valeurs de paramètre de l'Aperçu de la mise en page de table ouvre le dialogue Valeurs de paramètre XBRL, dans lequel les valeurs et les types de données de tous les paramètres de table (global et local) peuvent être édités.

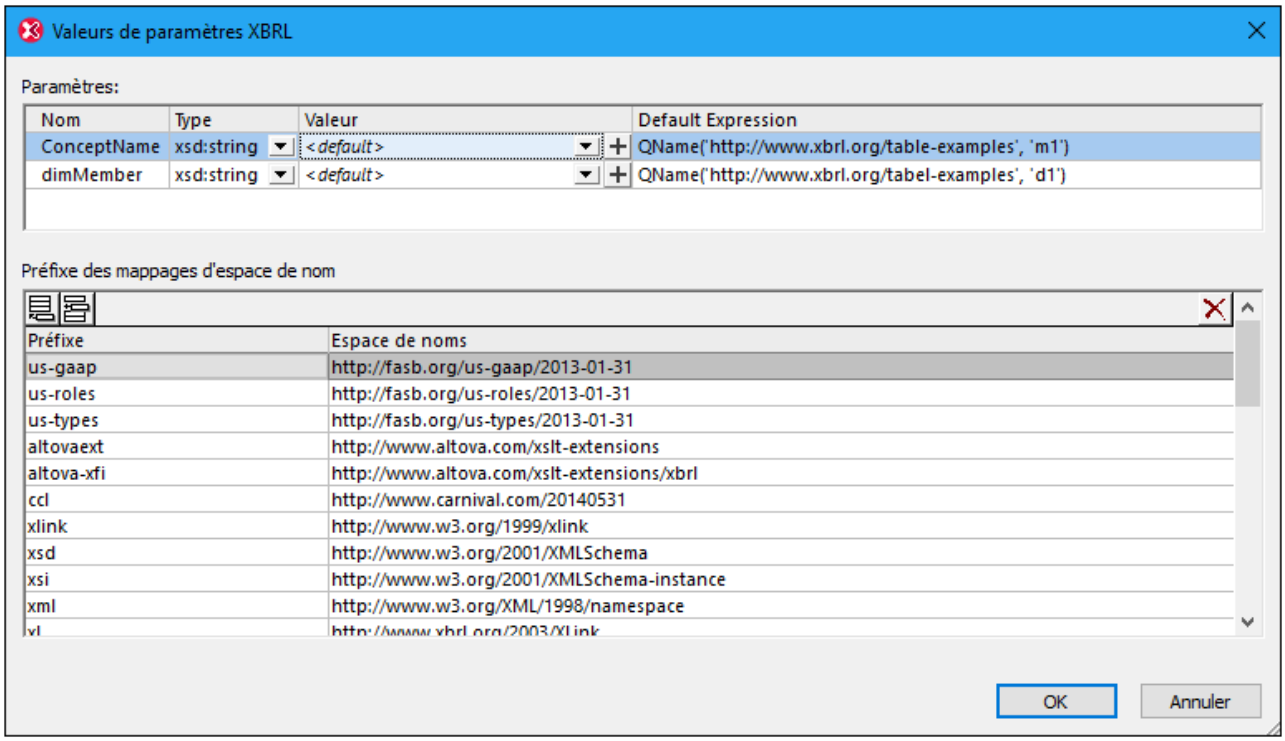
Définir les paramètres XBRL

Les paramètres XBRL peuvent être utilisés dans des expressions XPath dans des formules et dans des définitions de table. Les paramètres qui seront utilisés en tant que paramètres de formule (résidant dans la base de lien de la formule) sont créés dans l'onglet Formule, alors que les paramètres de table (résidant dans la base de lien de table) sont créés dans l'onglet Table. Les deux paramètres de formule et les paramètres de table peuvent être locaux ou globaux. Les paramètres locaux sont principalement des paramètres globaux qui sont liés au composant respectif (formule ou table) au moment de sa création. Les paramètres locaux sont créés en cliquant avec la touche de droite sur le composant (formule ou table) et en sélectionnant **Ajouter nouveau paramètre**, alors que des paramètres globaux sont créés en cliquant avec la touche de droite dans un espace vide de l'onglet respectif et en sélectionnant **Ajouter nouveau paramètre**. Vous ajoutez ainsi in nouveau paramètre nommé `parameter` dans le diagramme (*la capture d'écran ci-dessous montre un paramètre global*). Pour changer le nom du paramètre, double-cliquer le nom puis l'éditer.



Chaque paramètre a un flag *Requis*. Une fois défini, le paramètre est obligatoire, c'est à dire que sa valeur doit être fournie par l'application de traitement. Si le paramètre n'est pas obligatoire et qu'aucune valeur n'est fournie par l'application de traitement, la valeur fournie peut être calculée à l'aide d l'expression XPath donnée dans la propriété *Sélectionner*. Double-cliquer dans le champ *Sélect* pour saisir une expression XPath. Cette valeur sera la valeur par défaut du paramètre. La propriété optionnelle *Tel que* spécifie le type de données requis par le paramètre. Choisir un type de données depuis la liste déroulante de la liste de choix.

Dans le cas des paramètres qui seront utilisés en tant que paramètres de table, vous pouvez éditer le type de données du paramètre et fournir une valeur de paramètre qui surmonte la valeur par défaut. Pour ce faire, cliquer sur **XBRL | Valeurs de paramètres**. Ensuite, dans le dialogue qui apparaît (*capture d'écran ci-dessous*), saisir une valeur de paramètre. Cette valeur surmontera la valeur par défaut. Puisque les paramètres qui sont utilisés en tant que paramètres de table peuvent prendre des valeurs multiples, vous pouvez ajouter des valeurs de paramètre supplémentaires pour un paramètre en cliquant sur l'icône **+** dans la colonne *Valeur*.

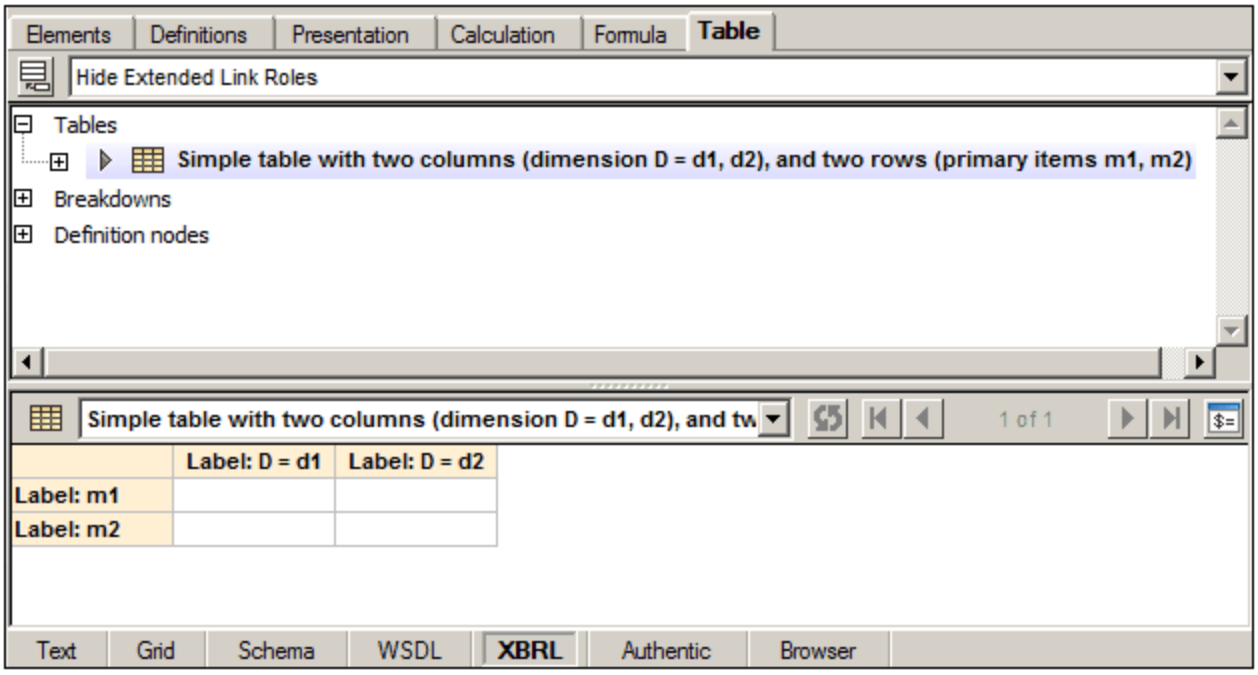


Les valeurs des paramètres globaux telles qu'attribuées dans ce dialogue sont uniquement évaluées pour les paramètres de table. Les valeurs des paramètres utilisés dans les formules ne sont pas éditables dans ce dialogue.

17.5.7 Aperçu de la mise en page de table

Le volet Aperçu de la mise en page de table XBRL est situé dans l'onglet Table en-dessous de l'arborescence de définitions de la table (*voir capture d'écran ci-dessous*). Une liste de choix dans le panneau de l'Aperçu de la mise en page recense toutes les tables dans la base de lien de la table de la taxonomie active. Pour visualiser la mise en page d'une table, choisir cette table dans la liste de choix du volet de l'aperçu (*voir capture d'écran ci-dessous*). Veuillez noter que l'aperçu ne montre que la mise en page. Les cellules de table ne sont pas remplies. Cela est dû au fait qu'il n'y a pas de données dans la taxonomie XBRL.

© 2016-2022 Altova GmbH



L'Aperçu de la mise en page vous permet de faire les opérations suivantes :

- Visualiser des mises en page de table, avec des aperçus mettant automatiquement à jour lorsque les définitions de table sont modifiées
- Aller directement à la définition d'un composant en cliquant sur une cellule de table et vice-versa (se rendre à la cellule de table en cliquant sur un composant dans l'arborescence de définition de la table)
- Accéder au dialogue Paramètres de table XBRL (par le biais de l'icône de barre d'outils Valeurs de paramètre) pour gérer les paramètres de table

Édition

Les modifications aux définitions de table et à la taxonomie sont gérées comme suit :

- *Modifications de table :* si la structure d'une définition de table est modifiée (dans le diagramme dans l'onglet Table ou par le biais de l'assistant à la saisie Détails), l'aperçu de la mise en page de table est mis à jour immédiatement. Les changements des définitions de paramètre ou des valeurs de paramètre déclencheront aussi cette mise à jour.
- *Modifications DTS :* l'Aperçu de la mise en page de table utilise le moteur XPath d'Altova pour évaluer les expressions XPath dans les nœuds de définition. Le modèle XPath est créé lors du chargement d'un schéma de taxonomie dans le mode XBRL et il est mis à jour pendant la validation. Si le DTS sous-jacent est modifié (par exemple, en éditant un concept ou une base de lien), l'aperçu de la table ne sera plus synchronisé avec le DTS modifié. L'icône *Table non synchronisée* dans la barre d'outils de l'aperçu indique cet état et ses astuces vous fourniront une indication : *L'aperçu doit être réinitialisé manuellement par le biais de la touche Réinitialiser*. La commande **Réinitialiser** invoque une redécouverte du DTS et est donc l'équivalent d'une validation complète de la taxonomie.

	*Table non synchronisée avec DTS*
	*Réinitialiser table*

Gestion d'erreur

Les erreurs liées à l'Aperçu de la mise en page de table sont gérées comme suit :

- *Expressions invalides dans les nœuds de définition de table :* si un nœud de définition de table contient une expression XPath qui ne peut pas être résolue, l'en-tête de la cellule de mise en page correspondante sera affichée en rouge. Dans ce cas, la contrainte d'aspect invalide est marquée dans l'onglet Contrainte de l'assistants à la saisie Détails.
- *Nœuds de relation non résolubles :* si un nœud de relation ne peut pas être résolu en raison de propriétés invalides ou un DTS invalide, la mise en page montre une cellule d'espace réservé avec le texte d'erreur marqué.
- *Nœuds de règle merged sans nœuds enfants :* si un nœud de règle merged n'a pas de nœud enfant, la mise en page montre une cellule d'espace réservé avec un texte d'erreur marqué.
- *DTS invalide :* si la taxonomie est invalide lors du chargement d'un schéma de taxonomie dans l'éditeur de taxonomie XBRL ou après validation, le modèle XPath n'est pas disponible. L'Aperçu de la mise en page de table sera dans un état d'erreur, qui est indiqué par l'icône de barre d'outils *Table non synchronisée avec DTS*. Malgré cela, la mise en page peut toujours être créée dans une certaine mesure. Les expressions XPath, néanmoins, ne peuvent pas être évaluées. Les indications de l'icône de la barre d'outils aideront l'utilisateur à résoudre ce problème (c'est à dire en réparant l'erreur de validation et en revalidant la taxonomie).

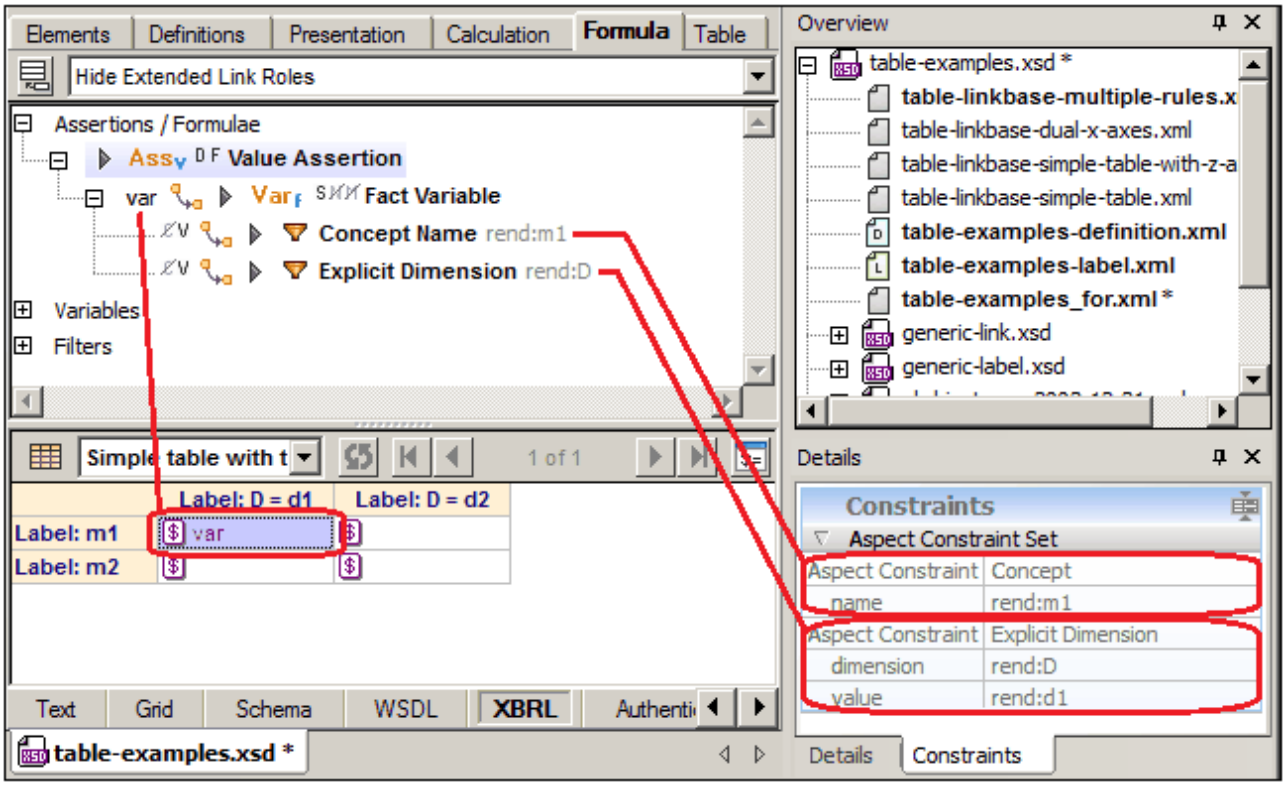
17.5.7.1 Élaborer des formules dans l'Aperçu de mise en page de table

L'Aperçu de mise en page de table est aussi affiché dans l'onglet Formule (*voir capture d'écran ci-dessous*) pour pouvoir prendre en charge la création des variables de fait sous des ensembles de variable (c'est à dire, sous des formules ou des assertions de valeur/existence). Dans ce cas, les cellules dans les axes de table ne peuvent pas être sélectionnés parce que les nœuds de définition de table correspondants ne sont pas visibles dans la base de lien de formule. Les cellules de données, en revanche, montrent une icône **Ajouter une variable de fait**, qui est activée dès qu'un ensemble de variable est sélectionné dans l'arborescence de formule (*voir capture d'écran ci-dessous*).

Ajouter une variable de fait à la formule ou l'assertion sélectionnée

Pour ajouter une variable de fait à un ensemble de variable, sélectionner cet ensemble de variable dans l'onglet Formule. (Un ensemble de variable est une formule ou une assertion de valeur/existence. Dans la capture d'écran ci-dessous, l'ensemble de variable sélectionné est une assertion de valeur.) Dans les cellules de l'Aperçu de mise en page de table, l'icône **Ajouter une variable de fait** sera activée. Cliquer sur l'icône pour ajouter la variable à l'ensemble de variable. Au cours de l'exécution, une nouvelle variable de fait contenant un filtre approprié pour chaque contrainte d'aspect définie par la cellule de données est créée sous l'ensemble de variable sélectionnée (c'est à dire, formule ou assertion).

© 2016-2022 Altova GmbH



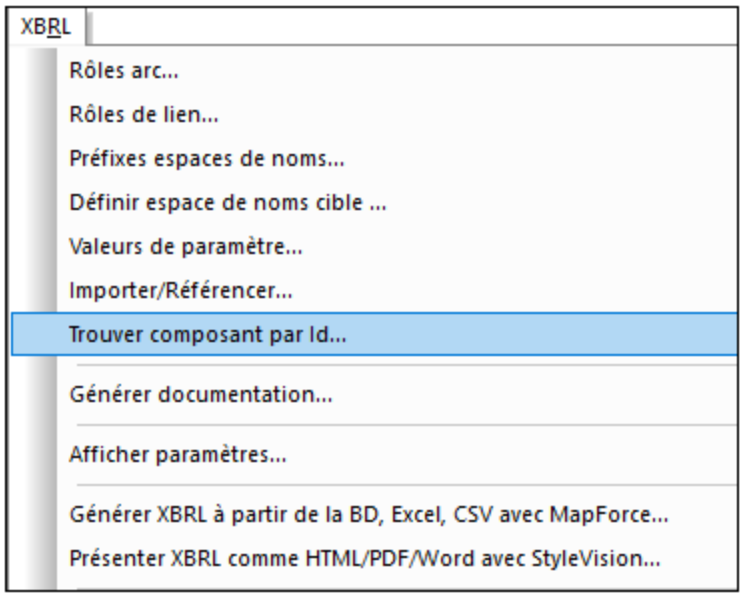
Si le formulaire ou l'assertion sélectionné présente une variable contenant des filtres qui correspondent aux contraintes d'aspect d'une cellule de données dans l'aperçu, le nom de variable est affiché par la cellule de données. Cela devrait être le cas, en particulier, après avoir créé une nouvelle variable de fait par le biais de l'icône **Ajouter une variable de fait** (*voir capture d'écran ci-dessus*).

17.5.8 Trouver des composants de table

Les composants de table peuvent être trouvés (i) en utilisant leurs ID et (ii) en navigant à travers les occurrences du composant dans le document.

Trouver le composant de table avec l'id

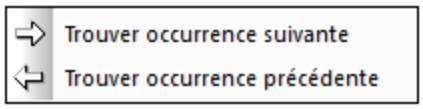
Dans les taxonomies à formule ou base de lien de table volumineux contenant plusieurs composants du même type (par ex. assertions, filtres, tables), il peut s'avérer utile de chercher un composant par le biais de son ID. La commande de menu **XBRL | Trouver composant par Id** permet une recherche par l'ID.



Cliquer sur la commande pour faire apparaître un dialogue demandant l'ID que vous cherchez.

Trouver les occurrences de composant

La plupart des composants de table sont affichés plusieurs fois dans le diagramme de base de lien de table : (i) la définition, qui est située directement sous le nœud de section approprié et (ii) toutes les références au composant (via les relations). Les commandes **Trouver occurrence suivante** et **Trouver occurrence précédente** dans le menu contextuel du composant (*capture d'écran ci-dessous*) pour vous rendre à tous les endroits où ce composant de table est référencé.



Ces commandes peuvent aussi être accédées par le biais de leurs icônes de barre d'outil (*capture d'écran ci-dessous*).



Une fois que la définition du composant est atteinte, un message à cet effet sera affiché.

© 2016-2022 Altova GmbH

17.6 XULE

XULE (provenant de XBRL Rule) est un langage de requête de rapports et taxonomies XBRL. L'objectif principal du langage est de fournir la capacité de requêter et de contrôler des rapports avant de les déposer, afin d'assurer la qualité des données. XULE vous permet de vérifier les rapports de deux manières :

- *Fournir une sortie à partir des données contenues dans les rapports :* en requêtant des données dans le rapport et en calculant les résultats à partir des données dans le rapport. La qualité de la sortie produite peut être ensuite évaluée.
- *Créer des assertions :* les données contenues dans le rapport peuvent être testées par rapport à ces assertions, et une action appropriée peut être prise ensuite sur la base des résultats

Liens Internet

- Page d'accueil XULE (contient un bref aperçu de XULE)
- Documentation de la syntaxe du langage XULE

Fonctions XMLSpy

XMLSpy propose les fonctions XULE suivantes :

- Un **processeur XULE intégré** qui traite les documents et les expressions XULE par rapport à un document d'instance XBRL
- Création de documents conformes à XULE
- Validation de documents XULE pour une syntaxe correcte par rapport à la spécification XULE
- Coloration de la syntaxe dans des documents XULE
- Remplissage automatique des structures de langage XULE lors de l'édition de documents XULE
- Une **fenêtre XULE** spéciale pour requêter des documents d'instance XBRL de manière interactive
- Le traitement d'une instance XBRL par rapport à un seul document XULE ou un ensemble de documents XULE stockés dans un archive zip ; le traitement peut être exécuté par le moteur XMLSpy ou le moteur RaptorXML(+XBRL) Server
- Intégration dans des projets XMLSpy de l'exécution des documents XULE et des ensembles de document

Les sous-sections de cette section décrivent ces fonctions en détail.

RaptorXML+XBRL Server d'Altova

Le RaptorXML+XBRL Server d'Altova permet un traitement XULE personnalisable et rapide qui vous permet de traiter des documents XULE depuis la ligne de commande, avec des scripts et par le biais d'un nombre d'API de serveur de moteur, y compris un API Python puissant. Pour plus d'informations, voir :

- La page RaptorXML+XBRL dans le site web Altova
- Certification DQC de RaptorXML+XBRL d'Altova
- Documentation de produit RaptorXML+XBRL
- documentation d'API Python de RaptorXML

17.6.1 Documents XULE

XMLSpy fourni une palette de fonctions qui prennent en charge la création, la validation et l'exécution de documents XULE. Cette section décrit ces fonctions.

Fichiers conformes à XULE (fichiers .xule)

Le type de fichier **.xule** est prédéfini dans XMLSpy comme étant conforme à XULE. Cela signifie que lorsqu'un fichier **.xule** est ouvert dans XMLSpy, l'aide à l'édition XULE sera disponible sous la forme de coloration de la syntaxe et remplissage automatique. Dans la section Types de fichier du dialogue Options, vous pouvez spécifier que d'autres extensions de fichier soient également conformes à XULE.

Ensembles de document XULE

Plusieurs documents XULE peuvent être comprimés dans une archive zip (généralement **.zip**). Cette archive zip est un ensemble de document XULE (ou ruleset XULE). Vous pouvez ensuite exécuter la totalité de l'ensemble du document XULE dans une instance XBRL, en spécifiant le fichier zip comme étant le fichier XULE à exécuter,

Une archive zip peut présenter n'importe quelle structure. Les fichiers XULE, dans tous les niveaux de l'archive, seront utilisés pendant l'exécution XULE ; les fichiers non-XULE seront filtrés.

Coloration syntaxique

Les Documents XULE peuvent être édités dans le Mode Texte de XMLSpy. La capture d'écran ci-dessous montre la coloration syntaxique par défaut d'un document XULE échantillon. Vous pouvez personnaliser la coloration syntaxique dans la section Polices et couleurs | Mode texte du dialogue Options.

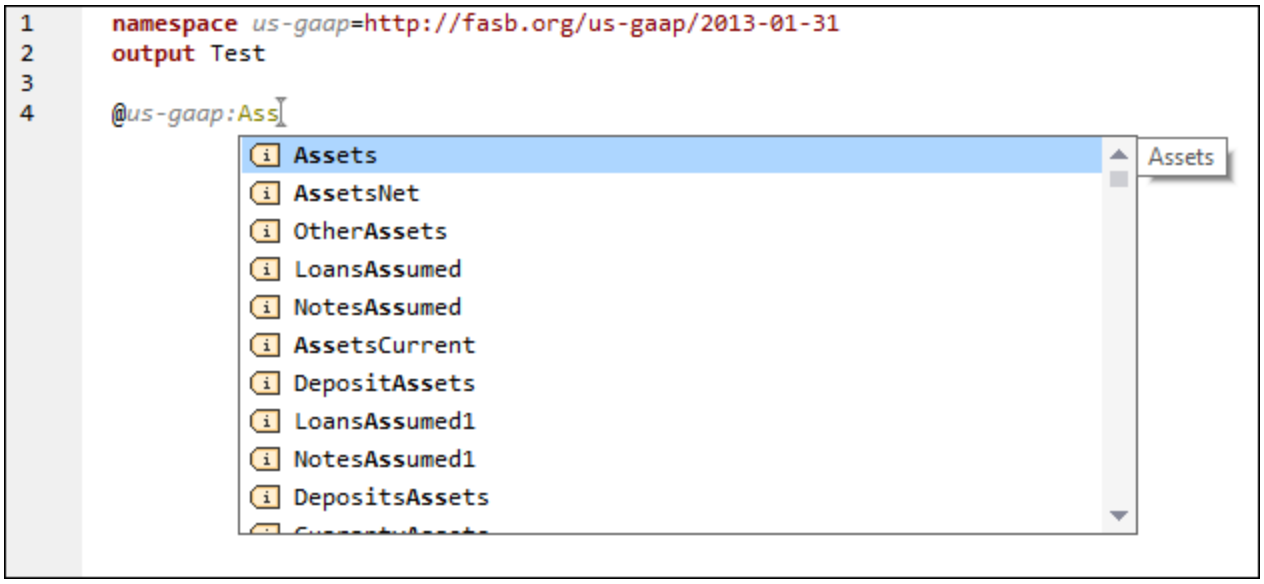
```
1   namespace us-gaap=http://fasb.org/us-gaap/2013-01-31
2
3   // Balance sheet test
4   assert balance-test unsatisfied
5
6   @us-gaap:Assets#assets == @us-gaap:LiabilitiesAndStockholdersEquity#liabilities
7
8   message
9
10  "Balance sheet is unbalanced in period {$assets.period}, {$assets} != {$liabilities}"
```

Remplissage automatique

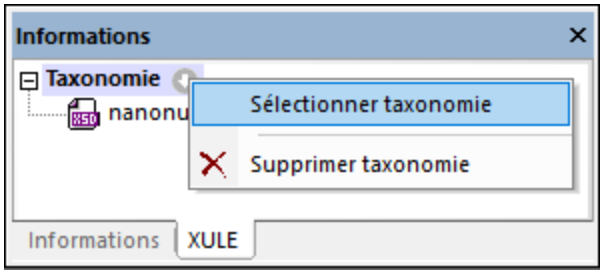
Pendant votre saisie des règles dans le document XULE, vous pouvez procéder à deux types d'aide au remplissage automatique :

- lié à la syntaxe de langage XULE
- lié à la structure d'une taxonomie XBRL sélectionnée

© 2016-2022 Altova GmbH



Vous pouvez sélectionner la taxonomie que vous souhaitez utiliser dans la fenêtre Info *(voir capture d'écran ci-dessous)*. Dans l'onglet XULE, cliquer sur l'icône située à la droite de l'item *Taxonomie*, et, dans le menu qui apparaît, cliquer sur **Sélectionner taxonomie**. Ensuite chercher la taxonomie et la sélectionner. Une seule taxonomie peut être ajoutée à la fois. Si vous ajoutez une nouvelle taxonomie, elle remplacera la taxonomie précédente. En alternative, vous pouvez supprimer une taxonomie *(voir capture d'écran ci-dessous)* avant d'en ajouter une nouvelle.



Note : Vous devez ajouter au document XULE des déclarations d'espace de noms de tous les composants de taxonomie dont vous avez besoin *(voir la capture d'écran ci-dessus document XULE)*. Les préfixes d'espace de noms ne doivent pas nécessairement correspondre à celles de la taxonomie, mais il vaut mieux garder les mêmes préfixes d'espace de noms pour éviter toute confusion. Si vous n'ajoutez pas de déclarations d'espace de noms, le remplissage automatique des composants de taxonomie ne fonctionnera pas.

Valider XULE

Un document XULE peut être validé pour une syntaxe correcte par rapport à la spécification de langage XULE en utilisant la commande **XML | Valider (F8)**.

Intégration dans des projets XMLSpy

Vous pouvez intégrer les documents XULE dans un projet XMLSpy de la manière suivante :

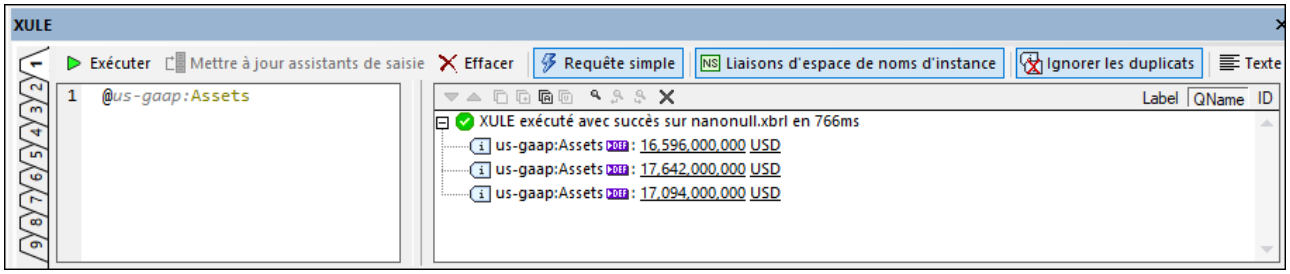
1. Ajouter le document XULE dans un dossier approprié dans un projet XMLSpy.

2. Dans les propriétés de projet de ce dossier, saisie le fichier d'instance XBRL sur lequel vous souhaitez exécuter le fichier XULE.
3. Cliquer avec la touche sur le document XULE dans le projet, et sélectionner la commande **XBRL | Exécuter XULE**. Le document XULE sera exécuter dans l'instance XBRL qui a été spécifié pour le dossier, et les résultats seront affichés dans la fenêtre Messages ou dans un nouveau document *(voir options d'exécution XULE)*.

17.6.2 Fenêtre XULE

La fenêtre XULE *(capture d'écran ci-dessous)* est une fenêtre de sortie. Elle vous permet de rechercher interactivement le document d'instance XBRL actif et de consulter les résultats de votre requête.

La fenêtre XULE est située par défaut en dessous de la fenêtre principale en bas de la GUI de XMLSpy. Elle a neuf onglets, dont chacun est divisé en deux volets : (i) un volet d'expression XULE où vous saisissez une expression XULE (ou une règle XULE) que vous voulez exécuter dans le document actif ; et (ii) un volet de Résultat, qui affiche le résultat de l'exécution.



Pour exécuter de manière interactive une expression XULE dans le document d'instance XBRL actif, suivez les étapes suivantes :

1. Faire du document d'instance XBRL que vous souhaitez requêter le document actif dans la fenêtre principale.
2. Saisir l'expression XULE dans le volet d'expression XULE *(volet de gauche)*. Les fonctions d'édition du volet d'édition incluent les couleurs de la syntaxe et la saisie automatique.
3. Cliquez sur **Exécuter** dans la barre d'outils de la fenêtre pour exécuter l'expression
4. Les résultats de l'exécution sont affichés dans le volet Résultats *(volet de droite)*. Vous pouvez cliquer sur un lien dans les résultats pour aller au nœud respectif dans le document d'instance XBRL.

Note : La coloration de syntaxe pour XULE peut être personnalisée dans le dialogue des Options (dans la section Polices et couleurs | Mode Texte). Pour des informations sur la saisie automatique, voir la description de la commande de la barre d'outils **Update Entry Helpers** ci-dessous.

Barre d'outils : commandes et options

La barre d'outils de la fenêtre XULE fournit des commandes et options utiles pour créer et exécuter des expressions XULE. Ils sont décrits ci-dessous.

Exécuter
Cliquez sur **Exécuter** pour exécuter une expression XULE.

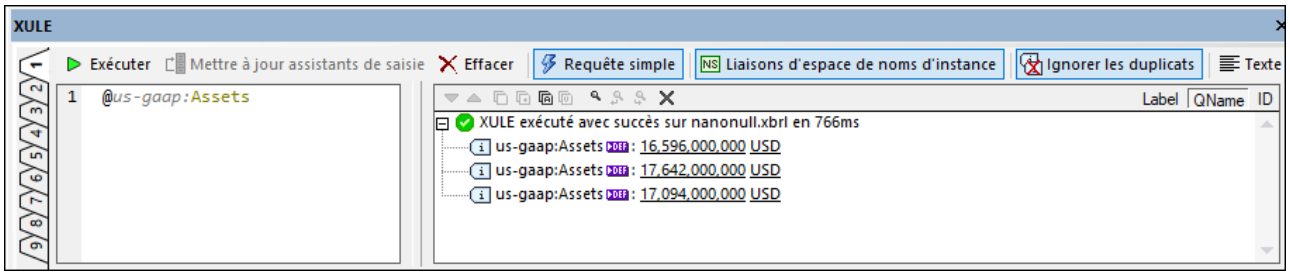
© 2016-2022 Altova GmbH

Mettre à jour les assistants à la saisie
Le volet d'expression XULE fournit deux types de saisie automatique lorsque vous saisissez : (i) celui lié à la syntaxe XULE, et (ii) celui lié à la structure du document d'instance XBRL actif. Toutefois, afin d'être conscient de la structure de l'instance XBRL, la fenêtre XULE doit lire la taxonomie XBRL qui est référencée par l'instance XBRL. Cliquez sur **Update Entry Helpers** afin de charger silencieusement la taxonomie qui est associée à l'instance XBRL. Une fois que la taxonomie a été chargée, la saisie automatique liée à la structure du document sera disponible et ce bouton de la barre d'outils (qui n'est plus nécessaire pour ce document d'instance XBRL) sera désactivé. Notez que la taxonomie est également lue à chaque moment quand vous cliquez sur **Exécuter**.

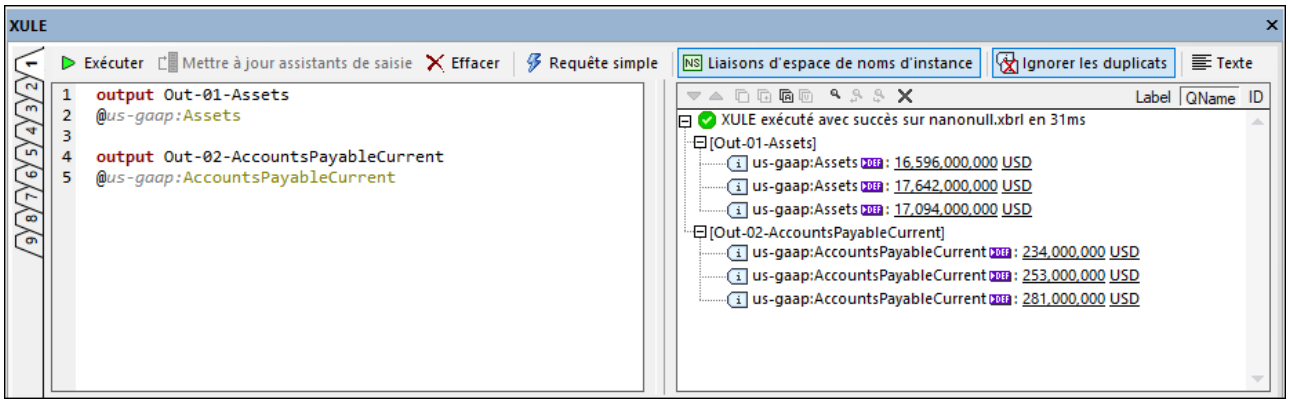
Supprimer
Cliquez sur **Supprimer** pour supprimer l'expression.

Requête simple
Le mode de Requête unique est la fonction qui est spécifique à la fenêtre XULE de XMLSpy. Elle vous permet : (i) de saisir une expression sans le mot-clé **`sortie`** et une requête simple et (ii) de générer le résultat comme sortie simple. Une syntaxe XULE valide requiert le mot-clé **`sortie`** mais si vous voulez rechercher le document XBRL rapidement, il est avantageux de pouvoir saisir une requête simple sans mot-clé **`sortie`**.

Les deux captures d'écran ci-dessous montrent comment utiliser les requêtes simples et les requêtes multiples.



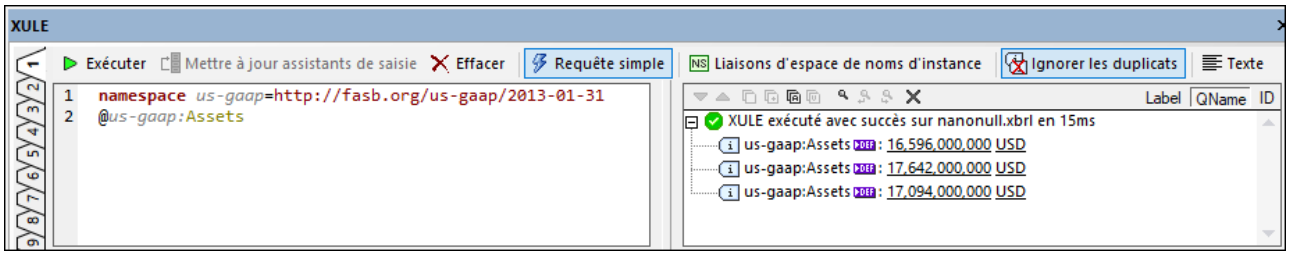
Mode Requête simple activé : pas de mot-clé 'sortie' requis.



Mode Requête simple désactivé : de multiples requêtes avec le mot-clé 'sortie', de multiples sorties.

Liaisons d'espace de noms d'instance
Si l'option **Liaisons d'espace de noms d'instance** est sélectionnée, alors vous ne devez pas déclarer les espaces de noms dans la requête XULE ; les préfixes d'espace de noms seront liés aux URI d'espaces de noms déclarés pour eux dans l'instance XBRL. Par exemple, dans la capture d'écran ci-dessus, le préfixe d'espace de nom **`us-gaap`** est lié à l'espace de nom défini pour le document d'instance. D'un autre côté, si

l'option **Liaisons d'espace de noms d'instance** est décochée, alors vous devez déclarer la requête XULE (avec le mot-clé `espaces de noms`, telle qu'affichée dans la capture d'écran ci-dessous).



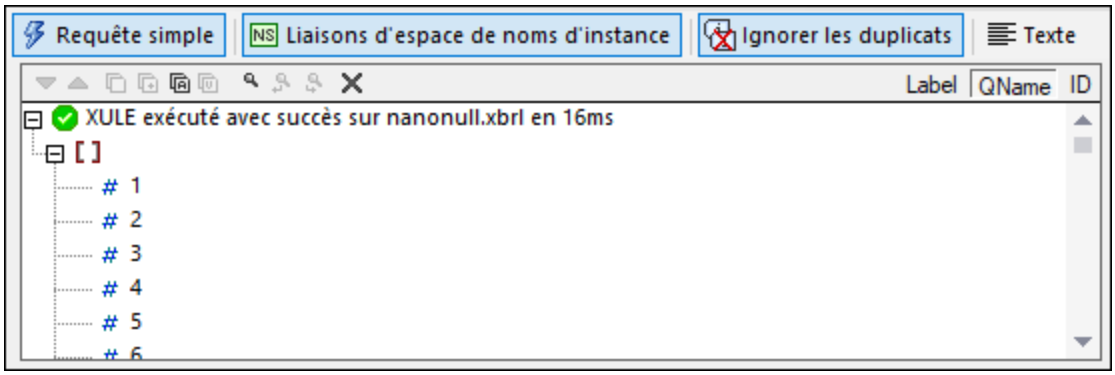
Cette option est utile car vous ne devez pas remplir la requête XULE avec des déclarations d'espace de noms.

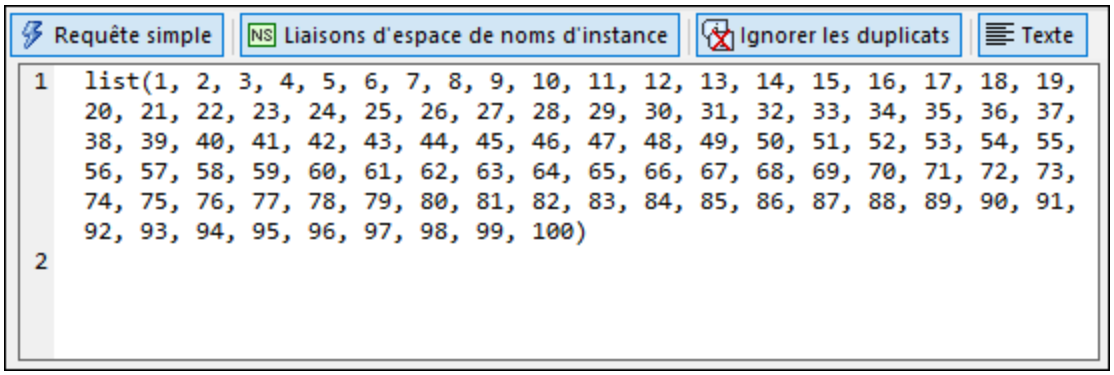
Ignorer les duplicats

Un fait dupliqué apparaît—généralement dans l'XBRL Inline—quand le même fait est noté plus d'une fois dans le code HTML. L'option **Ignorer les duplicats** spécifie que le fait dupliqué représente la sortie une seule fois.

Texte

L'option **Texte** bascule la sortie dans le volet Résultats entre les sorties texte et arborescence. Par exemple, les captures d'écran ci-dessous affichent les sorties sous forme d'arborescence *(gauche)* et de texte *(droite)*. La requête est : `list(for $i in range(100) $i)`.

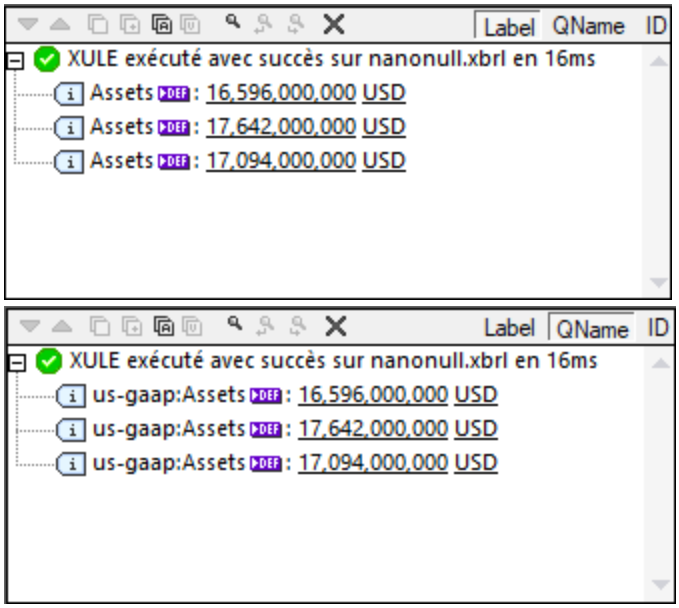




© 2016-2022 Altova GmbH

Le volet Résultat

Les résultats peuvent être affichés avec leurs labels, QNames ou ID. Sélectionnez l'option désirée dans la barre d'outils du volet Résultats, soit avant ou après avoir exécuté la requête. La capture d'écran ci-dessous affiche les résultats avec les labels *(gauche)* et QNames *(droite)*.



La barre d'outils du volet Résultats contient des icônes qui propose des fonctions de navigation, de recherche et de copie. Ces icônes sont décrites dans la table ci-dessous de gauche à droite. Les commandes correspondantes sont aussi disponibles dans le menu contextuel des items de la liste de résultat.

Icône	Fonction
Suivant, Précédent	Sélectionne, respectivement, l'item suivant et précédent dans la liste de résultats
Copier le message sélectionné	Copie l'item de résultat sélectionné dans le presse-papiers.
Copier le message sélectionné y compris les enfants	Copie l'item de résultat sélectionné dans le presse-papiers, de même que les items enfants. Chaque item est copié en tant que ligne séparée
Copier tous les messages	Copie tous les résultats d'item dans le presse-papiers.
Copier la valeur de la ligne sélectionnée dans le presse-papiers	Copie uniquement la valeur de l'item de résultat sélectionné dans le presse-papiers.
Recherche	Ouvre un dialogue *Recherche* pour rechercher tout string dans les résultats
Chercher précédent	Cherche l'occurrence précédente du terme qui a été saisi en dernier dans le dialogue *Chercher*

Chercher suivant	Cherche l'occurrence suivante du terme qui a été saisi en dernier dans le dialogue *Chercher*
Supprimer	Supprime la liste de résultats

17.6.3 Exécution XULE

Pour exécuter un document XULE dans un document d'instance XBRL, choisir la commande de menu **XBRL | Exécuter XULE**. La commande peut être utilisée dans les cas suivants :

- Lorsqu'un document XULE est le document actif, la sélection de la commande vous invite à choisir l'instance XBRL sur laquelle le document XULE doit être exécuté.
- Lorsqu'un document XBRL est le document actif, la sélection de la commande vous invite à choisir le document XULE ou le document XULE défini pour l'utilisation.
- Si (i) le document XULE (`.xule` file)—ou l'ensemble de document XULE (zip archive)—et le document d'instance XBRL font tous les deux partie d'un projet XMLSpy, et que (ii) le fichier d'instance XBRL a été défini en tant que le fichier XBRL cible dans les propriétés du projet XMLSpy, cliquer avec la touche de droite sur le fichier XULE du projet dans la fenêtre de projet de XMLSpy et choisir **Exécuter XULE**. Les documents XULE seront exécutés sur le XBRL qui est la cible du projet pour l'exécution XULE.

De plus, vous pouvez configurer XMLSpy de manière à ce qu'il exécute des commandes RaptorXML(+XBRL) Server, parmi lesquelles se trouvent des commandes de traitement XULE, depuis l'interface XMLSpy.

Options d'exécution XULE

Les options d'exécution XULE suivantes sont disponibles :

- Le résultat peut être envoyé soit : (I) à la fenêtre Messages, soit (ii) dans un nouveau document qui est affiché dans une nouvelle fenêtre XMLSpy et stocké temporairement dans la mémoire ; ce document peut être sauvegardé dans le fichier avec la commande **Fichier | Enregistrer sous**.
- Les faits doubles renvoient à des références multiples du même fait. (Un fait double se produit généralement dans Inline XBRL lorsque le même fait est noté plus d'une fois en raison du code HTML.) Vous pouvez choisir de ne rapporter les faits doubles qu'une seule fois.

Vous pouvez sélectionner les paramètres que vous souhaitez dans l'onglet XBRL XULE du dialogue Options (**Outils | Options | XBRL | XULE**).

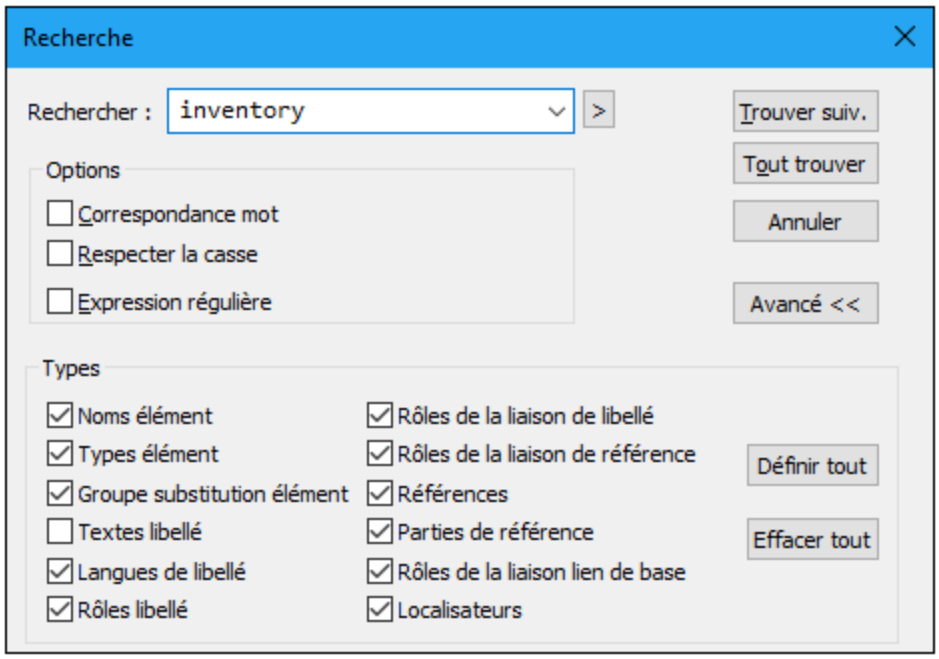
© 2016-2022 Altova GmbH

17.7 Chercher dans XBRL

Dans le Mode XBRL, les taxonomies XBRL peuvent être cherchées à l'aide de la fonction Chercher dans XBRL de XMLSpy, qui est activée lorsqu'une taxonomie XBRL est active dans le Mode XBRL. La fonction Chercher dans XBRL est accessible d'une des manières suivantes :

- Par le biais de la commande de menu **Édition | Recherche** lorsqu'une taxonomie XBRL est active dans le Mode XBRL.
- Par le biais de la touche **Chercher** dans la fenêtre Chercher dans XBRL.
- By pressing **Ctrl+F**.

Choisir une de ces méthodes d'accès permet d'afficher le dialogue de Recherche (capture d'écran ci-dessous).



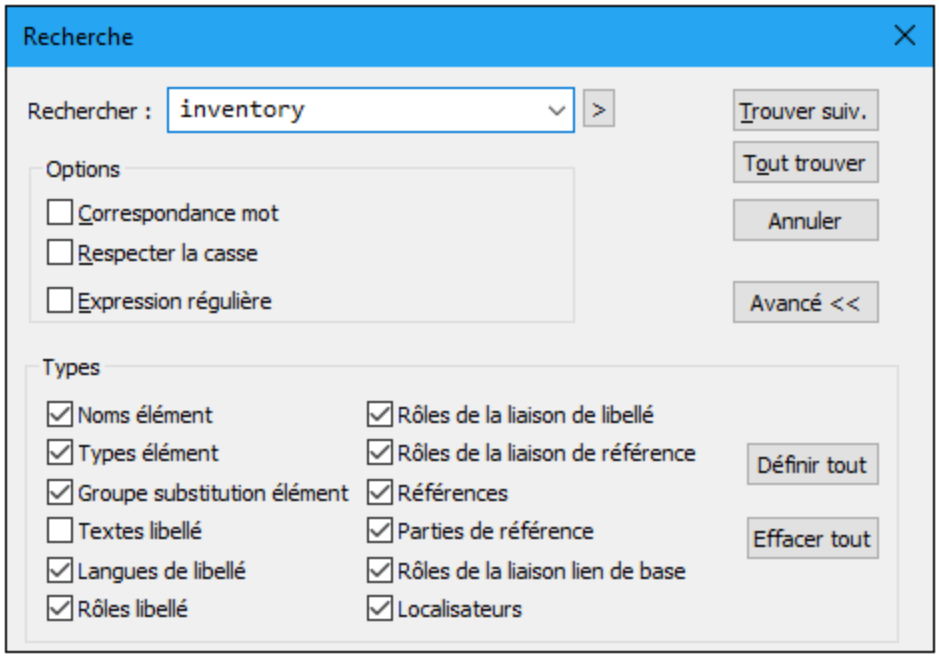
Procéder comme suit :

- Saisir le terme de la recherche dans le champ de saisie *Rechercher* du dialogue Chercher (*capture d'écran ci-dessus*) et cocher les options nécessaires
- Spécifier les types de composant XBRL à chercher dans le panneau Types
- Exécuter la commande à l'aide de la touche **Trouver suivant** ou **Trouver tout**
- Utiliser la fenêtre Chercher dans XBRL pour consulter les résultats de recherche et naviguer rapidement vers un composant.

17.7.1 Terme de Recherche

Un terme de recherche peut être spécifié pour être sensible à la casse ou pour correspondre à un mot entier en cochant les options correspondantes dans le volet Options (*voir capture d'écran ci-dessous*). Si vous souhaitez effectuer une recherche à l'aide d'une expression régulière, cocher l'option *Expression régulière* dans le volet

Options avant de cliquer sur la touche **Trouver suivant** ou **Trouver tout**. Voir ci-dessous pour plus de détails concernant les expressions régulières.



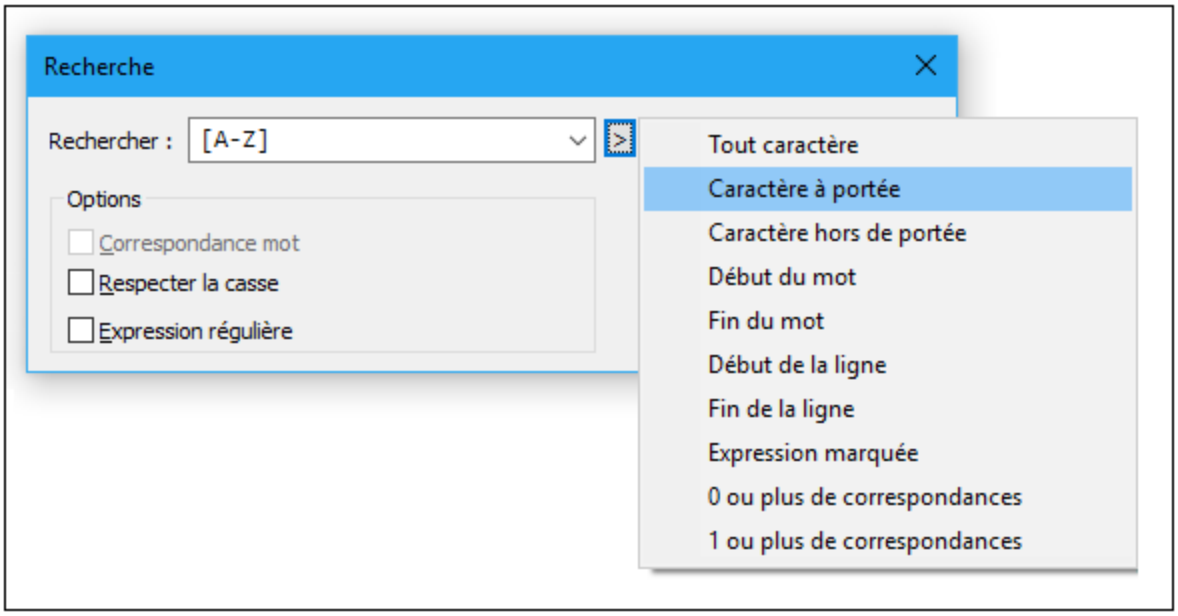
Note : Un mot entier est considéré être délimité par tout caractère non alphanumérique ou le caractère tiret bas. C'est ainsi que le terme de recherche `asset` retournera le texte `xbrl:asset`, puisque le double point (:) est considéré être un délimiteur de mot.

Dans le panneau types, spécifier les composants de recherche.

Expressions régulières

Vous pouvez utiliser les expressions régulières pour affiner vos critères de recherche. Une liste popup est disponible pour vous aider à construire des expressions régulières. Pour accéder à cette liste, cliquer sur la touche **>** située à droite du champ d'entrée pour le terme de recherche.

© 2016-2022 Altova GmbH

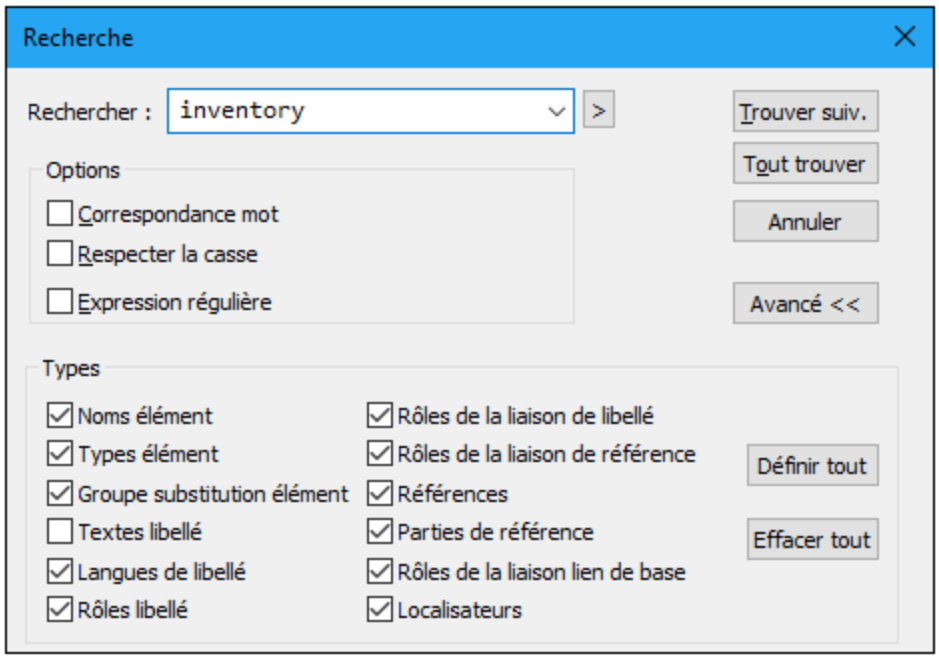


Cliquer sur la description d'expression requise pour insérer le caractère de syntaxe d'expression correspondant dans le champ d'entrée. Ci-dessous, vous trouverez une liste des caractères de syntaxe d'expression réguliers.

`.`	Correspond à tout caractère. Il s'agit d'un espace réservé pour un caractère unique.
`\(`	Marque le début d'une région pour marquer une correspondance.
`\)`	Marque la fin d'une région marquée.
`\<`	Correspond au début d'un mot.
`\>`	Correspond à la fin d'un mot.
`\x`	Vous permet d'utiliser un caractère `x`, qui aurait habituellement une signification particulière. Par exemple, `\[` serait interprété comme `[` et non pas en tant que le début d'un ensemble de caractères.
`[...]`	Indique un ensemble de caractères, par exemple, `[abc]` signifie un des caractères `a`, `b` ou `c`. Vous pouvez aussi utiliser des plages, par exemple `[a-z]` pour des caractère en minuscule.
`[^...]`	Le complément des caractères dans l'ensemble. Par exemple, `[^A-Za-z]` signifie tout caractère sauf un caractère alphabétique.
`^`	Correspond au début d'une ligne (sauf si utilisé dans un ensemble, voir ci-dessus).
`$`	Correspond à la fin d'une ligne. Exemple : `A+$` pour trouver un ou plusieurs `A` à la fin de la ligne.
`*`	Correspond à 0 ou plus de fois. Par exemple, `Sa*m` correspond à `Sm`, `Sam`, `Saam`, `Saaam` etc.
`+`	Correspond à 1 ou plus de fois. Par exemple, `Sa+m` correspond à `Sam`, `Saam`, `Saaam` etc.

17.7.2 Exécution de la commande

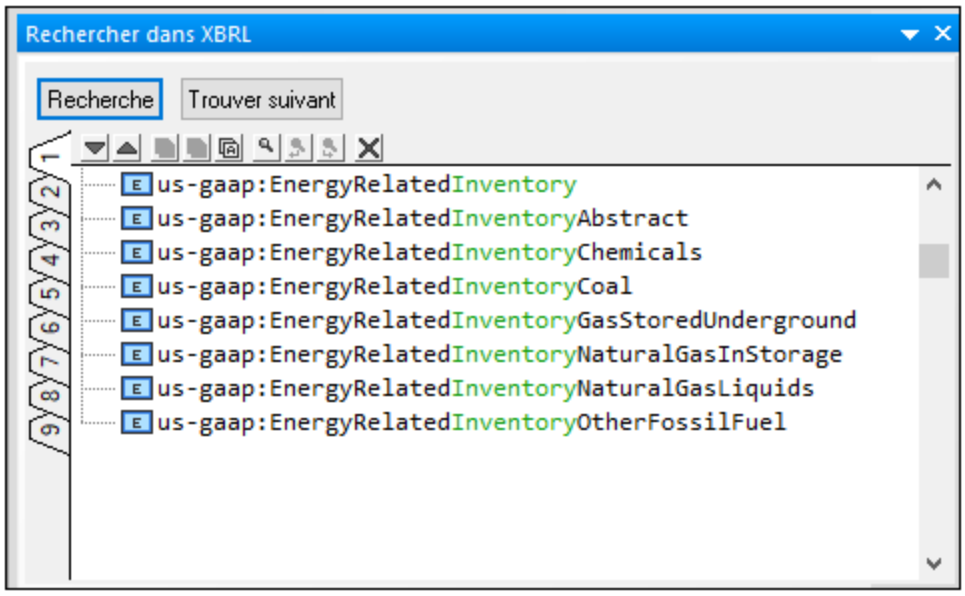
Une fois que le terme recherché a été saisi, et que les options de recherche et les filtres pour les types de composants ont été définis, vous disposez de deux options d'exécution de commande : **Trouver suivant** et **Tout trouver**. Ces commandes sont exécutées par le biais de touches dans le dialogue Chercher (*voir capture d'écran ci-dessous*).



Trouver suivant

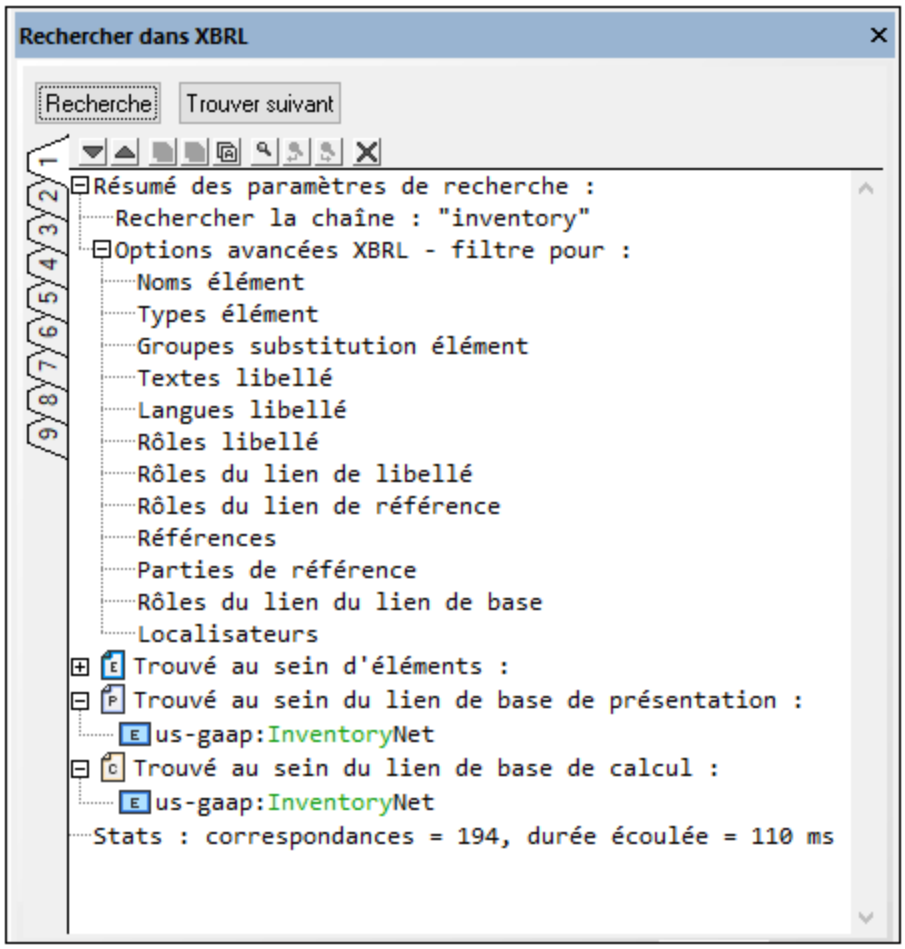
La commande **Trouver suivant** affiche, dans la fenêtre Chercher dans XBRL (*capture d'écran ci-dessous*), l'instance suivante du terme de recherche. La recherche de l'instance suivante commencera à la cellule suivante en partant de la position actuelle du curseur dans le document actif. Le processus **Trouver suivant** peut se poursuivre jusqu'à ce que toutes les instances dans le document soient affichées.

© 2016-2022 Altova GmbH



Tout trouver

La commande **Tout trouver** affiche toutes les instances du terme de recherche, avec un récapitulatif de recherche, dans la fenêtre Chercher dans XBRL (capture d'écran ci-dessous).



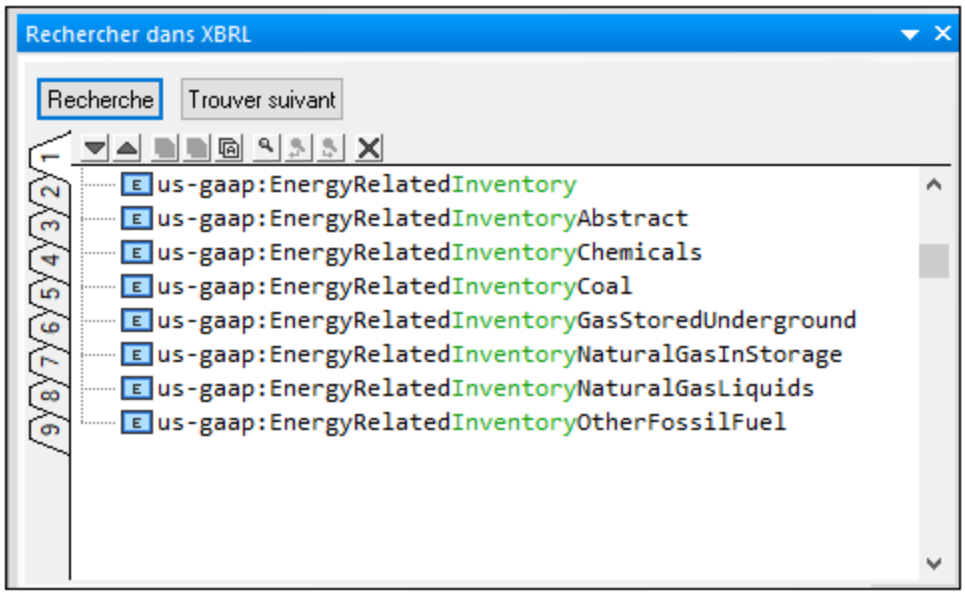
Le résultat produit : (i) un sommaire des types de composant XBRL qui ont été cherchés ; (ii) une liste des instances trouvées du terme en question, trié selon la base de lien ; et (iii) des statistiques concernant la recherche, y compris les correspondances trouvées et le temps qu'a pris la recherche. Chaque groupe de base de lien peut être agrandi ou réduit pour consulter les correspondances dans ce groupe. Cliquer sur une correspondance pour marquer l'élément correspondant dans le document XBRL dans la Fenêtre principale.

Pour plus d'informations concernant les fonctions de la fenêtre Chercher dans XBRL, consulter la section Résultats et Informations.

17.7.3 Résultats et informations

Chaque fois qu'une commande **Chercher** ou **Trouver suivant** est exécutée, les résultats de l'exécution de la commande sont affichés dans la fenêtre Chercher dans XBRL (*capture d'écran ci-dessous*). Le terme recherché est affiché en vert ; (dans la capture d'écran ci-dessous, on voit que le terme recherché était `Inventory`).

© 2016-2022 Altova GmbH



Les fonctions de la fenêtre Chercher dans XBRL

Les résultats sont affichés dans neufs onglets séparés (numérotés de 1 à 9). Afin de pouvoir garder les résultats d'une recherche dans un onglet, effectuer une nouvelle recherche et comparer les résultats. Cliquer sur un résultat dans la fenêtre Chercher dans XBRL et marquer le composant pertinent dans la Fenêtre principale du Mode XBRL. Ainsi, en utilisant la fenêtre Chercher dans XBRL, vous pouvez chercher et parcourir rapidement jusqu'au composant désiré.

Les commandes de la barre d'outils de Chercher dans XBRL suivants sont disponibles :

- Les icônes **Suivant** et **Précédent** permettent de sélectionner, respectivement, les résultats précédent et suivant de recherche pour le résultat sélectionné actuellement.
- Les commande **Copier messages** copient sur le presse-papiers, respectivement, le message sélectionné, le message sélectionné et ses messages enfants et tous les messages.
- Les commandes **Chercher** trouvent les chaînes de texte dans la fenêtre Chercher dans XBRL.
- La commande **Effacer** supprime tous les messages se trouvant dans l'onglet actif actuellement.

17.8 OIM

La spécification Open Information Model (OIM) 1.0 fournit un modèle indépendant de la syntaxe pour les données XBRL, permettant une transformation fiable de données XBRL dans d'autres représentations (OIM-XML, OIM-JSON et OIM-CSV). XMLSpy propose les fonctions liées à l'OIM suivantes :

- Convertir des documents de données XBRL en OIM xBRL-JSON et OIM xBRL-CSV (par le biais du menu Conversion).
- Convertir l'un des trois formats suivants en un des deux autres formats : OIM xBRL-XML, OIM xBRL-JSON et OIM xBRL-CSV (par le biais du menu Conversion).
- Valider tout document au format OIM, qui sera reconnu comme document XBRL et validé en conséquence.

© 2016-2022 Altova GmbH

17.9 Valider les instances XBRL et les taxonomies

Pour valider une instance XBRL ou une taxonomie XBRL, faire du document XBRL le document actif et choisir parmi une des méthodes de validation ci-dessous :

- **XML | Valider XML (F8)**. La validation est effectuée avec le moteur intégré de XMLSpy.
- **XML | Valider XML sur serveur (Ctrl+F8)**. La validation est effectuée par un RaptorXML+XBRL Server à distance (que vous pouvez configurer par le biais de la commande **Outils | Gérer les serveurs Raptor**)

L'onglet de Validation XBRL du dialogue d'Options (**Outils | Options**) fournit des options de validation pertinentes.
XSDB

18 Office Open XML, ZIP, EPUB

Les fichiers Office Open XML (OOXML), ZIP et EPUB ont en commun qu'ils sont tous des paquets contenant d'autres fichiers. Archive View de XMLSpy fournit une interface qui vous permet de consulter la structure interne de ces paquets, de modifier ces structures et d'accéder aux fichiers dans le paquet pour une édition dans XMLSpy. Dans le cas des fichiers EPUB, Archive View vous permet aussi de consulter directement le livre EPUB dans le Mode Navigateur de XMLSpy.

Office Open XML (OOXML)

OOXML est un format de fichier servant à décrire des documents, des feuilles de calcul et des présentations. Développé à l'origine par Microsoft pour leur suite de produits Office, il s'agit aujourd'hui d'une spécification ECMA ouverte.

Structure d'un fichier OOXML

Chaque document OOXML représente un paquet de fichiers multiples qui suit la convention Open Packaging. Un paquet consiste en des fichiers XML et d'autres données (comme des fichiers d'image) plus un fichier de relation qui spécifie les relations parmi les différents fichiers dans le paquet.

La structure interne, le dossier interne et les noms de fichier d'un fichier OOXML varient conformément au type de document. Néanmoins, il existe une structure de base commune : un fichier XML appelé `[Content_Types].xml` à la racine de la structure du répertoire et trois répertoires : `_rels`, `docProps`, et un répertoire spécifique au type de document (dans le cas des documents `.docx`, par exemple, ce dossier serait appelé `word`; `xl` dans des documents `.xlsx` et `ppt` dans des documents `.pptx`).

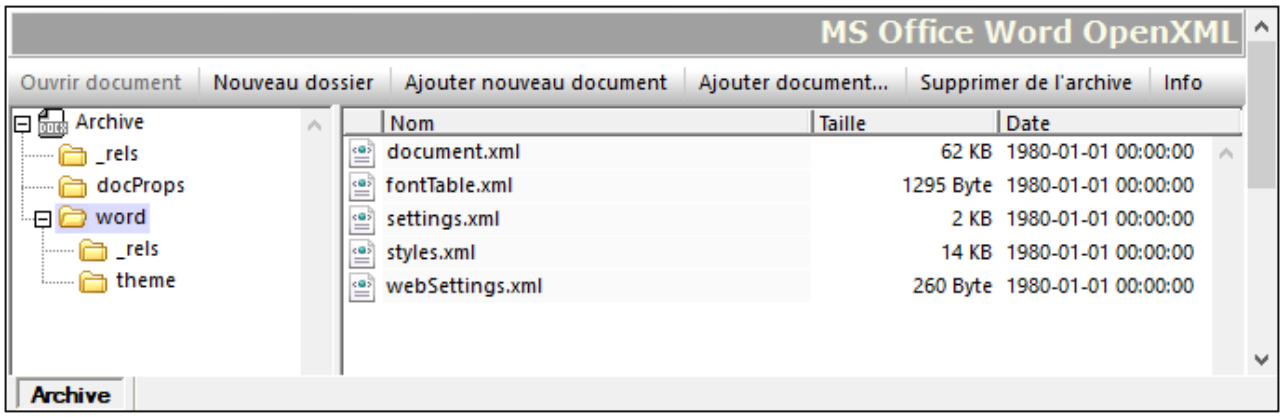
```
OOXML File
|-- File:            [Content_Types].xml
|-- Folder:   _rels
|-- Folder:   docProps
|-- Folder:   word/xl/ppt
```

- Le dossier `_rels` contient un fichier `rels.xml`, qui spécifie les relations entre les différents fichiers contenus dans le paquet.
- Le dossier `docProps` contient `app.xml` et `core.xml`, qui décrivent les propriétés du document clé.
- Les dossiers `word`, `xl` et `ppt` contiennent des fichiers XML qui recèlent le contenu du document. Par exemple, dans le document `word`, le fichier `document.xml` contient le contenu central du document.

OOXML dans le Mode Archive de XMLSpy

Dans le Mode Archive de XMLSpy (*capture d'écran ci-dessous*), vous pouvez consulter et éditer les contenus d'un fichier OOXML.

© 2016-2022 Altova GmbH



Le Mode Dossier affiché dans le volet de gauche montre les dossiers contenus dans le paquet, alors que la Fenêtre principale montre les fichiers contenus dans le dossier sélectionné dans le Mode Dossier. Dans le Mode Archive, les fichiers et les dossiers peuvent être ajoutés et supprimés des archives. De même, les fichiers peuvent être ouverts rapidement pour une édition dans XMLSpy en double cliquant sur le fichier dans le Mode Archive.

Édition intelligente de fichiers internes d'OOXML

Les documents XML dans les paquets OOXML sont basés sur des schémas standard. XMLSpy propose une prise en charge de l'édition intelligente pour les documents OOXML, sous la forme d'assistants à la saisie, le remplissage automatique et la validation.

Fichiers ZIP

Les fichiers ZIP archivent des fichiers multiples dans un paquet de compression sans perte. Ces fichiers peuvent être de type varié. Dans Archive View de XMLSpy, les fichiers ZIP peuvent être créés, la structure interne peut être modifiée et les fichiers contenus dans les archives peuvent être édités. Ces opérations sont décrites dans la sous-section fichiers ZIP de cette section.

Fichiers EPUB

Un fichier EPUB est un groupe comprimé de fichiers utilisés pour la distribution de publications numériques (livres EPUB). Dans Archive View, vous pouvez ouvrir des fichiers EPUB, créer et éditer des fichiers EPUB, prévisualiser un livre EPUB numérique, éditer des fichiers de composant pour les archives EPUB directement dans XMLSpy, valider le fichier EPUB et enregistrer les fichiers de composant dans les archives EPUB. Voir la section fichiers EPUB pour plus de détails.

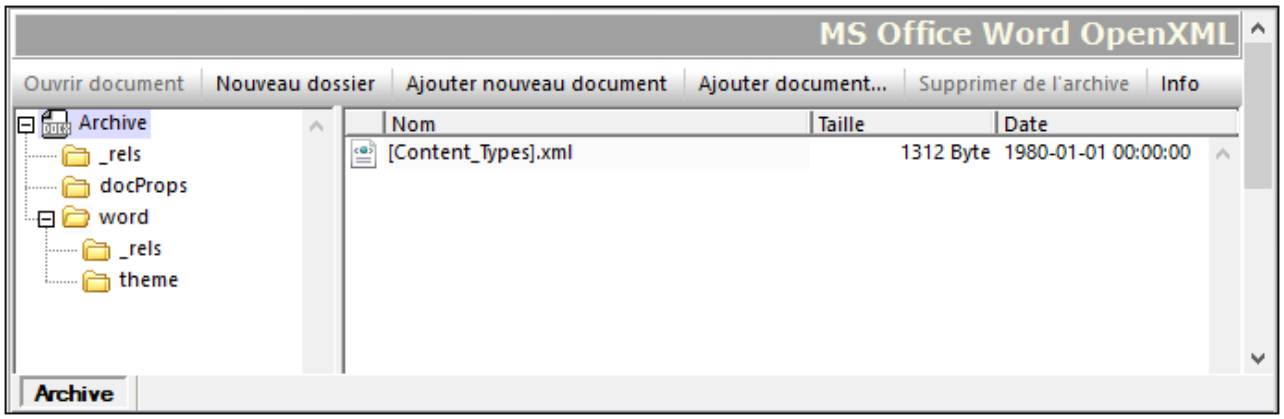
18.1 Travailler avec des fichiers OOXML

Cette section décrit comment travailler avec des documents OOXML dans Archive View. Les procédures suivantes seront discutées :

- Créer, ouvrir et enregistrer des fichiers OOXML
- Éditer la structure d'un fichier OOXML
- Ouvrir, éditer et enregistrer des documents internes OOXML
- Édition intelligente des documents internes OOXML
- Adresser des documents dans les fichiers OOXML
- Comparer les archives OOXML

Créer, ouvrir et enregistrer des fichiers OOXML

Les fichiers OOXML sont créés dans le dialogue Créer nouveau document (commande **Fichier | Nouveau**), dans lequel vous sélectionnez le type de fichier requis (`.docx`, `.pptx`, ou `.xlsx`). Vous serez invité à saisir un nom de fichier et un emplacement sous lequel enregistrer le fichier. Le nouveau fichier est créé à l'emplacement spécifié puis ouvert dans Archive View (*capture d'écran ci-dessous*). Veuillez noter que la structure interne de base du document OOXML a été créée.



Un fichier OOXML existant est ouvert dans le Mode Archive par le biais du dialogue Ouvrir (**Fichier | Ouvrir**) de XMLSpy. Enregistrer les fichiers OOXML avec la commande **Fichier | Enregistrer (Ctrl+S)**. Cette commande sauvegarde la structure et les relations avec le fichier OOXML.

Éditer la structure d'un fichier OOXML

Les contenus d'un fichier OOXML peuvent être modifiés en ajoutant et en supprimant les dossiers et les documents avec la fonction Archive View. Une fois que ces changements structurels ont été effectués, le fichier OOXML doit être enregistré (**Fichier | Enregistrer**) pour que les modifications prennent effet. Veuillez noter les points suivants :

- Lorsqu'un nouveau dossier ou document est ajouté par les boutons de commande dans Archive View, il devrait être nommé immédiatement après sa création. Il n'est pas possible de renommer un dossier ou un document dans Archive View.
- Une fois qu'un nouveau document a été ajouté à un dossier d'archive, il est enregistré dans les archives en l'enregistrant dans sa propre fenêtre ou en enregistrant le fichier OOXML.

© 2016-2022 Altova GmbH

Ouvrir, éditer et enregistrer des documents internes OOXML

Un document OOXML interne, c'est à dire un document contenu dans un paquet de fichier OOXML, est ouvert depuis Archive View en le double-cliquant, ou en le sélectionnant dans la Fenêtre principale et en cliquant sur la touche de commande Ouvrir le document. Le document s'ouvre dans une fenêtre XMLSpy séparée. Une fois l'avoir édité, il suffit d'enregistrer le document pour l'enregistrer dans l'archive OOXML ; plus besoin d'enregistrer le fichier OOXML.

Édition intelligente des documents internes OOXML

XMLSpy propose des fonctions d'édition intelligentes pour les documents Office Open XML internes ; c'est à dire pour des documents contenus dans un paquet de fichier OOXML. Ces fonctions comprennent des assistants à la saisie, un remplissage automatique et une validation.

Adresser des documents dans les fichiers OOXML

Les documents contenus dans les fichiers OOXML peuvent être adressés avec des chemins de fichier normaux plus le caractère de barre verticale. Par exemple, le chemin de fichier :

```
C:\Documents and Settings\<username>\My
Documents\Altova\XMLSpy2023\Examples\Office20XX\ExcelDemo.xlsx|zip\xl\tables\table1.xml
```

localise le fichier `table1.xml`, qui se trouve dans le dossier `xl\tables` du fichier OOXML `ExcelDemo.xlsx` situé dans le dossier `Examples\Office20XX` du dossier d'exemples XMLSpy.

Comparer les archives OOXML

Lorsqu'un fichier OOXML est ouvert dans Archive View, vous pouvez le comparer avec une autre archive en utilisant la commande **Outils | Comparer les répertoires**.

18.2 Fichiers d'exemples OOXML

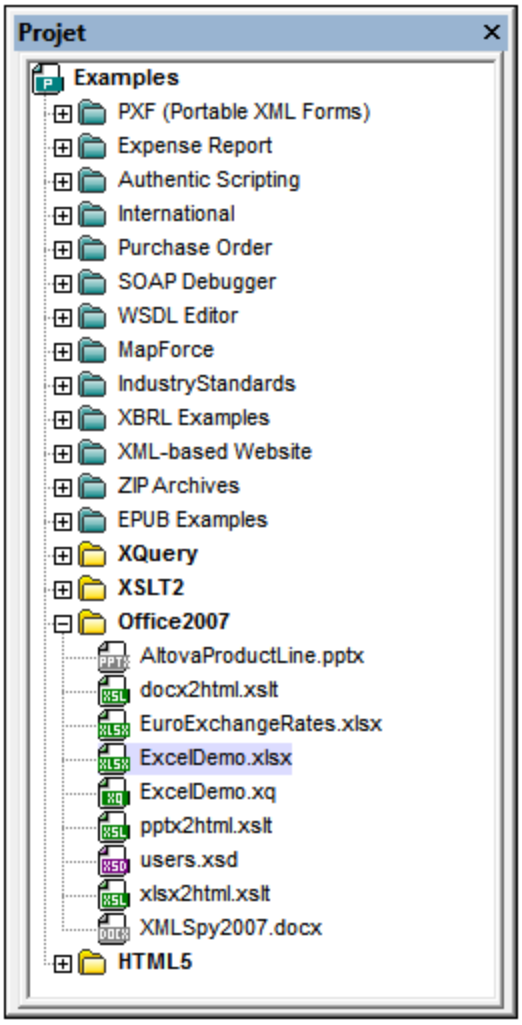
Dans le dossier `Examples\Office` de votre dossier d'application XMLSpy, vous trouverez les fichiers d'exemple suivants :

- Fichiers OOXML : (i) un fichier Word Open XML (`.docx`), (ii) un fichier Excel Open XML(`.xlsx`), et (iii) un fichier PowerPoint Open XML (`.pptx`)
- Fichiers XSLT : (i) `docx2html.xslt` (pour convertir le fichier échantillon `.docx` en HTML), (ii) `xslx2html.xslt` (pour convertir le fichier échantillon `.xslx` en HTML) et (iii) `pptx2html.xslt` (pour convertir le fichier échantillon `.pptx` en HTML)
- Un fichier XQuery : `ExcelDemo.xq` (pour extraire les données du fichier `.xslx`)

Les fichiers XSLT et XQuery visent à démontrer comment XSLT et XQ peuvent être utilisés pour accéder et transformer des données dans des fichiers OOXML. Pour exécuter les documents XSLT et XQuery, vous pouvez utiliser une des options suivantes :

- Ouvrir le fichier OOXML dans Archive View. Dans le Mode Dossier, sélectionner `Archive` puis cliquer sur la commande de menu **Transformation XSL/XQuery | XSL** (pour une transformation XSLT) ou **Exécution XSL/XQuery | XQuery** (pour une exécution XQuery). Parcourir le fichier XSLT ou XQuery et cliquer sur **OK**.
- Dans la fenêtre Projet de XMLSpy, cliquer avec la touche de droite sur le fichier `.xlsx`, `.pptx` ou `.docx` dans le dossier `Office2007` du projet `Examples` (*capture d'écran ci-dessous*) et sélectionner la commande de transformation. Chercher le fichier de transformation et cliquer sur **OK**.

© 2016-2022 Altova GmbH



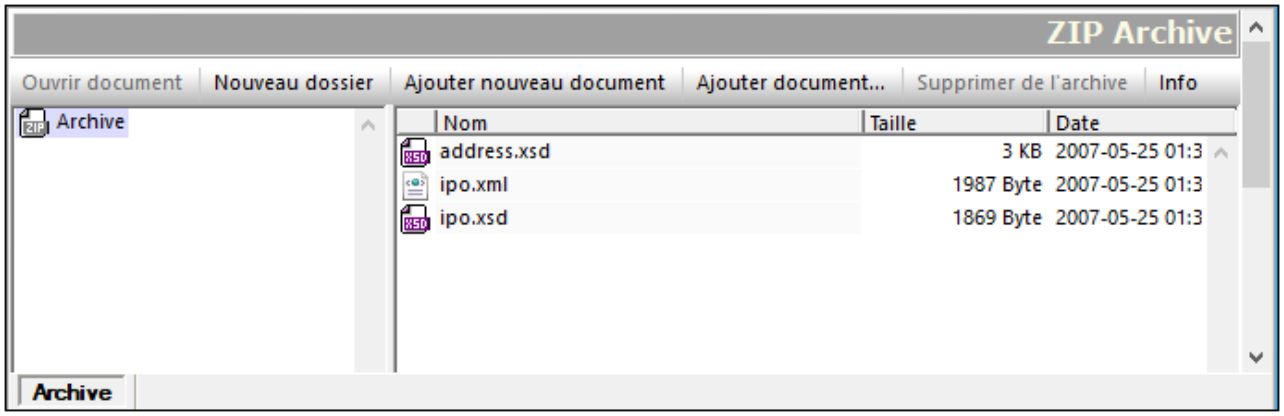
- Ouvrir le fichier XSLT ou XQuery dans XMLSpy et cliquer sur la commande de menu **Transformation XSL/XQuery | XSL** et **Exécution XSL/XQuery | XQuery**, respectivement. Lorsque vous serez invité à transformer le fichier XML, chercher le fichier `.docx`, `.xlsx` , ou `.pptx` (selon que le document XSLT/XQ soit destiné à fonctionner sur MS Word, MS Excel ou MS PowerPoint).

18.3 Fichiers ZIP

Dans Archive View, vous pouvez créer des fichiers WinZip, modifier la structure interne des fichiers ZIP (WinZip, WinRAR, etc), et éditer les fichiers dans le paquet ZIP directement dans XMLSpy et enregistrer les fichiers dans les archives ZIP.

Créer et enregistrer un fichier WinZip

Un fichier WinZip est créé par le biais du dialogue Créer nouveau document (commande **Fichier | Nouveau**), dans lequel vous sélectionnez le type de fichier `.zip`. Un archive WinZip vide est créé dans une nouvelle fenêtre dans XMLSpy (*capture d'écran ci-dessous*). À présent, vous devez enregistrer le fichier ZIP à l'emplacement souhaité avec la commande **Fichier | Enregistrer (Ctrl+S)**. Ajouter les dossiers et les fichiers comme décrit ci-dessous puis enregistrer le fichier ZIP pour enregistrer vos ajouts et changements.



Un fichier ZIP existant est ouvert dans le Mode Archive via le dialogue Ouvrir (**Fichier | Ouvrir**) de XMLSpy.

Note : la création d'un nouveau fichier ZIP est différente de la création d'un nouveau fichier OOXML étant donné que vous ne serez pas invité à préciser un emplacement pour enregistrer le fichier avant d'ouvrir l'archive dans Archive View. Pour enregistrer le fichier ZIP depuis l'archive vide qui est ouvert dans Archive View, vous devez utiliser explicitement la commande **Fichier | Enregistrer (Ctrl+S)**.

Ajouter des dossiers et des fichiers et modifier la structure d'archive

Vous pouvez ajouter des dossiers (cliquer sur la touche **Nouveau dossier**), des fichiers existants (**Ajouter document**), et de nouveaux fichiers (**Ajouter nouveau document**) au dossier Archive sélectionné. Veuillez noter que lorsque vous ajoutez un nouveau dossier ou un nouveau document, vous devez immédiatement saisir un nom pour le dossier ou le fichier ; il ne sera pas possible de renommer des dossiers ou des documents dans Archive View.

Adresser des documents dans les fichiers ZIP

Les documents contenus dans les fichiers ZIP peuvent être adressés avec des chemins de fichiers normaux complétés du symbole de barre verticale. Par exemple, le chemin de fichier :

```
C:\Documents and Settings\<username>\My
Documents\Altova\XMLSpy2023\Examples\Test.zip|zip\TestFolder\MyFile.xml
```


© 2016-2022 Altova GmbH

localise le fichier `MyFile.xml`, qui se trouve dans le dossier `TestFolder` du fichier ZIP `Test.zip` situé dans le dossier `Examples` du dossier d'exemples XMLSpy.

Comparer les archives ZIP

Lorsqu'un fichier ZIP est ouvert dans Archive View, vous pouvez le comparer avec une autre archive en utilisant la commande **Outils | Comparer répertoires**.

18.4 Fichiers EPUB

Un fichier EPUB est un groupe de fichiers zippés conformes au standard EPUB du International Digital Publishing Forum (IDPF). Ce standard est la norme de distribution et d'échange pour les publications web numériques. Dans Archive View, vous pouvez ouvrir des fichiers EPUB, consulter la publication numérique EPUB du fichier dans un onglet de pré-visualisation, éditer les fichiers de composants de l'archive EPUB directement dans XMLSpy, valider le fichier EPUB, et réenregistrer les fichiers de composant dans l'archive EPUB.

Note : (i) XMLSpy prend en charge EPUB 2.0.1. (ii) Un fichier EPUB modèle est disponible dans le projet `Examples` et dans le dossier `(My) Documents/Altova/XMLSpy2023/Examples`.

Terminologie

Dans les descriptions ci-dessous, les termes suivants sont utilisés comme suit :

- ***Fichier EPUB*** est utilisé pour indiquer le fichier EPUB comportant l'extension `.epub`. Il s'agit du fichier ZIP qui contient toute l'archive et il s'agit du fichier qui sera ouvert dans Archive View
- ***Fichier archive*** est un des fichiers contenus dans l'archive EPUB
- ***Livre EPUB*** est le terme utilisé pour indiquer la publication numérique générée par le fichier EPUB zippé

Dans cette section

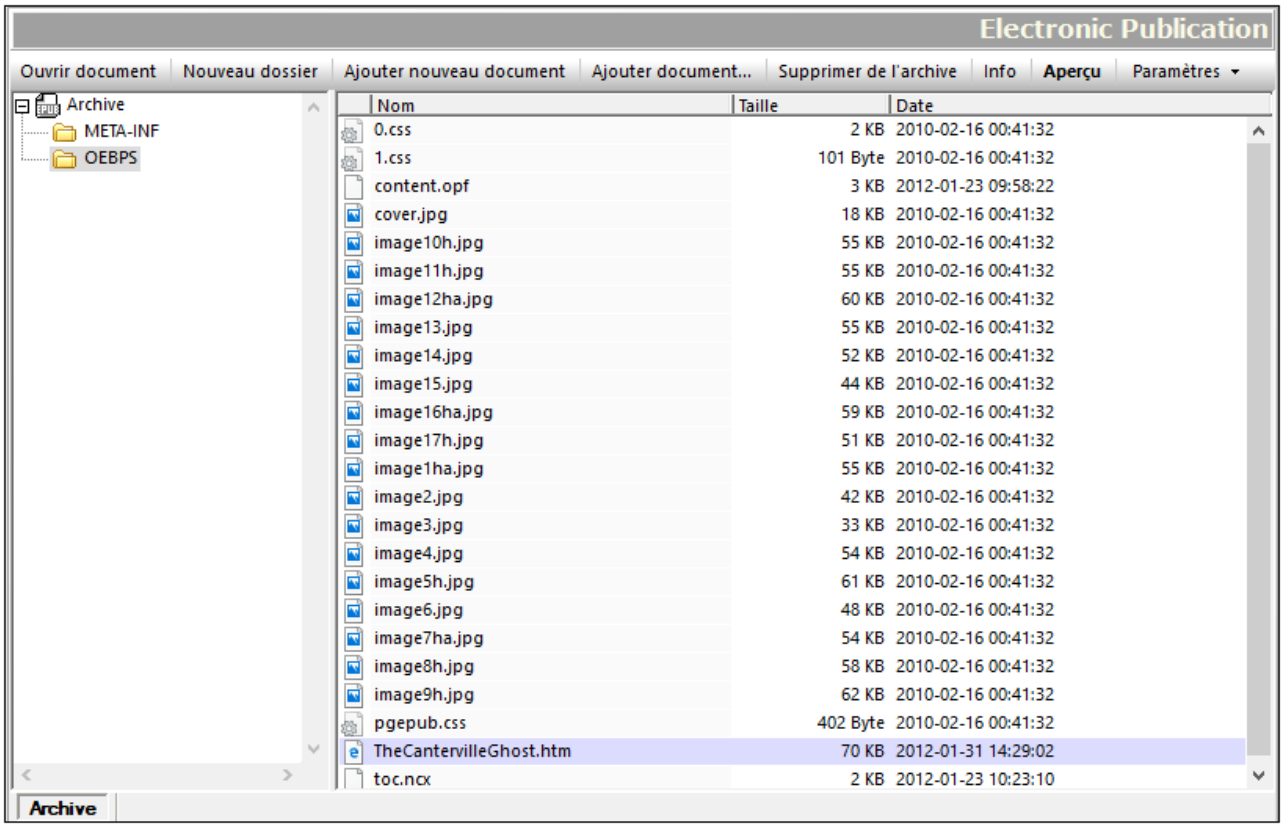
La description ci-dessous des fonctions EPUB dans XMLSpy est structurée dans les parties suivantes :

- Ouvrir des fichiers EPUB dans Archive View
- Créer un nouveau fichier EPUB
- Prévisualiser un livre EPUB
- Modifier les contenus et les structures d'une archive EPUB
- Infos et paramètres
- Éditer les fichiers d'archive directement dans XMLSpy
- Assistants à la saisie pour les fichiers d'archive
- Valider un fichier EPUB

Ouvrir des fichiers EPUB dans Archive View

Sélectionner la commande de menu **Fichier | Ouvrir**, se rendre sur le fichier EPUB et cliquer sur **Ouvrir**. Le fichier EPUB s'ouvre dans Archive View (*capture d'écran ci-dessous*). En alternative, vous pouvez cliquer avec la touche de droite sur le fichier EPUB dans Windows Explorer et sélectionner la commande de menu contextuel pour ouvrir le fichier avec XMLSpy. Si vous avez configuré XMLSpy en tant que l'éditeur par défaut des fichiers EPUB, double-cliquer sur le fichier EPUB pour ouvrir le fichier dans Archive View.

© 2016-2022 Altova GmbH



Le Mode Dossier affiché dans le volet de gauche montre les dossiers dans l'archive, alors que la Fenêtre principale montre les fichiers dans le dossier sélectionné dans le Mode Dossier. L'archive EPUB aura la structure et les composants clés suivants.

```
Archive
|-- Mimetype file
|
|-- META-INF folder
|     |-- container.xml
|
|-- DOCUMENT folder (dans la capture d'écran ci-dessus, OEBPS est le dossier Document.)
      |-- Contains HTML, CSS, image files, plus OPF and NCX files
```

Créer un nouveau fichier EPUB

Pour créer un nouveau fichier EPUB, sélectionner la commande de menu **Fichier | Nouveau**. Dans le dialogue Créer nouveau document qui s'ouvre, choisir le type de fichier `.epub`. Dans le dialogue Enregistrer sous qui s'affiche, donner un nom à votre document EPUB et cliquer sur **Enregistrer**. Une archive EPUB squelette contenant tous les dossiers et fichiers d'une archive EPUB valide (*voir structure de l'archive ci-dessus*) sera créée dans une nouvelle fenêtre dans Archive View. Ajouter les dossiers et les fichiers que vous souhaitez ajouter à l'archive, comme décrit ci-dessous, puis enregistrer le fichier EPUB. Pour éditer un fichier d'archive directement dans XMLSpy, double-cliquer sur le fichier dans Archive View. Le fichier s'ouvrira dans une nouvelle fenêtre XMLSpy. L'éditer puis l'enregistrer avec la commande **Fichier | Enregistrer (Ctrl+S)**.

Prévisualiser un livre EPUB

Afin de prévisualiser un livre EPUB, rendre le fichier EPUB actif dans Archive View, puis cliquer sur la touche **Aperçu** dans la barre outils de Archive View. Le livre EPUB s'ouvrira dans une fenêtre de navigateur séparée (Internet Explorer) dans XMLSpy. Si un des fichiers qui sera utilisé pour la prévisualisation, qu'il s'agisse d'un fichier de contenu ou d'un fichier lié à la structure, a été modifié mais pas enregistré, vous serez invité à enregistrer le fichier. Si vous n'enregistrez pas les modifications, l'aperçu utilisera les données précédemment enregistrées et il se pourrait qu'il ne soit pas à jour. Vous pouvez spécifier que tous les fichiers modifiés seront enregistrés automatiquement avant la prévisualisation en activant ce paramètre (par le biais de la touche **Paramètres** dans la barre outils de Archive View).

Veuillez noter les éléments suivants :

- Si la touche **Aperçu** dans Archive View est cliquée alors qu'une fenêtre d'Aperçu de cette publication EPUB est encore ouverte, alors la publication EPUB sera rechargée dans la fenêtre ouverte d'Aperçu.
- La réinitialisation de la fenêtre d'Aperçu elle-même (à l'aide de la commande **Refresh (F5)** d'Internet Explorer) ne met pas à jour la fenêtre d'Aperçu. La publication EPUB dans la fenêtre Aperçu doit être mise à jour à l'aide de la touche **Aperçu** (de Archive View) du fichier EPUB correspondant (*voir point précédent*).
- Pour fermer l'aperçu, fermer la fenêtre Aperçu.

Note : Toutes les balises EPUB ne sont pas prises en charge dans Internet Explorer, les aperçus peuvent donc être faussés. De plus, si le document de publication numérique est un XML et non un HTML, l'aperçu peut ne pas fonctionner. Des versions plus récentes d'Internet Explorer proposent une gestion améliorée des balises EPUB, donc si vous rencontrez des problèmes, essayez de vous mettre à jour de la dernière version d'Internet Explorer.

Modifier les contenus et les structures d'une archive EPUB

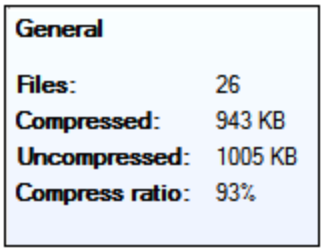
Vous pouvez ajouter des dossiers (cliquer sur la touche **Nouveau dossier**), de nouveaux fichiers (**Ajouter nouveau document**), et des fichiers existants (**Ajouter document**) au dossier d'archive sélectionné. Veuillez noter que lorsque vous ajoutez un nouveau dossier ou un nouveau document, vous devez immédiatement saisir un nom pour le dossier ou le fichier ; il ne sera pas possible de renommer des fichiers dans Archive View. Vous pouvez supprimer un fichier ou un dossier en le sélectionnant et en cliquant sur la touche **Supprimer depuis l'archive**.

Une fois avoir modifié l'archive, vous devez enregistrer le fichier EPUB (**Fichier | Enregistrer**) pour que les modifications soient enregistrées.

Infos et paramètres

Cliquer sur la touche **Info** pour afficher en bas de Archive View, un sommaire des informations d'archive clés (*capture d'écran ci-dessous*). Cliquer à nouveau sur la touche **Info** pour supprimer le sommaire. Le sommaire rapporte le nombre de fichiers contenus dans l'archive (y compris le fichier Mimetype et `container.xml`), la taille du fichier EPUB comprimé et la taille cumulative des fichiers décompressés.

© 2016-2022 Altova GmbH



La touche **Paramètres** contient deux options d'enregistrement de fichier automatiques qui peuvent être activées et désactivées : pour enregistrer automatiquement le fichier EPUB (i) avant la validation, et (ii) avant la prévisualisation du fichier EPUB (via la touche **Aperçu**) dans XMLSpy.

Éditer les fichiers d'archive directement dans XMLSpy

Pour éditer un fichier d'archive directement dans XMLSpy, double-cliquer le fichier dans Archive View. En alternative, sélectionner le fichier dans Archive View et cliquer sur la touche Ouvrir document dans la barre outils de Archive View. Le fichier s'ouvrira dans une nouvelle fenêtre XMLSpy. L'éditer puis l'enregistrer avec la commande **Fichier | Enregistrer (Ctrl+S)**.

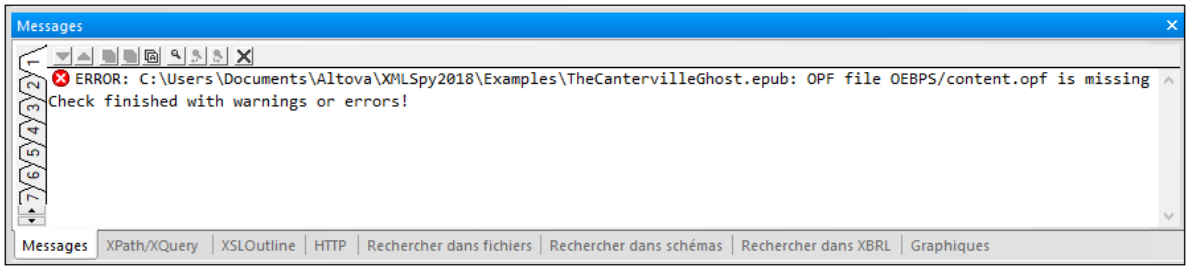
Assistants à la saisie pour les fichiers d'archive

Les assistants à la saisie pour les fichiers d'archive à base standard sont disponibles lorsque ces fichiers d'archive sont ouverts dans XMLSpy. Ces fichiers d'archive sont :

- Le fichier OPF, traditionnellement nommé `content.opf`, il contient les métadonnées EPUB du livre. Il est basé sur les spécifications Open Packaging Format (OPF).
- Le fichier NCX (fichier Navigation Control pour XML), traditionnellement nommé `toc.ncx`, il contient la table des matières de la publication. Il est basé sur la partie NCX des spécifications OPF.
- Le dossier nommé `META-INF` doit contenir le fichier `container.xml`, qui pointe vers le fichier définissant les contenus du livre (le fichier OPF). Le fichier `container.xml` spécifie comment les fichiers d'archive doivent être organisés conformément aux règles indiquées dans les spécifications Open Container Format (OCF).

Valider un fichier EPUB

Afin de valider un fichier EPUB, sélectionner la commande **XML | Valider XML (F8)**. Les résultats de validation sont affichés dans la fenêtre Messages (*capture d'écran ci-dessous*). Si un des fichiers d'archive, qu'il s'agisse d'un fichier de contenu ou d'un fichier lié à la structure, a été modifié mais pas enregistré, vous serez invité à enregistrer le fichier. Vous devez enregistrer les fichiers modifiés pour valider le fichier EPUB. Vous pouvez spécifier que tous les fichiers modifiés seront enregistrés automatiquement avant la validation en activant ce paramètre (par le biais de la touche **Paramètres** dans la barre outils de Archive View).



Affichage des messages d'erreur : (i) le fichier dans lequel l'erreur a été trouvée et, si applicable, le numéro de la ligne dans laquelle l'erreur s'est produite ; (ii) une description de l'erreur. Dans la capture d"écran ci-dessus, l'erreur surlignée s'est produite dans la ligne 21 du fichier `content.opf`. Cliquer sur la ligne d'erreur dans la fenêtre Messages pour ouvrir le fichier pertinent et marquer l'erreur.

Note : Le moteur de validation EPUB est une fonctionnalité Java, donc Java doit être installé sur votre appareil pour que le moteur de validation puisse être exécuté.

© 2016-2022 Altova GmbH

19 Bases de données

XMLSpy vous permet de vous connecter à un grand nombre de bases de données (BD) et d'effectuer des opérations comme par exemple la requête d'une BD, l'importation de structures de BD en tant que Schéma XML, la génération d'un fichier de données XML depuis la BD et l'exportation de données vers une BD. Chacune des fonctions liées à une BD est disponible dans XMLSpy en tant que commande de menu, et est décrite dans la section Référence de l'utilisateur de cette documentation sous la commande respective. Vous trouverez ci-dessous une liste complète de ces commandes, avec des liens vers les description respectives.

Dans cette section, nous allons :

- Décrire comment se connecter à une base de données, il s'agit là d'une opération nécessaire pour exécuter une des commandes XMLSpy liées aux BD ; et
- Lister les BD qui ont été testées avec succès pour une utilisation avec XMLSpy.

Note : Si vous utilisez la version 64-bit de XMLSpy, veuillez vous assurer que vous avez accès aux pilotes de base de données 64-bit nécessaires pour la base de données spécifique à laquelle vous vous connectez.

Fonction de XMLSpy liées aux BD

Les fonctions de XMLSpy liées aux BD sont exécutées avec les commandes dans les menus BD et Convertir.

- *Base de données Query :* Dans le menu **BD**. Charge la structure de la BD dans une fenêtre de requête de base de données séparée et permet les requêtes dans la BD. Les résultats sont affichés dans la fenêtre de Requête de base de données.
- *IBM DB2 :* Dans le menu **BD**. IBM DB2 est une BD XML, et XMLSpy active la gestion des Schémas XML de la BD XML ainsi que l'édition et la validation de la BD XML.
- *Oracle XML DB* : Dans le menu **BD**. Fournit une gamme étendue de fonctions pour les BD XML Oracle, y compris la gestion de Schéma XML, les requêtes de bases de données et la génération de fichiers XML sur la base de schémas de BD.
- *Importer des données de base de données* : Dans le menu **Convertir**. Importe des données de BD dans un fichier XML.
- *Créer un Schéma XML depuis la structure de BD* : Dans le menu **Convertir**. Génère un Schéma XML basé sur la structure de la BD.
- *Importation de BD basée sur le Schéma XML* : Dans le menu **Convertir**. Avec un document de Schéma XML activé dans XMLSpy, une connexion de BD est établie et les données d'une table de BD sélectionnée peuvent être importées. Le document XML résultant aura une structure basée sur le Schéma XML actif pendant que la connexion de BD est établie.
- *Créer une structure BD depuis le Schéma XML* : Dans le menu **Convertir**. Les tables de BD ne contenant pas de données sont créées sur la base de la structure d'un Schéma XML existant.
- *Exporter vers la base de données* : Dans le menu **Convertir**. Les données provenant d'un document XML peuvent être exportées vers une BD. Les tables de BD existantes peuvent être mises à jour avec les données XML, ou des nouvelles tables contenant les données XML peuvent être créées.

Conversions de type de données

Lors de la conversion de données entre les documents XML et des BD, les types de données doivent nécessairement être convertis en types appropriés pour les formats respectifs. La manière dont XMLSpy convertit les types de données est décrite dans les annexes Types de données dans les Schémas XML générés par BD et Types de données dans des BD générées depuis des Schémas XML.

Altova DatabaseSpy

Altova DatabaseSpy est un outil de conception de requête et de BD multiplateforme qui propose des fonctions supplémentaires de BD à celles disponibles dans XMLSpy. Pour plus de détails concernant Altova DatabaseSpy, consultez le site Internet Altova.

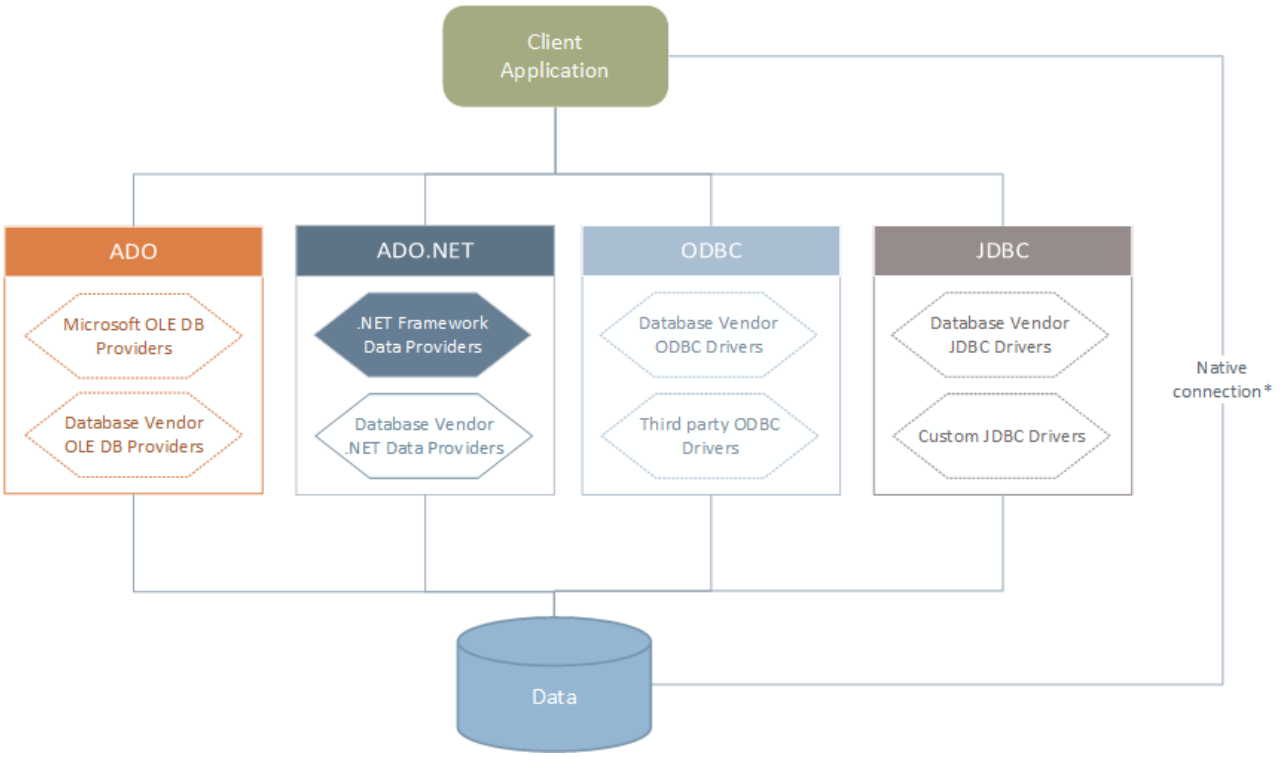
© 2016-2022 Altova GmbH

19.1 Connexion à une source de données

Dans les cas les plus simples, une base de données peut être un fichier local tel qu'un fichier de base de données Microsoft Access ou SQLite. Dans un scénario plus avancé, une base de données peut résider sur un serveur à distance ou un serveur de base données de réseau qui n'utilise pas nécessairement le même système d'exploitation que celui de l'application qui s'y connecte et qui consomme les données. Par exemple, alors que XMLSpy fonctionne sur un système d'exploitation Windows, la base de données à partir de laquelle vous souhaitez accéder aux données (par exemple MySQL) pourrait, elle, fonctionner sur une machine Linux.

Pour interagir avec des types de bases de données aussi bien locaux qu'à distance, XMLSpy repose sur les interfaces de connexion de données et les pilotes de base de données d'ores et déjà disponibles sur votre système d'exploitation ou publiés régulièrement par les distributeurs principaux de bases de données. Dans un environnement technologique de bases de données en constante évolution, cette approche permet une plus grande flexibilité et interopérabilité multiplateforme.

Le diagramme suivant illustre, de manière simplifiée, les options de connectivité des données disponibles entre XMLSpy (illustré en tant qu'application de client générique) et un stockage de données (qui peut être un serveur de base de données ou un fichier de base de données.



** Les connexions natives directes sont prises en charge pour les bases de données SQLite, PostgreSQL, Couch DB et MongoDB. Pour se connecter à ce type de bases de données, il n'est pas nécessaire d'installer un pilote supplémentaire sur votre système.*

Tel qu'indiqué dans le diagramme ci-dessus, XMLSpy peut accéder à tout type de base de données principale par le biais des technologies d'accès aux données suivantes :

- ADO (Microsoft® ActiveX® Data Objects), qui, à son tour, utilise un fournisseur OLE DB sous-jacent (Object Linking and Embedding, Database)
- ADO.NET (un ensemble de bibliothèques disponibles dans le Microsoft .NET Framework qui permet une interaction avec des données)
- JDBC (Java Database Connectivity)
- ODBC (Open Database Connectivity)

Note : certains fournisseurs ADO.NET ne sont pas pris en charge ou ne le sont que de manière limitée. See Notes de prise en charge d'ADO.NET.

À propos des technologies d'accès aux données

L'interface de connexion aux données que vous choisissez dépend largement de votre infrastructure logicielle existante. En règle générale, vous choisirez la technologie d'accès aux données et le pilote de base de données qui s'intègre le plus étroitement avec le système de base de données auquel vous souhaitez vous connecter. Par exemple, pour se connecter à une base de données Microsoft Access 2013, vous construiriez en principe une chaîne de connexion ADO qui utilise un fournisseur natif comme le **Microsoft Office Access Database Engine OLE DB Provider**. En revanche, pour vous connecter à Oracle, vous choisirez peut-être de télécharger et d'installer les dernières interfaces JDBC, ODBC ou ADO .NET depuis le site Internet Oracle.

Alors que les pilotes pour les produits Windows (comme Microsoft Access ou SQL Server) sont éventuellement déjà disponibles sur votre système d'exploitation Windows, ils peuvent ne pas être disponibles pour d'autres types de base de données. Les distributeurs des principales bases de données publient régulièrement des logiciels de client de base de données et des pilotes accessibles au public qui fournissent un accès multiplateforme aux bases de données respectives par le biais de combinaisons de ADO, ADO.NET, ODBC ou JDBC. De plus, plusieurs pilotes tiers peuvent être disponibles pour une ou plusieurs des technologies mentionnées ci-dessus. Dans la plupart des cas, il existe plusieurs manières de se connecter à la base de données requise depuis votre système d'exploitation, et, par conséquent, depuis XMLSpy. Les fonctions disponibles, paramètres de performance et les problèmes connus varieront généralement selon la technologie d'accès des données ou pilotes utilisés.

19.1.1 Lancer l'Assistant de la connexion

XMLSpy fournit un assistant de connexion à la base de données qui vous guide dans les étapes requises afin de configurer une connexion vers une source de données. Avant de suivre les étapes de l'assistant, veuillez noter que pour certains types de bases de données il est nécessaire d'installer et de configurer séparément plusieurs prérequis de base de données, tels qu'un pilote de base de données ou un logiciel client de base de données. Ceux-ci sont généralement fournis par les distributeurs de base de données respectifs, et comprennent la documentation taillée sur votre version Windows spécifique. Pour une liste de pilotes regroupés par base de données, voir Aperçu des pilotes de base de données.

Pour démarrer l'assistant de connexion à la base de données (*voir la capture d'écran ci-dessous*), suivez les étapes suivantes :

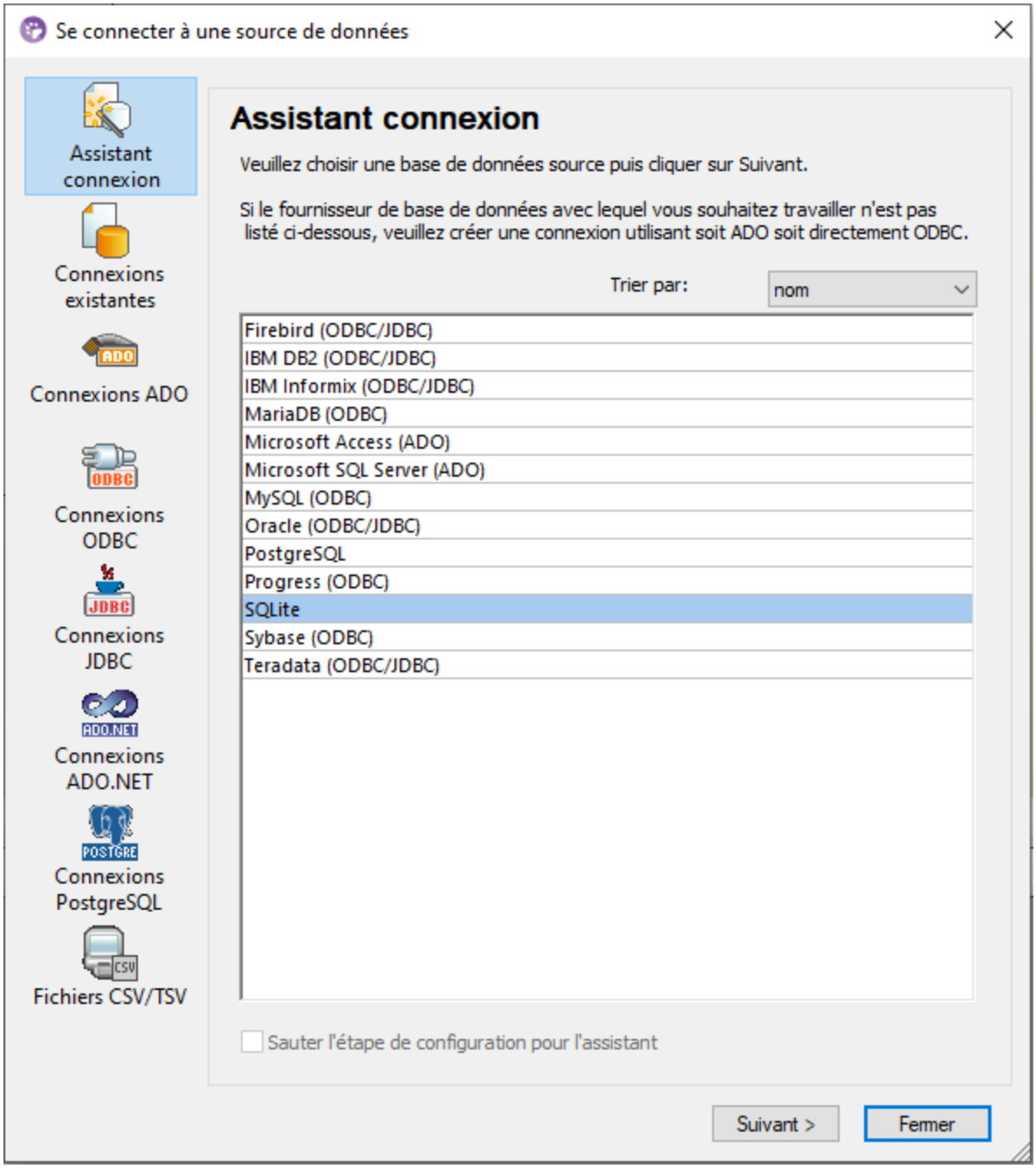
- Dans le menu **BD**, cliquez sur **Requête de base de données**.

L'assistant de connexion de connexion à la base de données (*voir la capture d'écran ci-dessous*) est lancé. À gauche de la fenêtre, vous pouvez sélectionner la manière la plus appropriée de connexion à la base de données parmi les possibilités suivantes :

© 2016-2022 Altova GmbH

- L'assistant de connexion qui vous invite à choisir votre type de base de données, puis vous guide à travers les étapes pour se connecter à une base de données de ce type
- Sélectionnez une connexion existante
- Sélectionnez une technologie d'accès aux données : ADO, ADO.NET, ODBC ou JDBC
- Utilisez une ressource globale Altova dans laquelle la connexion à la base de données est stockée
- Une connexion PostgreSQL native

Dans le volet de l'assistant de connexion (*voir la capture d'écran ci-dessous*), les bases de données peuvent être stockées de manière alphabétique par le nom du type de base de données ou par l'utilisation récente. Sélectionnez l'option que vous souhaitez dans la liste de choix *Trier par*. Après avoir sélectionné le type de base de données auquel vous voulez vous connecter, cliquez sur **Suivant**.



L'assistant vous guidera à travers les prochaines étapes conformément au type de base de données, à la technologie de connexion (ADO, ADO.NET, ODBC, JDBC) et au pilote qui sera utilisé. Pour consulter des exemples applicables aux différents types de base de données, voir Exemples de connexion à des bases de données.

En alternative à l'utilisation de l'assistant, vous pouvez utiliser une des technologies d'accès à la base de données suivantes :

- Configurer une connexion ADO

© 2016-2022 Altova GmbH

- Configurer une connexion ADO.NET
- Configurer une connexion ODBC
- Configurer une connexion JDBC

19.1.2 Aperçu des pilotes de base de données

Le tableau suivant regroupe les pilotes de base de données les plus communs que vous pouvez utiliser pour vous connecter à une base de données particulière par le biais d'une technologie d'accès aux données particulières. Veuillez noter que cette liste n'a pas la prétention d'être exhaustive ou prescriptive ; vous pouvez utiliser d'autres solutions natives ou tierces outre les pilotes indiqués ci-dessous.

Bien qu'un certain nombre de pilotes de base de données sont peut-être déjà disponibles sur votre système d'exploitation Windows, il se peut que vous nécessitiez le téléchargement d'un pilote alternatif. En ce qui concerne certaines bases de données, le dernier pilote fourni par le distributeur de bases de données est susceptible de fonctionner mieux que le pilote fourni avec le système d'exploitation.

Les distributeurs de bases de données peuvent fournir des pilotes soit en tant que pack séparément téléchargeables soit fournis avec le logiciel de client de la base de données. Dans ce cas, le logiciel de client de base de données inclut normalement tous les pilotes de base de données requis ou vous proposent une option au cours de l'installation pour sélectionner les pilotes et les composants que vous souhaitez installer. En règle générale, le logiciel de client de base de données consiste en des services d'administration et de configuration utilisés pour simplifier l'administration de base de données et la connectivité, et comporte une documentation expliquant comment installer et configurer le client de base de données et tous ses composants.

La configuration correcte du client de base de données est une étape cruciale pour la mise en place d'une connexion réussie à la base de données. Avant d'installer et d'utiliser votre logiciel de client de base de données, nous vous recommandons vivement de lire attentivement les instructions d'installation et de configuration du client de base de données, ceux-ci peuvent varier pour chaque version de base de données et pour chaque version Windows.

Afin de comprendre les possibilités et les limites de chaque technologie d'accès de données, pour chaque type de base de données, veuillez vous référer à la documentation de ce produit de base de données spécifique et testez également la connexion en comparaison avec votre environnement spécifique. Pour éviter les problèmes de connectivité les plus courants, veuillez consulter les notes et recommandations générales suivantes :

- Certains fournisseurs ADO.NET ne sont pas pris en charge ou ont une prise en charge limitée. Voir Notes de prise en charge ADO.NET.
- Lors de l'installation d'un pilote de base de données, nous vous recommandons d'en utiliser un qui ait la même plate-forme que l'application Altova (32-bit ou 64-bit). Par exemple, si vous utilisez une application Altova 32-bit sur un système d'exploitation 64-bit, installez le pilote 32-bit et configurez votre connexion de base de données à l'aide du pilote 32-bit, voir aussi Consulter les pilotes ODBC disponibles.
- Lors de la configuration d'une source de données ODBC, il est recommandé de créer le non de source de données (DSN) en tant que System DSN au lieu de l'User DSN. Pour plus d'informations, voir Configurer une connexion ODBC.
- Lors de la configuration d'une source de données JDBC, veuillez vous assurer que le JRE (Java Runtime Environment) ou le Java Development Kit (JDK) soit installé et que la variable d'environnement CLASSPATH du système d'exploitation soit configurée. Pour plus d'informations, voir Configurer une connexion JDBC Connection.

- En ce qui concerne les instructions et les détails de prise en charge de tout pilote ou logiciel de client de base de données que vous installez à partir d'un fournisseur de base de données, contrôlez la documentation fournie avec le paquet d'installation.

Base de données	Interface	Drivers
Firebird	ADO.NET	Fournisseur de données Firebird ADO.NET (https://www.firebirdsql.org/en/additional-downloads/)
	JDBC	Pilote Firebird JDBC (https://www.firebirdsql.org/en/jdbc-driver/)
	ODBC	Pilote Firebird ODBC (https://www.firebirdsql.org/en/odbc-driver/)
IBM DB2	ADO	Fournisseur IBM OLE DB pour DB2
	ADO.NET	Fournisseur IBM Data Server pour .NET
	JDBC	Pilote IBM Data Server pour JDBC et SQLJ
	ODBC	Pilote IBM DB2 ODBC
IBM DB2 for i	ADO	• Fournisseur IBM DB2 pour i5/OS IBMDA400 OLE DB • Fournisseur IBM DB2 pour i5/OS IBMDARLA OLE DB • Fournisseur IBM DB2 pour i5/OS IBMDASQL OLE DB
	ADO.NET	Fournisseur .NET Framework Data pour IBM i
	JDBC	Pilote IBM Toolbox pour Java JDBC
	ODBC	Pilote iSeries Access ODBC
IBM Informix	ADO	Fournisseur IBM Informix OLE DB
	JDBC	Pilote IBM Informix JDBC
	ODBC	Pilote IBM Informix ODBC
Microsoft Access	ADO	• Fournisseur Microsoft Jet OLE DB • Fournisseur Microsoft Access Database Engine OLE DB
	ADO.NET	Fournisseur de données .NET Framework pour OLE DB
	ODBC	• Pilote Microsoft Access
MariaDB	ADO.NET	En défaut d'un connecteur .NET dédié pour MariaDB, utiliser **Connector/NET** pour MySQL (https://dev.mysql.com/downloads/connector/net/).
	JDBC	Connecteur MariaDB /J (https://downloads.mariadb.org/)
	ODBC	Connecteur MariaDB /ODBC (https://downloads.mariadb.org/)
Microsoft SQL Server	ADO	• Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL) • Fournisseur Microsoft OLE DB pour SQL Server (SQLOLEDB) • SQL Server Native Client (SQLNCLI)

© 2016-2022 Altova GmbH

Base de données	Interface	Drivers
	ADO.NET	• .NET Framework Fournisseur de données pour SQL Server • .NET Framework Fournisseur de données pour OLE DB
	JDBC	• Pilote Microsoft JDBC pour SQL Server (https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server)
	ODBC	• Pilote ODBC pour Microsoft SQL Server (https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server)
MySQL	ADO.NET	Connecteur/NET (https://dev.mysql.com/downloads/connector/net/)
	JDBC	Connecteur/J (https://dev.mysql.com/downloads/connector/j
	ODBC	Connecteur/ODBC (https://dev.mysql.com/downloads/connector/odbc)
Oracle	ADO	• Fournisseur Oracle pour OLE DB • Fournisseur Microsoft OLE DB pour Oracle
	ADO.NET	Fournisseur de données Oracle pour .NET (http://www.oracle.com/technetwork/topics/dotnet/index-085163.html)
	JDBC	• JDBC Thin Driver • Pilote JDBC Oracle Call Interface (OCI) Ces pilotes sont généralement installés pendant l'installation de votre client de base de données Oracle. Connectez-vous par le biais du pilote OCI Driver (par le Thin Driver) si vous utilisez le composant Oracle XML DB.
	ODBC	• Microsoft ODBC pour Oracle • Le pilote Oracle ODBC (généralement installé pendant l'installation de votre client de base de données Oracle)
PostgreSQL	JDBC	Pilote PostgreSQL JDBC (https://jdbc.postgresql.org/download.html)
	ODBC	psqlODBC (https://odbc.postgresql.org/)
	Native Connection	Disponible. Il n'est pas nécessaire d'installer des pilotes si vous utilisez une connexion native.
Progress OpenEdge	JDBC	Connecteur JDBC (https://www.progress.com/jdbc/openedge)
	ODBC	Connecteur ODBC (https://www.progress.com/odbc/openedge)
SQLite	Native Connection	Disponible. Il n'est pas nécessaire d'installer des pilotes si vous utilisez une connexion native.
Sybase	ADO	Fournisseur Sybase ASE OLE DB

Base de données	Interface	Drivers
	JDBC	jConnect™ pour JDBC
	ODBC	Pilote Sybase ASE ODBC
Teradata	ADO.NET	Fournisseur de données .NET pour Teradata (https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata)
	JDBC	Pilote JDBC Teradata (https://downloads.teradata.com/download/connectivity/jdbc-driver)
	ODBC	Pilote ODBC Teradata pour Windows (https://downloads.teradata.com/download/connectivity/odbc-driver/windows)

19.1.3 Configurer une connexion ADO

Microsoft ActiveX Data Objects (ADO) est une technologie d'accès aux données qui vous permet de vous connecter à une variété de sources de données par le biais de OLE DB. OLE DB est une interface alternative à ODBC ou JDBC; elle fournit un accès uniforme aux données dans un environnement COM (Component Object Model). ADO est un précurseur du plus nouvel ADO.NET et l'une des manières possibles de se connecter aux bases de données natives de Microsoft comme Microsoft Access ou SQL Server, bien que vous puissiez aussi l'utiliser pour d'autres sources de données.

Ce qui est important, c'est que vous pouvez choisir entre de multiples fournisseurs ADO, et certains doivent être téléchargés et installés sur votre poste de travail avant de pouvoir les utiliser. Par exemple, pour se connecter au serveur SQL, les fournisseurs ADO suivants sont disponibles :

- Microsoft OLE DB *Driver* pour SQL Server (MSOLEDBSQL)
- Microsoft OLE DB *Provider* pour SQL Server (SQLOLEDB)
- SQL Server Native Client (SQLNCLI)

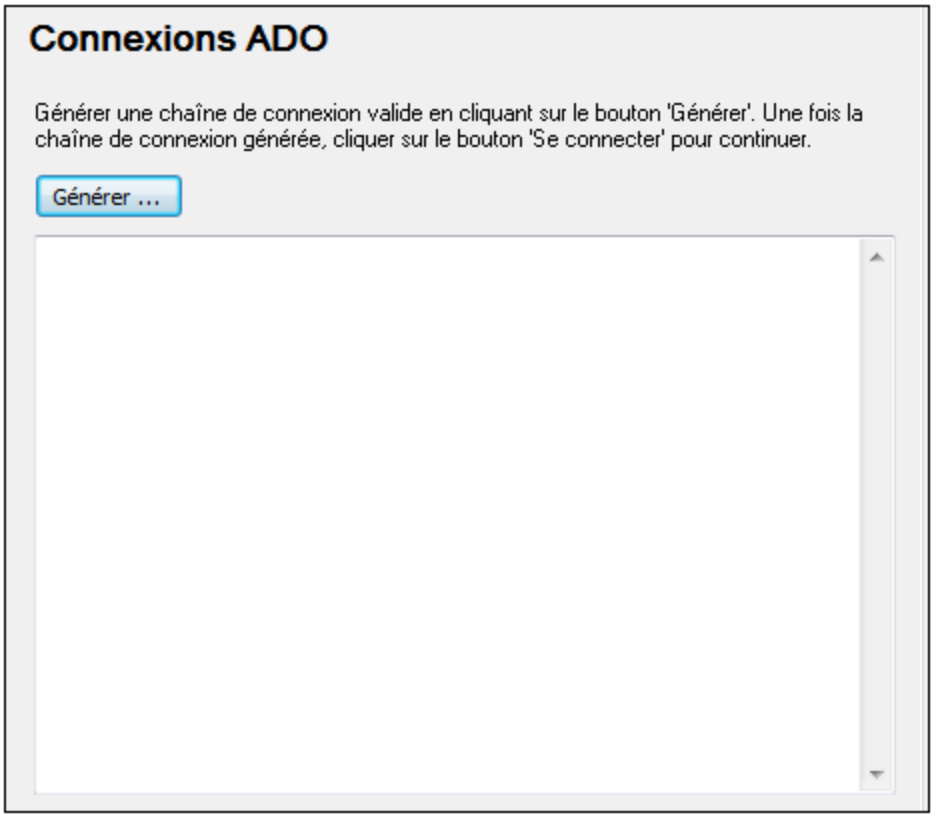
De la liste des fournisseurs ci-dessus, nous recommandons le MSOLEDBSQL; vous pouvez le télécharger depuis https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15. Veuillez noter qu'il doit correspondre avec la plate-forme de XMLSpy (32-bit ou 64-bit). Les fournisseurs SQLOLEDB et SQLNCLI sont considérés être dépréciés et ne sont donc pas recommandés.

Microsoft OLE DB Provider for SQL Server (SQLOLEDB) est connu pour avoir des soucis avec la liaison de paramètre de requêtes complexes comme Common Table Expressions (CTE) et des déclarations imbriquées SELECT.

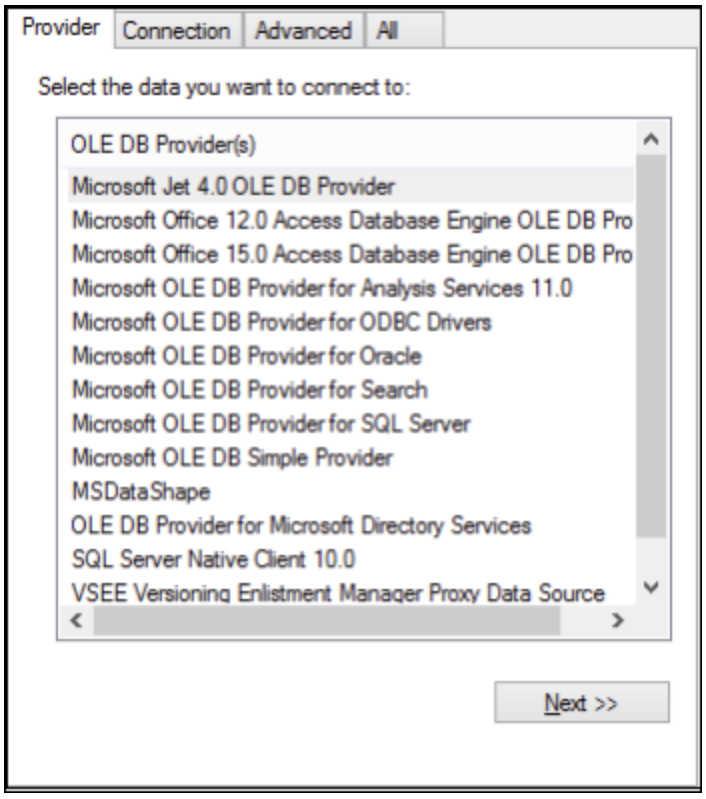
Pour configurer une connexion ADO :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **ADO Connections**.

© 2016-2022 Altova GmbH



3. Cliquez sur **Générer**.



4. Sélectionnez le fournisseur de données par le biais duquel vous souhaitez vous connecter. La table ci-dessous montre quelques scénarios communs.

Pour se connecter à cette base de données...	**Utiliser ce fournisseur...**
Microsoft Access	• **Fournisseur Microsoft Office Access Database Engine OLE DB** (recommandé) • **Microsoft Jet OLE DB Provider** Si le **Microsoft Office Access Database Engine OLE DB Provider** n'est pas disponible dans la liste, assurez-vous d'avoir installé soit Microsoft Access ou Microsoft Access Database Engine Redistributable (https://www.microsoft.com/en-us/download/details.aspx?id=54920) sur votre ordinateur.
Serveur SQL	• **Microsoft OLE DB Driver pour SQL Server (MSOLEDBSQL)** - c'est le fournisseur OLE DB recommandé. Si vous voulez que ce fournisseur apparaisse dans la liste, vous devez le télécharger depuis https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15 et l'installer. • **Microsoft OLE DB Provider pour SQL Server (OLEDBSQL)**

 © 2016-2022 Altova GmbH

Pour se connecter à cette base de données...	Utiliser ce fournisseur...
	• **SQL Server Native Client (SQLNCLI)**
Autre base de données	Sélectionnez le fournisseur applicable à votre base de données. Si un fournisseur OLE DB de votre base de données n'est pas disponible, veuillez installer le pilote requis depuis le distributeur de base de données (voir Aperçu des pilotes de base de données). En alternative, vous pouvez établir une connexion ADO.NET, ODBC ou JDBC. Si le système d'exploitation comporte un pilote ODBC pour la base de données requise, vous pouvez également utiliser le **Fournisseur Microsoft OLE DB pour les Pilotes ODBC** ou préférablement opter pour une connexion ODBC.

5. Après avoir sélectionné le fournisseur de votre choix, cliquez sur **Suivant** et achever l'assistant.

Les étapes suivantes de l'assistant sont spécifiques au fournisseur que vous avez choisi. En ce qui concerne SQL Server, vous devrez fournir ou sélectionner le nom d'hôte du serveur de base de données, la méthode d'authentification, le nom de la base de données ainsi que le nom d'utilisateur et le mot de passe de la base de données. Pour un exemple, consultez Se connecter à Microsoft SQL Server (ADO). En ce qui concerne Microsoft Access, vous serez invité à parcourir ou fournir le chemin du fichier de base de données. Pour un exemple, consultez Se connecter à Microsoft Access (ADO).

La liste complète des propriétés d'initialisation (paramètres de connexion) est disponible dans l'onglet **All** de la boîte de dialogue de connexion - ces propriétés varient dépendant du fournisseur choisi et doivent éventuellement être explicitement définies de telle manière que la connexion soit possible. Les prochaines sections fournissent des orientations sur la configuration des propriétés d'intitialisation de base pour les bases de données Microsoft Access et SQL Server :

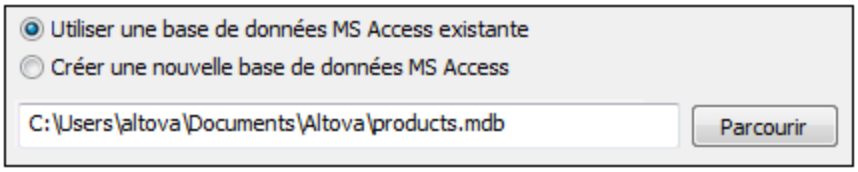
- Configurer les propriétés de liaison de données SQL Server
- Configurer les propriétés de liaison de données Microsoft Access

19.1.3.1 Se connecter à une base de données Microsoft Access existante

Cette approche est convenable si vous souhaitez vous connecter à une base de données Microsoft Access qui n'est pas protégée par un mot de passe. Si la base de données est protégée par un mot de passe, configurez le mot de passe de la base de données tel qu'indiqué dans Se connecter à Microsoft Access (ADO).

Pour se connecter à une base de données Microsoft Access existante :

1. Exécuter l'assistant de connexion de base de données (voir Lancer l'Assistant de la connexion de la base de données).
2. Choisir **Microsoft Access (ADO)**, puis cliquer sur **Suivant**.



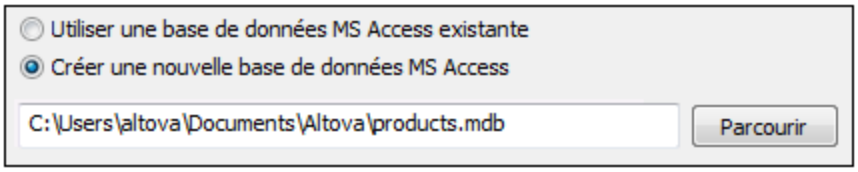
3. Choisir **Utiliser une base de données MS Access existante**.
4. Cherchez le fichier de base de données ou saisissez le chemin y menant (soit relatif, soit absolu).
5. Cliquez sur **Connexion**.

19.1.3.2 Créer une nouvelle base de données Microsoft Access

En alternative à la connexion à un fichier de base de données existant, vous pouvez créer un nouveau fichier de base de données Microsoft Access (.accdb, .mdb) et vous y connecter même si Microsoft Access n'est pas installé sur l'ordinateur. Le fichier de base de données créé par XMLSpy est vide. Pour créer la structure de base de données requise, utilisez Microsoft Access ou un outil comme DatabaseSpy (http://www.altova.com/databasespy.html).

Pour créer une nouvelle base de données Microsoft Access :

1. Exécutez l'assistant de connexion de base de données (voir Lancer l'Assistant de la connexion de la base de données).
2. Sélectionnez **Microsoft Access (ADO)**, puis cliquez sur **Suivant**.



3. Sélectionnez **Créer une nouvelle base de données MS Access**, puis saisissez le chemin (soit relatif ou absolu) du fichier de base de données à créer (par exemple, **c:\users\public\products.mdb**). En alternative, cliquez sur **Parcourir** pour choisir un dossier, saisissez le nom du fichier de base de données dans la fenêtre "Nom du fichier" (par exemple, **products.mdb**), et cliquez sur **Enregistrer**.

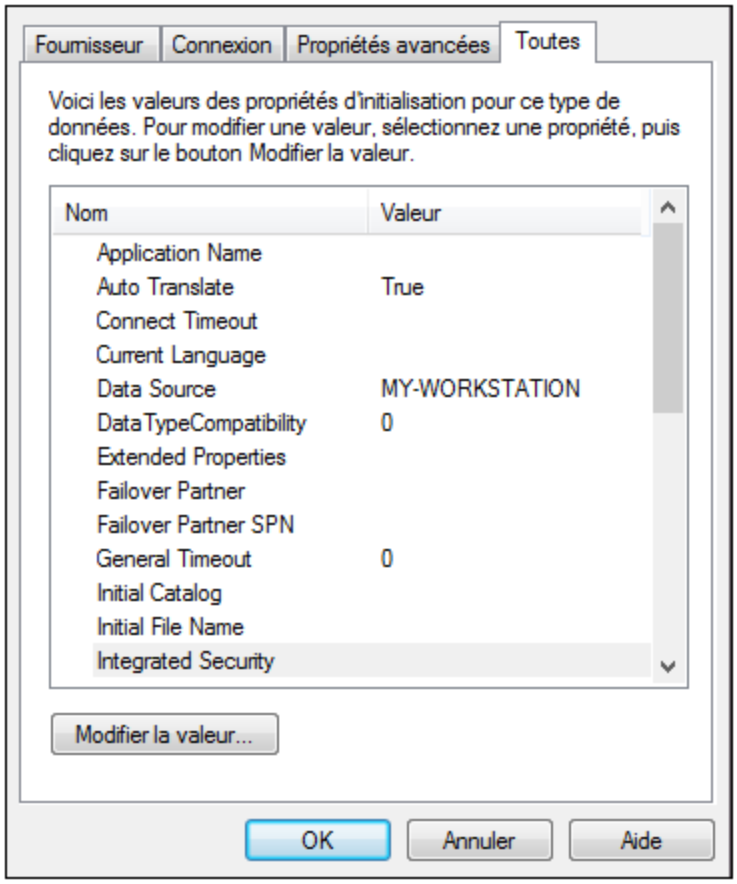
> **Notes**
>
> - Assurez-vous que vous possédez des permissions d'écriture pour le dossier dans lequel vous souhaitez créer le fichier de base de données.
> - Le nom de fichier de base de données doit présenter l'extension **.accdb** ou **.mdb**.

4. Cliquez sur **Se connecter**.

 © 2016-2022 Altova GmbH

19.1.3.3 Configurer les propriétés de liaison de données SQL Server

Lorsque vous vous connectez à une base de données Microsoft SQL Server par le biais d'ADO, veuillez vous assurer que les liaisons de données suivantes sont configurées dans l'onglet **Toutes** de la fenêtre de dialogue Propriétés des liaisons de données.

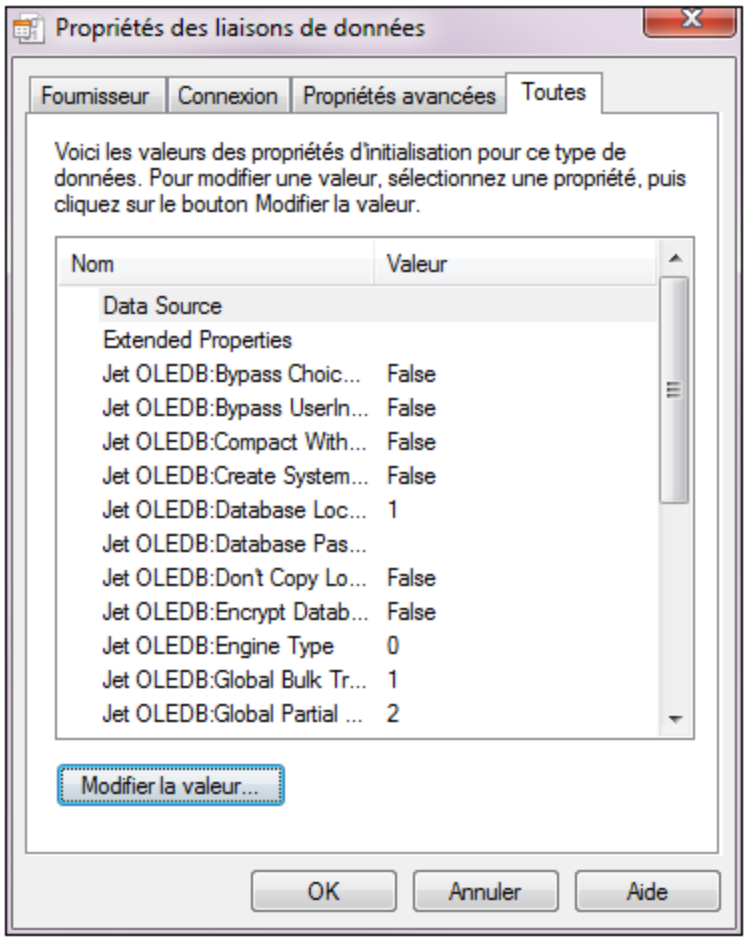


Fenêtre de dialogue Propriétés des liaisons de données

Propriété	Notes
Sécurité intégrée	Si vous avez sélectionné le fournisseur de données **SQL Server Native Client** dans l'onglet **Fournisseur**, configurez cette propriété sur un caractère d'espace.
Persister les infos de sécurité	Configurer cette propriété sur **True**.

19.1.3.4 Configurer les propriétés de liaison de données Microsoft Access

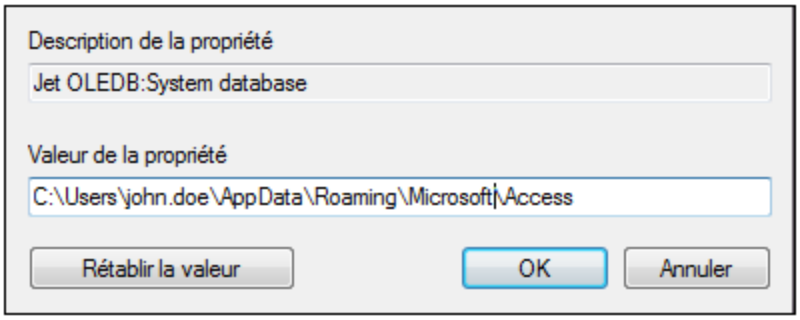
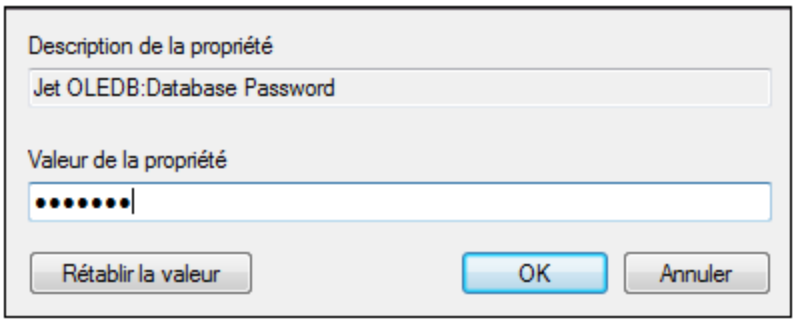
Lorsque vous vous connectez à une base de données Microsoft ADO, veuillez vous assurer que les propriétés suivantes sont configurées correctement dans l'onglet **Toutes** de la fenêtre de dialogue Propriétés des liaisons de données.



Fenêtre de dialogue Propriétés des liaisons de données

Propriété	Notes
Source de données	Cette propriété stocke le chemin vers le fichier de base de données Microsoft Access. Afin d'éviter des problèmes de connectivité, il est recommandé d'utiliser le format de chemin UNC (Universal Naming Convention), par exemple : `\\anyserver\share$\filepath`
Jet OLEDB : Base de données du système	Cette propriété stocke le chemin vers le fichier d'information du groupe de travail. Vous devrez éventuellement régler la valeur de cette propriété explicitement avant de pouvoir vous connecter à une base de données Microsoft Access. Si vous ne pouvez pas vous connecter en raison d'une erreur de "fichier d'information de groupe de travail", localisez le fichier d'information de groupe de travail (**System.MDW**) concernant votre profil d'utilisateur et configurez la valeur de propriété sur le chemin du fichier **System.MDW**.

© 2016-2022 Altova GmbH

Propriété	Notes
	
Jet OLEDB: Mot de passe de la Base de données	Si la base de données est protégée par un mot de passe, configurez la valeur de cette propriété sur le mot de passe de la base de données. 

19.1.4 Configurer une connexion ADO.NET

ADO.NET est un ensemble de bibliothèques Microsoft .NET Framework conçues pour interagir avec des données, y compris des données provenant de bases de données. Afin de se connecter à une base de données depuis XMLSpy par le biais de ADO.NET, il est nécessaire d'avoir Microsoft .NET Framework 4 ou plus élevé. Comme affiché ci-dessous, vous vous connectez à une base de données par le biais d'ADO.NET en sélectionnant un fournisseur .NET et en fournissant un string de connexion.

Un fournisseur de données .NET est une collection de classes qui permet de se connecter à une source de données de type particulier (par exemple, un Serveur SQL Server, ou une base de données Oracle), d'exécuter des commandes, et d'en extraire des données. En d'autres termes, avec ADO.NET, une application comme XMLSpy interagit avec une base de données par le biais d'un fournisseur de données. Chaque fournisseur de données est optimisé pour travailler avec le type spécifique de sources de données pour lequel il a été conçu. Il existe deux types de fournisseurs .NET :

1. offert par défaut avec Microsoft .NET Framework.
2. offert par des revendeurs des principales bases de données, en tant qu'une extension du .NET Framework. Ce genre de fournisseurs ADO.NET doit être installé séparément et peut être généralement téléchargé depuis le site Internet du revendeur de base de données respectif.

Note : certains fournisseurs ADO.NET ne sont pas pris en charge ou ne le sont que de manière limitée. Voir Notes de prise en charge d'ADO.NET.

Pour configurer une connexion ADO.NET :

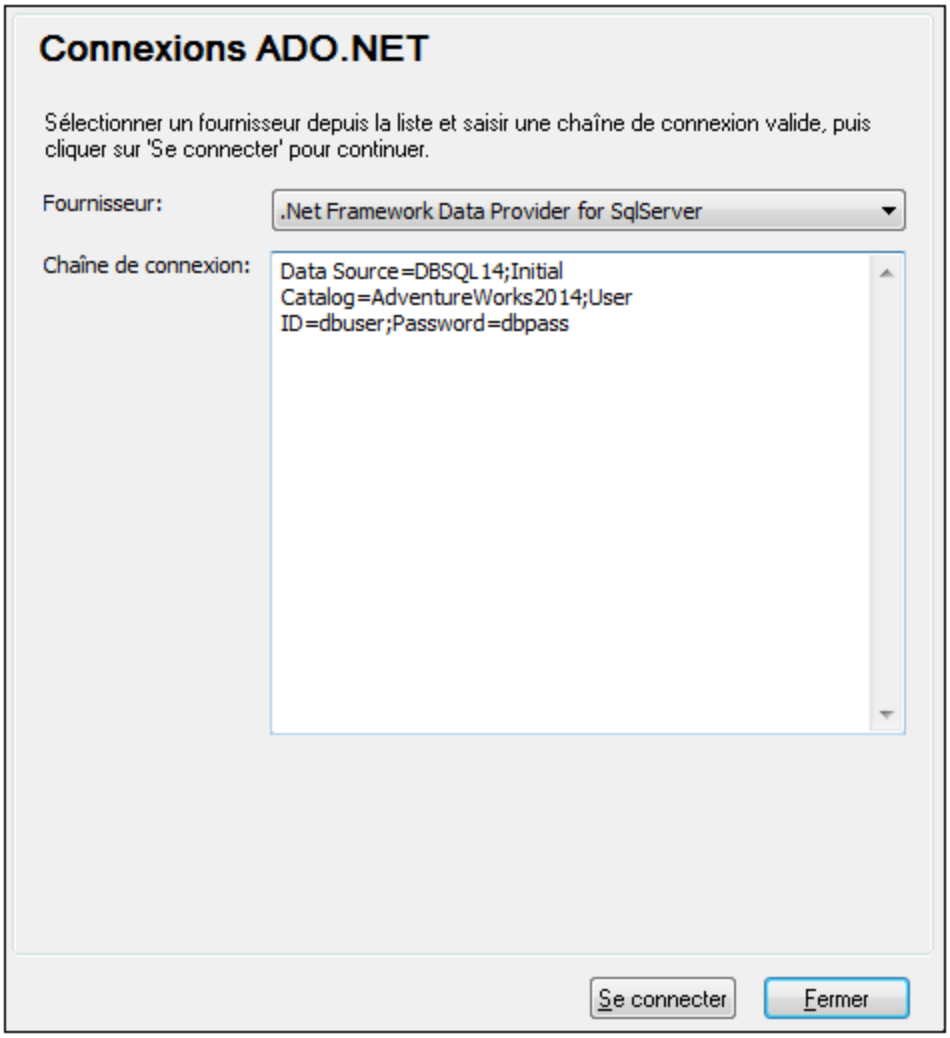
1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **ADO.NET Connections**.
3. Sélectionnez un fournisseur de données .NET depuis la liste.

   La liste de fournisseurs disponibles par défaut avec .NET Framework apparaît dans la liste des « Fournisseurs ». Les fournisseurs de données .NET spécifiques au revendeur apparaîtront uniquement dans la liste s'ils sont déjà installés sur votre système. Pour les rendre disponibles, les fournisseurs spécifiques au revendeur doivent être installés dans le GAC (Global Assembly Cache), en exécutant le fichier .msi ou .exe fourni par le revendeur de base de données.

4. Saisir un string de connexion à une base de données. Un string de connexion définit les informations de connexion à la base de données sous la forme de paramètres de connexion de paires de clés/valeurs contenues entre des points-virgules. Par exemple, une string de connexion comme **`Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass`** permet de se connecter à la base de données du serveur SQL `ProductsDB` sur le serveur `DBSQLSERV`, avec le nom d'utilisateur `dbuser` et le mot de passe `dbpass`. Vous pouvez créer une string de connexion en saisissant les paires de clé/valeur directement dans la fenêtre de dialogue « Connection String ». Une autre option est de la créer avec Visual Studio (voir Créer une string de connexion dans Visual Studio).

   La syntaxe de la string de connexion dépend du fournisseur sélectionné depuis la liste « Fournisseur ». Par exemple voir Strings de connexion ADO.NET d'échantillon.

© 2016-2022 Altova GmbH



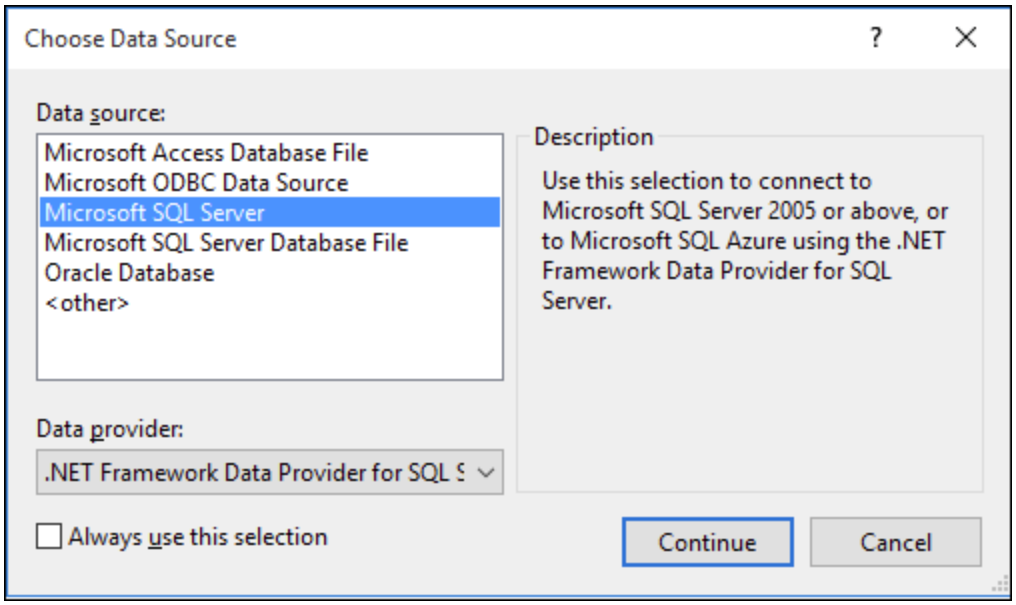
5. Cliquez sur **Se connecter**.

19.1.4.1 Créer une chaîne de connexion dans Visual Studio

Afin de vous connecter à une source de données en utilisant ADO.NET, il est nécessaire de disposer d'une chaîne de connexion de base de données valide. Les instructions suivantes vous montrent comment créer une chaîne de connexion depuis Visual Studio.

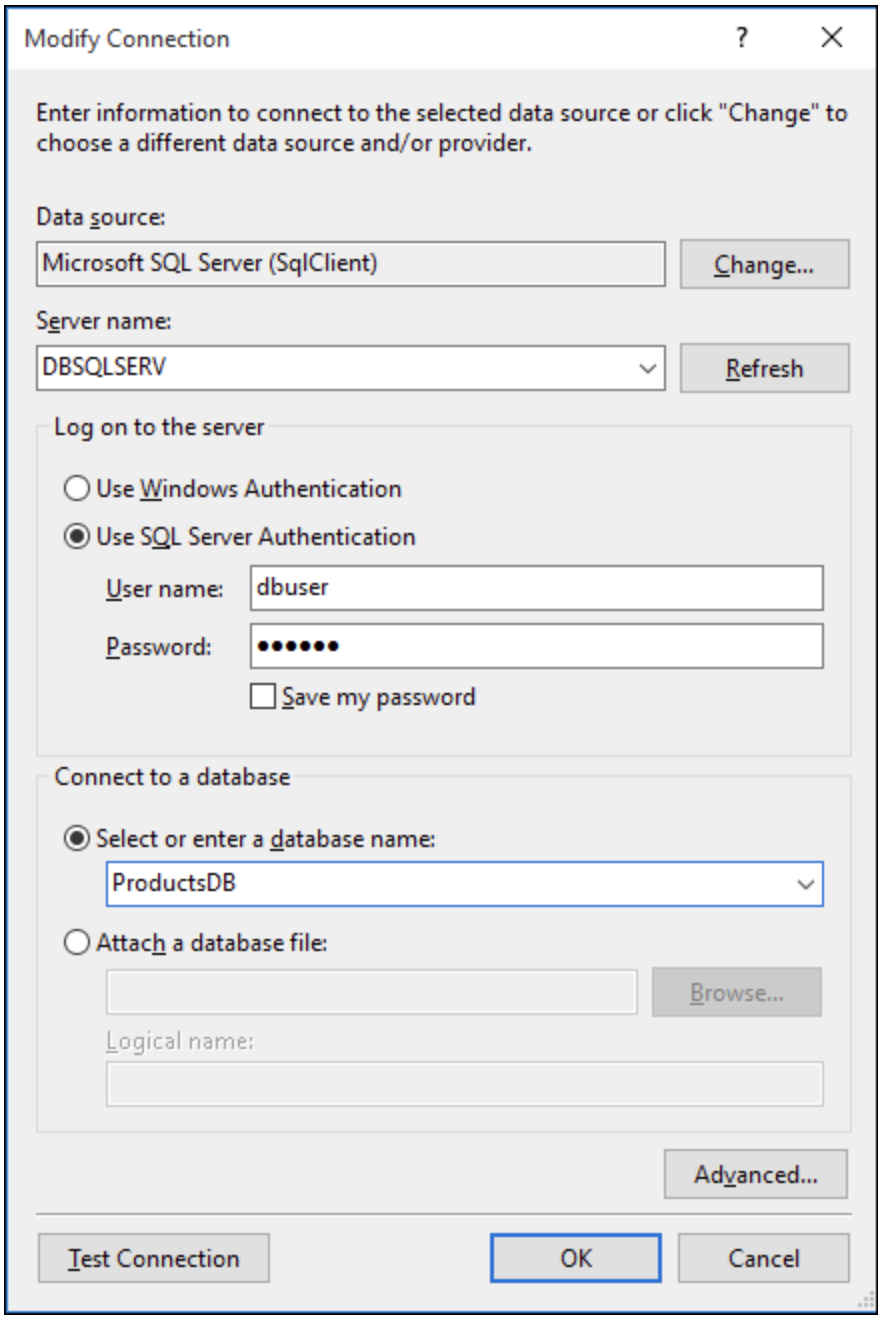
Pour créer une chaîne de connexion dans Visual Studio :

1. Dans le menu **Tools**, cliquer sur **Connect to Database**.
2. Choisir une source de données depuis la liste (dans cet exemple, Microsoft SQL Server). Le fournisseur de données est rempli automatiquement sur la base de votre choix.



3. Cliquer sur **Continue**.

© 2016-2022 Altova GmbH



4. Saisir le nom de l'hôte du serveur ainsi que le nom d'utilisateur et le mot de passe de la base de données. Dans cet exemple, nous nous connectons à la base de données `ProductsDB` sur le serveur `DBSQLSERV`, à l'aide de l'authentification de serveur SQL.
5. Cliquer sur **OK**.

Si la connexion à la base de données est établie avec succès, elle apparaît dans la fenêtre du Server Explorer. Vous pouvez afficher la fenêtre de Server Explorer à l'aide de la commande de menu **View | Server Explorer**. Pour obtenir la string de connexion à la base de données, cliquer avec la touche de droite sur la fenêtre Server Explorer et sélectionner **Properties**. La string de connexion est maintenant affichée dans la fenêtre Properties

de Visual Studio. Veuillez noter que, avant de coller la string dans la fenêtre « Connection String » de XMLSpy, vous devrez remplacer les astérisques (*) par le mot de passe.

19.1.4.2 Chaînes de connexion échantillons ADO.NET

Pour configurer une connexion ADO.NET, veuillez sélectionner un fournisseur ADO.NET depuis le dialogue de connexion à la base de données et saisir un string de connexion (voir aussi Configurer une connexion ADO.NET). Les strings de connexion échantillon ADO.NET pour les bases de données variées sont recensés ci-dessous, sous le fournisseur .NET pertinent.

Fournisseur de données .NET pour Teradata

Ce fournisseur peut être téléchargé depuis le site web Teradata (https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata). Un string de connexion d'échantillon ressemble à l'exemple suivant :

```
Data Source=ServerAddress;User Id=user;Password=password;
```

Fournisseur de données .NET Framework pour IBM i

Ce fournisseur est installé dans le *IBM i Access Client Solutions - Windows Application Package*. Un string de connexion d'échantillon ressemble à l'exemple suivant :

```
DataSource=ServerAddress;UserID=user;Password=password;DataCompression=True;
```

Pour plus d'informations, consulter le fichier d'aide ".Référence technique du fournisseur .NET" contenu dans le pack d'installation ci-dessus.

Fournisseur de données .NET Framework pour MySQL

Ce fournisseur peut être téléchargé depuis le site Internet MySQL (https://dev.mysql.com/downloads/connector/net/). Un string de connexion d'échantillon ressemble à l'exemple suivant :

```
Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;
```

Voir aussi : https://dev.mysql.com/doc/connector-net/en/connector-net-programming-connecting-connection-string.html

© 2016-2022 Altova GmbH

Fournisseur de données .NET Framework pour Serveur SQL

Un string de connexion d'échantillon ressemble à l'exemple suivant :

```
Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass
```

Voir aussi : https://msdn.microsoft.com/en-us/library/ms254500(v=vs.110).aspx

Fournisseur de données 10.1.2 IBM DB2 pour .NET Framework 4.0

```
Database=PRODUCTS;UID=user;Password=password;Server=localhost:50000;
```

Note : Ce fournisseur est généralement installé avec le paquet IBM DB2 Data Server Client. Si le fournisseur est absent de la liste des fournisseurs ADO.NET après l'installation du paquet IBM DB2 Data Server Client, veuillez vous référer à la note technique suivante : https://www-01.ibm.com/support/docview.wss?uid=swg21429586.

Voir aussi :
https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.1.0/com.ibm.swg.im.dbclient.adonet.ref.doc/doc/DB2ConnectionClassConnectionStringProperty.html

Fournisseur de données Oracle pour .NET (ODP.NET)

Le pack d'installation qui comprend le fournisseur ODP.NET peut être téléchargé depuis le site Internet Oracle (voir http://www.oracle.com/technetwork/topics/dotnet/downloads/index.html). Un string de connexion d'échantillon ressemble à l'exemple suivant :

```
Data Source=DSORCL;User Id=user;Password=password;
```

`DSORCL` est le nom de la source de données qui pointe vers un nom de service Oracle défini dans le fichier **tnsnames.ora**, comme décrit dans la section Connexion à Oracle (ODBC).

Pour vous connecter sans configurer un nom de service dans le fichier **tnsnames.ora**, utiliser un string comme :

```
Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=port)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));User
Id=user;Password=password;
```

Voir aussi : https://docs.oracle.com/cd/B28359_01/win.111/b28375/featConnecting.htm

19.1.4.3 Notes de prise en charge ADO.NET

La table suivante recense des pilotes de base de données ADO.NET connus qui ne sont pas pris en charge actuellement ou présentent une prise en charge limitée dans XMLSpy.

Base de données	Pilote	Notes de prise en charge
Toutes les bases de données	**.Net Framework Data Provider for ODBC**	Prise en charge limitée. Des problèmes connus existent avec des connexions Microsoft Access. Nous recommandons d'utiliser les connexions directes ODBC à la place.
	.Net Framework Data Provider for OleDb	Prise en charge limitée. Des problèmes connus existent avec des connexions Microsoft Access. Nous recommandons d'utiliser les connexions directes ADO à la place.
Firebird	**Firebird ADO.NET Data Provider**	Prise en charge limitée. Nous recommandons d'utiliser ODBC ou JDBC à la place.
Informix	**IBM Informix Data Provider for .NET Framework 4.0**	N'est pas pris en charge. Utiliser **DB2 Data Server Provider** à la place.
IBM DB2 for i (iSeries)	**.Net Framework Data Provider for i5/OS**	N'est pas pris en charge. Utiliser **.Net Framework Data Provider for IBM i** à la place, installé en tant que partie de *IBM i Access Client Solutions - Windows Application Package*.
Oracle	**.Net Framework Data Provider for Oracle**	Prise en charge limitée. Bien que ce pilote soit fourni avec le .NET Framework, Microsoft n'encourage pas son utilisation car il est obsolète.
PostgreSQL	**-**	Aucun pilote ADO.NET pour ce fournisseur n'est pris en charge. Utiliser une connexion native à la place.
Sybase	**-**	Aucun pilote ADO.NET pour ce fournisseur n'est pris en charge.

19.1.5 Configurer une connexion ODBC

ODBC (Open Database Connectivity) est une technologie d'accès de données largement utilisée qui vous permet de vous connecter à une base de données depuis XMLSpy. Elle peut être utilisée soit en tant que principal moyen pour se connecter à la base de données, ou comme alternative aux connexions pilotées native, OLE DB ou JDBC.

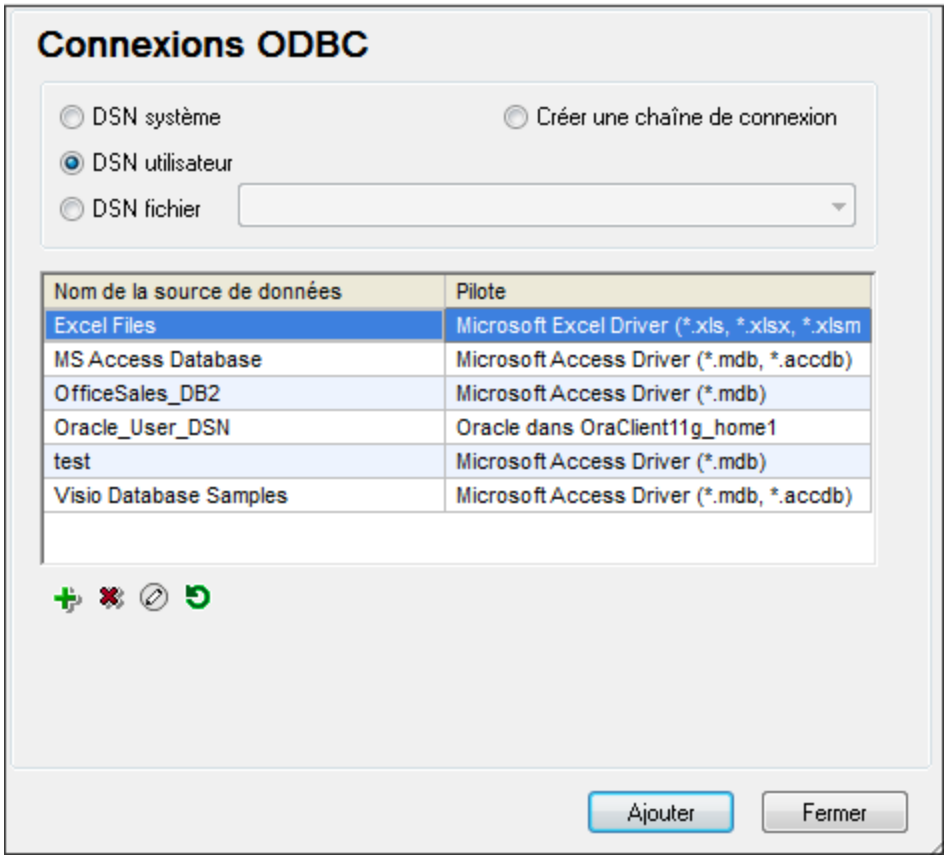
© 2016-2022 Altova GmbH

Pour se connecter à une base de données par le biais d'ODBC, vous devrez tout d'abord créer un nom de source de données (DSN) ODBC sur le système d'exploitation. Cette étape n'est pas nécessaire si le DSN a déjà été créé, par exemple par un autre utilisateur du système d'exploitation. Le DSN représente une manière uniforme de décrire la connexion de base de données à toute application client prise en charge ODBC du système d'exploitation, y compris XMLSpy. Les variables peuvent être de types suivants :

- DSN Système
- DSN Utilisateur
- DSN Fichier

Une source de données *System* est accessible par tous les utilisateurs avec des privilèges sur le système d'exploitation. Une source de données *Utilisateur* est disponible pour l'utilisateur l'ayant créée. Enfin, si vous créez un *DSN Fichier*, la source de données sera créée en tant que fichier avec une extension .dsn que vous pouvez partager avec d'autres utilisateurs, à condition qu'ils aient installé les pilotes utilisés par la source de données.

Tout DSN déjà disponible sur votre machine est regroupé par la boîte de dialogue de connexion de base de données lorsque vous cliquez sur **connexions ODBC** dans la boîte de dialogue de connexion ODBC.



Boîte de dialogue de connexions ODBC

Si un DSN vers la base de données requise n'est pas disponible, l'assistant de connexion de base de données de XMLSpy vous guidera pendant la création ; néanmoins, vous pouvez aussi le créer directement sur votre système d'exploitation Windows. Dans tous les cas, avant de commencer, assurez-vous que le pilote ODBC

applicable pour votre base de données se trouve dans la liste des pilotes ODBC disponibles pour le système d'exploitation (voir Consulter les pilotes ODBC disponibles).

Pour se connecter en utilisant un nouveau DSN :

1. Lancer l'assistant de connexion de base de données.
2. Dans le dialogue de connexion à la base de données, cliquez sur **Connexions ODBC**.
3. Sélectionnez un type de source de données (DSN Utilisateur, DSN Système, DSN Fichier).

   Pour créer un DSN Système, vous avez besoin de droits d'administration sur le système d'exploitation et XMLSpy doit être exécuté en tant qu'administrateur.

4. Cliquez sur **Ajouter** .
5. Sélectionnez un pilote, puis cliquez sur **DSN Utilisateur** ou **DSN Système** (dépendant du type de DSN que vous voulez créer). Si le pilote applicable pour votre base de données n'est pas dans la liste, téléchargez-le de la base de données et installez-le (voir Aperçu des pilotes de base de données).
6. Dans la boîte de dialogue qui apparaît, remplissez toute information de connexion spécifique au pilote pour compléter la configuration.

Pour que la connexion soit établie avec succès, vous devrez fournir le nom d'hôte (ou l'adresse IP) du serveur de base de données, ainsi que le nom d'utilisateur et le mot de passe de la base de données. Il peut y avoir d'autres paramètres de connexion optionnels - ces paramètres varient entre les fournisseurs de bases de données. Pour lire des informations plus détaillées concernant les paramètres spécifiques à chaque méthode de connexion, veuillez consulter la documentation du fournisseur du pilote. Pour se connecter en utilisant un DSN existant : Ceci vous permet de réutiliser les détails liés à la connexion de base de données à chaque fois que vous voulez vous connecter à la base de données. Veuillez noter que les DSN Utilisateur sont ajoutés à la liste des DSN Utilisateur alors que les DSN Système sont ajoutés à la liste des DSN Système.

Pour se connecter en utilisant un DSN existant :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **ODBC Connections**.
3. Sélectionnez le type de la source de données existante (DSN Utilisateur, DSN Système, DSN Fichier).
4. Cliquez sur l'enregistrement DSN existant, puis cliquez sur **Connexion**.

Pour construire une chaîne de connexion basée sur un fichier .dsn existant :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **ODBC Connections**.
3. Sélectionnez **Générer un string de connexion**, puis cliquez sur **Générer**.
4. Si vous souhaitez construire la chaîne de connexion en utilisant un DSN Fichier, cliquez sur l'onglet **Source de données de fichier**. Sinon, cliquez sur l'onglet **Source de données de l'appareil**. (Les DSN Système et DSN Utilisateur sont connus comme sources de données « Appareil ».)
5. Sélectionnez le fichier .dsn requis, puis cliquez sur **OK**.

© 2016-2022 Altova GmbH

Pour se connecter en utilisant une chaîne de connexion préparée :

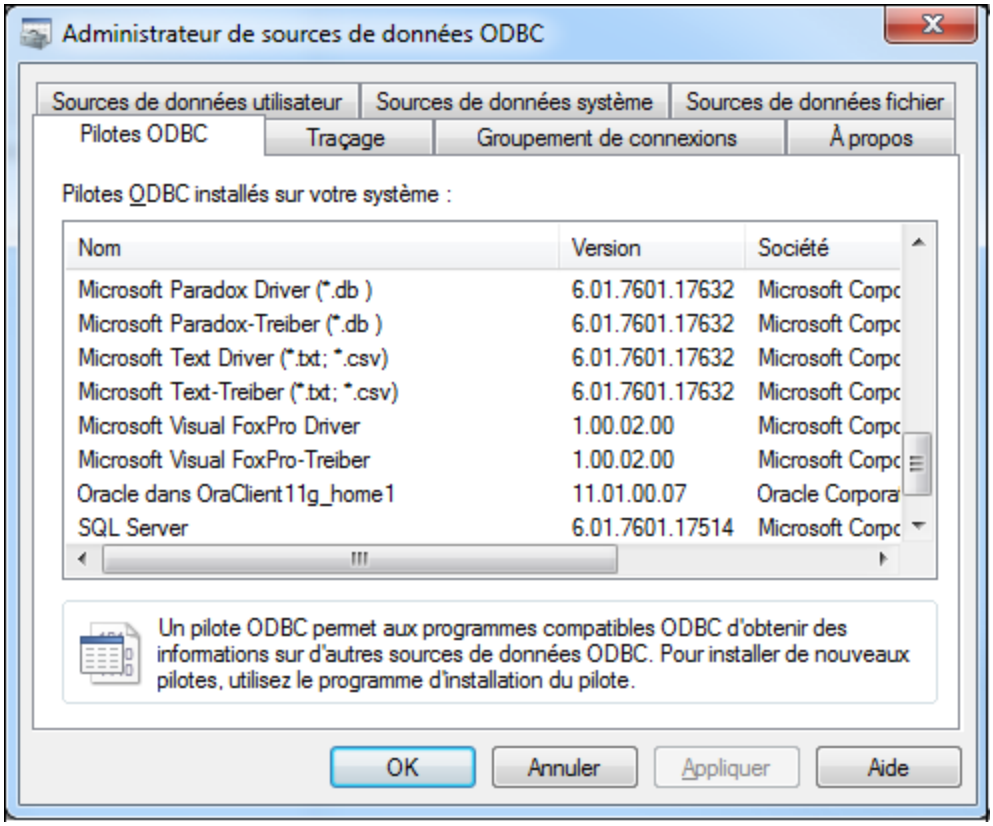
1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **ODBC Connections**.
3. Sélectionnez **Créer une chaîne de connexion**.
4. Collez la chaîne de connexion dans la fenêtre fournie, puis cliquez sur **Se connecter**.

19.1.5.1 Consulter les pilotes ODBC disponibles

Vous pouvez consulter les pilotes ODBC disponibles sur votre système d'exploitation dans l'Administrateur de source de données ODBC. Vous pouvez accéder à l'Administrateur de source de données ODBC (**Odbcad32.exe**) à partir du Panneau de configuration Windows, sous **Outils d'administration**. Dans les systèmes d'exploitation 64-bit, vous trouverez deux versions de ce programme d'exécution :

- La version 32-bit du fichier **Odbcad32.exe** est située dans le répertoire **C:\Windows\SysWoW64** (en partant du principe que **C:** est votre disque système).
- La version 64-bit du fichier **Odbcad32.exe** est située dans le répertoire **C:\Windows\System32**.

Tout pilote de base de données 32-bit installé sera visible dans la version 32-bit de l'Administrateur de source de données ODBC, alors que les pilotes 64-bit seront visibles dans la version 64-bit. Veuillez donc vous assurer que vous contrôlez les pilotes de base de données depuis les versions pertinentes de l'Administrateur de source de données ODBC.



Administrateur de source de données ODBC

Si le pilote de votre base de données cible n'existe pas dans la liste, ou si vous souhaitez ajouter un autre pilote, vous devrez le télécharger auprès du distributeur de base de données (voir Aperçu des pilotes de base de données). Une fois que le pilote ODBC est disponible sur votre système, vous êtes prêt à créer des connexions ODBC (voir Configurer une connexion ODBC).

19.1.6 Configurer une connexion JDBC

JDBC (Java Database Connectivity) est une interface d'accès à des bases de données, elle fait partie de la plateforme logicielle Java d'Oracle. Les connexions JDBC nécessitent généralement plus de ressources que les connexions ODBC, mais elles peuvent fournir des fonctions qui ne sont pas disponibles par le biais d'ODBC.

Prérequis

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) corresponde à celle du JRE/JDK.
- Les pilotes JDBC du distributeur de base de données doivent être installés. Il se peut que ce soient des pilotes JDBC installés en tant que partie de l'installation client de la base de données, ou de

 © 2016-2022 Altova GmbH

bibliothèques JDBC (.jar files) téléchargées séparément, si disponible ou pris en charge par la base de données, voir aussi Exemples de connexion à la base de données.
- La variable d'environnement `CLASSPATH` doit inclure le chemin du pilote JDBC (un ou plusieurs fichiers .jar) sur votre système d'exploitation Windows. Lorsque vous installez des clients de base de données, l'installateur peut configurer cette variable automatiquement. Voir aussi Configuration de CLASSPATH.

Se connecter à SQL Server par le biais de JDBC avec des identifiants Windows

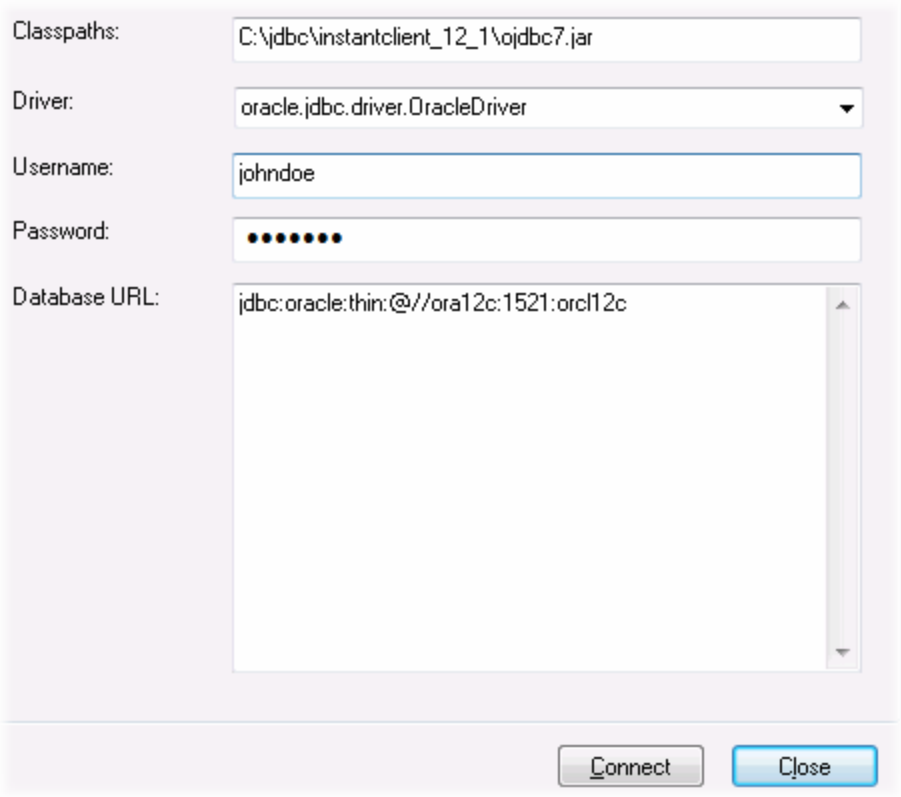
Si vous vous connectez à SQL Server par le biais de JDBC avec des identifiants Windows (sécurité intégrée), veuillez noter :

- Le fichier **sqljdbc_auth.dll** contenu dans le package du pilote JDBC doit être copié dans un répertoire qui se trouve dans la variable d'environnement du système PATH. Il y a deux fichiers de ce type, un pour la plateforme x86 et un pour la plateforme x64. Veuillez vous assurer que vous ajoutez à PATH celui qui correspond à votre plateforme JDK.
- Le string de connexion JDBC doit inclure la propriété **`integratedSecurity=true`**.

Pour plus d'informations, veuillez consulter la documentation *Microsoft JDBC driver for SQL Server*, https://docs.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url.

Configurer une connexion JDBC

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **Connexions JDBC**.
3. En option, saisir une liste de chemins d'accès de fichiers .jar séparées par des points-virgules dans la zone de texte « Chemins de classe ». Les bibliothèques .jar qui sont saisies ici seront chargées dans l'environnement en plus de celles déjà définies dans la variable d'environnement `CLASSPATH`. Une fois que vous avez terminé d'éditer la fenêtre de texte "Chemins de classe", tout pilote JDBC trouvé dans les librairies de source .jar sont automatiquement ajoutées à la liste « Pilote » (voir l'étape suivante).



4. À côté de « Pilote », choisir un pilote JDBC depuis la liste des pilotes ou saisir un nom de classe Java. Veuillez noter que cette liste contient tous les pilotes JDBC configurés par la variable d'environnement `CLASSPATH` (voir Configuration de CLASSPATH) ainsi que celles trouvées dans la zone de texte « Chemins de classe ».

   Les chemins de pilote JDBC définis dans la variable CLASSPATH, ainsi que dans tout chemin de fichier .jar saisis directement dans la boîte de dialogue de connexion de base de données sont tous fournis sur la Java Virtual Machine (JVM). La JVM décide ensuite quels pilotes utiliser pour établir une connexion. Il est recommandé de garder une trace des classes Java chargées dans la JVM pour ne pas créer de conflits de pilotes JDBC potentiels et éviter des résultats inattendus lors d'une connexion à la base de données.

5. Saisir le nom d'utilisateur et le mot de passe de la base de données dans les fenêtres correspondantes.
6. Dans la zone de texte URL de base de données, saisir le string URL de connexion JDBC dans le format spécifique à votre type de base de données. La table suivante décrit la syntaxe des URL de connexion JDBC (chaînes) pour les types de base de données communs.

Base de données	URL de connexion JDBC
Firebird	`jdbc:firebirdsql://<host>[:<port>]/<database path or alias>`
IBM DB2	`jdbc:db2://hostName:port/databaseName`

© 2016-2022 Altova GmbH

Base de données	URL de connexion JDBC
IBM DB2 for i	jdbc:as400://**[host]**
IBM Informix	jdbc:informix-sqli://**hostName**:**port**/**databaseName**:INFORMIXSERVER=**myserver**
MariaDB	jdbc:mariadb://**hostName**:**port**/**databaseName**
Microsoft SQL Server	jdbc:sqlserver://**hostName**:**port**;**databaseName**=name
MySQL	jdbc:mysql://**hostName**:**port**/**databaseName**
Oracle	jdbc:oracle:thin:@**hostName**:**port**:**SID** jdbc:oracle:oci:@//**hostName**:**port**:**service**
Oracle XML DB	jdbc:oracle:oci:@//**hostName**:**port**:**service**
PostgreSQL	jdbc:postgresql://**hostName**:**port**/**databaseName**
Progress OpenEdge	jdbc:datadirect:openedge://**host**:**port**;databaseName=**db_name**
Sybase	jdbc:sybase:Tds:**hostName**:**port**/**databaseName**
Teradata	jdbc:teradata://**databaseServerName**

Note : des variations de syntaxe pour les formats cités ci-dessus sont également possibles (par exemple, l'URL de base de données peut exclure le port ou peut inclure le nom d'utilisateur et le mot de passe de la base de données). Vérifiez la documentation du distributeur de la base de données pour plus de détails.

7. Cliquez sur **Se connecter**.

19.1.6.1 Configuration de CLASSPATH

La variable d'environnement CLASSPATH est utilisée par Java Runtime Environment (JRE) ou le Kit de développement Java (JDK) pour situer les classes Java et d'autres fichiers de ressource sur votre système d'exploitation. Lorsque vous vous connectez à une base de données par le biais de JDBC, cette variable doit être configurée pour contenir le chemin vers le pilote JDBC sur votre système d'exploitation, et, dans certains cas, le chemin vers des fichiers de librairie supplémentaires spécifiques au type de base de données que vous utilisez.

La table suivante recense les chemins de fichier échantillon qui sont généralement inclus dans la variable CLASSPATH. Attention ! Vous devrez éventuellement ajuster cette information sur la base de l'emplacement du pilote JDBC sur votre système, du nom de pilote JDBC, et de la version JRE/JDK présente sur votre système d'exploitation. Pour éviter tout problème de connectivité, contrôlez les instructions d'installation et les étapes de configuration de pré-installation ou de post-installation applicables au pilote JDBC installé sur votre système d'exploitation.

Base de données	Entrées CLASSPATH échantillons
Firebird	`C:\Program Files\Firebird\Jaybird-2.2.8-JDK_1.8\jaybird-full-2.2.8.jar`
IBM DB2	`C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar;C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar;`
IBM DB2 for i	`C:\jt400\jt400.jar;`
IBM Informix	`C:\Informix_JDBC_Driver\lib\ifxjdbc.jar;`
Microsoft SQL Server	`C:\Program Files\Microsoft JDBC Driver 4.0 for SQL Server\sqljdbc_4.0\enu\sqljdbc.jar`
MariaDB	**`<installation directory>`**`\mariadb-java-client-2.2.0.jar`
MySQL	`mysql-connector-java-`*`version`*`-bin.jar;`
Oracle	**`ORACLE_HOME`**`\jdbc\lib\ojdbc6.jar;`
Oracle (avec XML DB)	**`ORACLE_HOME`**`\jdbc\lib\ojdbc6.jar;`**`ORACLE_HOME`**`\LIB\xmlparserv2.jar;` **`ORACLE_HOME`**`\RDBMS\jlib\xdb.jar;`
PostgreSQL	**`<installation directory>`**`\postgresql.jar`
Progress OpenEdge	`%DLC%\java\openedge.jar;%DLC%\java\pool.jar;` Note : Si le Progress OpenEdge SDK est installé sur la machine, `%DLC%` est le répertoire où OpenEdge est installé.
Sybase	`C:\sybase\jConnect-7_0\classes\jconn4.jar`
Teradata	**`<installation directory>`**`\tdgssconfig.jar;`**`<installation directory>`**`\terajdbc4.jar`

- Un changement de la variable CLASSPATH peut influer sur le comportement des applications Java sur votre machine. Afin de comprendre les implications possibles avant de commencer, veuillez vous référer à la documentation Java.
- Les variables d'environnement peuvent être utilisateur ou système. Pour changer les variables d'environnement système, vous aurez besoin de droits administratifs sur le système d'exploitation.
- Après avoir changé la variable d'environnement, redémarrez les programmes en cours pour que la configuration puisse prendre effet. En alternative, déconnectez-vous ou redémarrez votre système d'exploitation.

Pour configurer CLASSPATH sur Windows 7 :

1. Ouvrir le menu **Démarrer** et cliquer avec le bouton de droite sur **Ordinateur**.
2. Cliquer sur **Propriétés**.
3. Cliquer sur **Configurations de système avancées**.
4. Dans l'onglet **Avancé**, cliquer sur **Variables d'environnement**,

© 2016-2022 Altova GmbH

5. Localiser la variable CLASSPATH sous les variables d'environnement utilisateur ou système, puis cliquer sur **Éditer**. Si la variable CLASSPATH n'existe pas, cliquer sur **Nouveau** pour la créer.
6. Éditer la valeur de variable pour inclure le chemin sur votre système d'exploitation où se trouve le pilote JDBC. Pour séparer le chemin de pilote JDBC des autres chemins qui pourraient déjà se trouver dans la variable CLASSPATH, utiliser un point-virgule en tant que séparateur (;).

Pour configurer CLASSPATH sur Windows 10 :

1. Cliquer sur la touche Windows, puis saisir **"variables d'environment"**.
2. Cliquer sur **Éditer les variables d'environnement de système**.
3. Cliquer sur **Variables d'environnement**.
4. Localiser la variable CLASSPATH sous les variables d'environnement utilisateur ou système, puis cliquer sur **Éditer**. Si la variable CLASSPATH n'existe pas, cliquer sur **Nouveau** pour la créer.
5. Éditer la valeur de variable pour inclure le chemin sur votre système d'exploitation où se trouve le pilote JDBC. Pour séparer le chemin de pilote JDBC des autres chemins qui pourraient déjà se trouver dans la variable CLASSPATH, utiliser un point-virgule en tant que séparateur (;).

19.1.7 Configurer une connexion PostgreSQL

Les connexions aux bases de données PostgreSQL peuvent être configurées soit en tant que connexions natives, soit en tant que connexions par le biais de ODBC, JDBC et d'autres pilotes. L'avantage d'une configuration native est qu'il n'est pas nécessaire d'installer d'autres pilotes sur votre système.

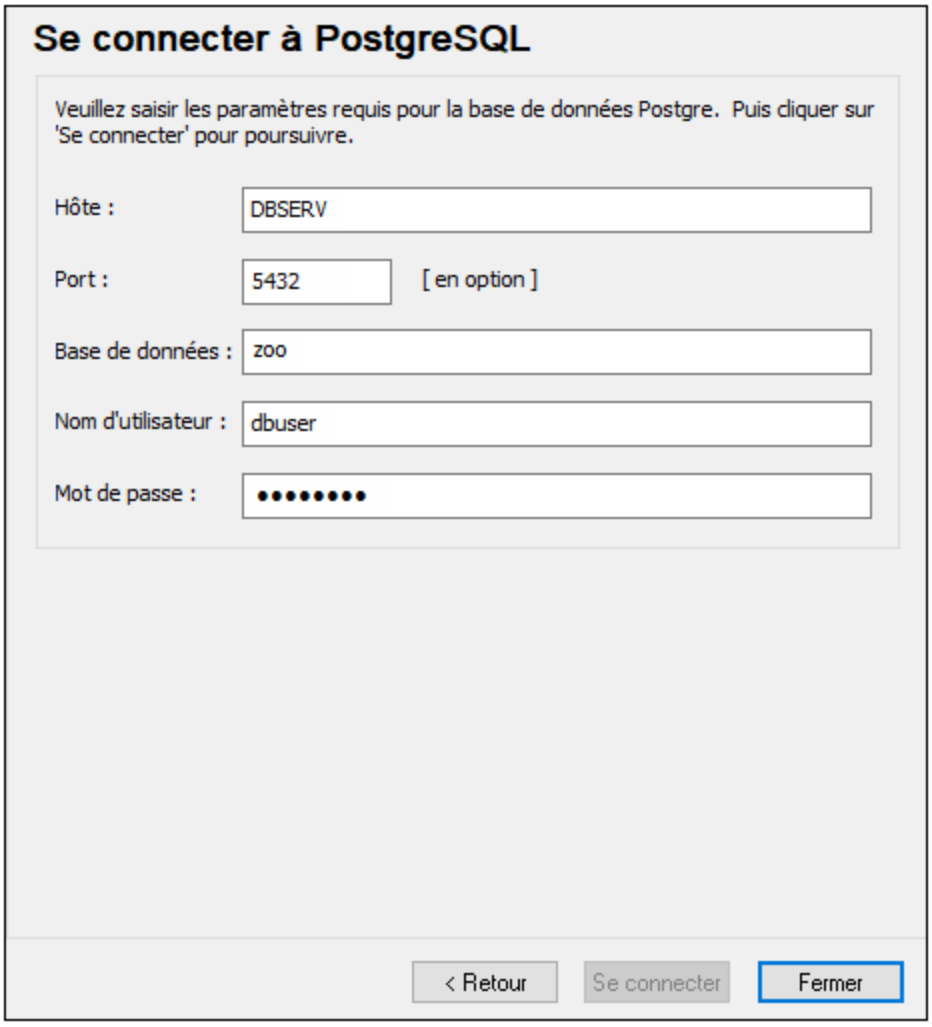
Si vous préférez établir une connexion par le biais d'un pilote non natif, voir les sections suivantes :

- Configurer une connexion JDBC
- Connexion à PostgreSQL (ODBC)

Sinon, si vous souhaitez configurer une connexion native à PostgreSQL, veuillez suivre les étapes suivantes. Avant de pouvoir continuer, il vous faudra les éléments suivants : nom d'hôte, port, nom de base de données, nom d'utilisateur et mot de passe.

Pour configurer une connexion PostgreSQL native :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **Connexions PostgreSQL**.
3. Saisir l'hôte (*localhost*, si PostgreSQL est exécuté sur la même machine), le port (en général 5432, cette étape est optionnelle), le nom de la base de données, le nom d'utilisateur et le mot de passe dans les fenêtres correspondantes.



4. Cliquez sur **Se connecter**.

Si le serveur de base de données PostgreSQL se trouve sur une machine différente, veuillez noter les points suivants :

- Le serveur de base de données PostgreSQL doit être configuré de manière à accepter les connexions provenant des clients. En particulier, le fichier **pg_hba.conf** doit être configuré de manière à permettre des connexions non locales. Ensuite, le fichier **postgresql.conf** doit être configuré de manière à écouter des adresses IP et des ports spécifiés. Pour plus d'information, consultez la documentation PostgreSQL (https://www.postgresql.org/docs/9.5/static/client-authentication-problems.html).
- Le serveur doit être configuré de manière à accepter les connexions sur le port désigné (en général 5432) par le biais du pare-feu. Par exemple, sur un serveur de base de données exécuté sur Windows, une règle peut être créée pour permettre des connexions sur le port 5432 par le biais du pare-feu, depuis **Panneau de configuration > Pare-feu Windows > Fonctions avancées > Règles entrantes**.

© 2016-2022 Altova GmbH

19.1.8 Configurer une connexion SQLite

SQLite est un type autonome de base de données sur base de fichier, idéal pour des scénarios mettant l'accent sur la portabilité et la simplicité de configuration. Puisque les bases de données SQLite sont prises en charge nativement par XMLSpy, vous n'aurez pas besoin d'installer des pilotes pour les connecter.

Notes de prise en charge de la base de données SQLite

- Sur Linux, le timeout pour l'exécution d'instruction n'est pas pris en charge pour les bases de données SQLite.
- La recherche plein texte n'est pas prise en charge pour les tables.
- SQLite permet des valeurs de types de données différents dans chaque ligne d'une table donnée. Toutes les valeurs traitées doivent être compatibles avec le type de colonne déclaré ; c'est pourquoi les valeurs non attendues peuvent être extraites et les erreurs d'exécution peuvent se produire si votre base de données SQLite a des valeurs de ligne qui ne sont pas les mêmes que le type de colonne déclaré.

Important

Il est recommandé de créer des tables avec le mot-clé `STRICT` afin de garantir un comportement plus prévisible de vos données. Autrement, les données ne sont pas écrites ou lues correctement lorsque les valeurs de différents types sont mélangées dans une colonne. Pour en savoir plus sur les tables STRICT, voir la documentation SQLite.

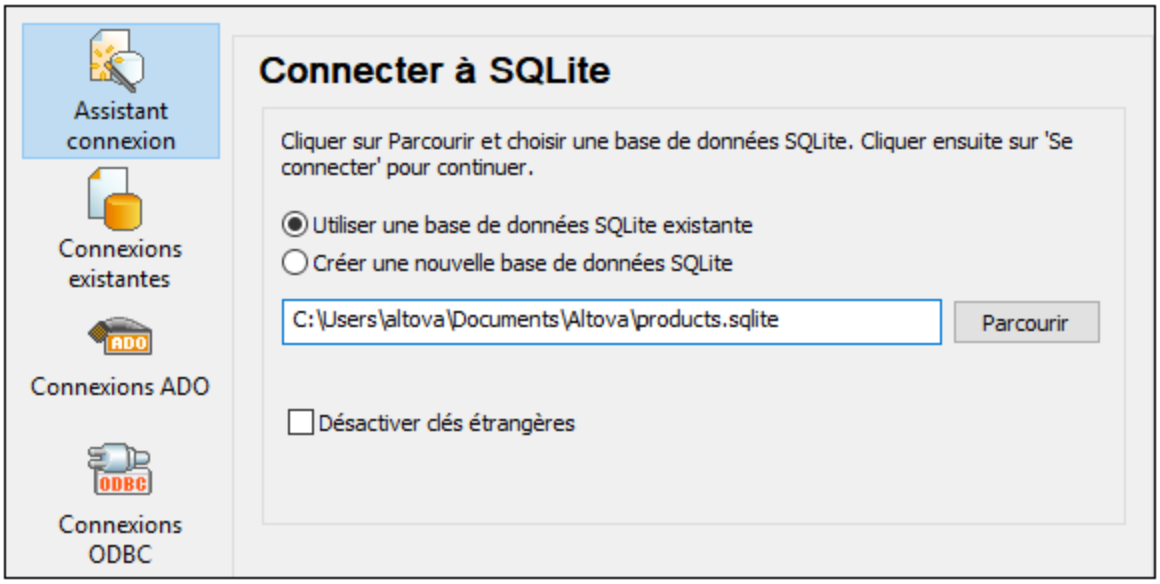
19.1.8.1 Connexion à une base de données SQLite existante

Vous pouvez vous connecter à une base de données SQLite existante soit en utilisant l'assistant de connexion, ou directement depuis Windows Explorer, en utilisant la commande **Ouvrir avec...**.

Utiliser l'Assistant de connexion

Pour se connecter à une base de données SQLite existante :

1. Exécuter l'assistant de connexion de base de données (voir Lancer l'Assistant de la connexion de la base de données).
2. Choisir **SQLite**, puis cliquer sur **Suivant**.



3. Choisir **Utiliser une base de données SQLite existante**, puis chercher le fichier de base de données SQLite, ou saisir le chemin (soit relatif ou absolu) à la base de données. La touche **Connexion** s'active une fois que vous avez saisi le chemin vers un fichier de base de données SQLite.
4. En option, choisir la case à cocher **Désactiver les clés étrangères**, voir Contraintes de clé étrangère.
5. Cliquer sur **Se connecter**.

Depuis Windows Explorer

Vous pouvez aussi ouvrir une base de données SQLite directement depuis Windows Explorer, comme suit :

Cliquer avec la touche de droite dans un fichier de base de données existant dans Windows Explorer et sélectionner **Ouvrir avec** depuis le menu contextuel.

1. Choisir **DatabaseSpy** depuis la liste des suggestions. Si cette suggestion n'est pas disponible, choisir **Choisir une autre appli**, et chercher le programme d'exécution de DatabaseSpy depuis le répertoire d'installation.
2. Si l'assistant de Connexion à la base de données, cliquer sur **Fermer** pour l'ignorer.

Vous pouvez aussi glisser/déposer le fichier de base de données dans DatabaseSpy si le second est déjà ouvert.

Notes :

- Vous pouvez aussi, en option, configurer DatabaseSpy depuis **Outils | Options** de manière à ne pas montrer l'assistant de connexion à la base de données à chaque fois que l'application démarre. Pour ce faire, décocher la case **Afficher créer un dialogue de connexion de base de données** dans le groupe de paramètres **Général**.
- Si DatabaseSpy est ouvert et qu'une source de données pour le même fichier de base de données existe déjà, elle sera réutilisée. En d'autres termes, une source de données dupliquée pour la même base de données ne sera pas créée. Cela vaut aussi bien lors de l'ouverture d'un fichier de base de données avec **Ouvrir avec** et si vous utilisez un glisser/déposer.

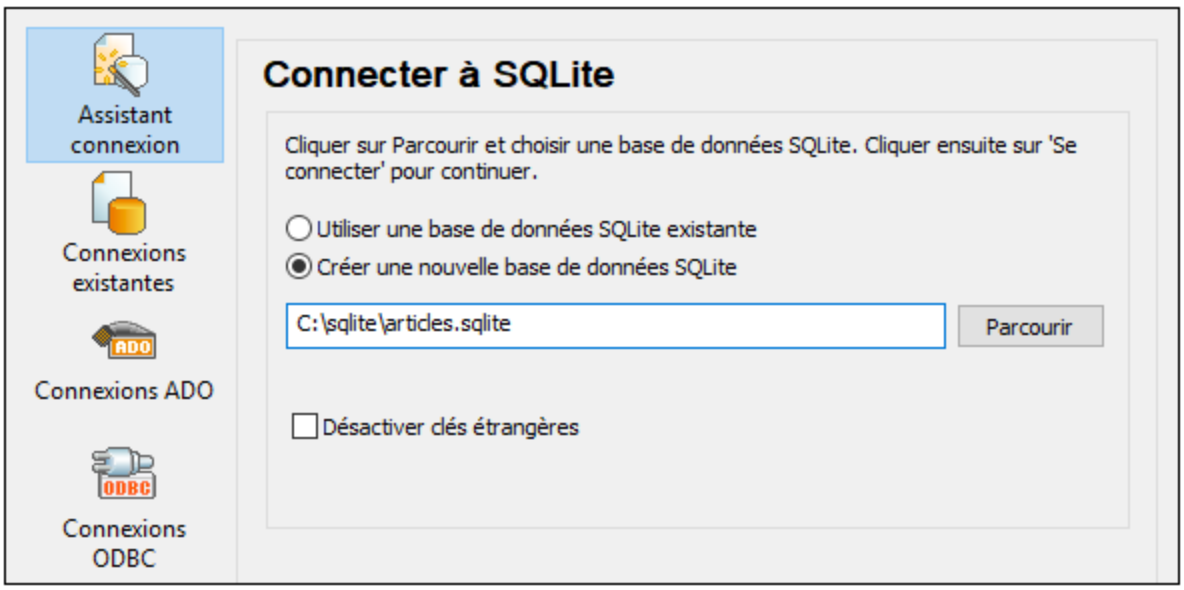
© 2016-2022 Altova GmbH

19.1.8.2 Créer une nouvelle base de données SQLite

Vous pouvez créer un nouveau fichier de base de données SQLite et vous y connecter, en tant qu'alternative à la connexion à un fichier de base de données existant. Le fichier de base de données créé par XMLSpy est vide ; utiliser les requêtes ou les scripts pour créer la structure de base de données requise et la remplir avec des données.

Pour créer une nouvelle base de données SQLite :

1. Lancer l'assistant de connexion de base de données (voir Lancer l'assistant de connexion de base de données).
2. Choisir **SQLite**, puis cliquer sur **Suivant**.



3. Sélectionner **Créer une nouvelle base de données SQLite**, puis saisir le chemin (soit relatif soit absolu) du fichier la base de données à créer (par exemple, **c:\users\public\products.sqlite**). En alternative, cliquer **Parcourir** pour sélectionner un dossier, saisir le nom du fichier de la base de données dans la fenêtre de texte "Nom de fichier" (par exemple, **products.sqlite**), et cliquer sur **Enregistrer**.

> Assurez-vous que vous détenez les droits d'écriture pour le dossier dans lequel vous souhaitez créer le ficher de base de données.

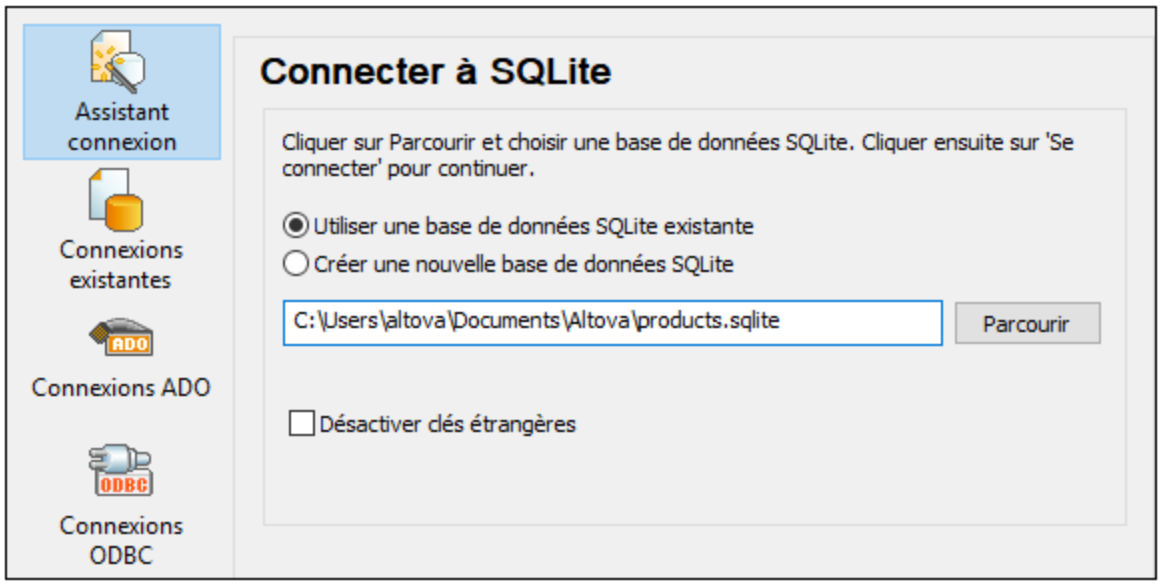
4. En option, choisir la case **Désactiver clés étrangères**, voir Contraintes de clé étrangère.
5. Cliquer sur **Se connecter**.

19.1.8.3 Contraintes de clé étrangère

Lorsque vous vous connectez à une base de données SQLite existante depuis XMLSpy, ou si vous en créez une nouvelle, les contraintes de clé étrangère sont activées par défaut. Elles aident à préserver l'intégrité des

données dans votre base de données. Par exemple, lorsque des clés étrangères sont activées, il n'est pas possible de supprimer un enregistrement d'une table si elle a des dépendances dans une autre table.

Dans certains cas, vous souhaiterez éventuellement contourner temporairement ce comportement et désactiver les clés étrangères, peut-être pour mettre à jour ou insérer plusieurs lignes de données sans recevoir d'erreurs de validation des données. Afin de désactiver explicitement des clés étrangères avant de vous connecter à la base de données SQLite, choisir l'option **Désactiver clés étrangères** disponible dans l'assistant de la base de données.



Page d'assistant "Se connecter à SQLite"

Lorsque les clés étrangères sont désactivées, vous pourrez effectuer des opérations par rapport aux données qui ne seraient habituellement pas possible en raison des contrôles de validation. Néanmoins, vous courez tout de même le risque d'introduire des données erronées dans la base de données ou de créer des lignes "orphelines". (Un exemple d'une ligne "orpheline" est une adresse dans la table "addresses" qui n'est liée à aucune personne dans la table "person" parce que la personne a été supprimée mais pas son adresse.)

19.1.9 Ressources globales

Après avoir créé une base de données en tant que ressource globale, ses détails de connexion sont stockés et peuvent être utilisés dans tous les produits Altova installés sur votre appareil.

Créer une base de données en tant que ressource globale

Pour créer une base de données en tant que ressource globale, suivez les étapes suivantes

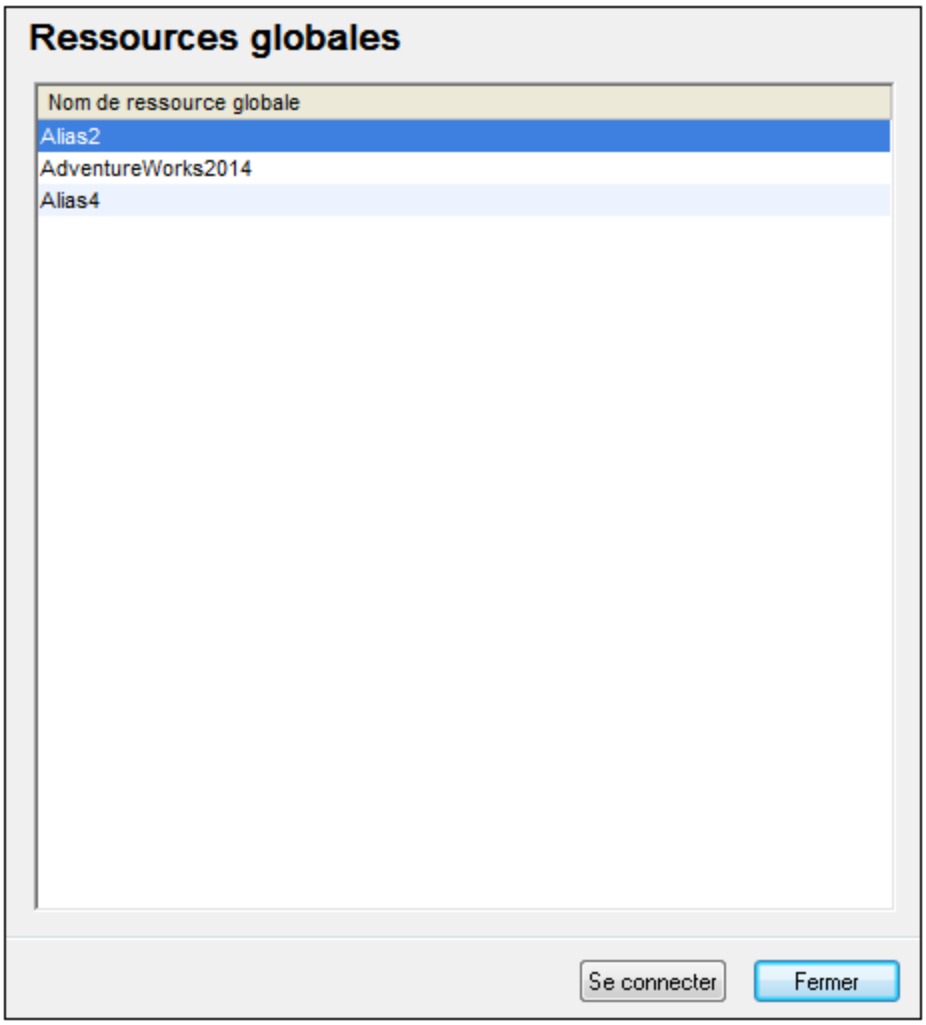
1. Dans le menu **Outils** de XMLSpy, cliquez sur les **Ressources globales**.
2. Cliquez sur **Ajouter**, puis cliquez sur Base de données.
3. Saisissez un nom pour la ressource globale dans le champ *Resource Alias*.
4. Cliquer sur **Choose Database**. L'assistant de connexion apparait.
5. Utiliser l'assistant de connexion pour ajouter une connexion de base de données, tel que décrit ci-dessus.

© 2016-2022 Altova GmbH

Utiliser une base de données ressource globale

Pour utiliser une base de données qui a été créée en tant que ressource globale (*voir-ci-dessus*), suivez les étapes suivantes :

1. Lancer l'assistant de connexion de base de données, tel que décrit ci-dessus.
2. Sélectionnez les Ressources globales. Toutes les bases de données qui ont été créées en tant que ressources globales seront recensées par leurs noms dans le volet des Ressources globales (*voir la capture d'écran ci-dessous*).



3. Sélectionnez la ressource globale de votre choix. Astuce : déplacez le curseur sur une ressource globale dans la liste afin de voir l'information sur les bases de données.

19.1.10 Exemples de connexion à la base de données

Cette section comprend des exemples de connexion à une base de données provenant de XMLSpy par le biais de ADO, ODBC ou JDBC. Les exemples de connexion ADO.NET sont recensés séparément, voir Chaînes de

connexion échantillons ADO.NET. Pour consulter des instructions concernant la mise en place d'une connexion native sur PostgreSQL et SQLite, voir Configurer une connexion PostgreSQL et Configurer une connexion SQLite, respectivement.

Noter :

- Les instructions peuvent différer si votre configuration Windows, l'environnement de réseau et le client de base de données ou le logiciel de serveur ne sont pas les mêmes que ceux décrits dans chaque exemple.
- En ce qui concerne la plupart des types de base de données, il est possible de se connecter en utilisant plus d'une technologie d'accès aux données (ADO, ADO.NET, ODBC, JDBC) ou pilote. La performance de la connexion de la base de données, ainsi que ses fonctions et ses limitations dépendront du pilote sélectionné, du logiciel client de base de données (le cas échéant), et de tout autre paramètre de connectivité que vous êtes susceptible d'avoir configuré outre XMLSpy.

19.1.10.1 Firebird (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données Firebird par le biais de JDBC.

Conditions préalables :

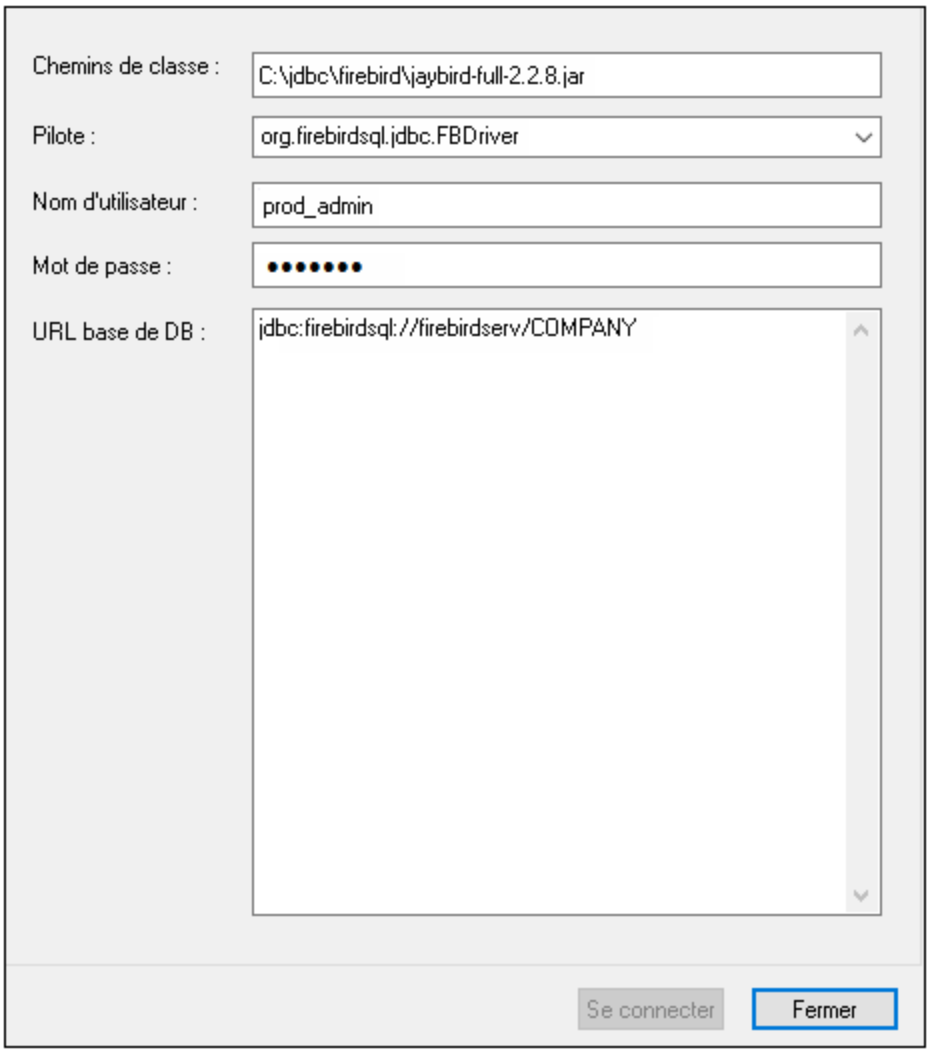
- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) corresponde à celle du JRE/JDK.
- Le pilote Firebird JDBC doit être disponible sur votre système d'exploitation (il prend la forme d'un fichier .jar qui offre une connectivité à la base de données). Le pilote peut être téléchargé depuis le site web Firebird (https://www.firebirdsql.org/). Cet exemple utilise *Jaybird 2.2.8*.
- Vous disposez des détails de connexion pour la base de données suivante : l'hôte, le chemin de base de données ou alias, le nom d'utilisateur et le mot de passe.

Pour se connecter à Firebird par le biais de JDBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquer sur **JDBC Connections**.
3. À côté de « Chemins de classe », saisir le chemin menant au fichier .jar qui fournit la connexion à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par des points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : **C:\jdbc\firebird\jaybird-full-2.2.8.jar**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le chemin d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans le champ « Pilote », choisir **org.firebirdsql.jdbc.FBDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de

 © 2016-2022 Altova GmbH

classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



5. Saisir le nom d'utilisateur et le mot de passe de la base de données dans les champs correspondants.
6. Saisir la chaîne de connexion du serveur de base de données dans le champ de saisie URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

```
jdbc:firebirdsql://<host>[:<port>]/<database path or alias>
```

7. Cliquer sur **Se connecter**.

19.1.10.2 Firebird (ODBC)

Cet exemple illustre comment se connecter à une base de données Firebird 2.5.4 fonctionnant sur un serveur Linux.

Conditions préalables :

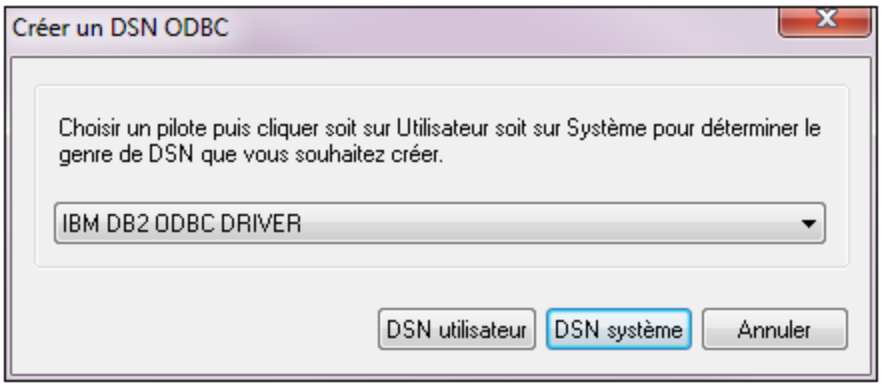
- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Le pilote Firebird ODBC doit être installé sur votre système d'exploitation. Cet exemple utilise le pilote Firebird ODBC version 2.0.3.154 téléchargé depuis le site web Firebird (https://www.firebirdsql.org/).
- Le client Firebird doit être installé sur votre système d'exploitation. Veuillez noter qu'aucun installateur autonome n'est disponible pour le client Firebird 2.5.4 ; le client fait partie du paquet d'installation du serveur Firebird. Vous pouvez télécharger le paquet d'installation du serveur Firebird depuis le site web Firebird (https://www.firebirdsql.org/), chercher « Windows executable installer for full Superclassic/Classic or Superserver ». Pour n'installer que les fichiers client, choisir « **Minimum client install - no server, no tools** » lorsque vous suivrez les instructions de l'assistant.

  Important :

  - La plateforme du pilote Firebird ODBC et du client (32-bit ou 64-bit) doit correspondre à celle de XMLSpy.
  - La version du client Firebird doit correspondre à la version du serveur Firebird auquel vous vous connectez.
- Vous disposez des détails de connexion à la base de données suivantes : le nom d'hôte du serveur ou l'adresse IP, le chemin de base de données (ou alias) sur le serveur, le nom d'utilisateur et le mot de passe.

Pour se connecter à Firebird via ODBC :

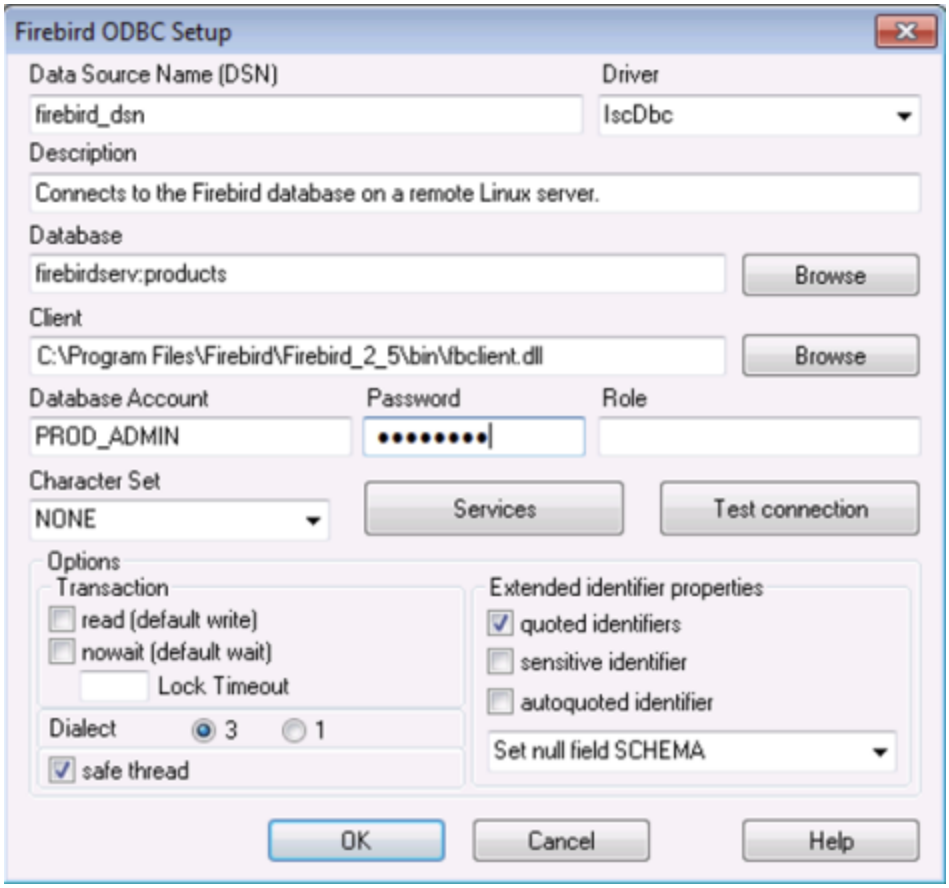
1. Lancer l'assistant de connexion de base de données.
2. Cliquer sur **Connexions ODBC**.
3. Choisir **DSN utilisateur** (ou **DSN système**, si vous disposez de privilèges d'administrateur), puis cliquez sur **Ajouter** .



4. Choisir le pilote Firebird, puis cliquer sur **DSN utilisateur** (ou **DSN système**, selon ce que vous aurez sélectionné dans l'étape précédente). Si le pilote Firebird n'est pas disponible dans la liste, veuillez

 © 2016-2022 Altova GmbH

vous assurer qu'il est installé dans votre système d'exploitation (voir aussi Consulter les pilotes ODBC disponibles).



5. Saisir les détails de connexion de base de données comme suit :

Nom de source de données (DSN)	Saisir un nom descriptif pour la source de données que vous êtes en train de créer.
Base de données	Saisir le nom d'hôte du serveur ou l'adresse IP, suivi d'un double point, suivi d'un alias de base de données (ou chemin). Dans cet exemple, le nom d'hôte est `firebirdserv`, et l'alias de base de données est `products`, comme suit : `firebirdserv:products` L'utilisation d'une base de données assume que, du côté serveur, l'administrateur de la base de données a configuré les *produits* d'alias pour pointer vers le fichier de base de données Firebird (.fdb) sur le serveur (voir la documentation Firebird pour plus de détails). Vous pouvez également utiliser l'adresse IP du serveur à la place du nom d'hôte et un chemin à la place d'un alias ; donc n'importe laquelle des chaînes de connexion d'échantillon est valide :

	`firebirdserver:/var/Firebird/databases/butterflies.fdb`
	`127.0.0.1:D:\Misc\Lenders.fdb`
	
	Si la base de données se trouve sur la machine Windows locale, cliquer sur **Parcourir** et sélectionner directement le fichier de base de données Firebird (.fdb).

Client	Saisir le chemin vers le fichier **fbclient.dll**. Par défaut, il s'agit du sous-répertoire `bin` du répertoire d'installation Firebird.
Compte de base de données	Saisir le nom d'utilisateur de base de données fourni par l'administrateur de la base de données (dans cet exemple, `PROD_ADMIN`).
Mot de passe	Saisir le mot de passe de la base de données fourni par l'administrateur de la base de données.

6. Cliquer sur **OK**.

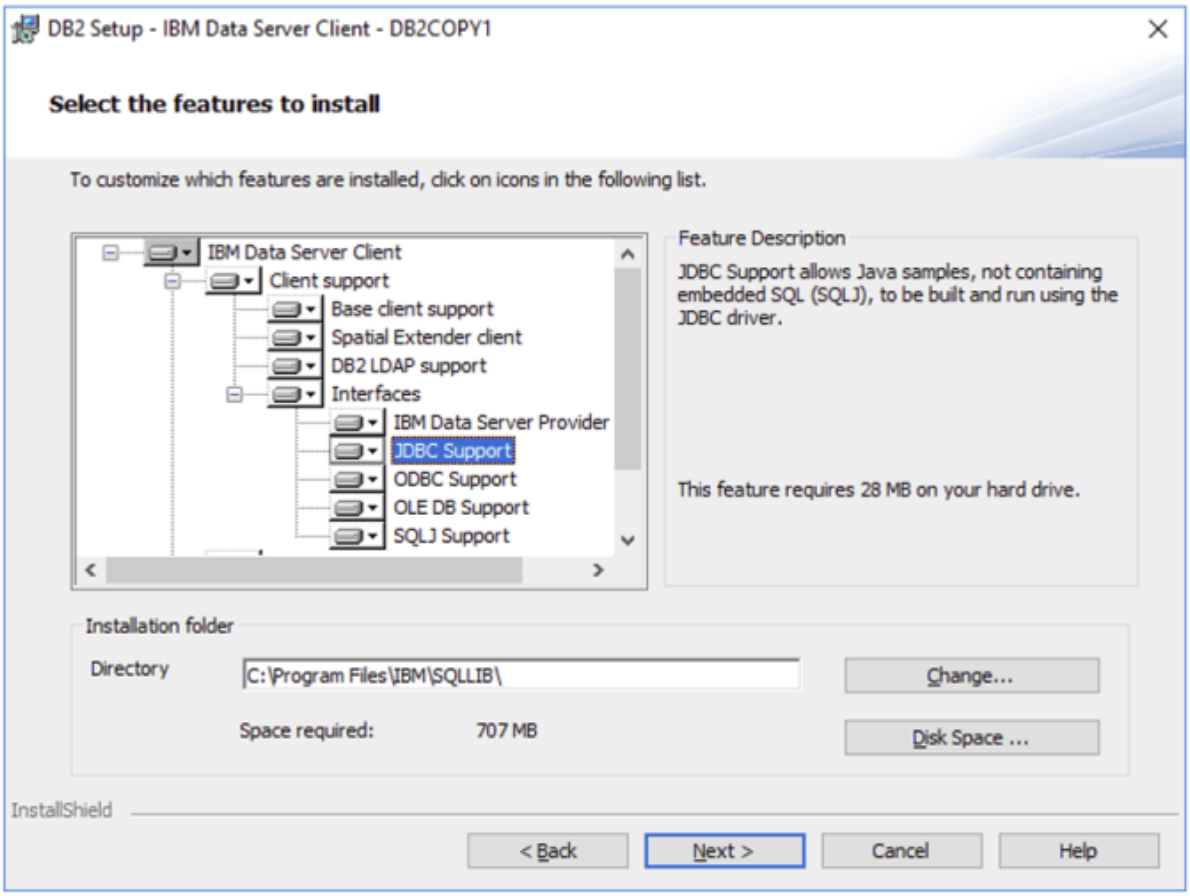
19.1.10.3 IBM DB2 (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données **IBM DB2** par le biais de JDBC.

Conditions préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) corresponde à celle du JRE/JDK. Cet exemple utilise l'OpenJDK 11.0 64-bit d'Oracle, et, par conséquent, la version 64-bit de XMLSpy.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui fournissent une connectivité dans la base de données) doit être disponible sur votre système d'exploitation. Cet exemple utilise le pilote JDBC disponible après avoir installé la version 10.1 (64-bit) **IBM Data Server Client**. Pour installer les pilotes JDBC, choisir une installation **Typical**, ou sélectionner cette option explicitement sur l'assistant d'installation.

© 2016-2022 Altova GmbH

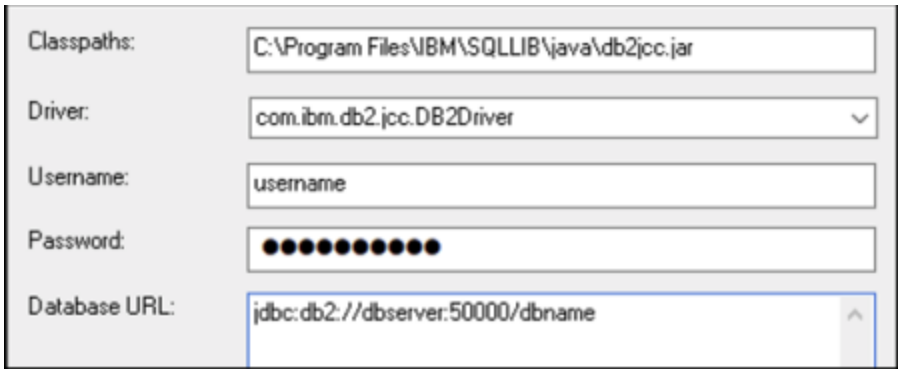


Si vous n'avez pas changé le chemin d'installation par défaut, les fichiers .jar requis se trouvent dans le répertoire **C:\Program Files\IBM\SQLLIB\java** après l'installation.

- Vous nécessitez les détails de connexion de base de données suivants : hôte, port, nom d'utilisateur et mot de passe.

Pour se connecter à IBM DB2 par le biais de JDBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquer sur **JDBC Connections**.
3. À côté de « Classpaths », saisir le chemin menant au fichier .jar qui fournit la connexion à la base de données. Cet exemple se réfère à **C:\Program Files\IBM\SQLLIB\java\db2jcc.jar.** Vous devrez éventuellement vous référer au pilote **db2jcc4.jar**, selon la version du serveur de base de données. Pour une compatibilité de pilote, se référer à la documentation IBM (http://www-01.ibm.com/support/docview.wss?uid=swg21363866). Veuillez noter que vous pouvez laisser le champ de texte "Classpaths" vide si vous avez ajouté les chemins de fichier .jar dans la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configurer le CLASSPATH).
4. Dans la case "Driver", choisir **com.ibm.db2.jcc.DB2Driver**. Cette entrée devient disponible uniquement si un chemin de fichier .jar valide a été trouvé soit dans le champ de texte "Classpaths", soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



5. Saisir le nom d'utilisateur et le mot de passe de l'utilisateur de la base de données dans le champ de texte correspondant.
6. Saisir le string de connexion JDBC dans le champ de texte de l'**URL de base de données**. Veuillez vous assurer de remplacer les détails de connexion avec ceux applicable dans votre serveur de base de données.

```
jdbc:db2://hostName:port/databaseName
```

7. Cliquer sur **Se connecter**.

19.1.10.4 IBM DB2 (ODBC)

Cet exemple illustre comment se connecter à une base de données IBM DB2 via ODBC.

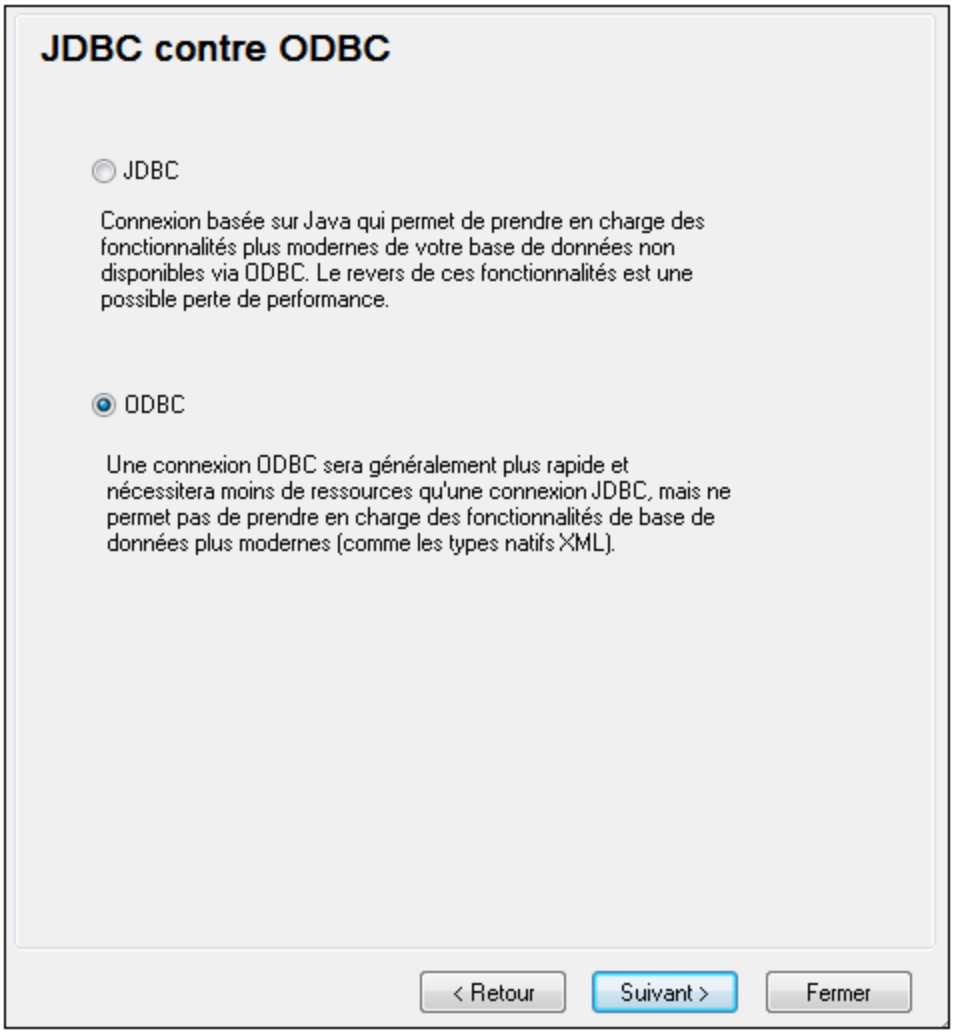
Conditions préalables :

- L'IBM Data Server Client doit être installé et configuré sur votre système d'exploitation (cet exemple utilise IBM Data Server Client 9.7). Pour travailler avec des instructions d'installation, veuillez consulter la documentation fournie avec votre logiciel IBM DB2. Après avoir installé IBM Data Server Client, vérifiez si les pilotes ODBC sont disponibles sur votre machine (voir Consulter les pilotes ODBC disponibles).
- Créer un alias de base de données. Vous pouvez procéder de plusieurs manières :
  - Depuis l'Assistant de configuration IBM DB2
  - Depuis le IBM DB2 Command Line Processor
  - Depuis l'assistant de source de données ODBC (dans ce cas, les instructions sont affichées ci-dessous)
- Vous disposez des détails de connexion de base de données suivants : hôte, base de données, port, nom d'utilisateur et mot de passe.

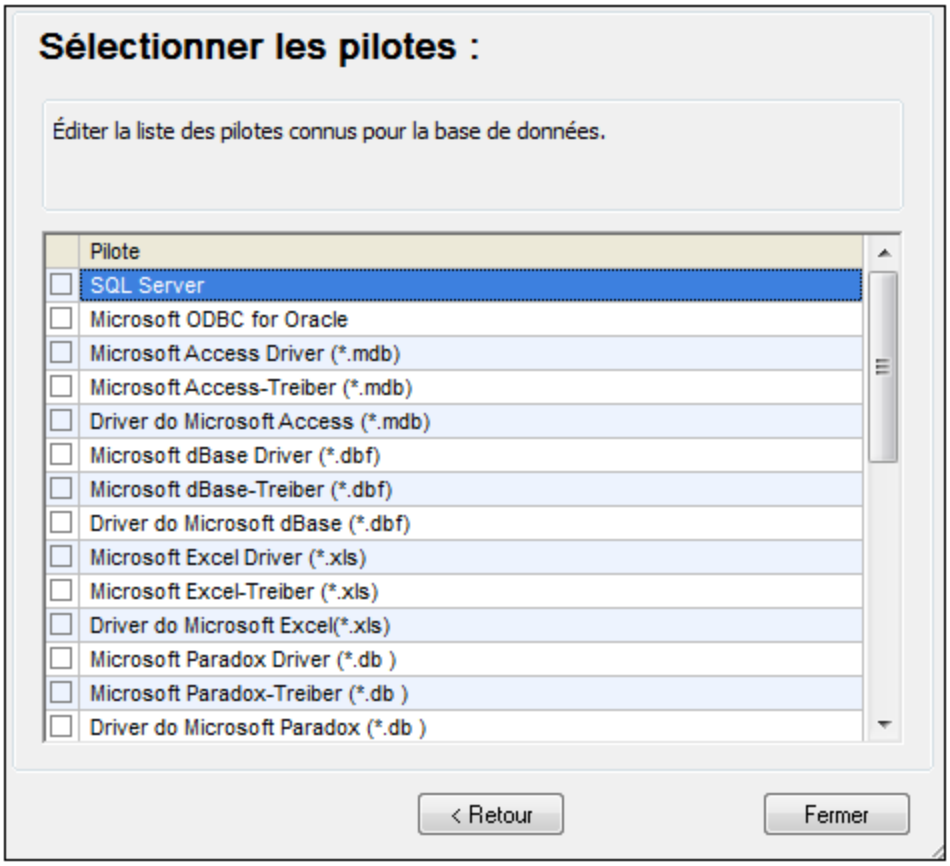
Pour se connecter à IBM DB2 :

1. Lancer l'assistant de connexion de base de données et sélectionner **IBM DB2 (ODBC/JDBC)**.
2. Cliquer sur **Suivant**.

© 2016-2022 Altova GmbH



3. Choisir **ODBC** et cliquer sur **Suivant**. Si vous êtes invité à éditer la liste des pilotes connus pour la base de données, sélectionnez les pilotes de base de données applicables pour IBM DB2 (voir les Conditions préalables), et cliquer sur **Suivant**.

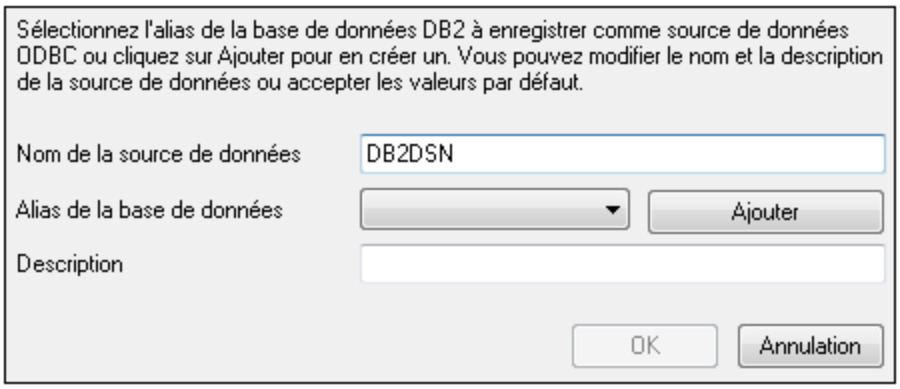


4. Choisir le pilote IBM DB2 depuis la liste et cliquer sur **Se connecter**. (Pour éditer la liste des pilotes disponibles, cliquer sur **Éditer pilotes**, puis cocher ou décocher les pilotes IBM DB2 que vous souhaitez ajouter ou supprimer, respectivement.)

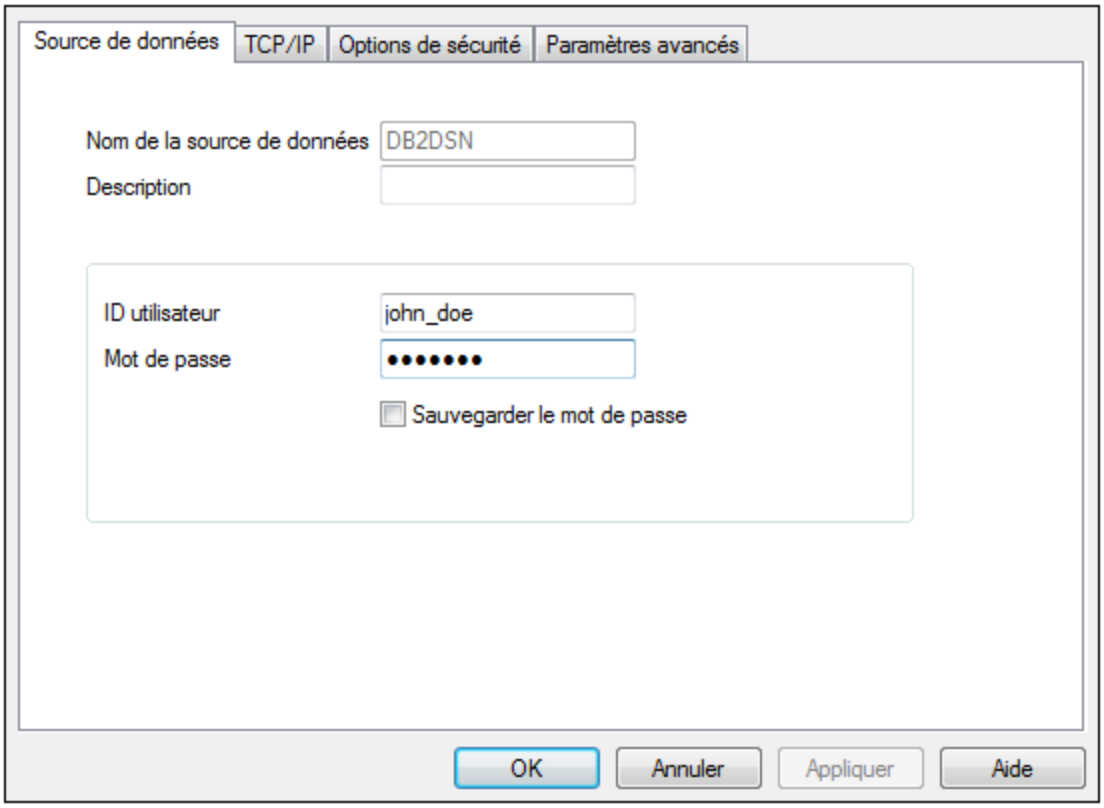
© 2016-2022 Altova GmbH



5. Saisir un nom de source de données (dans cet exemple, **DB2DSN**), puis cliquer sur **Ajouter**.

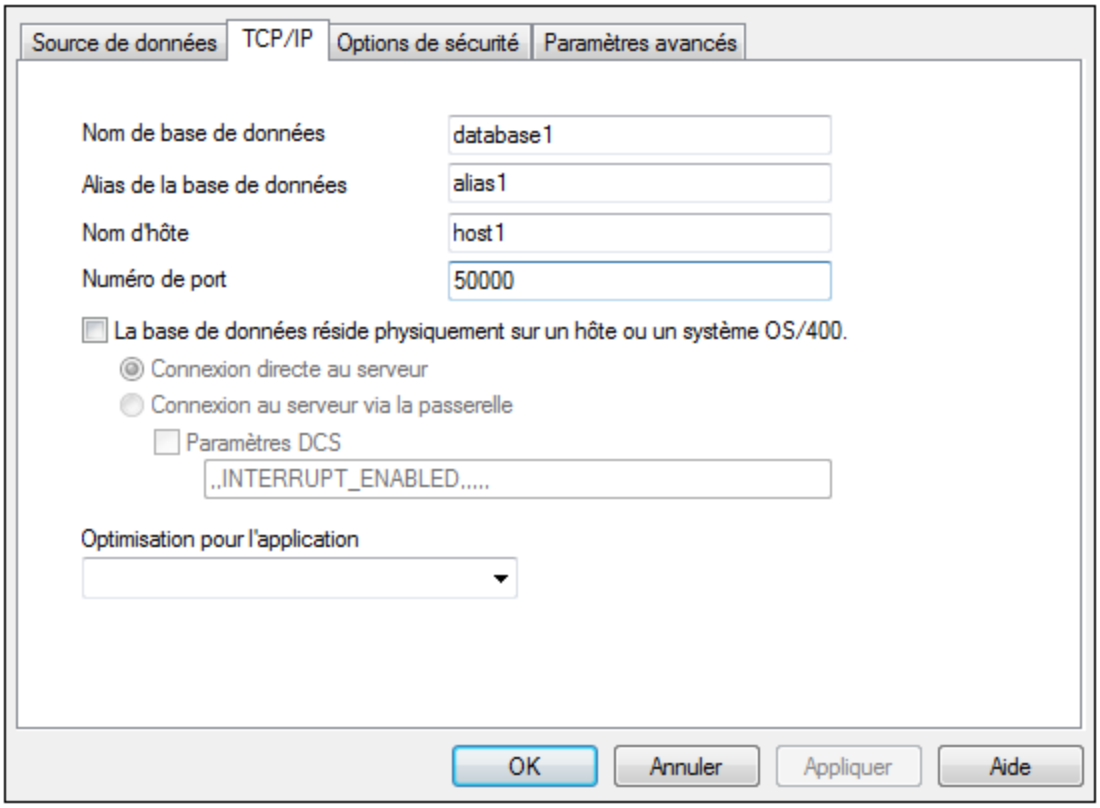


6. Dans l'onglet **Source de données**, saisir le nom d'utilisateur et le mot de passe de la base de données.

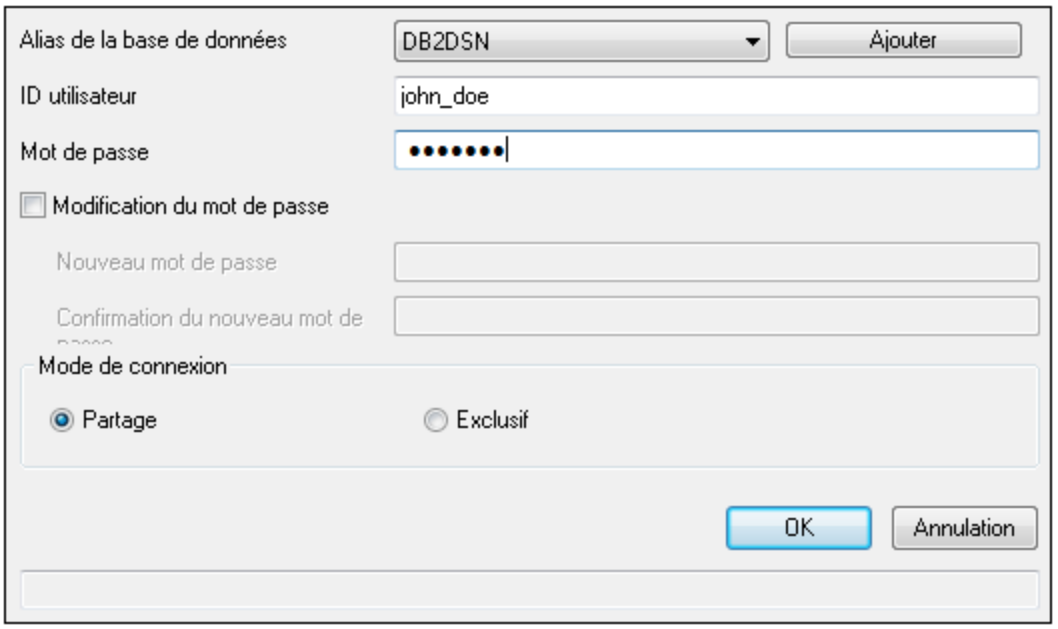


7. Dans l'onglet **TCP/IP**, saisir le nom de base de données, un nom pour l'alias, le nom d'hôte et le numéro de port, puis cliquer sur OK.

© 2016-2022 Altova GmbH



8. Saisir à nouveau le nom d'utilisateur et le mot de passe puis cliquer sur **OK**.



19.1.10.5 IBM DB2 pour i (JDBC)

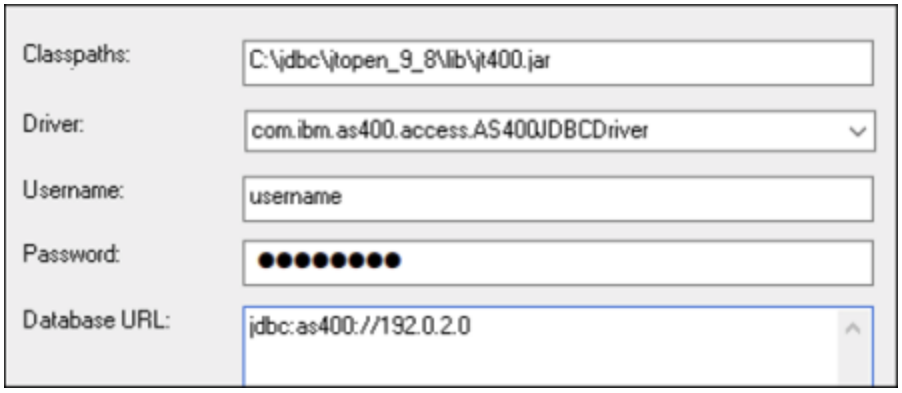
Cet exemple illustre comment se connecter à un serveur de base de données **IBM DB2 for i** par le biais de JDBC.

Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) corresponde à celle du JRE/JDK. Cet exemple utilise l'OpenJDK 11.0 64-bit d'Oracle, et, par conséquent, la version 64-bit de XMLSpy.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui offrent une connectivité à la base de données) doit être disponible sur votre système d'exploitation. Cet exemple utilise la source ouverte **Toolbox for Java/JTOpen** version 9.8 (http://jt400.sourceforge.net/). Après avoir installé le package et l'avoir décompressé vers un répertoire local, les fichiers .jar requis seront disponibles dans le sous-répertoire **lib**.
- Vous nécessitez les détails de connexion de base de données suivants : hôte, port, nom d'utilisateur et mot de passe.

Pour se connecter à IBM DB2 pour i par le biais de JDBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **JDBC Connections**.
3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : **C:\jdbc\jtopen_9_8\jt400.jar**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans la case« Driver », choisir **com.ibm.as400.access.AS400JDBCDriver**. Cette entrée devient disponible uniquement si un chemin de fichier .jar valide a été trouvé soit dans le champ de texte « Classpaths », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



 © 2016-2022 Altova GmbH

5. Saisissez le nom d'utilisateur et le mot de passe de l'utilisateur de la base de données dans la zone de texte correspondante.
6. Saisissez le string de connexion JDBC dans la zone de texte **Base de données URL** Assurez-vous de remplacer l'**hôte** par le nom de l'hôte ou l'adresse IP de votre serveur de base de données.

```
jdbc:as400://host
```

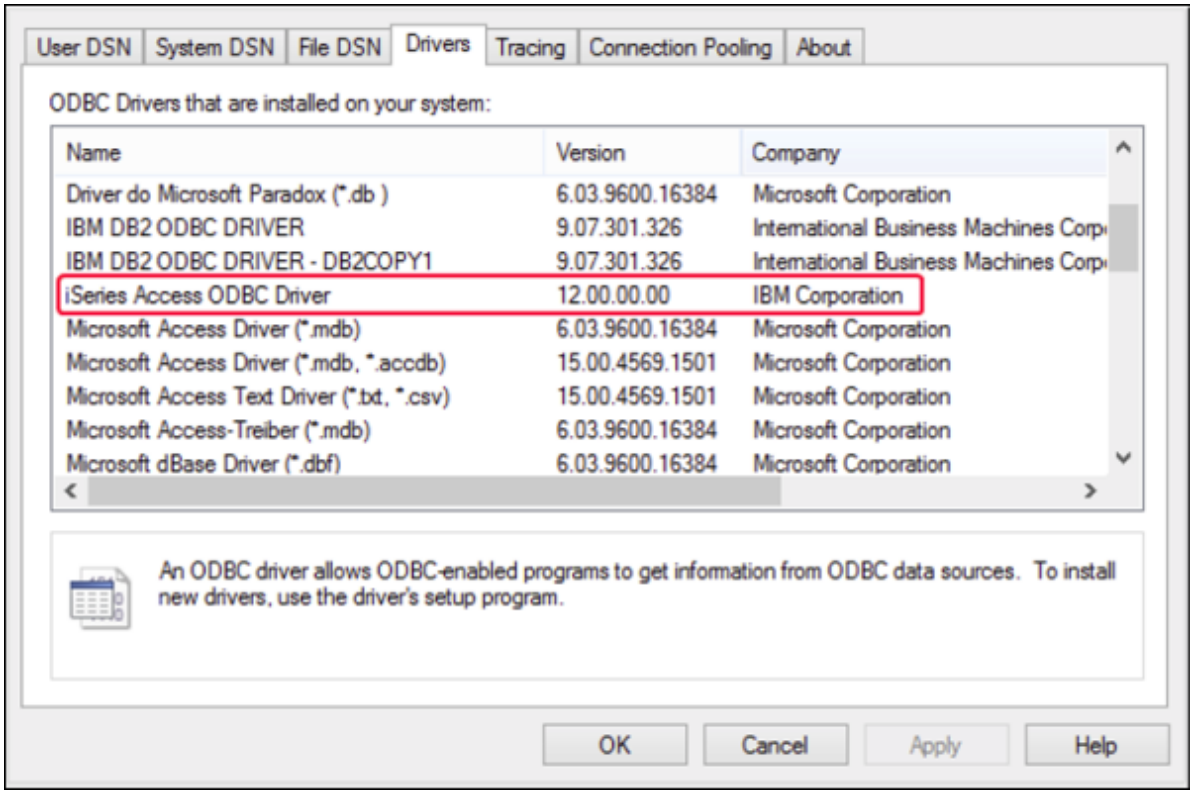
7. Cliquez sur **Connexion**.

19.1.10.6 IBM DB2 pour i (ODBC)

Cet exemple illustre comment se connecter à une base de données *IBM DB2 pour i* par le biais d'ODBC.

Exigences préalables :

- *IBM System i Access pour Windows* doit être installé sur votre système d'exploitation (cet exemple utilise *IBM System i Access pour Windows V6R1M0*). Pour les instructions d'installation, vérifiez la documentation fournie avec votre logiciel *IBM DB2 pour i*. Après l'installation, vérifier si le pilote ODBC est disponible sur votre appareil (voir Consulter les pilotes ODBC disponibles).

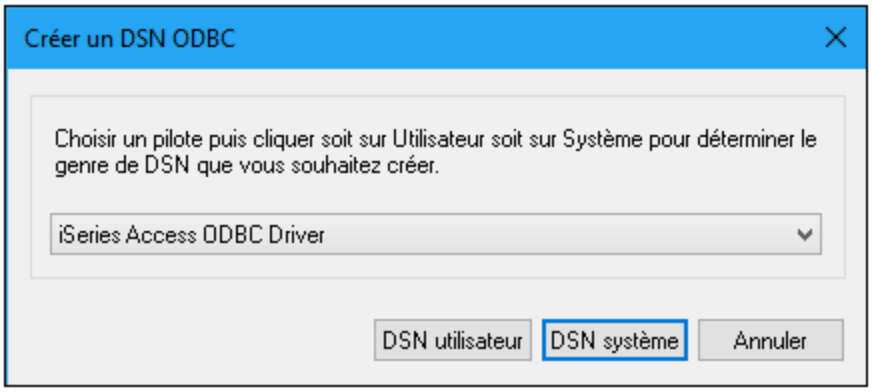


- Vous disposez des détails de connexion à la base de données suivants : l'adresse IP du serveur de la base de données, le nom d'utilisateur de la base de données et le mot de passe.
- Exécuter *System i Navigator* et suivez l'assistant pour créer une nouvelle connexion. Sur invitation à spécifier un système, saisissez l'adresse IP du serveur de la base de données. Après avoir établi la connexion, il est recommandé de la vérifier (cliquez sur la connexion, et choisissez **Fichier** >

Diagnostics > **Vérifier connexion**). Si vous obtenez des erreurs de connectivité, contacter l'administrateur de serveur de la base de données.

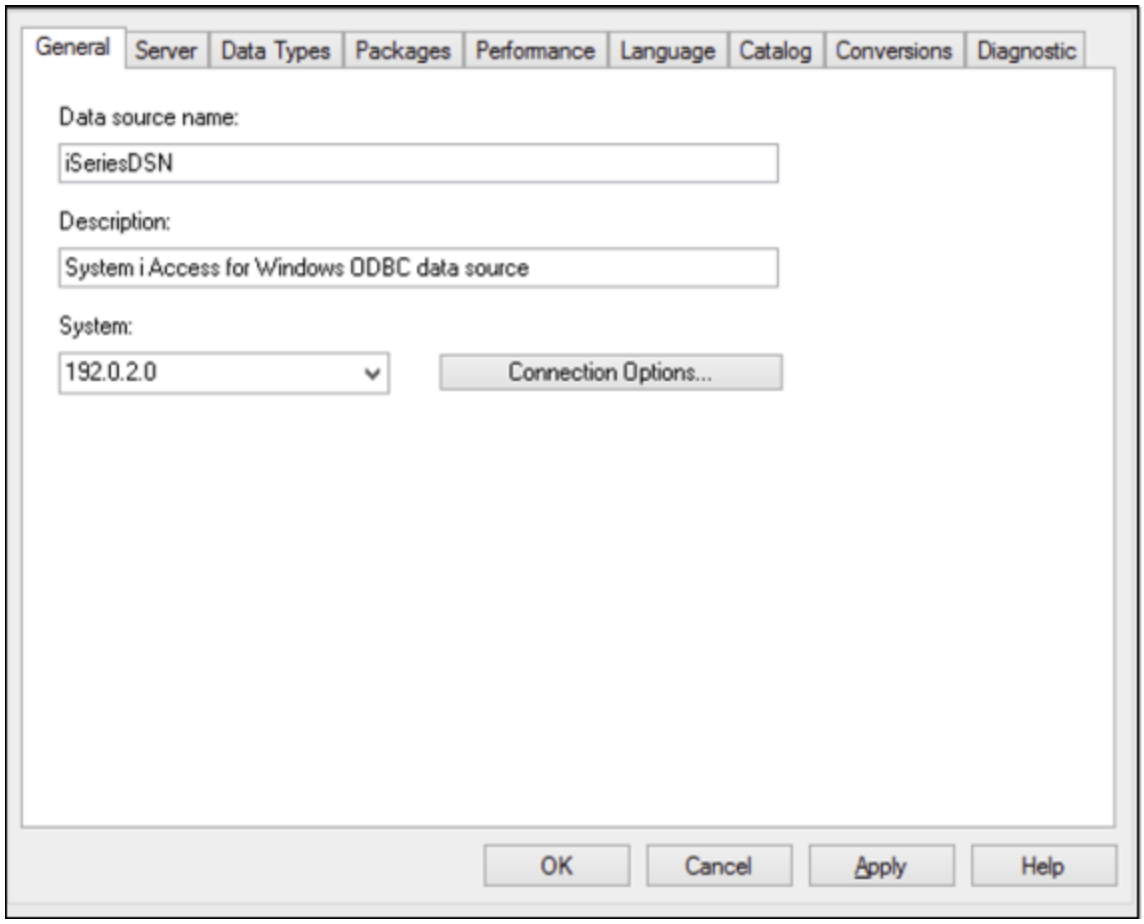
Pour se connecter à IBM DB2 for i:

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **connexions ODBC**.
3. Cliquez sur**DSN Utilisateur** (en alternative, cliquez sur **DSN Système** ou **DSN Fichier**, dans un tel cas, les instructions suivantes seront semblables).
4. Cliquez sur **Ajouter** .
5. Sélectionnez le **pilote iSeries Access ODBC** depuis la liste et cliquez sur **DSN Utilisateur** (ou **DSN Système**, le cas échéant).

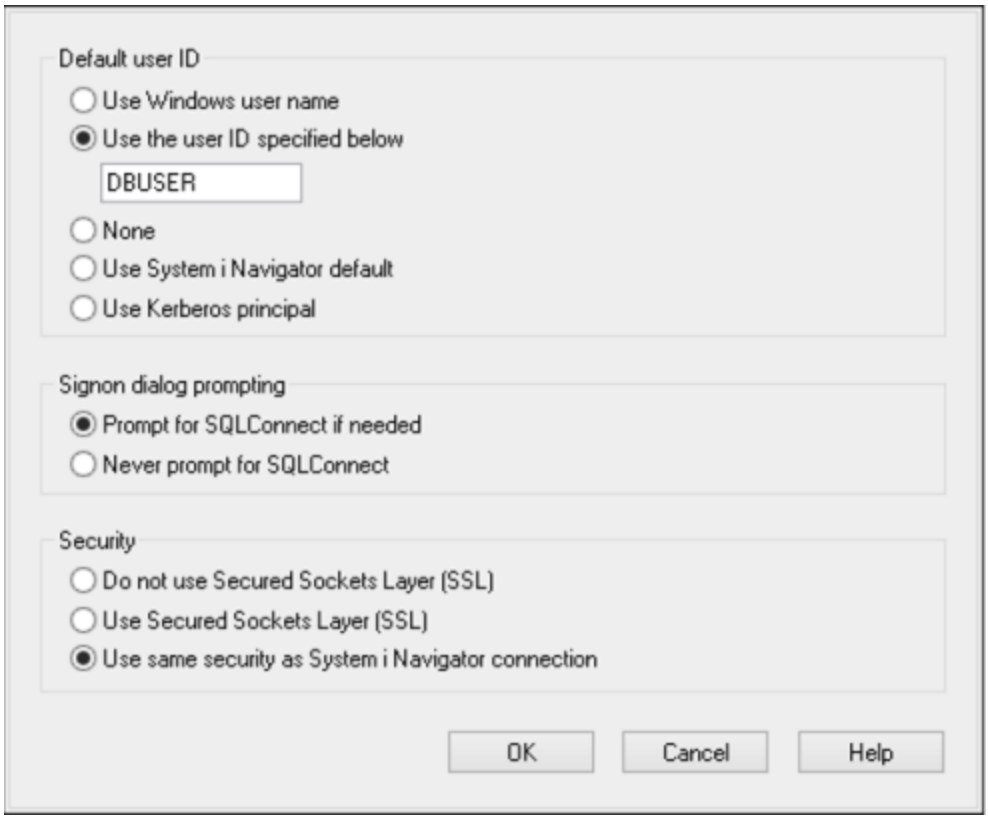


6. Saisissez un nom de source de données et sélectionnez la connexion depuis la zone de liste du Système. Dans cet exemple, le nom de source de données est **iSeriesDSN** et le Système est **192.0.2.0**.

© 2016-2022 Altova GmbH



7. Cliquez sur les options de connexion, sélectionnez **Utiliser l'ID utilisateur spécifiée ci-dessous** et saisissez le nom de l'utilisateur de la base de données (dans cet exemple, **DBUSER**).



8. Cliquez sur **OK**. La nouvelle source de données devient disponible dans la liste des DSN.
9. Cliquez sur **Connexion**.
10. Saisissez le nom utilisateur et le mot de passe de la base de données quand vous y êtes invité, puis cliquez sur **OK**.

19.1.10.7 IBM Informix (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données IBM Informix par le biais de JDBC.

Exigences préalables :

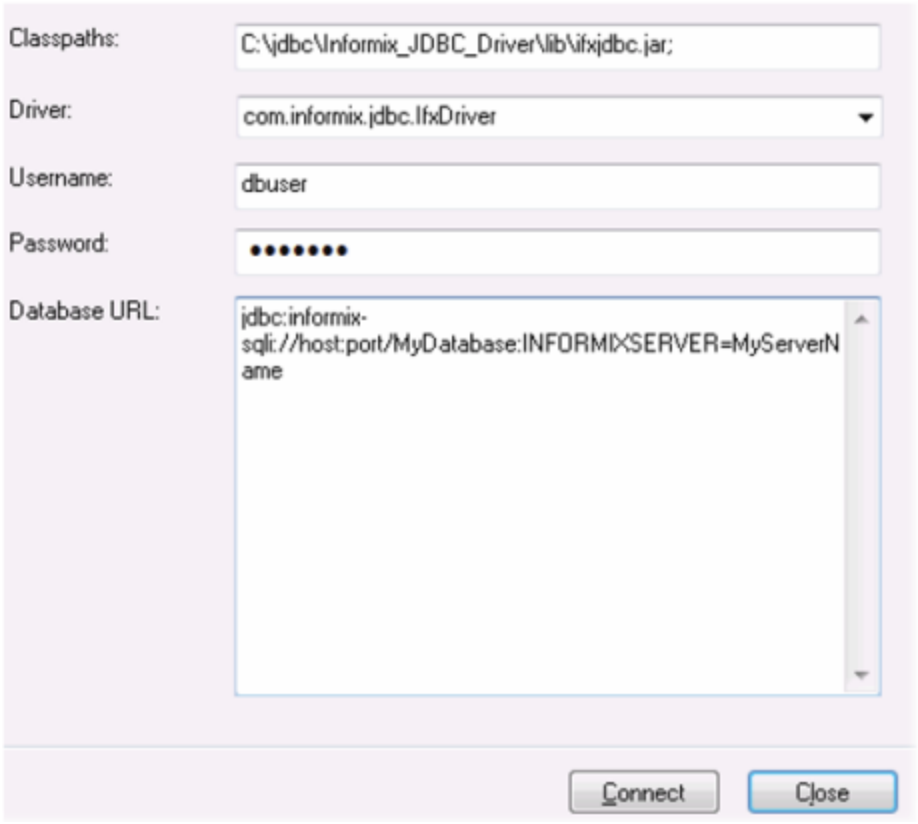
- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) correspond à celle du JRE/JDK.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui offrent une connectivité à la base de données) doit être disponible sur votre système d'exploitation. Dans cet exemple, le pilote IBM Informix JDBC version 3.70 est utilisé. En ce qui concerne les instructions d'installation du pilote, voir la documentation fournie avec le pilote ou bien le manuel « IBM Informix JDBC Driver Programmer's Guide »).

 © 2016-2022 Altova GmbH

- Vous disposez des détails de connexion à la base de données suivants : hôte, nom du serveur Informix, base de données, port, nom d'utilisateur et mot de passe.

Pour se connecter à IBM Informix par le biais de JDBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **JDBC Connections**.
3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : **C:\Informix_JDBC_Driver\lib\ifxjdbc.jar**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le chemin d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans le champ « Pilote », sélectionnez **com.informix.jdbc.IfxDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
6. Saisir la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

```
jdbc:informix-sqli://hostName:port/databaseName:INFORMIXSERVER=myserver;
```

7. Cliquez sur **Connexion**.

19.1.10.8 MariaDB (ODBC)

Cet exemple indique comment se connecter à un serveur de base de données MariaDB par le biais d'ODBC.

Exigences préalables :

- Le connecteur/ODBC MariaDB (https://downloads.mariadb.org/connector-odbc/) doit être installé.
- Vous disposez des détails de connexion à la base de données suivants : hôte, base de données, port, nom utilisateur et mot de passe.

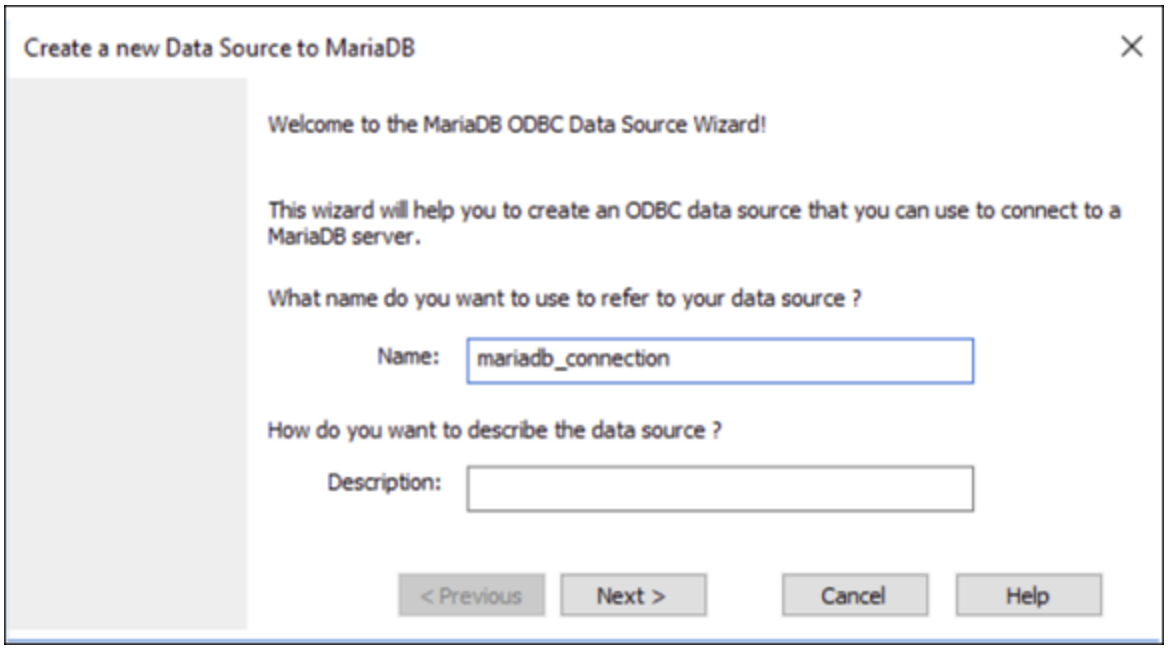
Pour se connecter à MariaDB avec ODBC :

1. Lancer l'assistant de connexion de base de données.
2. Choisissez **MariaDB (ODBC)** et cliquez sur **Suivant**.

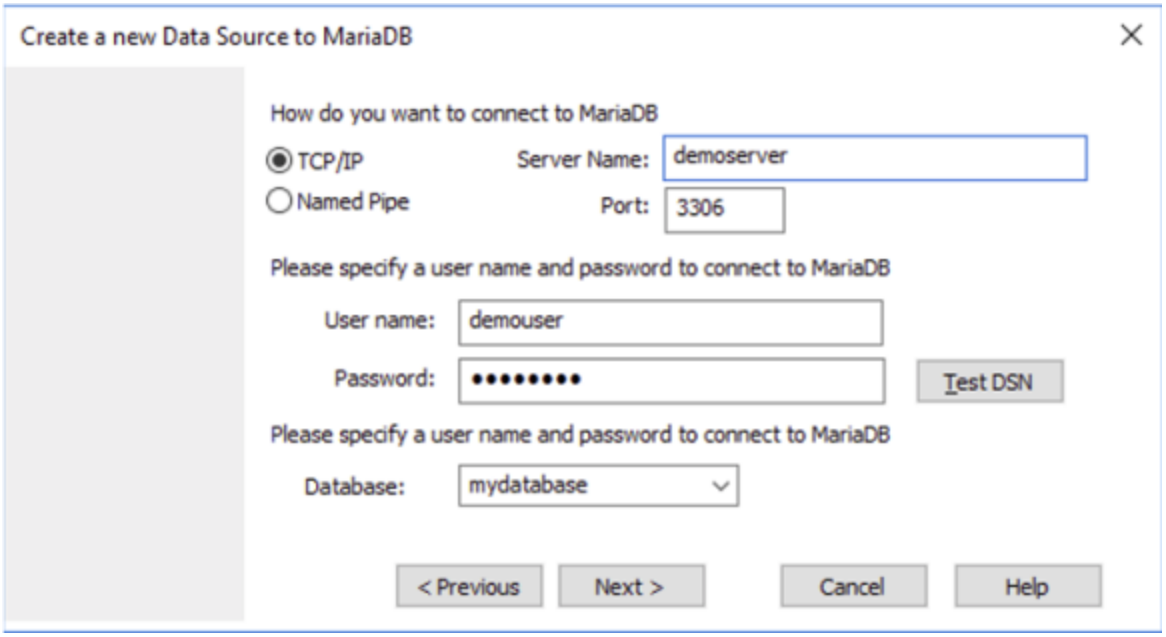


3. Choisissez **Créer un nouveau nom de source de données (DSN) avec le pilote** et choisissez le **pilote MariaDB ODBC 3.0**. Si ce pilote n'est pas disponible dans la liste, cliquez sur **Éditer pilotes**, et choisissez un des pilotes MariaDB disponibles (la liste contient tous les pilotes ODBC installés dans votre système d'opération).
4. Cliquez sur **Se connecter**.

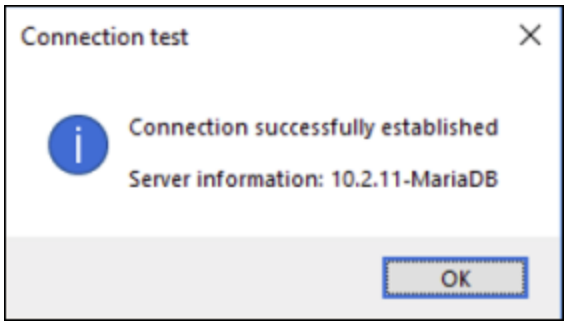
© 2016-2022 Altova GmbH



5. Saisir le nom et, éventuellement, une description qui vous aidera à identifier cette source de données ODBC à l'avenir.



6. Remplissez les identifiants de connexion dans la base de données (Serveur TCP/IP, Utilisateur, Mot de passe), choisissez une base de données et cliquez sur **Tester DSN**. Une fois la connexion établie, un message apparaît :



7. Cliquez sur **Suivant** et terminez l'assistant. D'autres paramètres seront éventuellement nécessaires, selon les cas (par exemple, des certificats SSL si vous vous êtes connecté à MariaDB par le biais d'une connexion sécurisée).

Note : si le serveur de base de données est à distance, il doit être configuré par l'administrateur du serveur pour accepter les connexions à distance depuis l'adresse IP de votre appareil.

19.1.10.9 Microsoft Access (ADO)

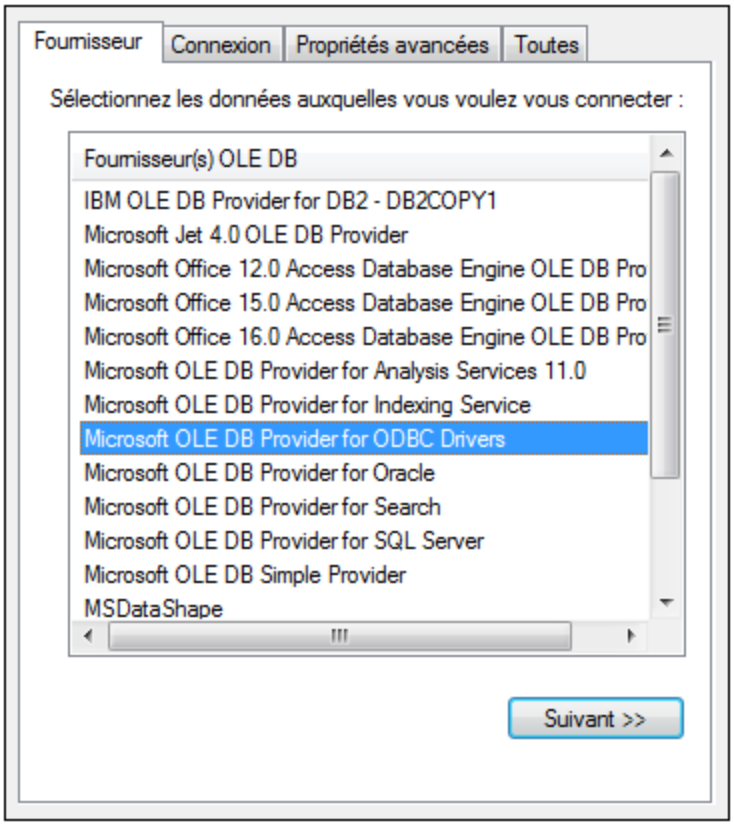
Une manière simple de se connecter à une base de données Microsoft Access est de suivre les instructions de l'assistant et de parcourir le fichier de base de données, tel que montré dans la section Connexion à une base de données Microsoft Access existante. Une approche alternative est de définir explicitement une connexion ADO, tel que montré dans cette section. Cette approche est utile si votre base de données est protégée par un mot de passe.

Il est également possible de se connecter à Microsoft Access par le biais d'une connexion ODBC, mais ce scénario a des limites, il vaut donc mieux l'éviter.

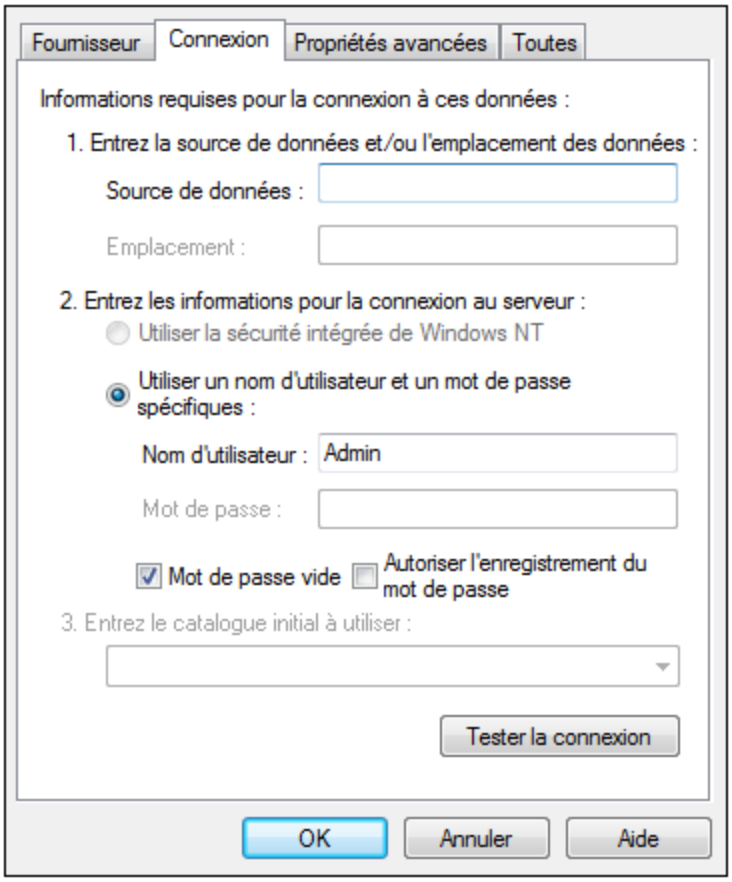
Pour se connecter à une base de données Microsoft Access protégée par un mot de passe :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **Connexions ADO**.
3. Cliquer sur **Générer**.

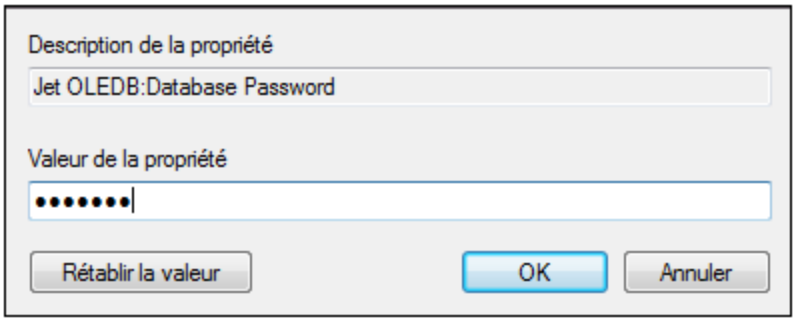
© 2016-2022 Altova GmbH



4. Sélectionnez le **fournisseur Microsoft Office 15.0 Access Database Engine OLE DB**, puis cliquez sur **Suivant**.



5. Dans la case de source de données, saisissez le chemin vers le fichier Microsoft Access dans le format UNC, par exemple **\\myserver\\mynetworkshare\Reports\Revenue.accdb**, où **myserver** est le nom du serveur et **mynetworkshare** est le nom du réseau partagé.
6. Dans l'onglet **Toutes**, double-cliquez la propriété **Jet OLEDB:Database Password** et saisissez le mot de passe de la base de données en tant que valeur de propriété.



Note : si vous ne parvenez toujours pas à vous connecter, localisez le fichier d'information du groupe de travail (**System.MDW**) applicable à votre profil d'utilisateur, et réglez la valeur de la propriété **Jet OLEDB : System database** sur le chemin d'accès du fichier **System.MDW**.

© 2016-2022 Altova GmbH

19.1.10.10 Microsoft SQL Server (ADO)

Cet exemple indique comment se connecter à une base de données SQL Server avec ADO. Ces instructions sont applicables quand vous utilisez **Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL)** recommandé, qui est disponible pour le téléchargement au lien https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15.

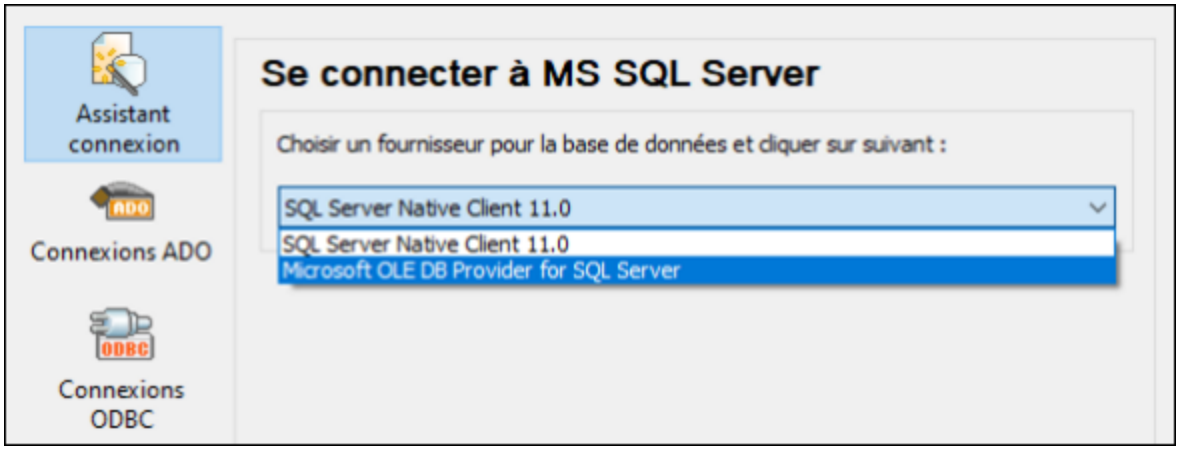
Avant de suivre ces instructions, assurez-vous avoir téléchargé et installé le fournisseur ci-dessus sur votre poste de travail. Le fournisseur ADO doit correspondre à la plate-forme de XMLSpy (32-bit ou 64-bit).

Si vous souhaitez utiliser d'autres fournisseurs ADO tels que **SQL Server Native Client (SQLNCLI)** ou **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)**, les instructions sont semblables, mais ces fournisseurs sont dépréciés et donc pas recommandés. Aussi, pour que le connexion réussisse avec un fournisseur, vous allez devoir définir des propriétés de connexion additionnelles telles que décrites dans Configurer les propriétés de SQL Server Data Link.

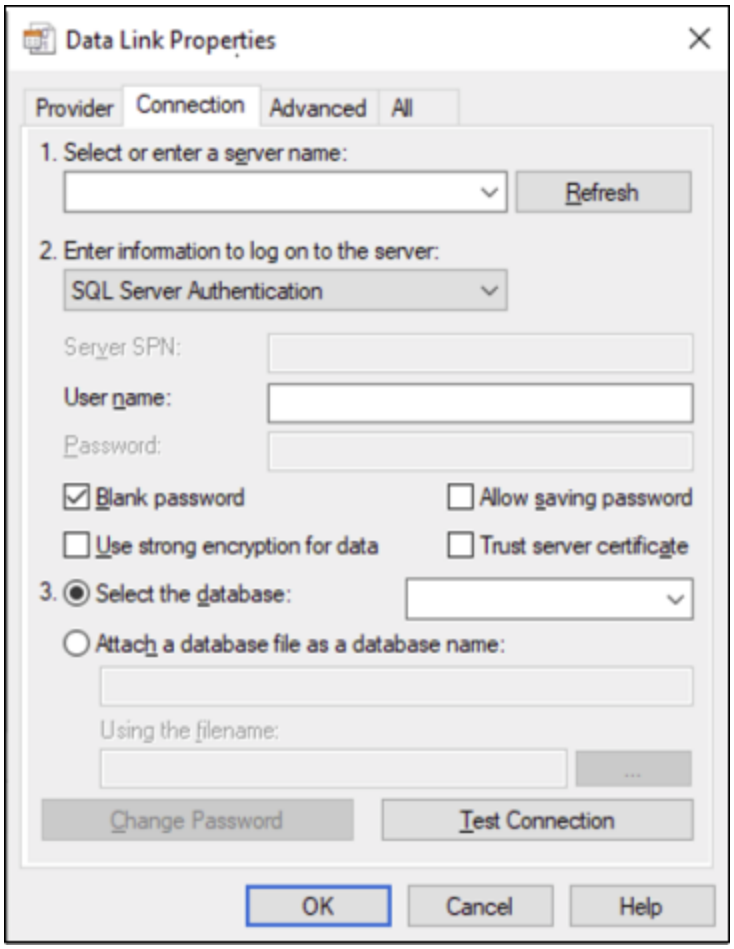
Microsoft OLE DB Provider for SQL Server (SQLOLEDB) est connu pour avoir des soucis avec la liaison de paramètre de requêtes complexes comme Common Table Expressions (CTE) et des déclarations imbriquées SELECT.

Pour se connecter à SQL Server :

1. Lancer l'assistant de connexion de base de données.
2. Choisissez **Microsoft SQL Server (ADO)**, puis cliquez sur **Suivant**. La liste des fournisseurs ADO disponibles est affichée. Dans cet exemple, le **Microsoft OLE DB Driver for SQL Server** est utilisé. S'il ne figure pas dans la liste, assurez-vous qu'il est installé sur votre ordinateur, tel que mentionné ci-dessus.

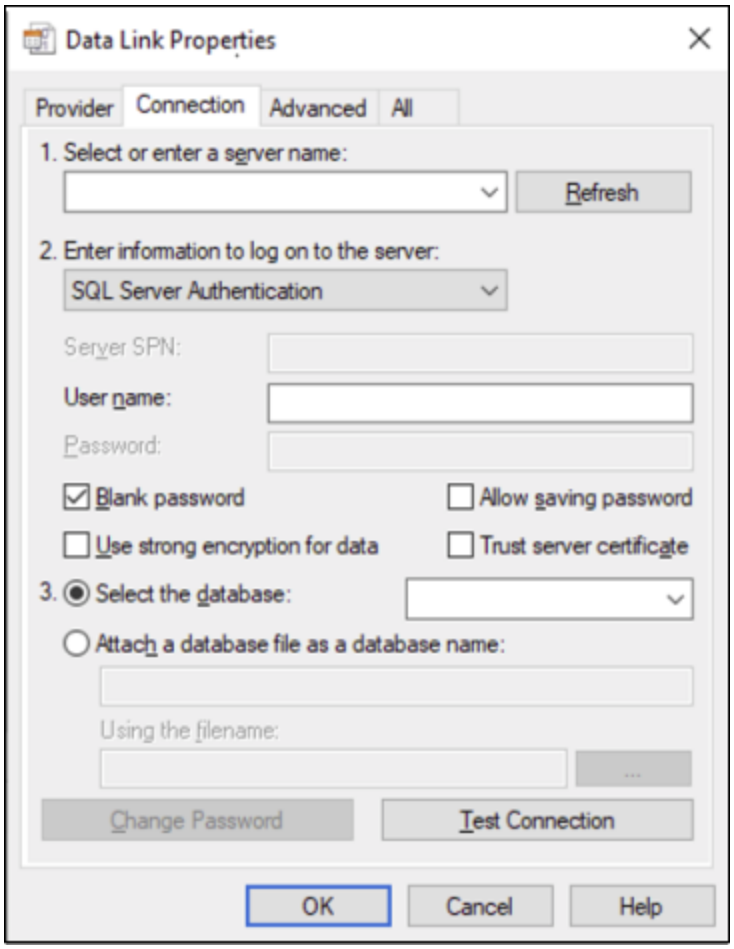


3. Cliquez sur **Next**. Le dialogue Data Link Properties apparaît.



4. Choisissez ou saisissez le nom du serveur de base de données, par exemple **SQLSERV01**. Si vous vous connectez à une instance SQL Server nommée, le nom du serveur ressemblera à **SQLSERV01\SOMEINSTANCE**.
5. Si le serveur de la base de données a été configuré pour permettre des connexions depuis des utilisateurs authentifiés dans le domaine Windows, choisissez **Windows Authentication**. Sinon, sélectionnez **SQL Server Authentication**, décochez la case **Blank password** et saisissez les identifiants de base de données dans les champs pertinents.
6. Sélectionnez la case **Autoriser enregistrer mot de passe** et la base de données à laquelle vous vous connectez (dans cet exemple, « Nanonull »).

© 2016-2022 Altova GmbH



7. Pour tester la connexion à cet endroit, cliquez sur **Tester la connexion**. Il s'agit d'une étape optionnelle recommandée.
8. Cliquez sur **OK**.

19.1.10.11 Microsoft SQL Server (ODBC)

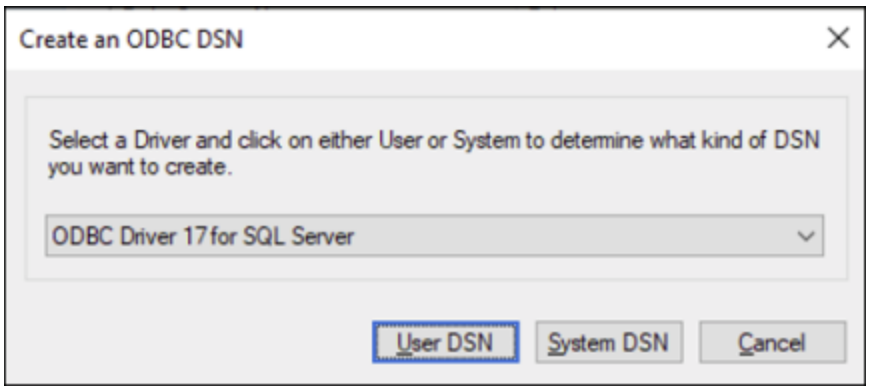
Cet exemple indique comment se connecter à une base de données SQL Server avec ODBC.

Exigences préalables :

- Pour télécharger et installer **Microsoft ODBC Driver for SQL Server** depuis le site web Microsoft, voir https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server. Cet exemple utilise le **Pilote Microsoft ODBC 17 pour SQL Server** pour se connecter à une base de données **SQL Server 2016**. Vous pouvez tout à fait télécharger une version de pilote ODBC différente, selon la version du Serveur SQL auquel vous souhaitez vous connecter. Pour plus d'informations concernant les versions de pilote ODBC prises en charge par votre base de données Serveur SQL, veuillez consulter les exigences de système du pilote.

Pour se connecter à SQL Server avec ODBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **ODBC Connections**.
3. Sélectionnez **DSN Utilisateur** (ou **DSN Système** si vous avez des privilèges d'administrateur), puis cliquez sur **Ajouter** .
4. Sélectionnez le pilote depuis la liste. Veuillez noter que le pilote apparaît dans la liste uniquement après qu'il ait été installé.

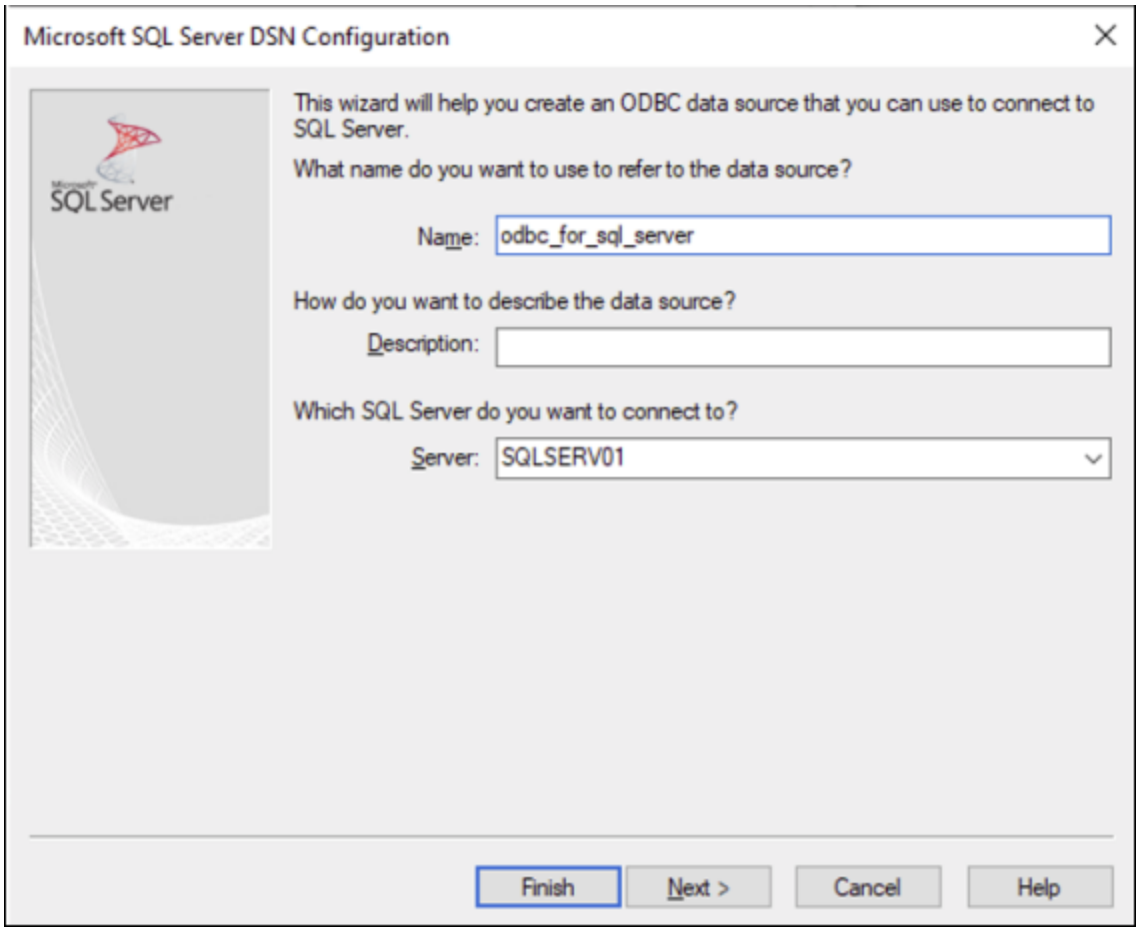


5. Cliquez sur **DSN Utilisateur** (ou **System DSN** si vous créez un DNS Système).

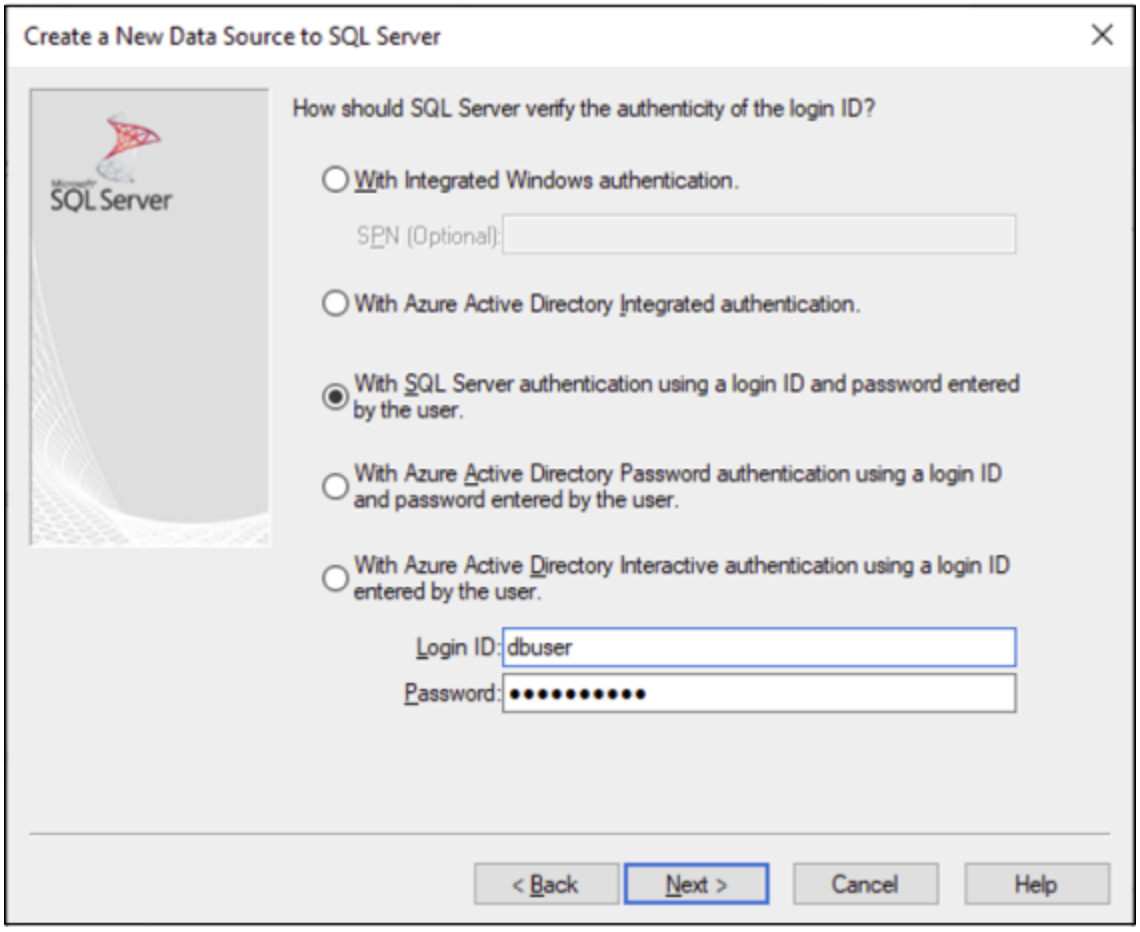
> La création d'un **DNS Système** exige que XMLSpy soit exécuté en tant qu'administrateur. C'est pourquoi, afin de créer un **DNS Système**, annulez l'assistant, assurez-vous d'exécuter XMLSpy en tant qu'administrateur et effectuez les étapes ci-dessus à nouveau.

6. Saisissez un nom et, en option, une description pour identifier cette connexion, puis sélectionner le serveur SQL depuis la liste sur lequel vous voulez vous connecter (**SQLSERV01** dans cet exemple).

© 2016-2022 Altova GmbH

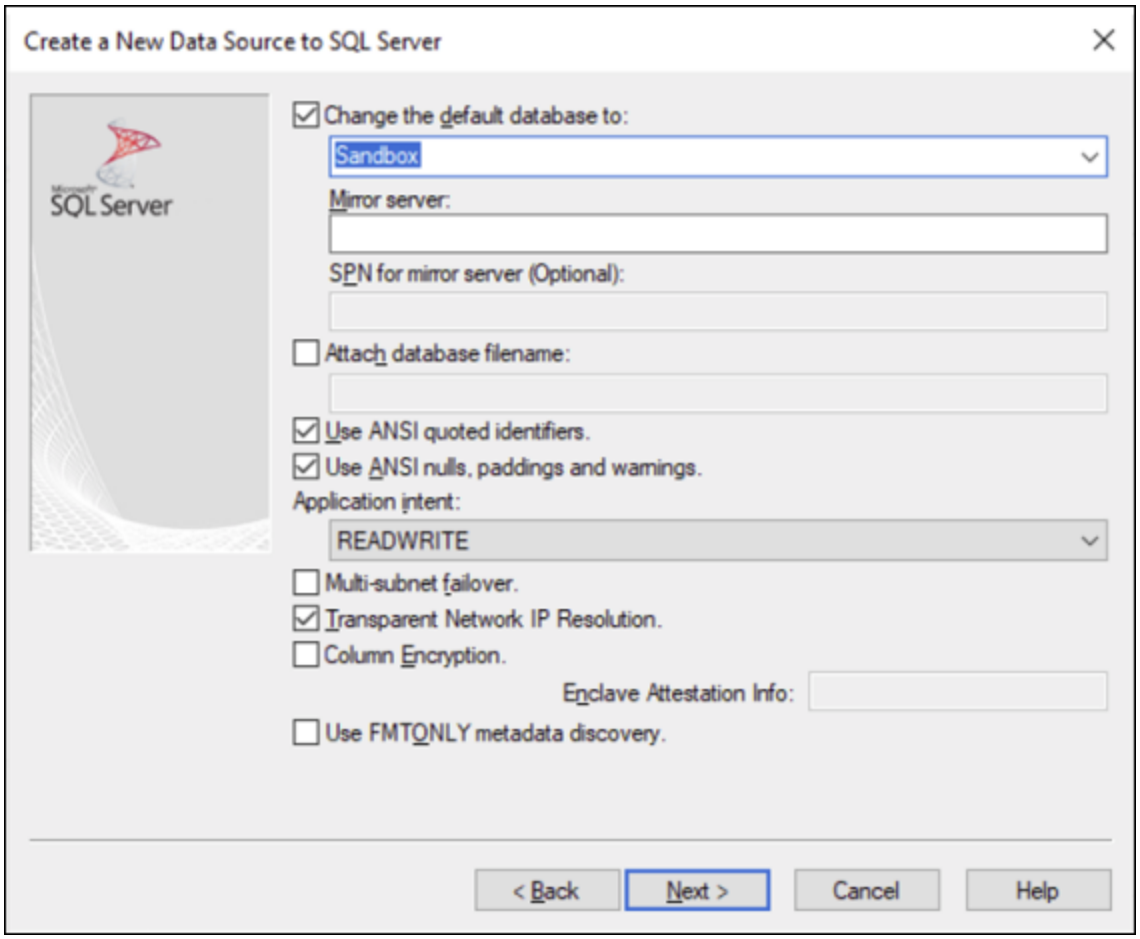


7. Si le serveur de la base de données a été configuré pour permettre des connexions depuis les utilisateurs authentifiés sur le domaine Windows, sélectionnez **Avec l'authentification Windows intégrée**. Sinon, sélectionnez une des autres options, le cas échéant. Cet exemple utilise **Avec authentification SQL Server...** , qui requiert que le nom utilisateur et le mot de passe soient saisis dans les cases pertinentes.

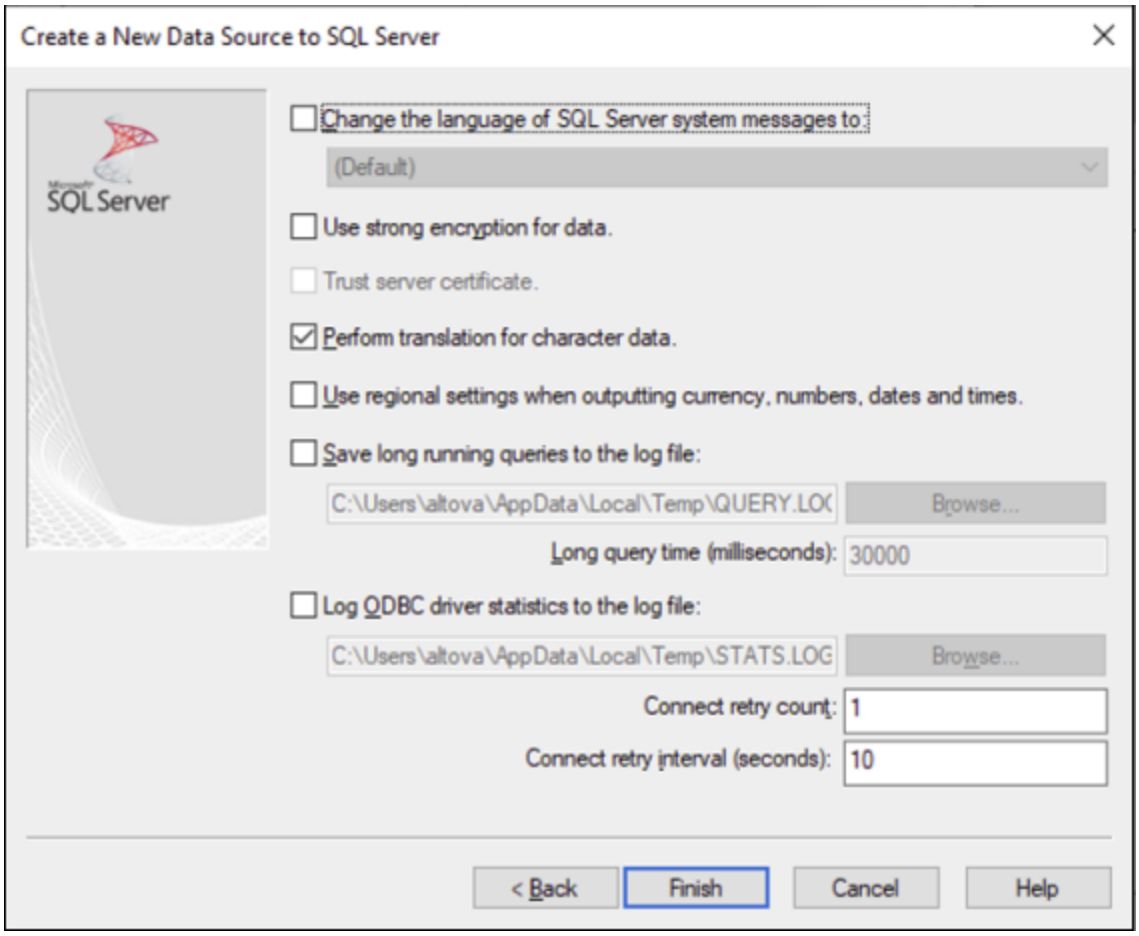


8. En option, cochez la case **Changer la base de données par défaut en** et saisissez le nom de la base de données à laquelle vous vous connectez (dans cet exemple, **Sandbox**).

© 2016-2022 Altova GmbH

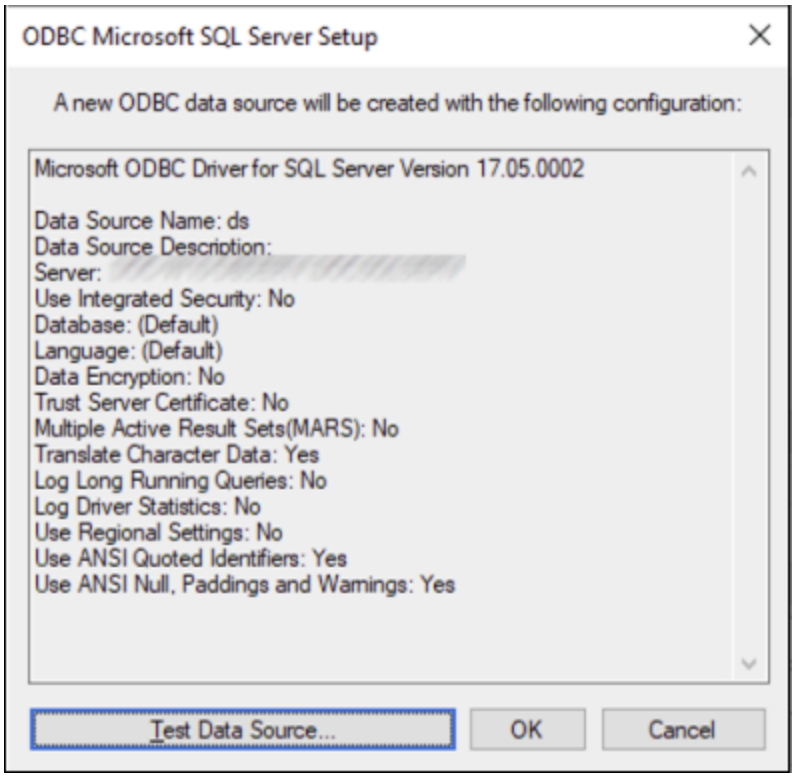


9. Cliquez sur **Suivant** et, en option, configurez des paramètres supplémentaires pour cette connexion.

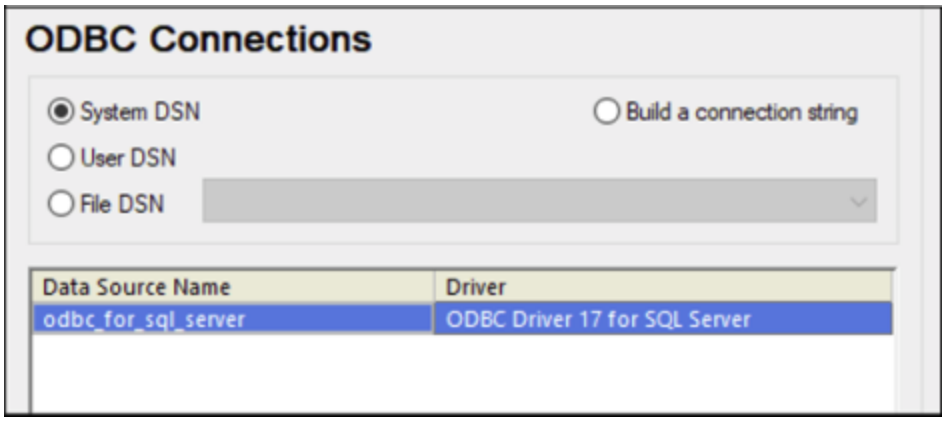


10. Cliquez sur **Terminer**. Un dialogue de confirmation recensant les détails de connexion s'ouvre.

© 2016-2022 Altova GmbH



11. Cliquez sur **OK**. La source de données apparaît maintenant dans la liste de sources de données **Utilisateur** ou **Système**, tel que configuré, par exemple :



19.1.10.12 MySQL (ODBC)

Cet exemple illustre comment se connecter à un serveur de base de données MySQL depuis une machine Windows par le biais du pilote ODBC. Le pilote ODBC MySQL n'est pas disponible sur Windows, il doit donc être téléchargé et installé séparément. Cet exemple utilise MySQL Connecteur/ODBC 8.0.

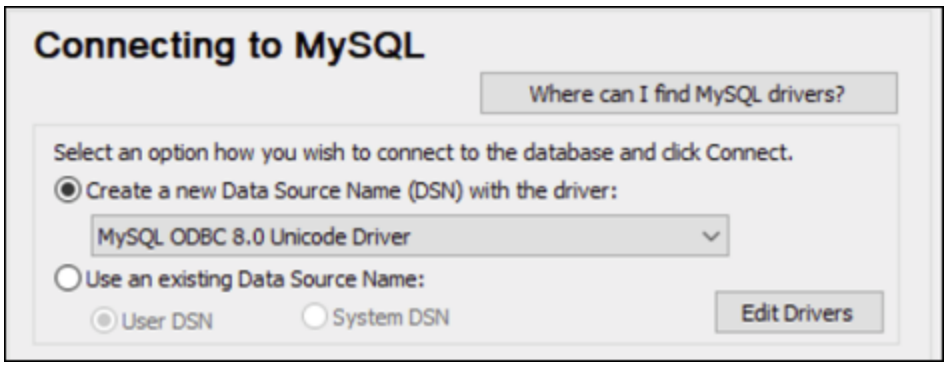
Exigences préalables :

- Le pilote ODBC MySQL doit être installé sur votre système d'exploitation. Vérifiez la documentation MySQL pour la version du pilote recommandée de votre version de serveur de base de données (voir https://dev.mysql.com/downloads/connector/odbc/).
- Vous disposez des détails de connexion à la base de données suivants : hôte, base de données, port, nom utilisateur et mot de passe.

Si vous avez installé MySQL Connector/ODBC pour la plateforme 64-bit, assurez-vous d'installer aussi XMLSpy pour la plateforme 64-bit.

Pour se connecter à MySQL via ODBC :

1. Lancer l'assistant de connexion de base de données.
2. Sélectionnez **MySQL (ODBC)**, puis cliquez sur **Suivant**.

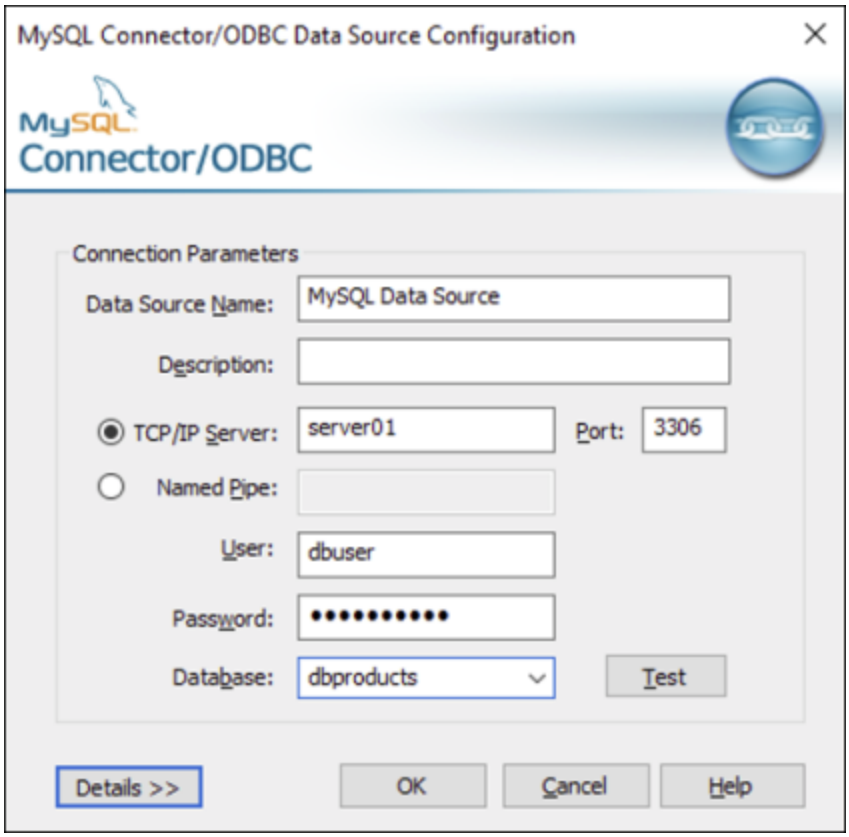


3. Sélectionnez **Créer un nouveau Nom de source de données (DSN) avec le pilote**, puis sélectionnez un pilote MySQL. Si aucun pilote MySQL n'est disponible dans la liste, cliquez sur **Éditer les pilotes**, puis sélectionnez tout pilote MySQL disponible (la liste contient tous les pilotes ODBC installés sur votre système d'exploitation).

   Si vous avez installé XMLSpy 64-bit, alors les pilotes ODBC 64-bit sont affichés dans la liste. Sinon, les pilotes ODBC 32-bit sont affichés. Voir aussi Consulter les pilotes ODBC disponibles.

4. Cliquez sur **Connexion**.

© 2016-2022 Altova GmbH



5. Dans le champ du nom de la source de données, saisissez un nom descriptif qui vous permettra d'identifier cette source de données ODBC à l'avenir.
6. Remplir les identifiants de connexion de base de données (Serveur TCP/IP, Utilisateur, Mot de passe), sélectionnez une base de données et cliquez sur **OK**.

Note : si le serveur de base de données est à distance, il doit être configuré par l'administrateur du serveur pour accepter les connexions à distance depuis l'adresse IP de votre appareil. De même, si vous cliquez sur **Détails>>**, vous trouverez plusieurs paramètres supplémentaires disponibles pour la configuration. Vérifiez la documentation du pilote avant de changer leurs valeurs par défaut.

19.1.10.13 Oracle (JDBC)

Cet exemple montre comment vous connecter à un serveur de base de données Oracle à partir d'une machine client, à l'aide de l'interface JDBC. La connexion est créée en tant qu'une connexion Java pure, par le biais de **Oracle Instant Client Package (Basic)** disponible sur le site Internet Oracle. L'avantage de ce type de connexion est qu'elle ne nécessite que l'environnement Java et les bibliothèques .jar fournies par l'Oracle Instant Client Package, vous évitant d'installer et de configurer un client de base de données plus complexe.

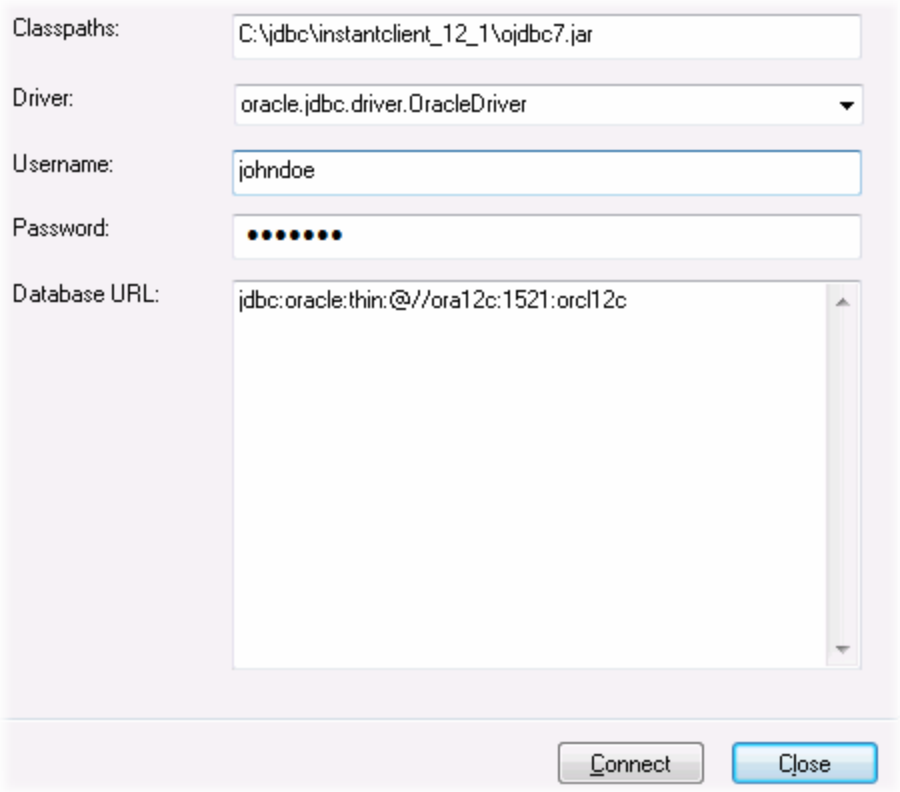
Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le

chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) correspond à celle du JRE/JDK.
- L'**Oracle Instant Client Package (Basic)** doit être disponible sur votre système d'exploitation. Le paquet peut être téléchargé depuis le site Internet officiel d'Oracle. Cet exemple utilise l'Oracle Instant Client Package version 12.1.0.2.0, pour Windows 32-bit et par conséquent, Oracle JDK 32-bit.
- Vous disposez des détails de connexion à la base de données suivants : hôte, port, nom de service, nom d'utilisateur et mot de passe.

Pour se connecter à Oracle par le biais de Instant Client Package :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **JDBC Connections**.
3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : **C:\jdbc\instantclient_12_1\ojdbc7.jar**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans le champ « Pilote », sélectionnez soit **oracle.jdbc.OracleDriver**, soit **oracle.jdbc.driver.OracleDriver**. Veuillez noter que ces saisies sont disponibles si un chemin d'accès .jar valid est trouvé soit dans le champ de saisie « Classpath », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir étape précédente).
5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.



 © 2016-2022 Altova GmbH

6. Saisissez la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

```
jdbc:oracle:thin:@//host:port:service
```

7. Cliquez sur **Connexion**.

19.1.10.14 Oracle (ODBC)

Cet exemple illustre un scénario commun au cours duquel vous vous connectez depuis XMLSpy à un serveur de base de données Oracle sur une machine de réseau par le biais d'un client de base de données Oracle installé sur le système d'exploitation local.

L'exemple comprend des instructions pour configurer une source de données ODBC (DSN) en utilisant l'assistant de connexion de base de données dans XMLSpy. Si vous avez déjà créé un DSN, ou si vous préférez le créer directement depuis l'**administrateur de ODBC Data Source** dans Windows, vous pouvez le faire, puis le sélectionner si vous êtes invité à le faire par l'assistant. Pour plus d'informations concernant les sources de données ODBC, voir Configurer une connexion ODBC.

Exigences préalables :

- Le client de base de données Oracle (qui inclut le pilote ODBC Oracle) doit être installé et configuré sur votre système d'exploitation. Pour obtenir des instructions d'installation et de configuration d'un client de base de données Oracle, veuillez vous référer à la documentation fournie avec votre logiciel Oracle.
- Le fichier **tnsnames.ora** situé dans le répertoire personnel Oracle contient une entrée qui décrit les paramètres de connexion de la base de données dans un format semblable au suivant :

```
ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server01)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = orcl)
      (SERVER = DEDICATED)
    )
  )
```

Le chemin d'accès menant au fichier **tnsnames.ora** dépend de l'emplacement où le répertoire personnel Oracle a été installé. Pour le client de base de données Oracle 11.2.0, le chemin d'accès du répertoire personnel Oracle par défaut pourrait être comme suit :

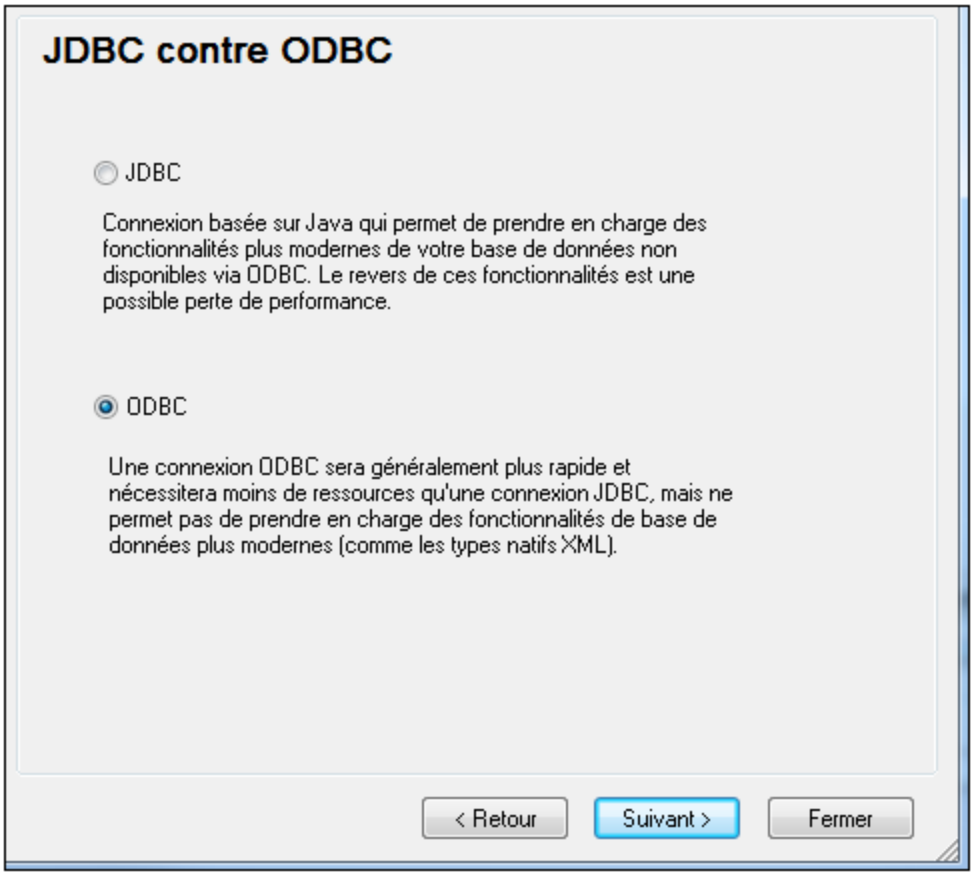
```
C:\app\username\product\11.2.0\client_1\network\admin\tnsnames.ora
```

Vous pouvez ajouter de nouvelles entrées au fichier **tnsnames.ora** soit en collant les détails de connexion et en enregistrant le fichier, ou en exécutant l'assistant d'Oracle *Net Configuration Assistant* (si disponible). Si vous voulez que ces valeurs apparaissent dans des listes déroulantes

durant le processus de configuration, alors vous allez éventuellement devoir ajouter le chemin au dossier admin en tant que variable d'environnement**TNS_ADMIN**.

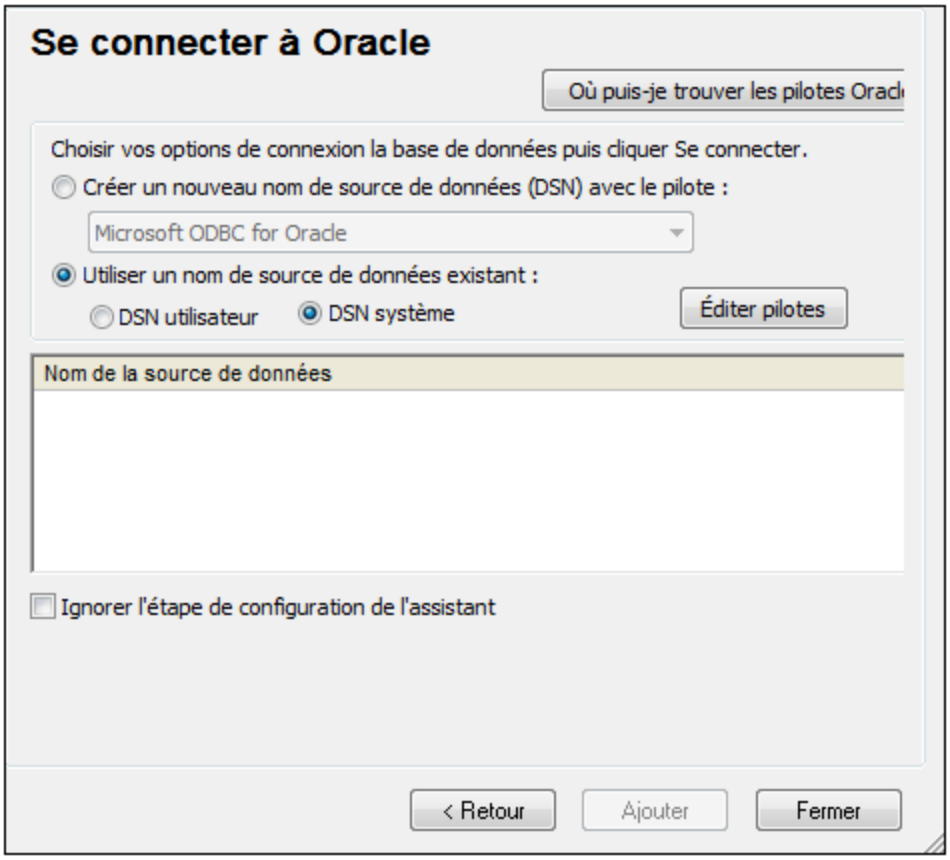
Pour se connecter à Oracle avec ODBC :

1. Lancer l'assistant de connexion de base de données.
2. Sélectionnez **Oracle (ODBC / JDBC)**, puis cliquez sur **Suivant**.

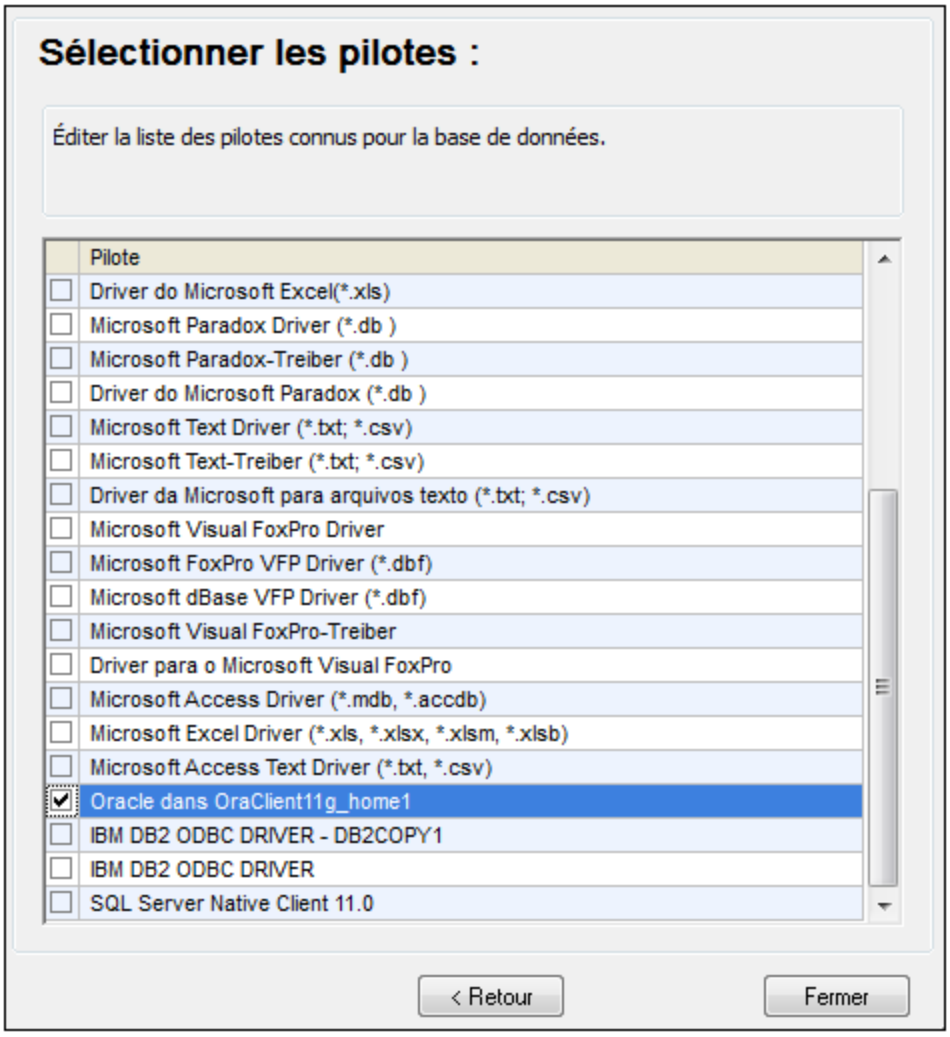


3. Sélectionnez **ODBC**.

© 2016-2022 Altova GmbH

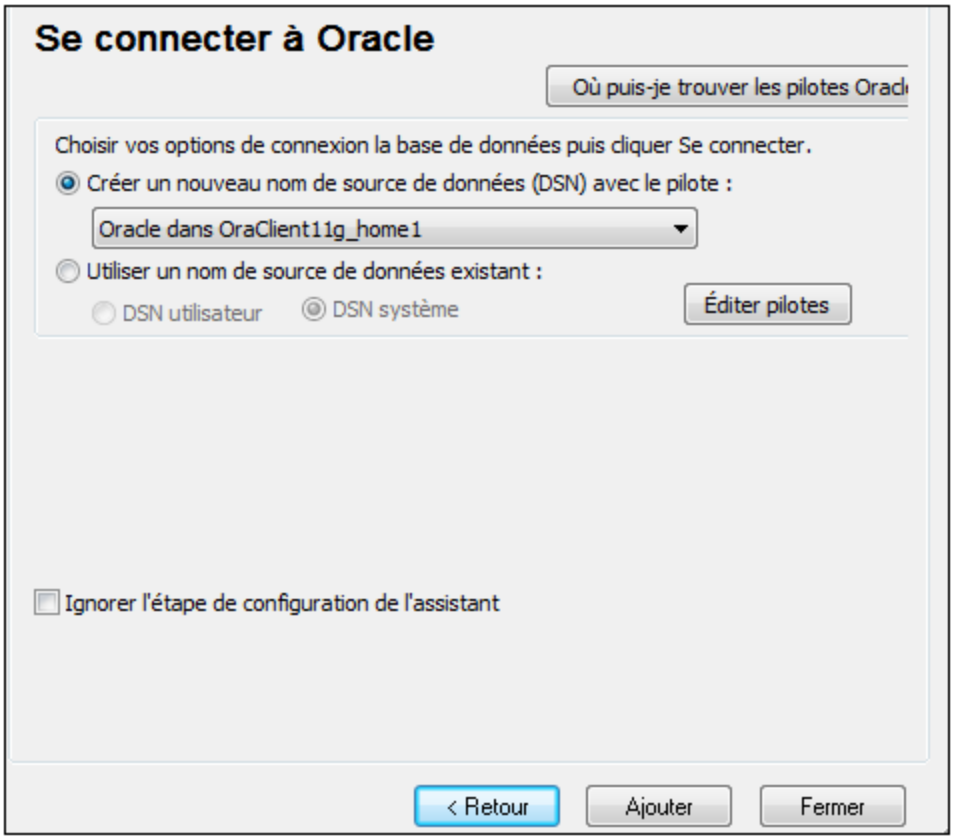


4. Cliquez sur **Éditer les pilotes**.



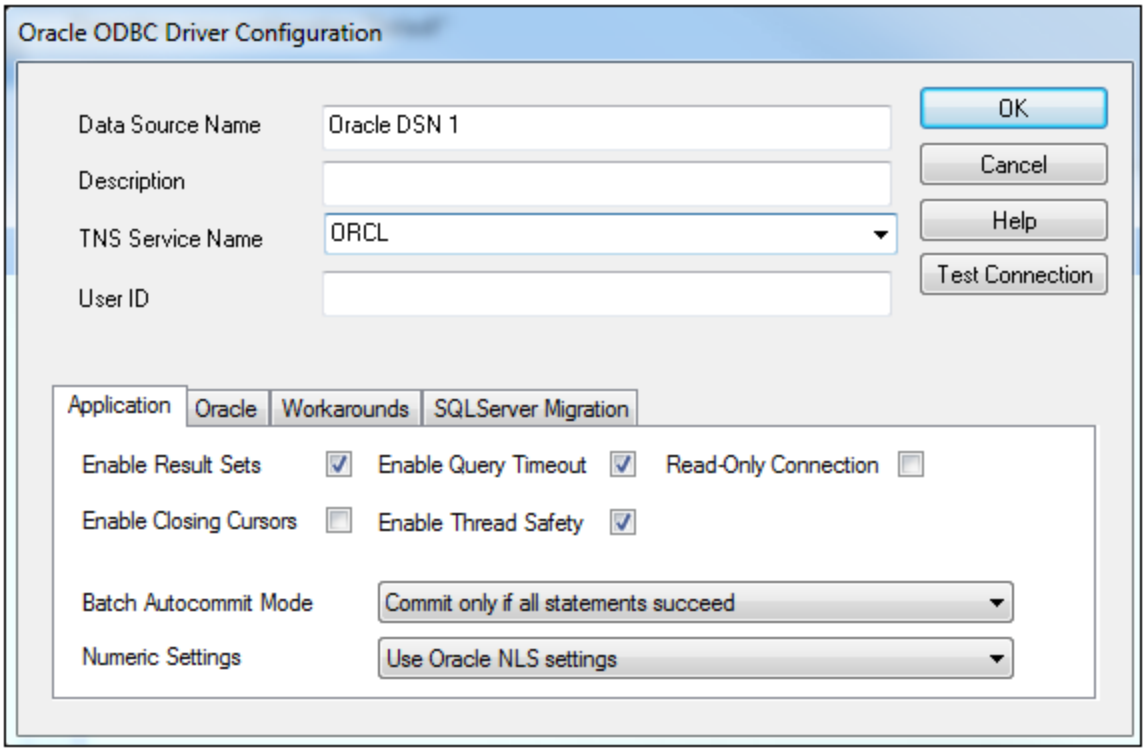
5. Sélectionnez les pilotes Oracle que vous souhaitez utiliser (dans cet exemple, **Oracle dans OraClient11g_home1**). La liste affiche les pilotes disponibles sur votre système après l'installation du client Oracle.
6. Cliquez sur **Retour**.
7. Sélectionner **Créer un nouveau nom de source de données (DSN) avec le pilote**, puis sélectionner le pilote Oracle choisi dans l'étape .4

© 2016-2022 Altova GmbH

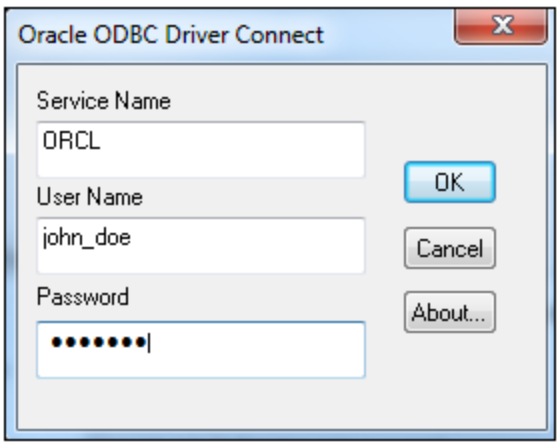


Évitez d'utiliser le pilote fourni par Microsoft appelé **Microsoft ODBC for Oracle**. Microsoft recommande d'utiliser le pilote ODBC fourni par Oracle (voir http://msdn.microsoft.com/en-us/library/ms714756%28v=vs.85%29.aspx)

8. Cliquez sur **Se connecter**.



9. Dans le champ de saisie Nom de source de données, saisissez un nom pour identifier la source de données (dans cet exemple, **Oracle DSN 1**).
10. Dans le champ TNS Service Name, saisissez le nom de connexion tel qu'il est défini dans le fichier **tnsnames.ora** (voir conditions préalables). Dans cet exemple, le nom de connexion est **ORCL**. *Note :* Si vous souhaitez avoir une liste déroulante de la combo box peuplée avec les valeurs du fichier **tnsnames.ora**, alors vous allez éventuellement devoir ajouter le chemin au dossier admin en tant que variable d'environnement **TNS_ADMIN**.
11. Cliquez sur **OK**.



12. Saisissez le nom d'utilisateur et le mot de passe de la base de données, puis cliquez sur OK.

 © 2016-2022 Altova GmbH

19.1.10.15 PostgreSQL (ODBC)

Cet exemple illustre comment se connecter à un serveur de base de données PostgreSQL depuis un appareil Windows machine par le biais des pilotes ODBC. Le pilote ODBC PostgreSQL n'est pas disponible sur Windows, il doit donc être téléchargé et installé séparément. Cet exemple utilise le pilote psqlODBC (version 11.0) téléchargé depuis le site web officiel (voir aussi Aperçu des pilotes de base de données).

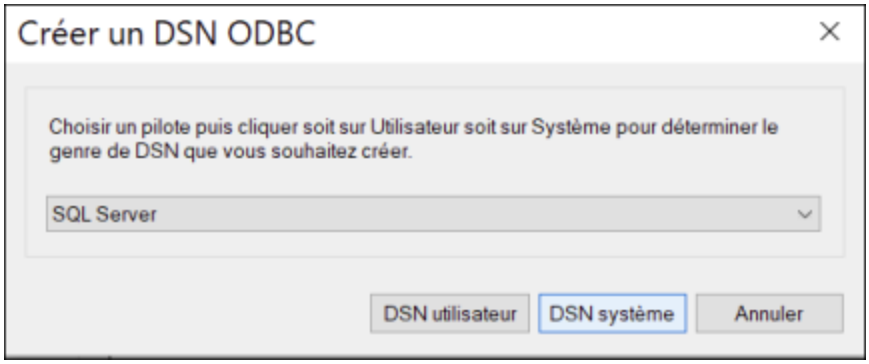
Note : vous pouvez aussi vous connecter à un serveur de base de données PostgreSQL directement (sans le pilote ODBC), voir Configurer une connexion PostgreSQL.

Exigences préalables :

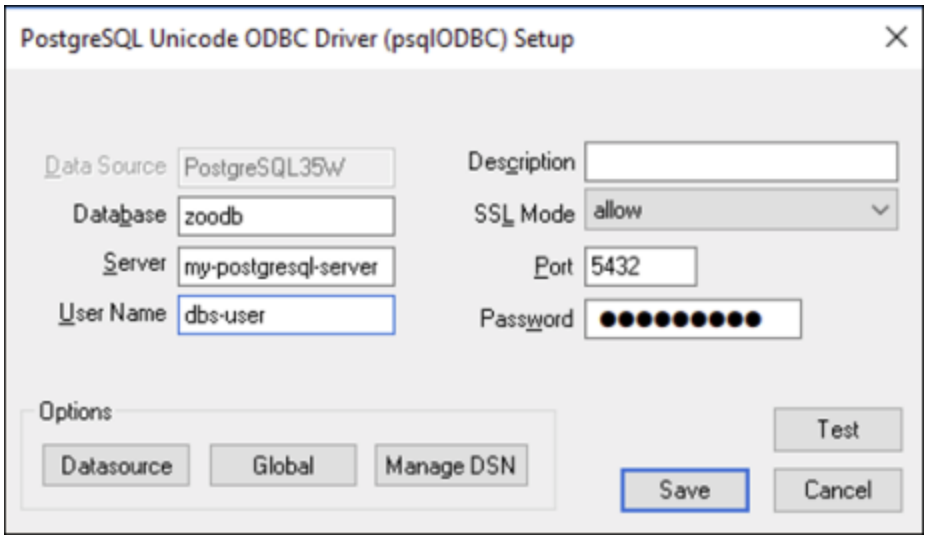
- Le pilote *psqlODBC* doit être installé sur votre système d'exploitation.
- Vous disposez des détails de connexion à la base de données suivants : serveur, port, base de données, nom d'utilisateur et mot de passe.

Pour établir une connexion à PostgreSQL avec ODBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **Connexions ODBC**.
3. Sélectionnez l'option **DSN Utilisateur**.
4. Cliquez sur **Créer un nouveau DSN** et sélectionnez le pilote depuis la liste déroulante. Si aucun pilote PostgreSQL n'est disponible dans la liste, veuillez vous assurer que le pilote ODBC PostgreSQL est installé sur votre système d'exploitation, comme mentionné dans les conditions préalables ci-dessus.

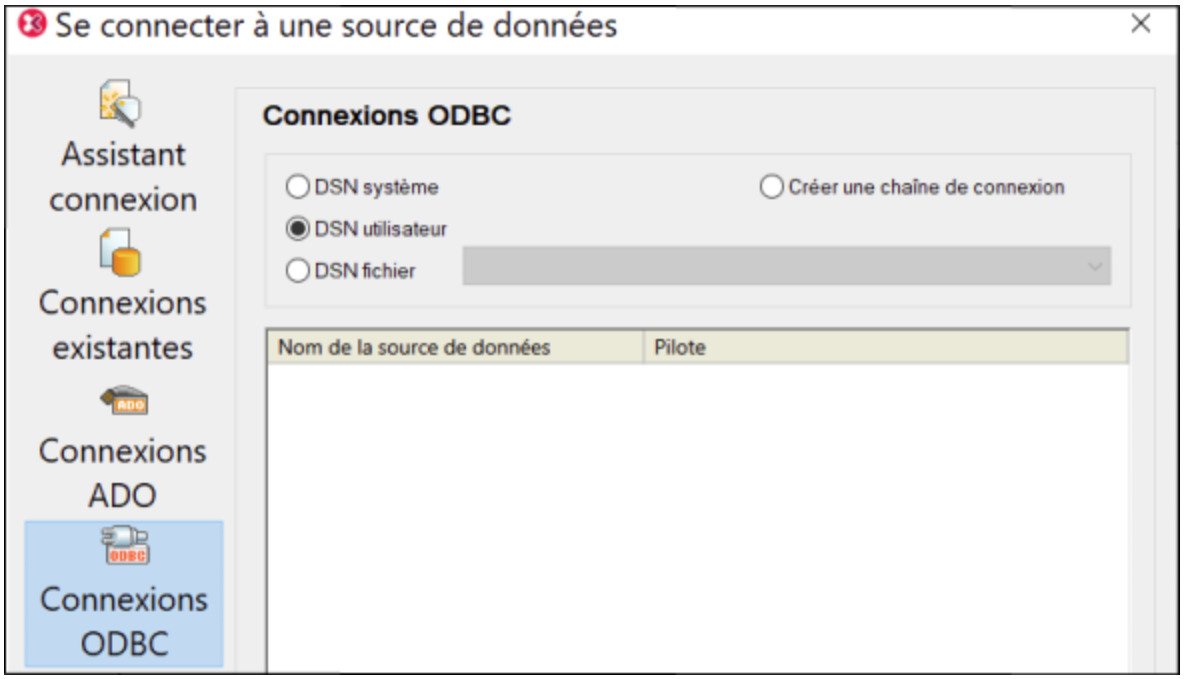


5. Cliquez sur **DSN Utilisateur**.



6. Remplissez les identifiants de connexion à la base de données (ils doivent être fournis par le propriétaire de la base de données), puis cliquez sur **Enregistrer**.

La connexion est désormais disponible dans la liste des connexions ODBC. Pour se connecter à la base de données, vous pouvez soit double-cliquer sur la connexion ou la sélectionner, puis cliquer sur **Connexion**.



19.1.10.16 Progress OpenEdge (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données Progress OpenEdge 11.6 par le biais de JDBC.

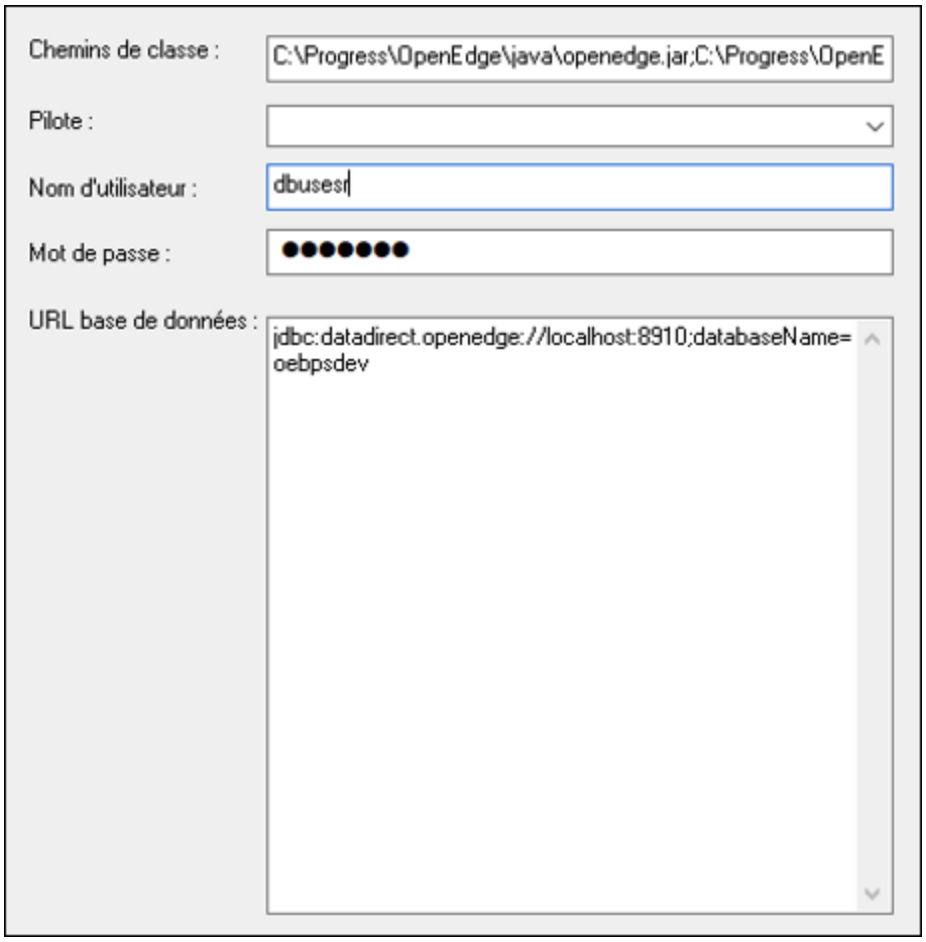
© 2016-2022 Altova GmbH

Prérequis

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK, soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Veuillez vous assurer que la plateforme de XMLSpy (32-bit, 64-bit) correspond à celle du JRE/JDK.
- La variable d'environnement `PATH` du système d'exploitation doit inclure le chemin vers le répertoire `bin` du répertoire d'installation JRE ou JDK, par exemple `C:\Program Files (x86)\Java\jre1.8.0_51\bin`.
- Le pilote Progress OpenEdge JDBC doit être disponible sur votre système d'exploitation. Dans cet exemple, la connectivité JDBC est fournie par les fichiers de composant de pilote **openedge.jar** et **pool.jar** disponibles dans **C:\Progress\OpenEdge\java** en tant que partie intégrante de l'installation OpenEdge SDK.
- Vous disposez des détails de connexion à la base de données suivants : hôte, port, nom de la base de données, nom d'utilisateur et mot de passe.

Pour se connecter à OpenEdge par le biais de JDBC

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **JDBC Connections**.
3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, les chemins de fichier .jar exigés se trouvent sous : **`C:\Progress\OpenEdge\java\openedge.jar;C:\Progress\OpenEdge\java\pool.jar;`**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le chemin d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans le champ « Driver », choisir **com.ddtek.jdbc.openedge.OpenEdgeDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
6. Saisissez la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

```
jdbc:datadirect:openedge://host:port;databaseName=db_name
```

7. Cliquez sur **Connexion**.

19.1.10.17 Progress OpenEdge (ODBC)

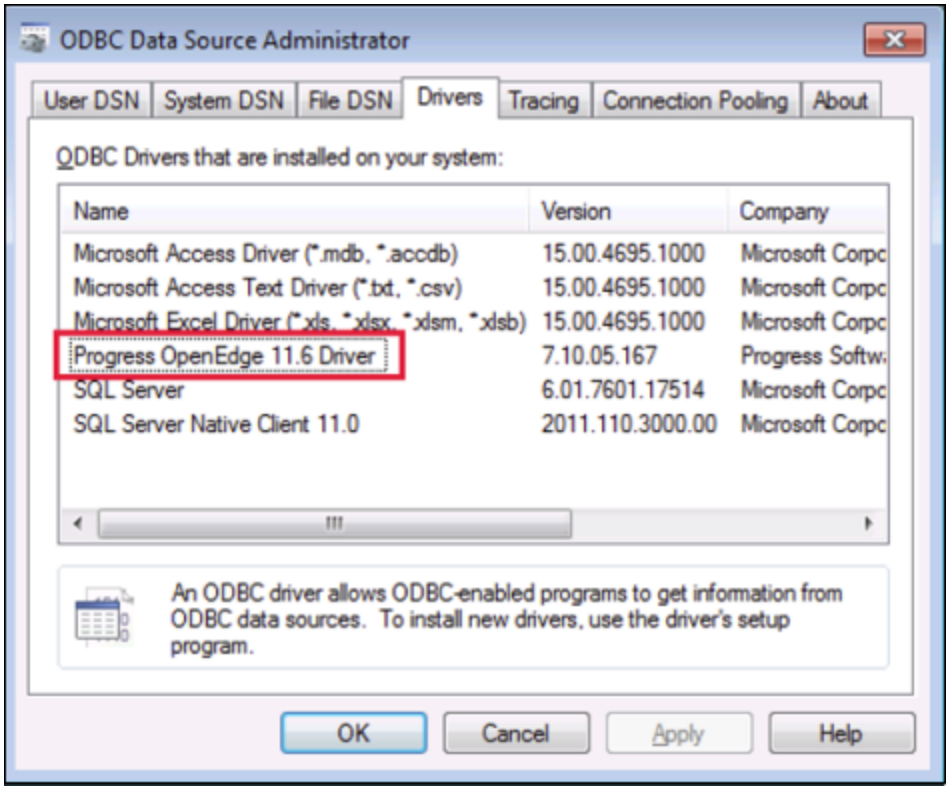
Cet exemple illustre comment se connecter à un serveur de base de données Progress OpenEdge par le biais du pilote Progress OpenEdge 11.6 ODBC.

Exigences préalables :

- Le pilote *ODBC Connector for Progress OpenEdge* doit être installé sur votre système d'exploitation. Le pilote Progress OpenEdge ODBC peut être téléchargé depuis le site web du vendeur (voir aussi

© 2016-2022 Altova GmbH

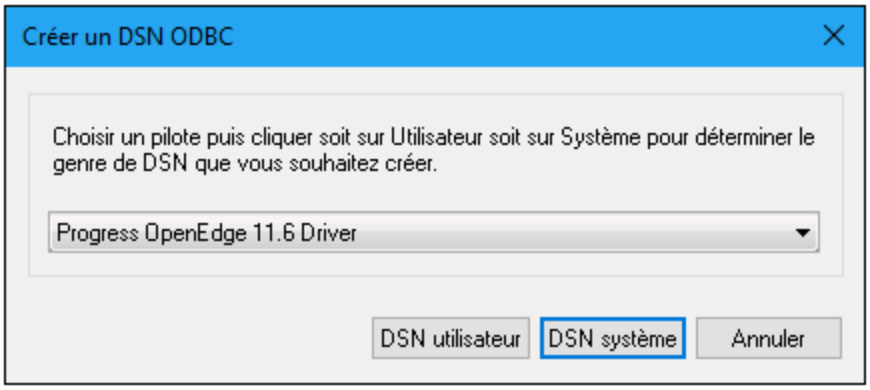
Aperçu des pilotes de base de données). Veillez à télécharger le pilote 32-bit lorsque vous exécutez la version 32-bit de XMLSpy, et le pilote 64-bit lorsque vous exécutez la version 64-bit. Après l'installation, vérifiez si le pilote ODBC est disponible sur votre appareil (voir aussi Aperçu des pilotes ODBC disponibles).



- Vous disposez des détails de connexion de base de données suivants : nom d'hôte, numéro de port, nom de base de données, ID utilisateur et mot de passe.

Pour se connecter à Progress OpenEdge par le biais d'ODBC

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **Connexions ODBC**.
3. Cliquez sur**DSN Utilisateur** (en alternative, cliquez sur **DSN Système** ou **DSN Fichier**, dans un tel cas, les instructions suivantes seront semblables).
4. Cliquez sur **Ajouter** .
5. Sélectionnez le **pilote Progress OpenEdge** depuis la liste et cliquez sur **DSN Utilisateur** (ou **DSN Système**, le cas échéant).



6. Remplissez les identifiants de connexion à la base de données (base de données, serveur, port, nom utilisateur, mot de passe) et cliquez sur **OK**. Pour vérifier la connectivité avant d'enregistrer les données saisies, cliquez sur **Tester la connexion**.



7. Cliquer sur OK. La nouvelle source de données apparait désormais dans la liste des sources de données d'ODBC.

© 2016-2022 Altova GmbH



8. Cliquez sur **Se connecter**.

19.1.10.18 Sybase (JDBC)

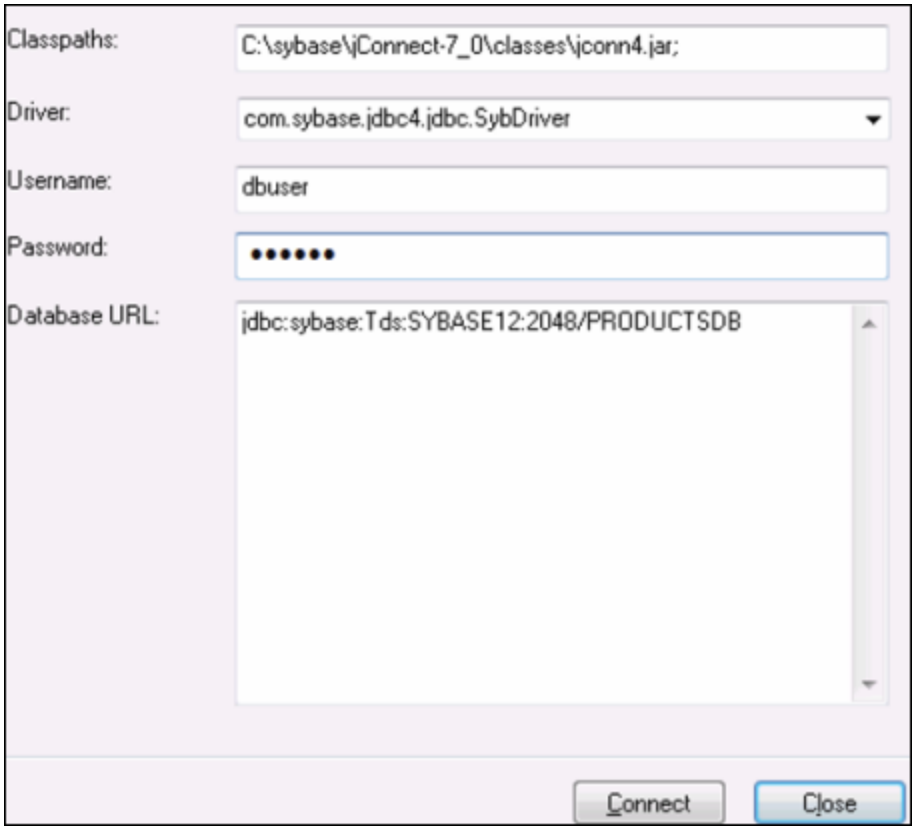
Cet exemple illustre comment se connecter à un serveur de base de données Sybase par le biais de JDBC.

Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK, soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Assurez-vous que la plateforme de XMLSpy (32-bit, 64-bit) correspond à celle du JRE/JDK.
- Le composant Sybase *jConnect* doit être installé sur votre système d'exploitation (dans cet exemple, *jConnect 7.0* est utilisé, installé en tant que partie intégrante de l'installation *Sybase Adaptive Server Enterprise PC Client*). En ce qui concerne les instructions d'installation du client de base de données, référez-vous à la documentation Sybase.
- Vous disposez des détails de connexion à la base de données suivants : hôte, port, nom de la base de données, nom d'utilisateur et mot de passe.

Connexion à Sybase par le biais de JDBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **JDBC Connections**.
3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le chemin de fichier .jar exigé se trouve sous : **C:\sybase\jConnect-7_0\classes\jconn4.jar**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans la case "Driver", choisir **com.sybase.jdbc4.jdbc.SybDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
6. Saisissez la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

```
jdbc:sybase:Tds:hostName:port/databaseName
```

7. Cliquez sur **Connexion**.

© 2016-2022 Altova GmbH

19.1.10.19 Teradata (JDBC)

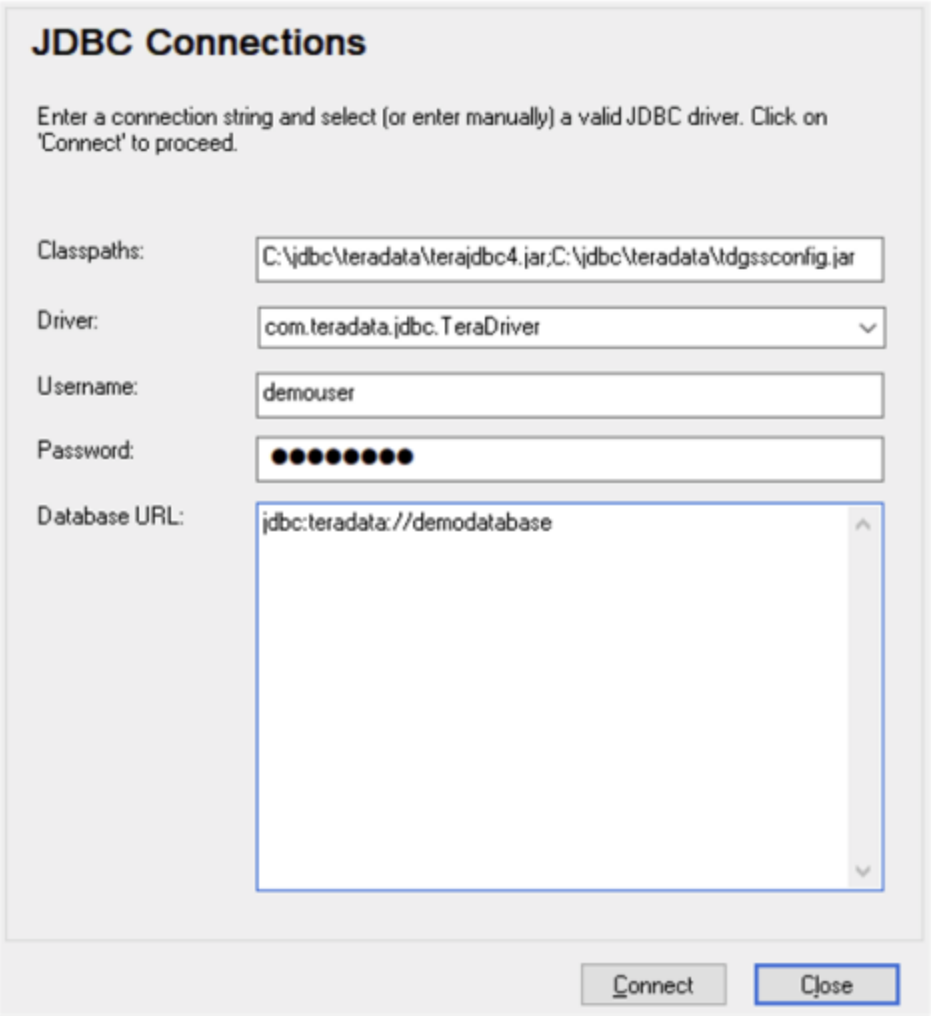
Cet exemple indique comment se connecter à un serveur de base de données Teradata avec JDBC.

Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK, soit d'un build open source comme Oracle OpenJDK. XMLSpy déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement `JAVA_HOME`.
- Assurez-vous que la plateforme de XMLSpy (32-bit, 64-bit) correspond à celle du JRE/JDK.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui fournissent une connectivité à la base de données) doit être installé sur votre système d'exploitation. Dans cet exemple, le pilote JDBC Teradata 16.20.00.02 est utilisé. Pour plus d'informations, voir https://downloads.teradata.com/download/connectivity/jdbc-driver.
- Vous disposez des détails de connexion à la base de données suivants : hôte, base de données, port, nom utilisateur et mot de passe.

Connexion à Teradata par le biais de JDBC :

1. Lancer l'assistant de connexion de base de données.
2. Cliquez sur **JDBC Connections**.
3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : **C:\jdbc\teradata**. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH).
4. Dans le champ « Pilote », choisir **com.teradata.jdbc.TeraDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).



5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
6. Saisissez la chaîne de connexion du serveur de la base de données dans le champ de saisie Database URL, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

```
jdbc:teradata://databaseServerName
```

7. Cliquez sur **Connexion**.

© 2016-2022 Altova GmbH

19.1.10.20 Teradata (ODBC)

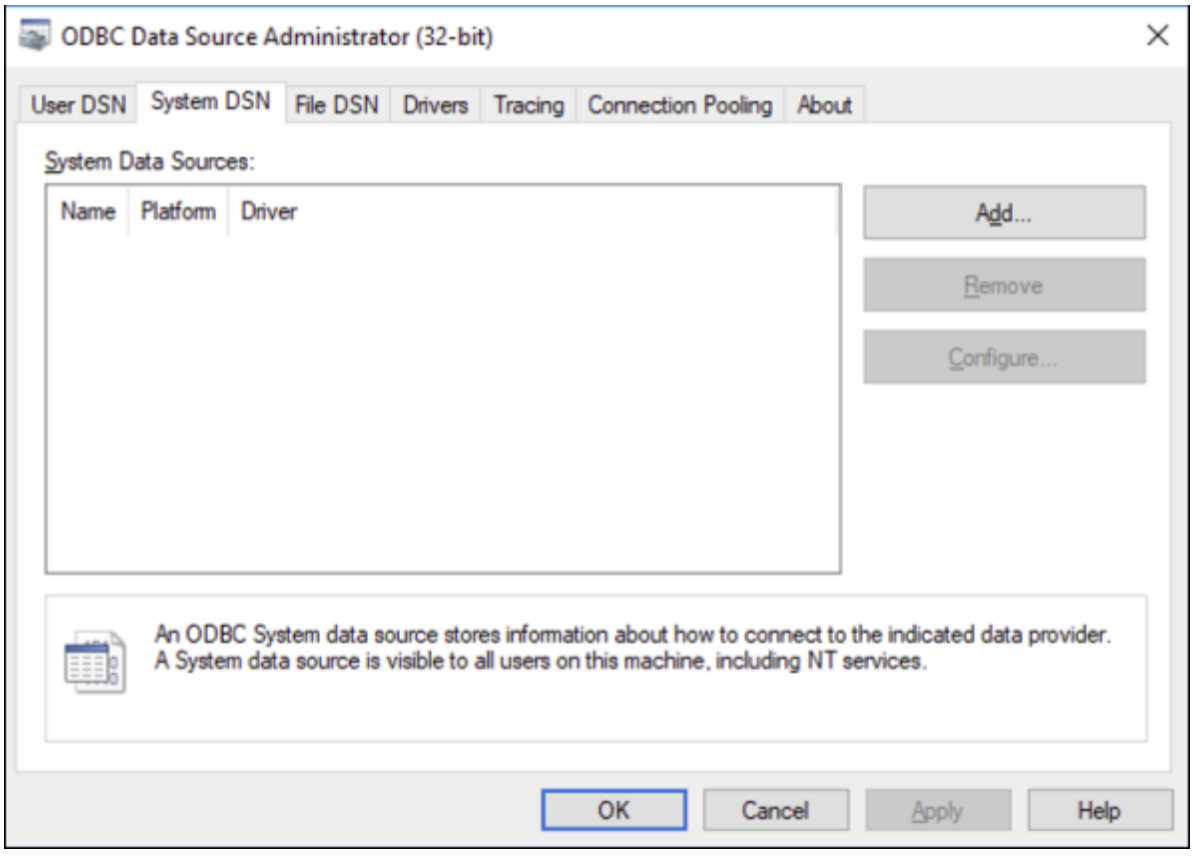
Cet exemple indique comment se connecter à un serveur de base de données Teradata avec ODBC.

Exigences préalables :

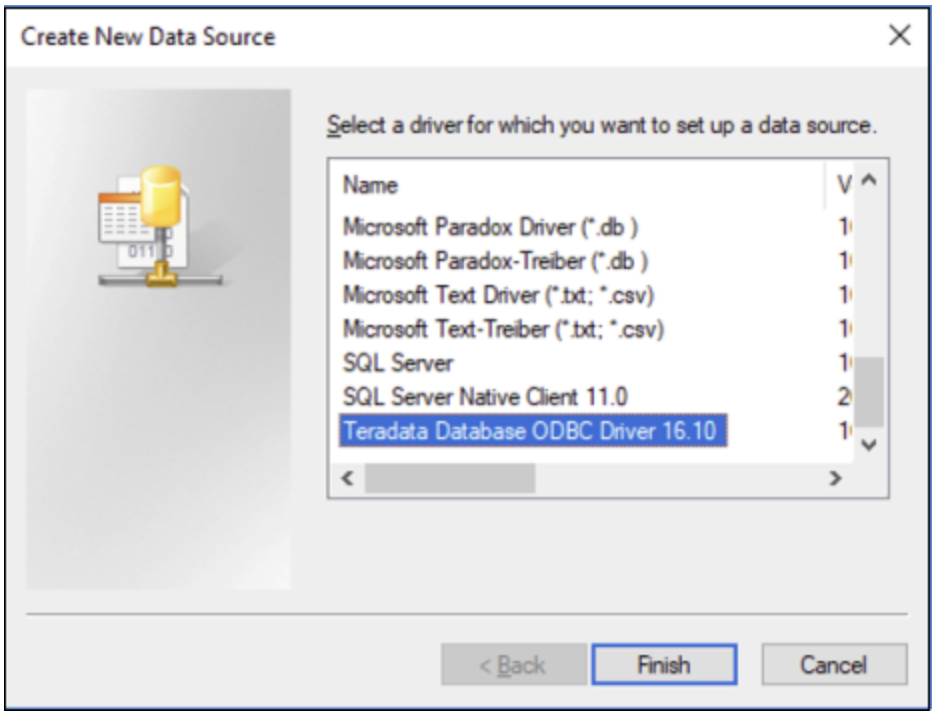
- Le pilote Teradata ODBC doit être installé (voir https://downloads.teradata.com/download/connectivity/odbc-driver/windows. Cet exemple utilise le pilote Teradata ODBC pour Windows version 16.20.00.
- Vous nécessitez les détails de connexion de base de données suivants : hôte, port, nom d'utilisateur et mot de passe.

Connexion à Teradata par le biais de ODBC :

1. Appuyez sur la clé **Windows**, saisissez « ODBC » et sélectionnez **Configurer les sources de données ODBC (32-bit)** depuis la liste de suggestions. Si vous disposez d'un pilote 64-bit, sélectionnez **Configurer les sources de données ODBC (64-bit)** et utilisez XMLSpy 64-bit aux étapes suivantes.

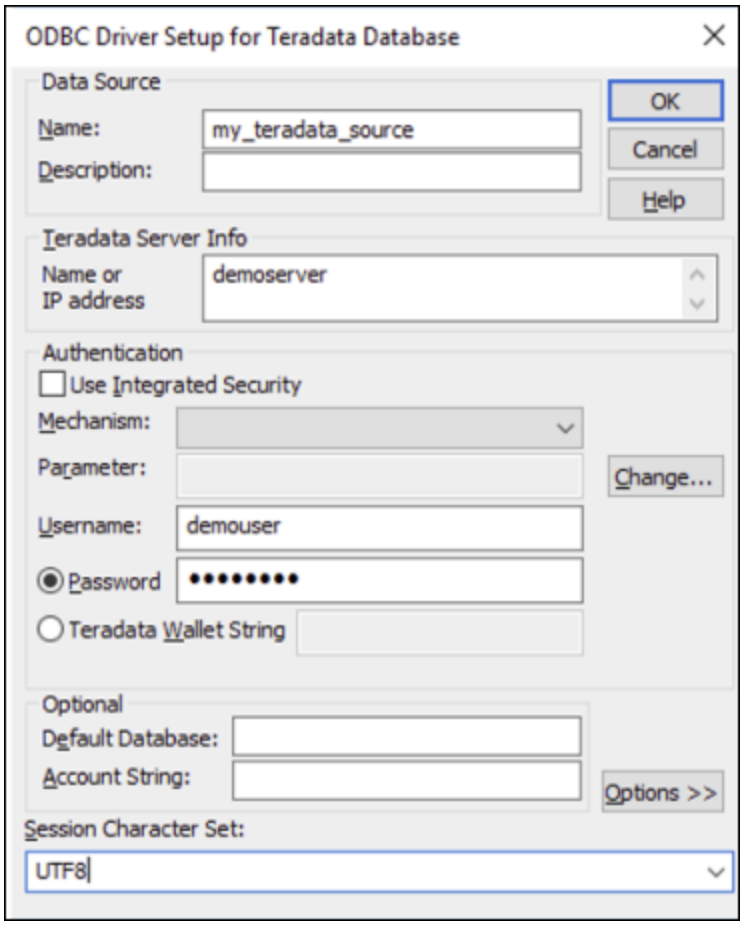


2. Cliquez sur **System DSN**, puis cliquez sur **Ajouter**

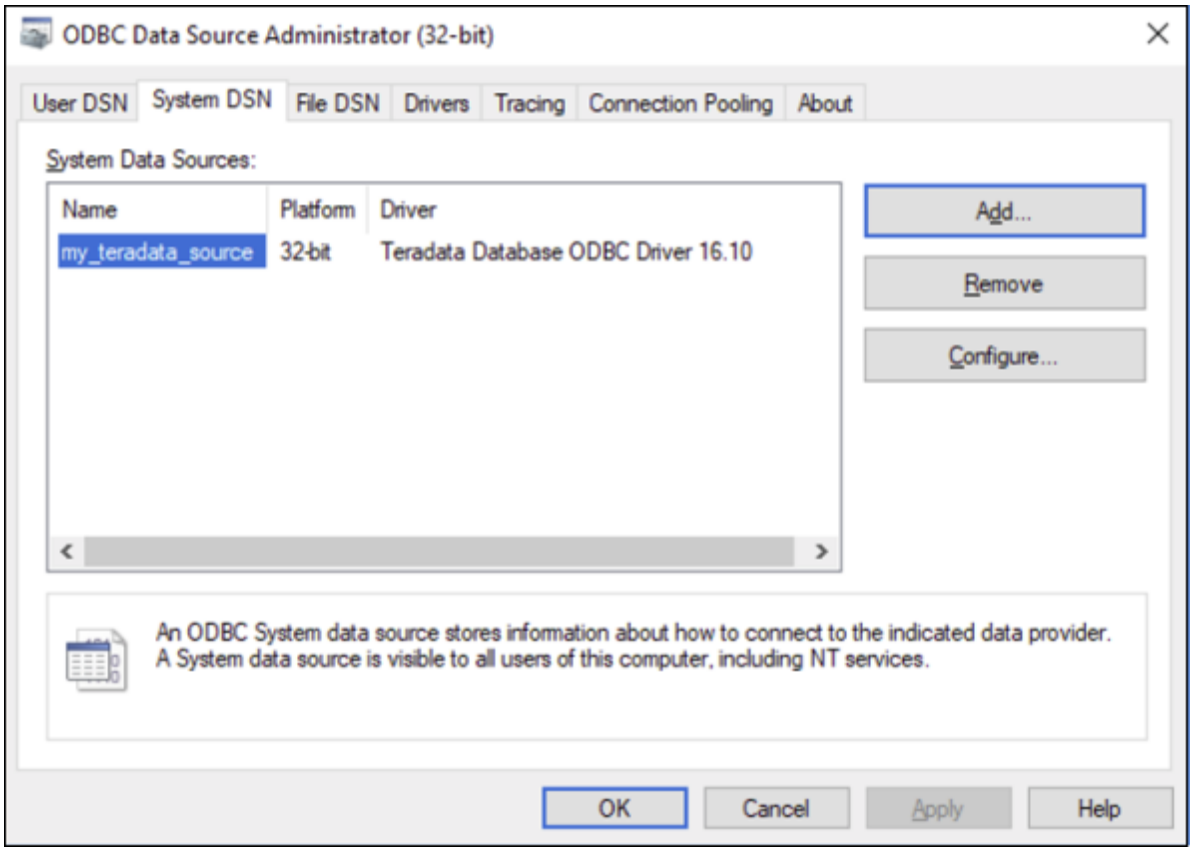


3. Choisissez le **pilote Teradata Database ODBC** et cliquez sur **Terminer**.

© 2016-2022 Altova GmbH

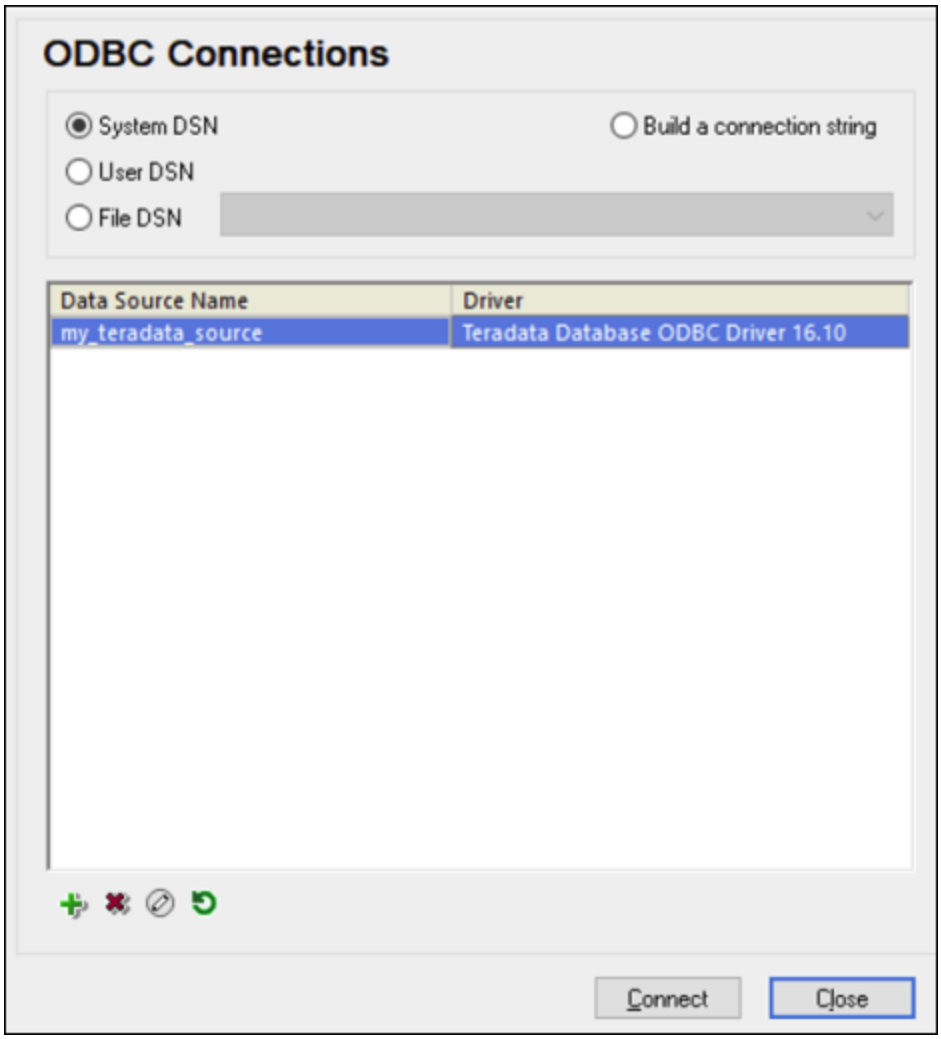


4. Saisir le nom et, éventuellement, une description qui vous aidera à identifier cette source de données ODBC à l'avenir. De même, saisissez les identifiants de connexion de base de données (serveur de base de données, utilisateur, mot de passe), et, en option, choisissez une base de données.
5. Cliquez sur **OK**. La source de données apparaît maintenant dans la liste.



6. Exécutez XMLSpy et lancez l'assistant de connexion à la base de données.
7. Cliquez sur **Connexions ODBC**.

© 2016-2022 Altova GmbH



8. Cliquez sur **System DSN**, sélectionnez la source de données créée précédemment, puis cliquez sur **Se connecter**.

Note : si vous obtenez l'erreur suivante : « Le pilote a retourné (ou n'a pas pu retourné) SQL_DRIVER_ODBC_VER: 03.80 invalide », veuillez vous assurer que le chemin menant au client ODBC (par exemple, **C:\Program Files\Teradata\Client\16.10\bin**, si vous l'avez installé à cet endroit, par exemple) existe bien dans votre variable d'environnement PATH de votre système. Si ce chemin est manquant, ajoutez-le manuellement.

19.2 Bases de données prises en charge

La table ci-dessous énumère toutes les bases de données prises en charge. Bien qu'Altova œuvre pour prendre en charge d'autres bases de données, une connexion et un traitement des données réussis ont uniquement été testés avec les bases de données énumérées ci-dessous. Si votre application Altova est une version 64-bit, assurez-vous que vous avez accès aux pilotes de base de données 64-bit nécessaires pour la base de données spécifique à laquelle vous tentez de vous connecter.

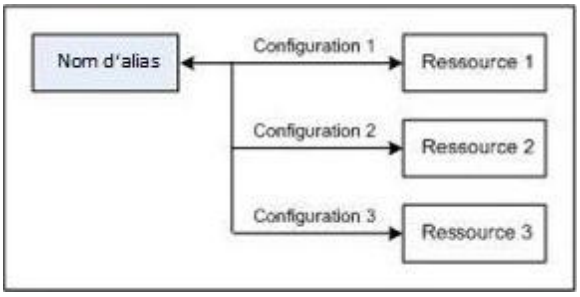
Base de données	Notes
Firebird 2.x, 3.x	
IBM DB2 DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 i6.x, 7.4	Les fichiers logiques sont pris en charge et présentés en tant qu'affichage.
IBM Informix 11.70 et ultérieur	
MariaDB 10 et ultérieur	
Microsoft Access 2003 et ultérieur	Au moment de la rédaction de ce manuel (début septembre 2019), il n'a y pas de Microsoft Access Runtime disponible pour Access 2019. Vous pouvez vous connecter à une base de données Access 2019 depuis les produits Altova uniquement si Microsoft Access 2016 Runtime est installé et uniquement si la base de données n'utilise pas le type de données « Large Number ».
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005 et ultérieur Microsoft SQL Server on Linux	
MySQL 5 et ultérieur	
Oracle 9i et ultérieur	
PostgreSQL 8 et ultérieur	Les connexions PostgreSQL sont prises en charge aussi bien en tant que connexions natives qu'en tant que connexions sur base pilote par le biais d'interfaces (pilotes) comme ODBC ou JDBC. Les connexions natives ne nécessitent pas de pilotes.
Progress OpenEdge 11.6	
SQLite 3.x	Les connexions SQLite sont prises en charge en tant que connexions directes natives du fichier de base de données SQLite. Aucun pilote séparé n'est nécessaire. Dans l'affichage Authentic, les données provenant d'une base de données SQLite ne sont pas éditables. Si vous essayez d'enregistrer des données SQLite à partir de l'affichage Authentic, une boîte de messages vous informera de cette

© 2016-2022 Altova GmbH

Base de données	Notes
	limitation connue.
Sybase ASE 15, 16	
Teradata 16	

20 Ressources Globales Altova

Les ressources globales Altova sont une collection d'alias pour les ressources de fichier, dossiers et bases de données. Chaque alias peut posséder plusieurs configurations et chaque configuration mappe vers une ressource unique (*voir capture d'écran ci-dessous*). C'est pourquoi, lorsqu'une ressource globale est utilisée en tant qu'entrée, la ressource globale peut être changée dans le cadre de ses configurations. Pour ce faire, il suffit d'utiliser les commandes dans la GUI qui vous permettent de sélectionner la configuration active. Par exemple, si une feuille de style XSLT de transformation d'un document XML est attribuée via une ressource globale (un alias), nous pourrons définir plusieurs configurations pour la ressource globale, chacune pointant vers un fichier XSLT différent. Après avoir configuré la ressource globale de cette manière, une modification de la configuration changera le fichier XSLT utilisé pour la transformation.



Une ressource globale peut non seulement être utilisée pour changer de ressources dans le cadre d'une application Altova, mais peut aussi être utilisée pour générer et utiliser des ressources depuis d'autres applications Altova. Il est donc possible de générer des fichiers automatiquement dans une application Altova pour l'utiliser dans une autre application Altova. Tout ceci facilite et accélère considérablement le processus de développement et de test. Par exemple, une feuille de style XSLT dans XMLSpy peut être utilisée pour transformer un fichier XML généré automatiquement par un mappage Altova MapForce.

L'utilisation des Ressources globales Altova implique deux processus :

- Définir les Ressources globales : les ressources sont définies et les définitions sont stockées dans un fichier XML. Ces ressources peuvent être partagées avec plusieurs applications Altova.
- Utiliser les Ressources globales : dans le cadre de XMLSpy, les fichiers peuvent être situés par le biais d'une ressource globale et non par le biais d'un chemin de fichier. L'avantage est que la ressource peut être modifiée en changeant la configuration active dans XMLSpy.

Les ressources globales dans d'autres produits Altova

Actuellement, les ressources globales peuvent être définies et utilisées dans les produits Altova suivants : XMLSpy, StyleVision, MapForce, Authentic Desktop, MobileTogether Designer, et DatabaseSpy.

© 2016-2022 Altova GmbH

20.1 Définir les Ressources globales

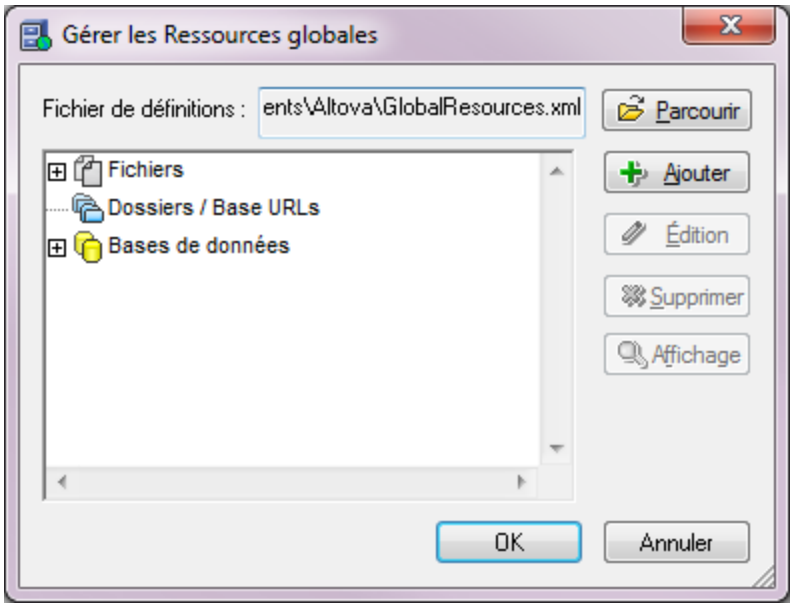
Les Ressources globales Altova sont définies dans le dialogue de Gestion des Ressources globales, qui peut être accédé de deux manières :

- Cliquer sur la commande de menu **Outils | Ressources globales**.
- Cliquer sur l'icône **Gérer les Ressources globales** dans la barre outils des Ressources globales (*capture d'écran ci-dessous*).



Le fichier de Définitions des Ressources globales

Les informations concernant les ressources globales sont stockées dans un fichier XML appelé le Fichier de Définitions des Ressources globales. Ce fichier est créé lorsque la première ressource globale est définie dans le dialogue Gérer les Ressources globales (*capture d'écran ci-dessous*) et enregistré.



Lorsque vous ouvrez le dialogue Gérer les Ressources globales pour la première fois, l'emplacement par défaut et le nom du fichier Définitions des Ressources globales est spécifié dans la fenêtre de texte *Fichier de définitions* (*voir capture d'écran ci-dessous*) :

```
C:\Users\<username>\My Documents\Altova\GlobalResources.xml
```

Ce fichier est configuré en tant que le fichier de Définitions des ressources globales par défaut pour toutes les applications Altova. Donc une ressource globale peut être enregistrée depuis n'importe quelle application Altova sur ce fichier et elle sera immédiatement disponible pour toutes les autres applications Altova en tant que ressource globale. Afin de définir et enregistrer une ressource globale dans le fichier Définition des Ressources globales, ajouter la ressource globale dans le dialogue Gérer les Ressources globales et cliquer sur **OK** pour enregistrer.

Pour sélectionner un fichier de Définitions des ressources globales existant qui deviendra le fichier de définitions actif d'une application Altova particulière, chercher le fichier à l'aide du bouton **Parcourir** de la fenêtre de texte de *Fichier de définitions* (*voir capture d'écran ci-dessus*).

Note : Vous pouvez renommer le fichier de Définitions des ressources globales comme vous le souhaitez et l'enregistrer à tout endroit accessible de vos applications Altova. Tout ce que vous devez faire dans chaque application est de spécifier ce fichier en tant que le fichier de Définitions des ressources globales pour cette application (dans la fenêtre de texte *Fichier de définitions*). Les ressources deviennent globales pour tous les produits Altova si vous utilisez un seul fichier de définitions pour tous les produits Altova.

Note : Vous pouvez aussi créer des fichiers de Définitions des ressources globales multiples. Néanmoins, seul un d'entre eux peut être actif à tout moment dans une application Altova donnée, et seules les définitions contenues dans ce fichier seront disponibles pour l'application. La disponibilité des ressources peut donc être restreinte ou peut toucher plusieurs produits, tel que vous le souhaitez.

Gérer les ressources globales : ajouter, éditer, supprimer, enregistrer

Dans le dialogue Gérer les Ressources globales (*capture d'écran ci-dessus*), vous pouvez ajouter une ressource globale au fichier de Définitions des Ressources globales sélectionné ou bien éditer ou supprimer une ressource globale sélectionnée. Le fichier de Définitions des Ressources globales organise les ressources globales que vous ajoutez dans des groupes différents : fichiers, dossiers et bases de données (*voir capture d'écran ci-dessus*).

Pour ***ajouter une ressource globale***, cliquer sur le bouton **Ajouter** et définir la ressource globale dans le dialogue appropriée **Ressource globale** qui s'affiche (*voir les descriptions des fichiers, dossiers, et bases de données dans les sous-sections de cette section*). Après avoir défini une ressource globale et l'avoir enregistrée (en cliquant sur **OK** dans le dialogue Gérer les Ressources globales), la ressource globale est ajoutée à la bibliothèque des définitions globales dans le fichier des Définitions des Ressources globales. La ressource globale sera identifiée au moyen d'un alias.

Pour ***éditer une ressource globale***, la sélectionner et cliquer sur **Éditer,** le dialogue **Ressource globale** pertinent s'affiche dans lequel vous pourrez procéder aux changements nécessaires (*voir les descriptions des fichiers, dossiers et bases de données dans les sous-sections de cette section*).

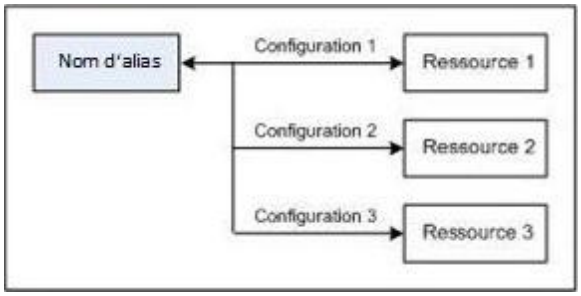
Pour ***supprimer une ressource globale***, la sélectionner et cliquer sur **Supprimer**.

Une fois que vous avez terminé l'ajout, l'édition et la suppression, n'oubliez pas de cliquer sur **OK** dans le dialogue de Gestion des Ressources globales et ***d'enregistrer vos modifications*** dans le fichier de Définition des Ressources globales.

© 2016-2022 Altova GmbH

Lier des ressources globales à des noms d'alias par le biais de configuration

La définition d'une ressource globale requiert le mappage d'un alias à une ressource (fichier, dossier ou base de données). Un seul nom d'alias peut être mappé à plusieurs ressources. Chaque mappage est appelé une configuration. Un seul nom d'alias peut donc être associé avec plusieurs ressources par le biais de configurations différentes (*capture d'écran ci-dessous*).



Dans une application Altova, vous pouvez ensuite attribuer des alias au lieu de fichiers. Pour chaque alias, vous pouvez passer entre les ressources mappées à cet alias en changeant tout simplement la configuration de la Ressource globale active de l'application (configuration active). Par exemple, dans l'application XMLSpy d'Altova, si vous souhaitez exécuter une transformation XSLT sur le document XML `MyXML.xml`, vous pouvez lui attribuer l'alias `MyXSLT` en tant que la ressource globale à utiliser pour les transformations XSLT. Dans XMLSpy, vous pouvez ensuite changer la configuration active pour utiliser plusieurs fichiers XSLT. Si `Configuration-1` mappe `First.xslt` à `MyXSLT` et que `Configuration-1` est sélectionné en tant que la configuration active, `First.xslt` sera utilisé pour la transformation. Cela vous permet d'utiliser des configurations multiples pour accéder à des ressources multiples par le biais d'un seul alias. Ce mécanisme peut être utile lors des tests et en ce qui concerne les ressources de comparaison. De plus, puisque les ressources globales peuvent être utilisées avec les produits Altova, les ressources peuvent aussi être testées et comparées sur plusieurs produits Altova.

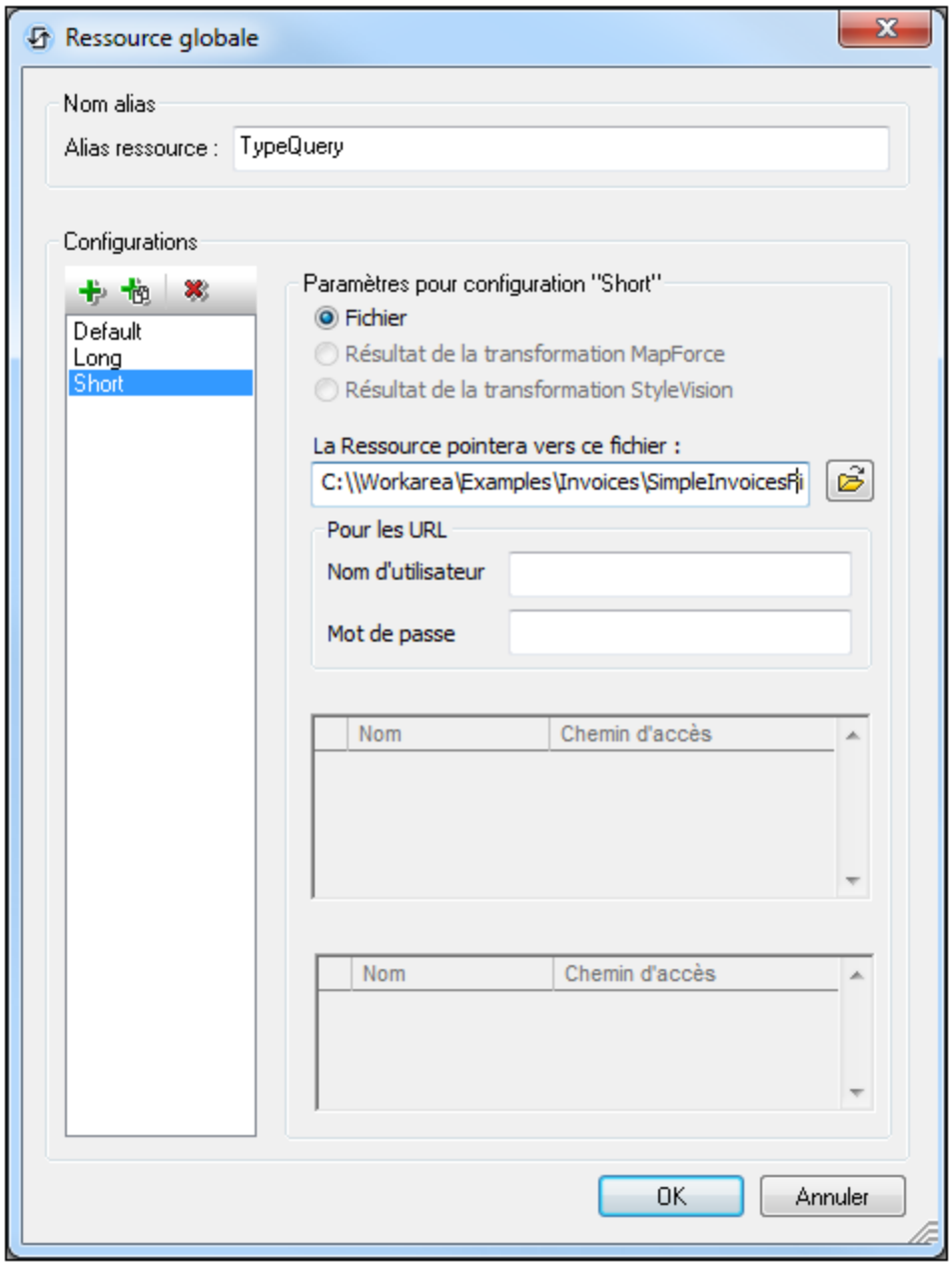
20.1.1 Fichiers

Le dialogue de Ressources globales pour Fichiers (*capture d'écran ci-dessous*) peut être accédé par le biais de la commande **Ajouter | Fichiers** dans le dialogue Gérer les Ressources globales. Dans ce dialogue, vous pouvez définir les configurations de l'alias nommé dans la fenêtre de texte *Alias de ressource*. Après avoir spécifié les propriétés des configurations tel qu'expliqué ci-dessous, enregistrer la définition de l'alias en cliquant sur **OK**.

Après avoir enregistré une définition d'alias, vous pouvez ajouter un autre alias en répétant les étapes indiquées ci-dessus (en commençant avec la commande **Ajouter | Fichiers** dans le dialogue Gérer les Ressources globales).

Dialogue Ressources globales

Un alias est défini dans le dialogue de Ressources globales (*capture d'écran ci-dessous*).



Icônes de dialogue des Ressources globales

Ajouter une configuration : ouvre le dialogue Ajouter configuration dans lequel vous pouvez saisir le nom de la configuration à ajouter.

Ajouter une configuration en tant que copie : ouvre le dialogue Ajouter configuration dans lequel vous pouvez saisir le nom de la configuration à créer en tant que copie de la configuration sélectionnée.

Supprimer : supprime la configuration sélectionnée.

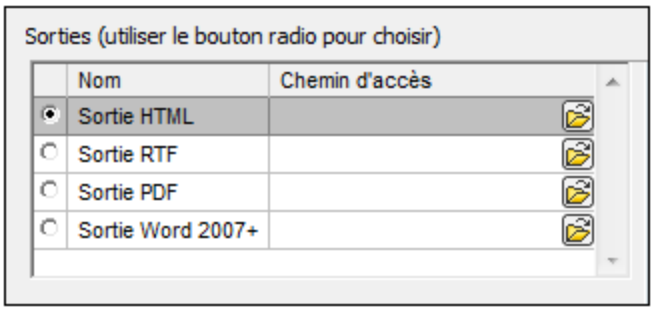
© 2016-2022 Altova GmbH

Ouvrir : chercher le fichier à créer en tant que ressource globale.

Définir l'alias

Définir l'alias (son nom et ses configurations) comme suit :

1. *Nommer l'alias :* saisir le nom de l'alias dans le champ de saisie *Alias de ressource*.
2. *Ajouter les configurations :* le panneau Configurations contiendra par défaut une configuration nommée `Default` (*voir capture d'écran ci-dessus*), qui ne peut pas être supprimée ou renommée. Vous pouvez ajouter autant de configurations supplémentaires que vous le souhaitez en : (i) cliquant sur les icônes **Ajouter configuration** ou **Ajouter configuration en tant que copie**, et (ii) en donnant à la configuration un nom dans le dialogue qui apparaît. Chaque configuration ajoutée sera affichée dans la liste des Configurations. Dans la capture d'écran ci-dessus, deux configurations supplémentaires, nommées `Long` et `Short`, ont été ajoutées à la liste Configurations. La commande Ajouter Configuration en tant que copie vous permet de copier la configuration sélectionnée puis de la modifier.
3. *Sélectionner un type de ressource pour chaque configuration :* sélectionner une configuration depuis la liste Configurations, et, dans le panneau *Paramètres pour la configuration*, spécifier une ressource pour la configuration : (i) Fichier, (ii) Résultat d'une transformation Altova MapForce, ou (iii) Résultat d'une transformation Altova StyleVision. Sélectionner le bouton radio approprié. Si une option de transformation MapForce ou StyleVision est choisie, une transformation sera effectuée par MapForce ou StyleVision en utilisant, respectivement, le fichier `.mfd` ou `.sps` et le fichier d'entrée respectif. Le résultat de la transformation sera la ressource.
4. *Sélectionner un fichier pour le type de ressource :* si la ressource est un fichier directement sélectionné, chercher le fichier dans la fenêtre de texte *Sélection du fichier de ressource*. Si la ressource est le résultat d'une transformation, dans le champ de saisie *Sélection du fichier*, chercher le fichier `.mfd` (pour des transformations MapForce) ou `.sps` (pour les transformations StyleVision). Lorsque des entrées ou sorties multiples pour la transformation sont possibles avec MapForce, une sélection des options sera présentée. Par exemple, les options de sortie d'une transformation StyleVision sont affichées conformément à l'édition de StyleVision installée (*la capture d'écran ci-dessous montre les sorties pour l'Édition Enterprise*).



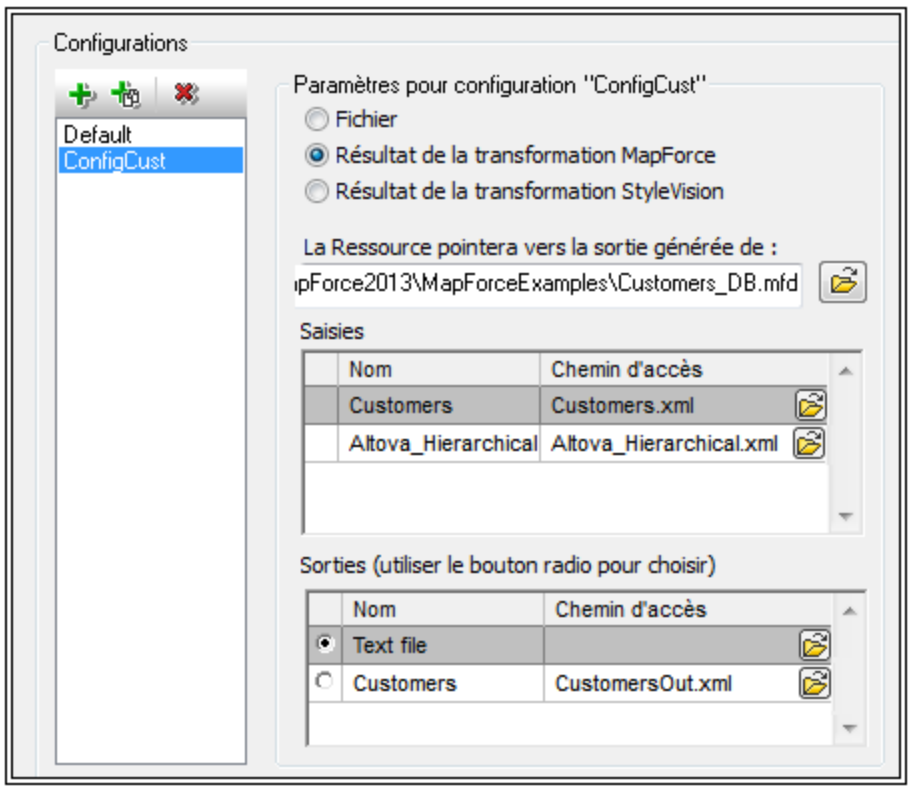
Choisir le bouton radio de l'option désirée (dans la capture d'écran ci-dessus, 'Sortie HTML' est sélectionnée). Si la ressource est le résultat d'une transformation, la sortie peut être enregistrée en tant qu'un fichier ou bien elle-même peut constituer une ressource globale. Cliquer sur l'icône et sélectionner, respectivement, Ressource globale (pour enregistrer la sortie en tant que ressource globale) ou Chercher (pour enregistrer la sortie en tant que fichier). Si aucune de ces options d'enregistrement n'est sélectionnée, le résultat de transformation sera chargé en tant que fichier temporaire lorsque la ressource globale sera invoquée.

5. *Définir des configurations multiples le cas échéant :* vous pouvez ajouter plus de configurations et spécifier une ressource pour chacune d'entre elles. Pour ce faire, répéter les étapes 3 et 4 ci-dessus pour chaque configuration. Vous pouvez ajouter une nouvelle configuration à la définition d'alias à tout moment.
6. *Enregistrer la définition d'alias :* cliquer sur **OK** pour enregistrer l'alias et toutes ses configurations en tant que ressource globale. La ressource globale sera recensée sous Fichiers dans le dialogue Gérer les Ressources globales.

Résultat de la transformation MapForce

Altova MapForce mappe un ou plusieurs schémas de document d'entrée (existant) vers un ou plusieurs (nouveaux) schémas de document de sortie. Ce mappage, créé par un utilisateur de MapForce, est connu sous la désignation de MapForce Design (MFD). Les fichiers XML, fichiers de texte, bases de données, etc., qui correspondent au/x schéma/s d'entrée peuvent être utilisés en tant que sources de données. MapForce génère des fichiers de données de sortie qui correspondent au schéma de document de sortie. Ce document de sortie est le fichier *Résultat de la transformation MapForce* qui deviendra une ressource globale.

Si vous souhaitez configurer un fichier de données généré par MapForce en tant qu'une ressource globale, les points suivants doivent être spécifiés dans le dialogue de ressource globale (*voir capture d'écran ci-dessous*) :



- ***Un fichier `.mfd` (MapForce Design).*** Vous devez spécifier ce fichier dans la fenêtre *La ressource va pointer vers la sortie générée de* (*voir capture d'écran ci-dessus*).
- ***Un ou plusieurs fichiers de données d'entrée.*** Une fois le fichier MFD spécifié, il est analysé et, sur la base de l'information de schéma d'entrée qui y est contenue, le/ fichier/s de données par défaut

 © 2016-2022 Altova GmbH

est/sont saisi/s dans le panneau *Entrées* (*voir capture d'écran ci-dessus*). Vous pouvez modifier la sélection du fichier par défaut pour chaque schéma d'entrée en spécifiant un autre fichier.

- ***Un fichier de sortie.*** Si le document MFD comporte plusieurs schémas de sortie, ils sont tous recensés dans le panneau *Sorties* (*voir capture d'écran ci-dessus*). Vous devez en choisir un. Si l'emplacement de fichier de sortie d'un schéma de sortie individuel est spécifié dans le document MFD, cet emplacement de fichier est saisi pour ce schéma de sortie dans le panneau *Sorties*. À partir de la capture d'écran ci-dessus, nous constatons que le document MFD spécifie que le schéma de sortie `Customers` comprend un fichier de données XML (`CustomersOut.xml`), alors que le schéma de sortie `Text file` n'a pas d'association de fichier dans le fichier MFD. Vous pouvez utiliser l'emplacement de fichier par défaut dans le panneau Sorties ou en spécifier un vous-même. Le résultat de la transformation MapForce sera enregistré dans l'emplacement de fichier du schéma de sortie sélectionné. Il s'agit du fichier qui sera utilisé en tant que ressource globale.

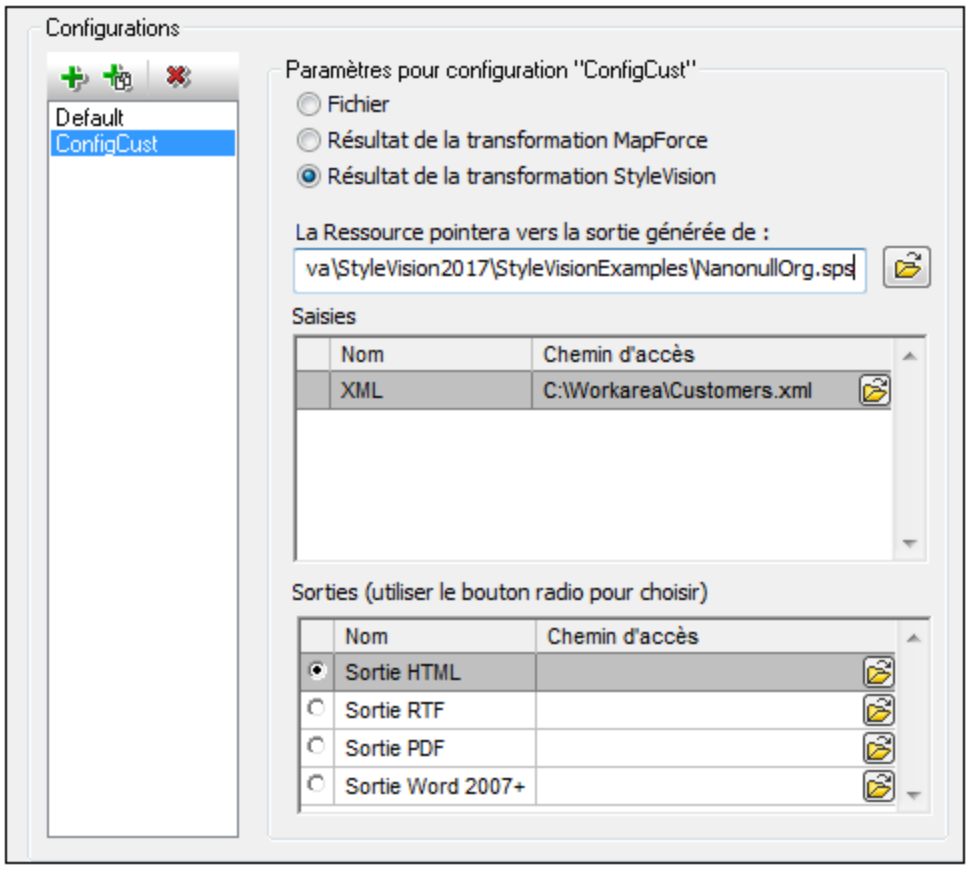
Note : L'avantage de cette option (Résultat de la transformation MapForce) est que la transformation est effectuée au moment où la ressource globale est invoquée. Cela signifie que la ressource globale contiendra les données les plus récentes (depuis le/s fichier/s d'entrée).

Note : Puisque MapForce est utilisé pour exécuter la transformation, vous devrez avoir préalablement installé Altova MapForce.

Résultat de la transformation StyleVision

Altova StyleVision est utilisé pour créer des fichiers StyleVision Power Stylesheet (SPS). Ces fichiers SPS génèrent des feuilles de style XSLT utilisées pour transformer des documents XML en documents de sortie dans des formats variés (HTML, PDF, RTF, Word 2007+, etc.). Si vous sélectionnez l'option *Résultat de la transformation StyleVision*, le document de sortie créé par StyleVision sera la ressource globale associée avec la configuration sélectionnée.

Pour l'option *Transformation StyleVision* dans le dialogue Ressource globale (*voir capture d'écran ci-dessous*), les fichiers suivants doivent être spécifiés.



- ***Un fichier `.sps` (SPS).*** Vous devez spécifier ce fichier dans la fenêtre *La ressource va pointer vers la sortie générée de* (*voir capture d'écran ci-dessus*).
- ***Fichier/s d'entrée.*** Le fichier d'entrée est peut-être déjà spécifié dans le fichier SPS. Si c'est le cas, il apparaîtra automatiquement dans le panneau *Entrées* une fois que le fichier SPS a été sélectionné. Vous pouvez modifier cette entrée. S'il n'y a pas d'entrée, vous devrez en ajouter une.
- ***Fichier/s de sortie.*** Sélectionner le format de sortie dans le panneau *Sorties*, et spécifier un emplacement de fichier de sortie pour ce format.

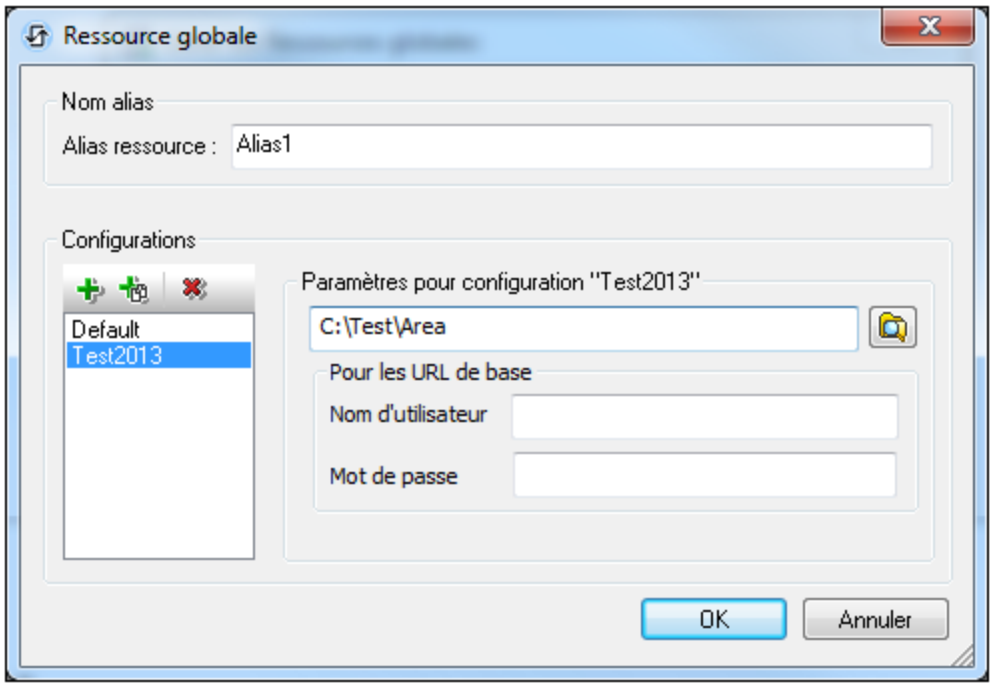
Note : L'avantage de cette option (Résultat de la transformation StyleVision) est que la transformation sera effectuée lors de l'invocation de la ressource globale. Cela signifie que la ressource globale contiendra les données les plus récentes (depuis le/s fichier/s d'entrée).

Note : Puisque StyleVision est utilisé pour exécuter la transformation, vous devrez avoir préalablement installé Altova StyleVision pour que cette fonction puisse marcher.

20.1.2 Dossiers

Dans le dialogue de Ressources globales pour Dossiers (*capture d'écran ci-dessous*), vous pouvez ajouter une ressource de dossier tel que décrit ci-dessous.

© 2016-2022 Altova GmbH



Icônes de dialogue des Ressources globales

Ajouter une configuration : ouvre le dialogue Ajouter configuration dans lequel vous pouvez saisir le nom de la configuration à ajouter.

Ajouter une configuration en tant que copie : ouvre le dialogue Ajouter configuration dans lequel vous pouvez saisir le nom de la configuration à créer en tant que copie de la configuration sélectionnée.

Supprimer : supprime la configuration sélectionnée.

Ouvrir : chercher le fichier à créer en tant que ressource globale.

Définir l'alias

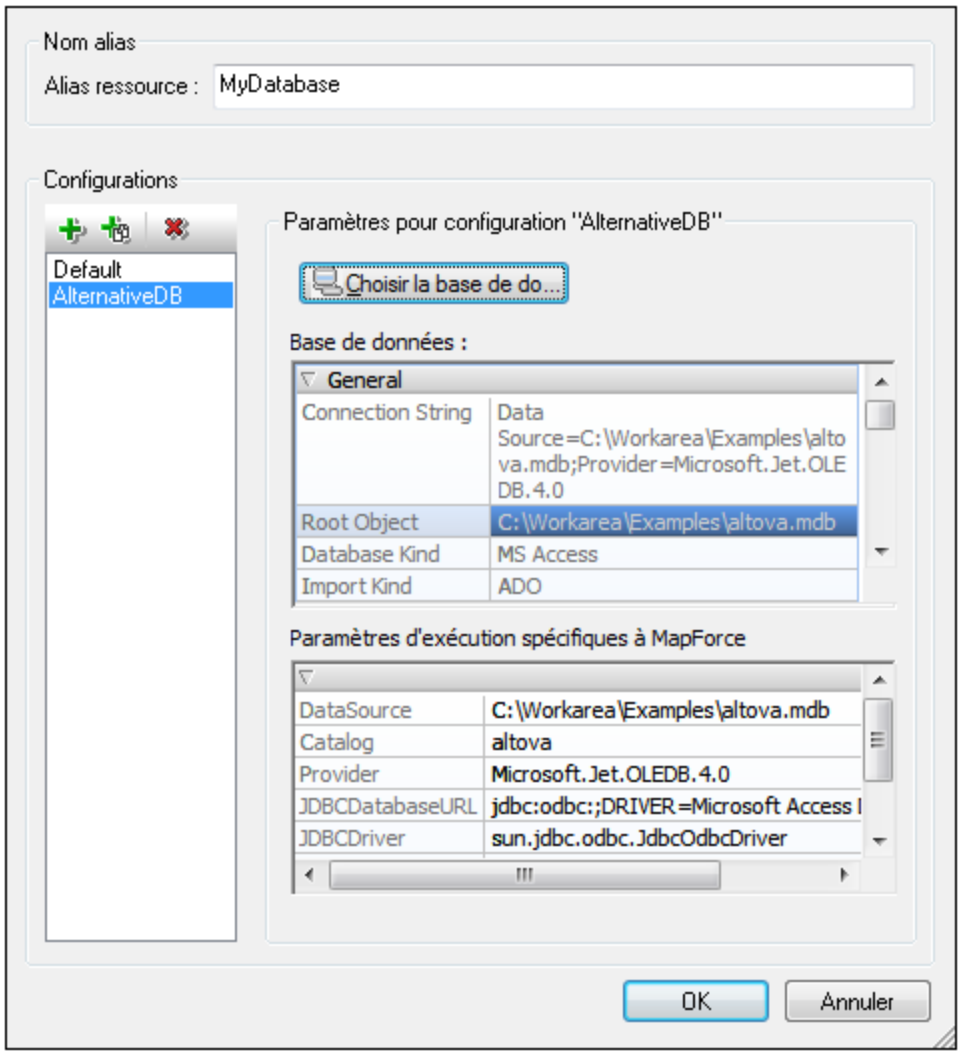
Définir l'alias (son nom et ses configurations) comme suit :

1. *Nommer l'alias :* saisir le nom de l'alias dans la fenêtre de texte *Alias de ressource*.
2. *Ajouter les configurations :* le panneau Configurations contiendra une configuration nommée `Default` (*voir capture d'écran ci-dessus*), qui ne peut pas être supprimée ou renommée. Vous pouvez ajouter autant de configurations supplémentaires pour l'alias sélectionné que vous le souhaitez. Ajouter une configuration en cliquant sur l'icône **Ajouter configuration** ou **Ajouter configuration en tant que copie**. Dans le dialogue qui s'ouvre, saisir le nom de configuration. Cliquer sur **OK**. La nouvelle configuration sera recensée dans le panneau Configurations. Répéter l'opération pour le nombre de configurations que vous souhaitez.
3. *Sélectionner un dossier en tant que ressource d'une configuration :* sélectionner une des configurations dans le panneau Configurations et cherchez le dossier que vous souhaitez créer en tant que ressource globale.

4. *Définir des configurations multiples le cas échéant :* spécifier une ressource de dossier pour chaque configuration que vous avez créée (répéter l'étape 3 ci-dessus pour chaque configuration que vous avez créée). Vous pouvez ajouter une nouvelle configuration à la définition d'alias à tout moment.
5. *Enregistrer la définition d'alias :* cliquer sur **OK** dans le dialogue de Ressources globales pour enregistrer l'alias et toutes ses configurations en tant que ressource globale. La ressource globale sera recensée sous Dossiers dans le dialogue Gérer les Ressources globales.

20.1.3 Bases de données

Dans le dialogue des Ressources globales pour les bases de données (*voir la capture d'écran ci-dessous*), vous pouvez ajouter une ressource de base de données comme suit :



Icônes du dialogue Ressource globale

© 2016-2022 Altova GmbH



Ajouter une configuration : Affiche la boîte de dialogue Ajouter une configuration dans laquelle vous saisissez le nom de la configuration à ajouter.



Ajouter une configuration en tant que copie : Affiche la boîte de dialogue Ajouter une configuration dans laquelle vous pouvez saisir le nom de la configuration à créer en tant que une copie de la configuration choisie.

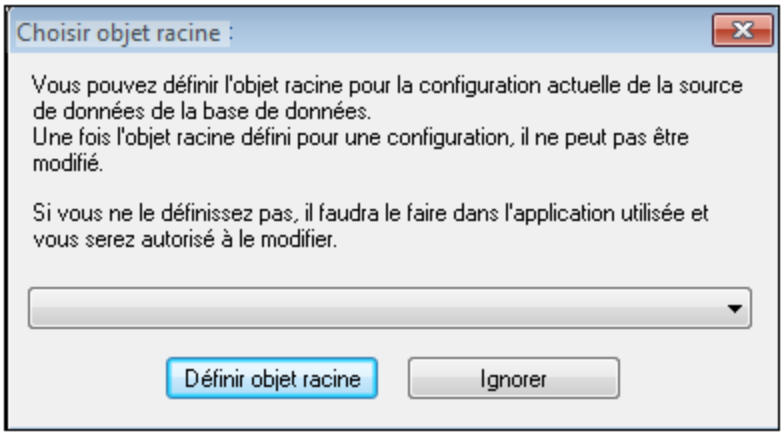


Supprimer : Supprime la configuration sélectionnée.

Définir l'alias

Définir l'alias (son nom et ses configurations) comme suit :

1. *Donner un nom à l'alias :* Saisir le nom de l'alias dans la zone de texte *Alias de ressource*.
2. *Ajouter des configurations :* Le volet de configurations aura une configuration appelée Par défaut (*voir la capture d'écran ci-dessus*). Cette configuration par défaut ne peut pas être supprimée ni changer de nom. Vous pouvez saisir autant de configurations supplémentaires pour l'alias sélectionné que vous voulez. Ajouter une configuration en cliquant sur les icônes **Ajouter Configuration** ou **Ajouter une configuration en tant que copie**. Dans le dialogue qui apparaît, saisissez le nom de configuration. Cliquez sur **OK**. La nouvelle configuration sera recensée dans le volet de configurations. Répéter pour autant de configurations que vous voulez.
3. *Lancer la sélection d'une base de données comme ressource d'une configuration :* Sélectionnez une des configurations dans le volet de configurations et cliquez sur l'icône **Choisir base de données**. Ceci fait apparaître le dialogue de connexion des Ressources globales.
4. *Connexion à la base de données :* Sélectionnez si vous voulez créer une connexion à la base de données en utilisant l'Assistant de connexion, une connexion existante, des connexions ADO, ODBC ou JDBC. Complétez la définition de la méthode de connexion telle que décrite dans la section Se connecter à une base de données. Si une connexion a déjà été établie à la base de données depuis XMLSpy, vous pouvez cliquer sur l'icône Connexions existantes et sélectionner la BD depuis la liste de connexions qui est affichée.
5. *Sélectionner l'objet racine :* Si vous vous connectez à un serveur de base de données où un objet racine peut être sélectionné, vous serez invité dans le dialogue Choisir Objet racine (*capture d'écran ci-dessous*) à choisir un objet racine sur le serveur. Sélectionnez l'objet racine et cliquez sur **Définir l'objet racine**. L'objet racine que vous sélectionnez sera l'objet racine qui est chargé lorsque cette configuration est utilisée.



Si vous décidez de ne pas sélectionner un objet racine (en cliquant sur le bouton **Skip**), alors vous pouvez sélectionner l'objet racine au moment du chargement de la ressource globale.

6. *Définissez de multiples configurations, le cas échéant :* Spécifiez une ressource de base de données pour toute autre configuration que vous avez créé (à savoir, répétez les étapes 3 à 5 ci-dessus pour les diverses configurations que vous avez créées). Vous pouvez ajouter une nouvelle configuration à la définition de l'alias à tout moment.
7. *Enregistrer la définition de l'alias :* Cliquez sur **OK** dans le dialogue Ressources globales pour enregistrer l'alias et toutes ses configurations comme ressource globale. La ressource globale sera recensée sous la base de données Gérer les Ressources globales.

© 2016-2022 Altova GmbH

20.2 Utiliser les Ressources globales

Il existe plusieurs types de ressources globales (type fichier, type dossier, et type de base de données). Certains scénarios dans lesquels vous pouvez utiliser des ressources globales dans XMLSpy sont recensés ici : Fichiers et Dossiers et les Bases de données.

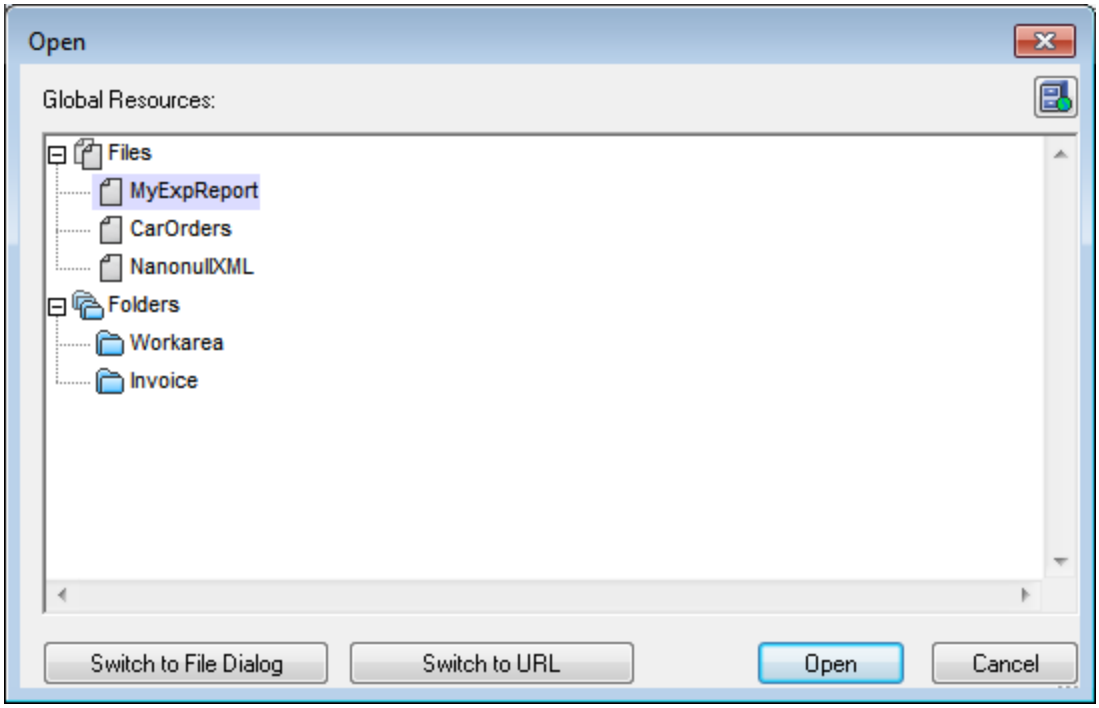
Sélections qui déterminent quelle ressource est utilisée

Il y a deux sélections au sein des applications qui déterminent quelles ressources globales peuvent être utilisées et quelles ressources globales peuvent être réellement utilisées à tout moment donné :

- *Le Fichier XML de Ressources globales actif* est sélectionné dans le dialogue Ressources globales. Les définitions de ressources globales qui sont présentes dans le Fichier XML de ressources globales sont disponibles dans tous les fichiers ouverts dans l'application. Seules les définitions dans le Fichier XML de Ressources globales actif sont disponibles. Le Fichier XML de Ressources globales actif peut être changé à tout moment et les définitions de ressources globales contenues dans le nouveau fichier actif remplaceront immédiatement les fichiers actifs précédents. Le Fichier XML de Ressources globales actif détermine donc : (i) quelles ressources globales peuvent être attribuées et (ii) quelles ressources globales sont disponibles pour une consultation (par exemple, si une ressource globale dans un Fichier de Ressources globales est attribuée mais qu'il n'y a pas de ressource globale de ce nom dans le Fichier XML de Ressources globales actif actuellement, alors la ressource globale attribuée (alias) ne pourra pas être consultée).
- *La configuration active est* sélectionnée via l'élément de menu **Outils | Configuration active** ou via la barre outils Ressources globales . Cliquer sur cette commande (ou liste déroulante dans la barre outils) pour ouvrir une liste de configurations à travers tous les alias. La sélection d'une configuration activera cette configuration dans toutes les applications. Cela signifie que quel que soit l'endroit où une ressource globale (ou alias) est utilisée, la ressource correspondant à la configuration active de chaque alias utilisé sera chargée. La configuration active est appliquée à tous les alias utilisés. Si un alias n'a pas de configuration avec le nom de la configuration active, alors la configuration par défaut de cet alias sera utilisée. La configuration active n'est pas pertinente lors de l'attribution de ressources ; elle n'est uniquement déterminante que lorsque les ressources sont réellement utilisées.

20.2.1 Attribuer des Fichiers et des Dossiers

Les ressources de type Fichier et de type Dossier sont attribuées différemment. Dans l'un des scénarios d'utilisation ci-dessous, cliquer sur le bouton **Passer aux ressources globales** pour afficher le dialogue Ouvrir la Ressource globale (*capture d'écran ci-dessous*).



Gérer les Ressources globales : affiche le dialogue Gérer les Ressources globales.

La sélection d'une *ressource globale de type fichier* permet d'attribuer le fichier. La sélection d'une *ressource globale de type dossier* entraîne l'ouverture du dialogue Ouvrir dans lequel vous pouvez chercher le fichier nécessaire. Le chemin d'accès du fichier sélectionné est saisi de manière relative à la ressource de dossier. Donc si une ressource globale de type dossier doit comporter deux configurations, chacune pointant vers des dossiers différents, les fichiers possédant le même nom mais placés dans des dossiers différents peuvent être ciblés par les deux configurations. Cela peut être utile pour les tests.

Vous pouvez passer au dialogue de fichier ou au dialogue URL en cliquant sur le bouton respectif en bas du dialogue. L'icône **Gérer les Ressources globales** située dans le coin en haut à droite affiche le dialogue Gérer les Ressources globales.

Scénarios d'utilisation

Les ressources de type Fichier et de type Dossier peuvent être utilisées dans les scénarios suivants :

- Ouvrir les ressources globales
- Enregistrer en tant que ressource globale
- Attribuer des fichiers pour des transformations XSLT
- Transformations XSLT
- Exécutions XQuery
- Attribuer un SPS

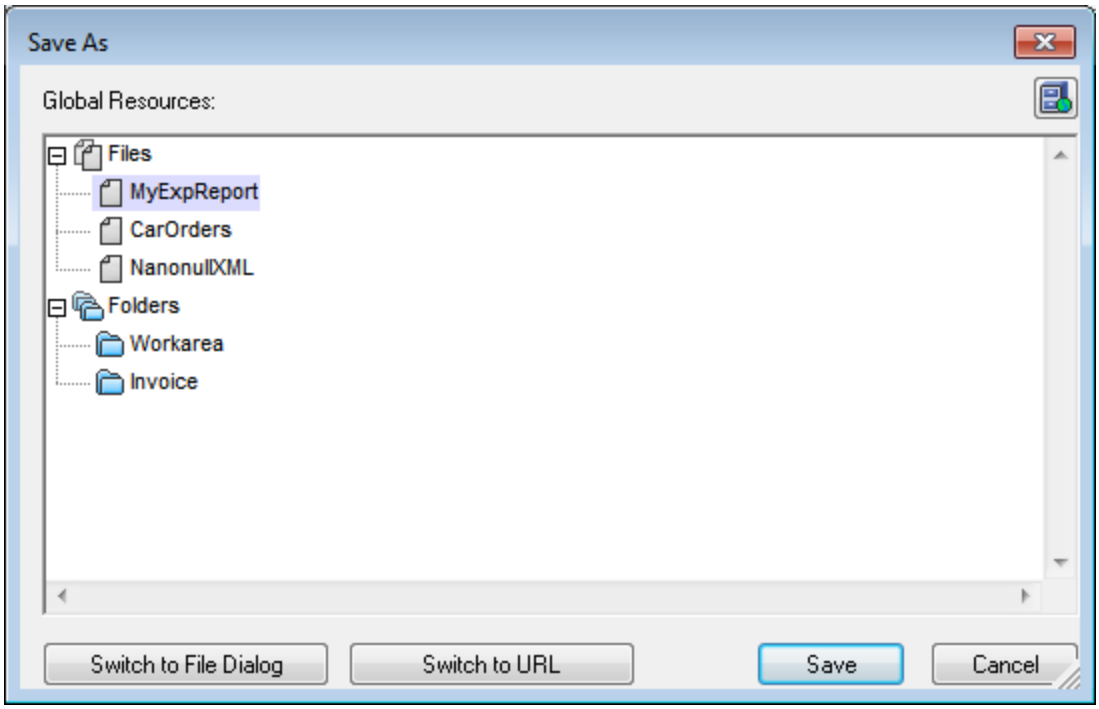
© 2016-2022 Altova GmbH

Ouvrir les ressources globales

Une ressource globale peut être ouverte dans XMLSpy avec la commande **Fichier | Ouvrir (Passer à la Ressource globale)** et peut être éditée. Dans le cas d'une ressource globale de type fichier, le fichier est ouvert directement. Dans le cas d'une ressource globale de type dossier, un dialogue Ouvrir s'ouvre avec le dossier associé sélectionné. Vous pouvez ensuite chercher le fichier nécessaire dans les dossiers descendants. Un des avantages lors de l'adressage des fichiers pour les éditer par le biais de ressources globales, est que les fichiers associés peuvent être enregistrés sous les différentes configurations d'une seule ressource globale et qu'ils peuvent être accédés simplement en changeant les configurations. Toutes les modifications d'édition devraient être enregistrées avant de modifier la configuration.

Enregistrer en tant que ressource globale

Un nouveau fichier peut être enregistré en tant que ressource globale. De même, un fichier déjà existant peut être ouvert puis enregistré en tant que ressource globale. Cliquer sur les commandes **Fichier | Enregistrer** ou **Fichier | Enregistrer sous**, pour faire apparaître le dialogue Enregistrer. Cliquer sur le bouton **Passer à la Ressource globale** pour accéder aux ressources globales disponibles (*capture d'écran ci-dessous*), qui sont les alias définis dans le Fichier XML de Ressources globales actuel.



Choisir un alias et cliquer sur **Enregistrer**. Si l'alias est un alias de fichier, le fichier sera enregistré directement. S'il s'agit d'un alias de dossier, un dialogue apparaîtra qui vous invitera à saisir le nom du fichier sous lequel vous souhaitez l'enregistrer. Dans tous les cas, le fichier sera sauvegardé à l'emplacement défini pour la configuration active actuellement.

Note : Chaque configuration pointe vers un emplacement de fichier spécifique, précisé dans la définition de cette configuration. Si le fichier que vous enregistrez en tant que ressource globale ne possède pas le même type d'extension que le fichier à l'emplacement actuel de la configuration, il pourrait se produire

des erreurs d'édition et de validation lorsque cette ressource globale est ouverte dans XMLSpy. Cela est dû au fait que XMLSpy ouvrira le fichier en présumant que le fichier présente le type de fichier spécifié dans la définition de la configuration.

Attribuer des fichiers pour les transformations XSLT

Les fichiers XSLT peuvent être attribués à des documents XML et des fichiers XML pour des documents XSLT via les ressources globales. Lorsque les commandes pour l'attribution des fichiers XSLT (**XSL/XQuery | Attribuer XSL** et **XSL/XQuery | Attribuer XSL:FO**) et les fichiers XML (**XSL/XQuery | Attribuer XML modèle**) sont cliqués, le dialogue d'attribution s'ouvre. Cliquer sur le bouton **Parcourir** pour ouvrir le dialogue Ouvrir, dans lequel vous pouvez passer au dialogue Ouvrir la Ressource globale et choisir la ressource globale nécessaire. Un des principaux avantages d'une utilisation d'une ressource globale pour spécifier des fichiers pour les transformations XSLT est que le fichier XSLT (ou le fichier XML) peut être modifié pour une transformation en changeant tout simplement la configuration active dans XMLSpy ; aucune attribution de nouveau fichier ne doit être réalisée à chaque fois qu'une transformation est nécessaire avec un fichier différent. Lors du lancement d'une transformation XSLT, celle-ci utilisera le/s fichier/s associé/s avec la configuration active.

Transformation XSLT et exécutions XQuery

Cliquer sur la commande **XSL/XQuery | Transformation XSL** ou **XSL/XQuery | Transformation XSL:FO** ou **XSL/XQuery | XQuery Update Execution** pour ouvrir un dialogue dans lequel vous pouvez chercher le fichier XSLT, XQuery, ou XML requis. Cliquer sur le bouton **Parcourir** puis sur le bouton **Passer à la Ressource globale** pour ouvrir le dialogue Ouvrir la Ressource globale (*capture d'écran en haut de la section*). Le fichier associé avec la configuration actuellement active de la ressource globale sélectionnée est utilisé pour la transformation.

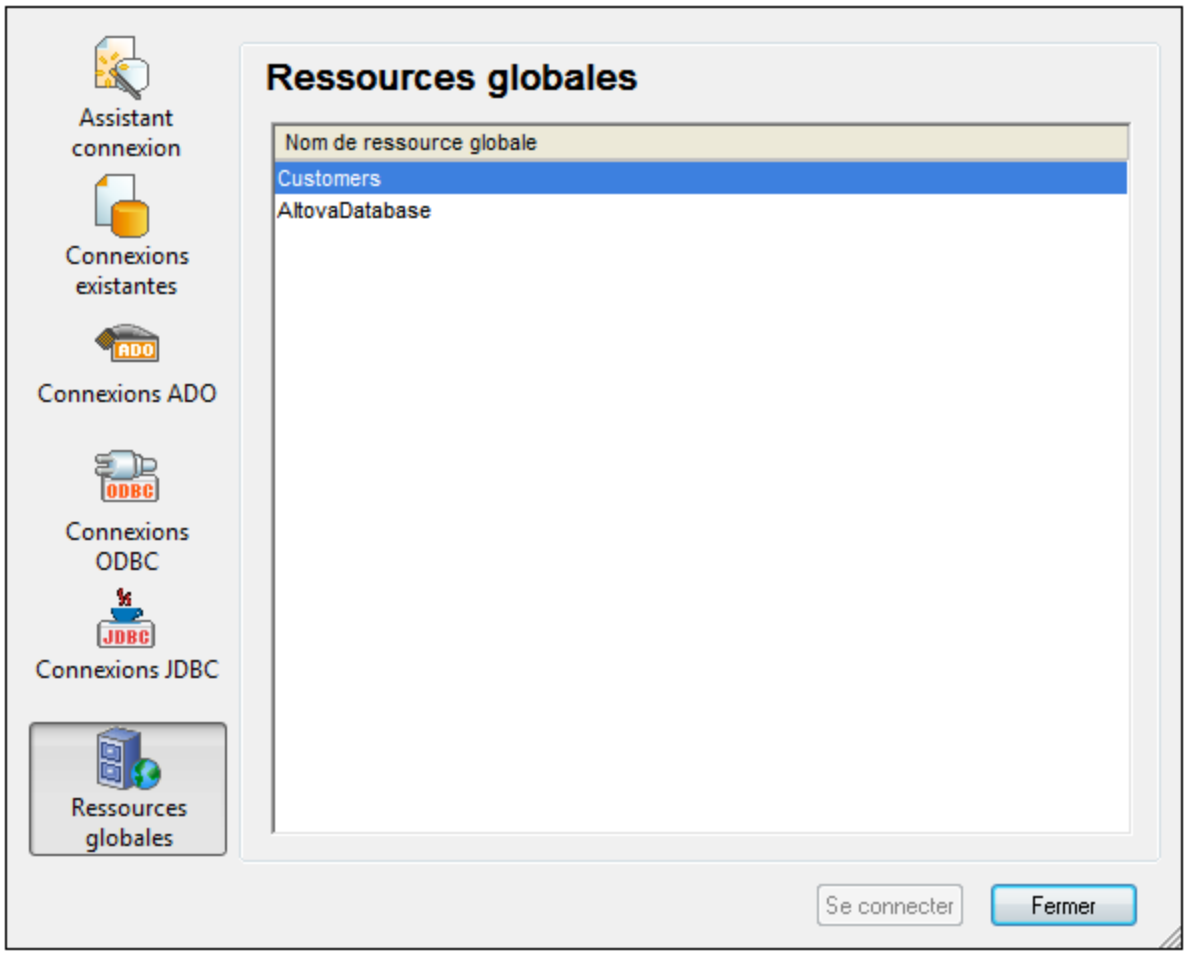
Attribuer un SPS

Lors de l'assignation d'une feuille de style StyleVision à un fichier XML (**Authentic | Attribuer la Feuille de style StyleVision**), vous pouvez sélectionner une ressource globale pour situer la feuille de style. Cliquer sur le bouton **Parcourir** puis sur le bouton **Passer à la Ressource globale** pour ouvrir le dialogue Ouvrir la Ressource globale (*capture d'écran en haut de la section*). Avec une ressource globale sélectionnée en tant que l'attribution, le Authentic View du document XML peut être modifié en changeant uniquement la configuration active dans XMLSpy.

20.2.2 Attribuer des bases de données

Lors de l'exécution d'une commande qui importe des données ou une structure de données (comme un Schéma XML) depuis une BD dans XMLSpy (par exemple, avec la commande **Convertir | Importer les données de base de données**), vous pouvez sélectionner l'option d'utilisation d'une ressource globale (*capture d'écran ci-dessous*). D'autres commandes, dans les endroits où une ressource globale de type base de données peuvent être utilisées, sont des commandes liées à des bases de données dans le menu.

© 2016-2022 Altova GmbH



Dans le dialogue Connexion (*capture d'écran ci-dessus*), toutes les ressources globales de type base de données qui ont été définies dans le Fichier XML de Ressources globale actuellement actif sont affichées. Choisir la ressource globale nécessaire et cliquer sur **Connecter**. Si la ressource globale sélectionnée comporte plus d'une configuration, la ressource de base de données pour la configuration actuellement active est utilisée (contrôler **Outils | Configuration Active** ou la barre outils de Ressources globales), et la connexion est établie. Vous devez maintenant choisir les structures de données et les données à utiliser tel que décrit dans Création d'un schéma XML depuis une BD et Importer les données BD.

20.2.3 Changer la Configuration active

Une configuration d'une ressource globale peut être active à tout moment. Cette configuration est appelée la configuration active et elle est active pour toutes les applications. Cela signifie que la configuration active est active pour tous les alias de ressources globales dans tous les fichiers ouverts actuellement et les connexions de source de données. Si un alias n'a pas de configuration avec le nom de la configuration active, la configuration par défaut de cet alias sera utilisée. Prenons en guise d'exemple pour le changement des configurations, le cas suivant : un fichier a été attribué par le biais d'une ressource globale avec plusieurs configurations. Chaque configuration mappe vers un fichier différent. Donc le fichier sélectionné dépend de la configuration sélectionnée en tant que la configuration active de l'application.

Il existe deux manières de passer la configuration active :

- Par la commande de menu **Outils | Configuration active**. Sélectionnez la configuration depuis le sous-menu de la commande.
- Dans la liste de choix de la barre outils de la Ressource globale (*capture d'écran ci-dessous*), sélectionnez la configuration requise.

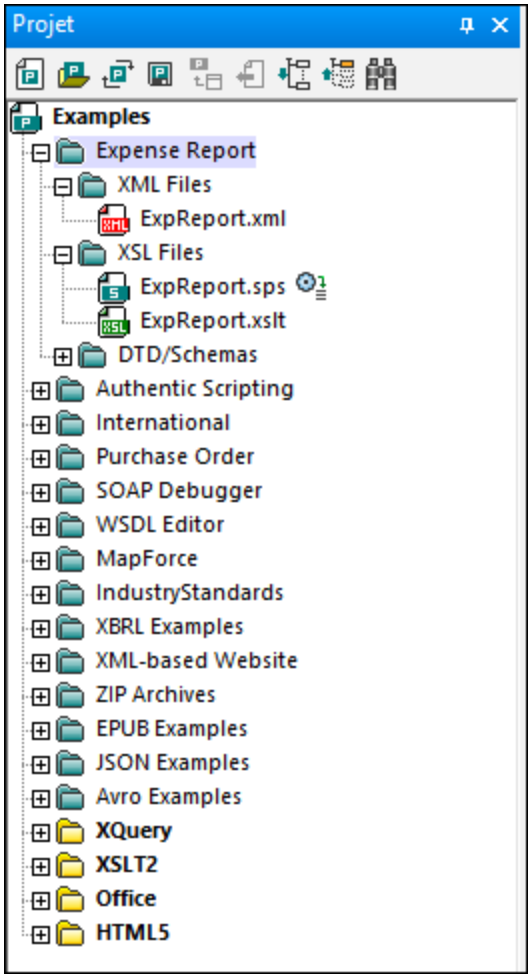


Ainsi, en changeant la configuration active, vous pouvez changer des fichiers sources qui sont attribués par le biais d'une ressource globale.

© 2016-2022 Altova GmbH

21 Projets

Un projet est une collection de fichiers qui sont liés l'un à l'autre d'une manière que vous pouvez déterminer. Par exemple, dans la capture d'écran ci-dessous, un projet nommé `Examples` rassemble les fichiers pour des exemples divers dans des dossiers d'exemples séparés, dont chacun peut être organisé dans des sous-dossiers. Dans le cadre du projet `Examples`, par exemple, le dossier d'exemples `OrgChart` est classé dans des sous-dossiers pour les fichiers XML, XSL et Schéma.

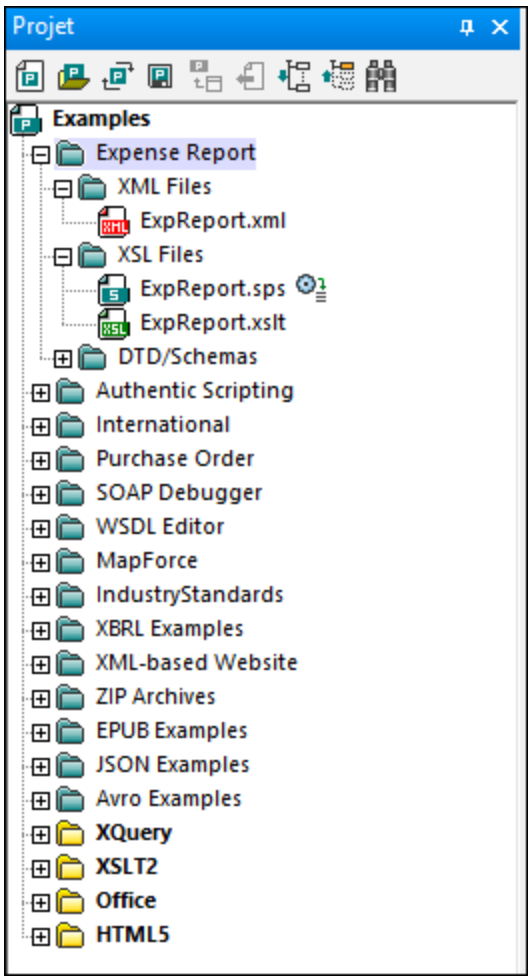


Les projets vous permettent donc de rassembler des fichiers qui sont utilisés ensembles et d'y accéder plus rapidement. De plus, vous pouvez définir des fichiers de schémas et de XSLT pour les dossiers individuels, permettant ainsi le traitement par lots de fichiers dans un dossier.

Cette section décrit comment créer et éditer des projets et comment utiliser des projets.

21.1 Créer et éditer des projets

Les projets sont gérés dans la fenêtre de projet (*capture d'écran ci-dessous*) et le menu Projet. Un projet à la fois peut être ouvert dans l'application. Le projet ouvert est affiché dans la fenêtre de projet.



Créer de nouveaux projets, ouvrir des projets existants

Un nouveau projet est créé avec la commande de menu **Projet | Nouveau projet**. Un projet existant est ouvert avec la commande de menu **Projet | Ouvrir projet**. Le projet ouvert récemment (qu'il soit nouveau ou existant) remplace le projet ouvert précédemment dans la fenêtre Projet. Si le projet ouvert précédemment contient des changements non enregistrés (indiqués par un astérisque à côté du nom de dossier ; *voir capture d'écran ci-dessous*), vous serez invité à enregistrer ces changements.

Nommer et enregistrer des projets

Un nouveau projet sera nommé lorsque vous l'enregistrerez. Un projet est enregistré avec la commande **Projet | Enregistrer projet** et porte l'extension de fichier `.spp`. Lorsqu'un projet est modifié, il doit être enregistré pour que les modifications soient stockées. Notez qu'un projet (indiqué par le dossier de niveau supérieur dans le

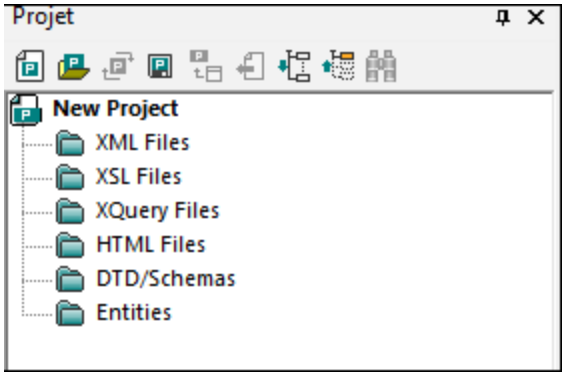
© 2016-2022 Altova GmbH

projet de fenêtre) ne peut être renommé qu'en modifiant son nom dans Windows File Explorer ; le nom ne peut pas être modifié dans la GUI. (Les noms des sous-dossiers, quant à eux, peuvent être modifiés dans la GUI.)

Project structure

Un projet présente une structure arborescente de dossiers et de fichiers. Les dossiers et les fichiers peuvent être créés à tous niveaux et à une profondeur sans limite. Pour ce faire, sélectionnez un dossier dans la fenêtre Projet et utilisez les commandes dans le menu **Projet** ou le menu contextuel pour ajouter des dossiers, des fichiers ou des ressources. Les dossiers, fichiers ou ressources qui ont été ajoutés à un projet peuvent être supprimés ou glissés vers d'autres emplacements dans l'arborescence de projet.

Lors de la création d'un nouveau projet, la structure du projet par défaut organise le projet en types de fichiers (XML, XSL, etc.) (*voir capture d'écran ci-dessous*).



Les extensions de type Fichier sont associées à un dossier par le biais des définitions de propriété pour ce dossier. Lorsqu'un fichier a été ajouté à un dossier, il est ajouté automatiquement au dossier enfant approprié correspondant à l'extension du type de fichier. Vous pouvez définir pour chaque dossier les extensions de type de fichier que vous souhaitez lui associer.

Ce qui peut être ajouté à un projet

Des dossiers, des fichiers et d'autres ressources peuvent être ajoutées soit au dossier de projet de niveau supérieur ou à un dossier à tout niveau dans le projet. Il existe trois types de dossiers : (i) dossiers de projet ; (ii) dossiers externes ; (iii) dossiers web externes.

Pour ajouter un objet, sélectionner le dossier pertinent puis la commande désirée depuis le menu **Projet** ou le menu contextuel du dossier sélectionné. Les objets suivants sont disponibles pour un ajout au dossier de projet.

- *Les Dossiers de projet* (vert) sont des dossiers que vous ajoutez au projet pour structurer le contenu du projet. Vous pouvez définir quelles extensions de fichier vous souhaitez associer avec un dossier de projet (dans les propriétés de ce dossier). Lorsque des fichiers sont ajoutés à un dossier, ils sont automatiquement ajoutés au premier dossier enfant auquel cette extension de fichier a été associée. Par conséquent, lorsque de multiples fichiers sont ajoutés au dossier, ils seront distribués par une extension de fichier parmi les dossiers enfant qui ont des associations d'extension de fichier correspondantes.
- *Les Dossiers externes* (jaune) sont des dossiers situés dans un système de fichier. Lorsqu'un dossier externe est ajouté à un dossier, le dossier externe et tous ses fichiers, sous-dossiers et fichiers de sous-dossiers sont inclus dans le projet. La définition des extensions de fichier dans un dossier externe sert à filtrer les fichiers disponibles dans le projet.

- *Les Dossiers web externes* sont comme des dossiers externes, sauf qu'ils sont situés sur un serveur web et leur accès nécessite une authentification d'utilisateur. La définition d'extensions de fichier dans un dossier web externe sert à filtrer les fichiers disponibles dans le projet.
- *Les Fichiers* peuvent être ajoutés à un dossier en choisissant le dossier et puis en utilisant une des trois commandes d'Ajout de fichier : (i) **Ajouter des fichiers**, pour sélectionner le(s) fichier(s) par le biais d'un dialogue Ouvrir ; (ii) **Ajouter fichier actif**, pour ajouter le fichier qui est actif dans la fenêtre principale ; (iii) **Ajouter des fichiers liés et actifs**, ajoute additionnellement des fichiers liés à un fichier XML actif, par exemple, un schéma XML ou DTD. Notez que les fichiers associés par les moyens d'une instruction de traitement (par exemple, fichiers XSLT), ne sont pas considérés être des fichiers liés.
- Les *Ressources globales*sont des alias pour des fichiers, dossiers et ressources de bases de données. Vous trouverez une description de leur définition et de leur utilisation dans la section concernant les Ressources Globales.
- *Les URL* identifient un objet de ressource par le biais d'une URL.
- *Un projet de scripts d'Altova*, qui est un fichier `.asprj`, peut être assigné à un projet XMLSpy. Ce rendra les macros et autres scripts disponibles pour le projet. Vous trouverez une description de la création d'un Projet de Script et de l'attribution d'un projet à un projet XMLSpy dans la section, Scripts.

Propriétés de dossier et de projet

Les propriétés (telles que le schéma pour la validation et XSLT pour la transformation) peuvent être définies non seulement pour le projet entier, mais aussi pour les dossiers individuels. Puis, vous pouvez exécuter des actions, telles que la validation et la transformation, dans l'ensemble du projet ou les dossiers individuels. Pour exécuter une action, cliquez de la touche droite sur le projet ou le dossier, et choisissez l'action que vous voulez réaliser depuis le menu contextuel qui apparait.

Les propriétés d'un dossier sont stockées dans le dialogue Propriétés de ce dossier qui est accessible en sélectionnant tout d'abord le dossier, puis la commande **Projet | Propriétés** (ou les Propriétés de commande du menu contextuel du dossier. Les propriétés suivantes d'un dossier peuvent être définies et éditées dans le dialogue Propriétés :

- *Nom de dossier* : ne peut pas être édité pour le dossier de projet de niveau supérieur (pour lequel, au lieu d'un nom, le chemin de fichier s'affiche).
- *Extensions de fichier* : ne peut pas être édité pour le projet de niveau supérieur. Il s'agit d'une liste d'extensions de fichiers séparées par un point-virgule (par exemple, **`xml;svg;wml`**). Cette liste détermine quels fichiers sont ajoutés au dossier lorsque des fichiers sont ajoutés à un projet. Par exemple, lorsque des fichiers actifs et liés sont ajoutés à un projet, les *Extensions de fichier* déterminent dans quels dossiers les fichiers ajoutés seront placés.
- *Validation* : spécifie le fichier de schéma XML ou DTD qui devrait être utilisé pour valider les fichiers XML dans un dossier.
- *Transformations* : spécifie (i) les fichiers XSLT à être utilisés pour transformer les fichiers XML dans le dossier, et (ii) les fichiers XML à être transformés avec des fichiers XSLT dans le dossier.
- *Fichiers de destination* : pour les transformations de sortie, spécifie l'extension de fichier et le dossier où les fichiers doivent être enregistrés.
- *Fichiers SPS pour le Mode Authentic* : spécifie les fichiers SPS à utiliser afin de pouvoir consulter et éditer des fichiers XML dans un dossier dans le Mode Authentic.

Veuillez prendre note des points suivants :

- Une propriété qui est définie dans un dossier écrase la même propriété du projet.
- Si une propriété est définie dans un projet, elle est appliquée à tous les autres dossiers qui n'ont pas le même ensemble de propriété.

© 2016-2022 Altova GmbH

- Si une action est exécutée dans un projet, elle est appliquée à tous les types de fichiers applicables dans tous les dossiers du projet. Par exemple, si une validation est réalisée dans un projet, celle-ci est exécutée sur tous les fichiers XML dans tous les dossiers du projet.. Dans ce cas, le schéma qui a été défini pour le projet est utilisé pour toutes les validations, à l'exception des fichiers XML se trouvant dans des dossiers qui sont dotés de la propriété de validation de schéma définie pour un autre schéma.

Voir la description de la commande **Projet | Propriétés** pour des informations plus détaillées.

Contrôle de source dans les projets

Les systèmes de contrôle de la source qui sont compatibles avec Microsoft Visual Source-Safe sont pris en charge dans les projets. Vous trouverez une description pour utiliser cette fonction dans la section Référence de l'utilisateur du manuel.

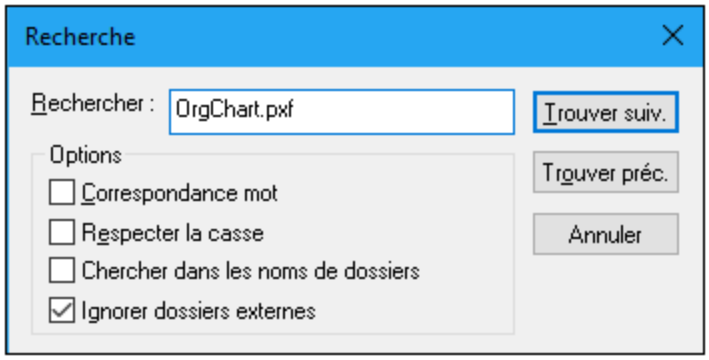
Enregistrer les projets

Toutes les modifications que vous effectuez à un projet, comme l'ajout ou la suppression d'un fichier, ou la modification d'une propriété de projet, doit être enregistrée avec la commande **Enregistrer Projet**.

Rechercher dans le projet

Vous pouvez chercher les fichiers de projet et les dossiers en utilisant leurs noms ou une partie de leur nom. Si la recherche aboutit, les fichiers ou les dossiers trouvés seront marqués un par un.

Pour lancer une recherche, activer la fenêtre du Projet en cliquant dessus (ou dedans), puis sélectionner la commande **Édition | Recherche** (ou le raccourcis **Ctrl+F**). Dans le dialogue Recherche qui apparaît (*capture d'écran ci-dessous*), saisir le texte que vous souhaitez chercher et sélectionner ou désélectionner les options de recherche (*explications ci-dessous*) conformément à vos exigences.



Les options de recherche suivantes sont disponibles :

- La correspondance mot est plus limitée puisque le texte entier doit correspondre à un mot entier dans le nom du fichier ou du dossier. Dans les noms du fichier, les parties avant et après le point (sans le point) sont toutes traitées en tant qu'un mot.
- Vous pouvez préciser que la casse du mot recherché doit correspondre exactement au string contenu dans le nom du fichier ou du dossier.
- Les noms de dossier peuvent être inclus à la recherche. Dans le cas contraire, seuls les noms de fichiers seront recherchés.

- Les dossiers externes peuvent être inclus ou exclus de la recherche. Les dossiers externes sont des dossiers réels dans le système ou le réseau, contrairement aux dossiers de projet qui sont créés dans le cadre du projet et non dans le système.

Si la recherche aboutit, le premier item correspondant est marqué dans la barre latérale du Projet. Vous pouvez ensuite chercher dans tous les items correspondants retournés en cliquant sur les touches **Chercher suivant** et **Trouver précédent** dans le dialogue Recherche.

Réinitialiser les projets

Si vous effectuez une modification dans un dossier externe, cette modification ne sera pas réfléchie dans la Fenêtre du Projet tant que le projet est réinitialisé.

© 2016-2022 Altova GmbH

21.2 Utiliser les projets

Les projets sont très utiles pour mieux organiser votre espace de travail, appliquer des paramètres à des fichiers multiples et pour configurer et exécuter des commandes par lot. L'utilisation de projets peut donc contribuer à accélérer considérablement votre travail et vous faciliter la vie. Pour toute information sur la gestion de projets, voir Créer et éditer les projets et la description des

Avantages d'utiliser des projets

La liste suivante regroupe les avantages d'une utilisation des projets.

- Les fichiers et les dossiers sont regroupés en dossiers par le biais de l'extension de fichier ou par tout autre critère.
- Les fichiers Schémas et XSLT peuvent être attribués à un dossier. Cela peut être utile si vous souhaitez valider ou transformer rapidement un fichier XML unique en utilisant des fichiers Schéma ou XSLT différents. Ajoutez un fichier XML file à différents dossiers et définissez différents dossiers schémas et fichiers XSLT pour les différents dossiers.
- Un traitement par lots peut être appliqué à des dossiers individuels. Les commandes disponibles pour le traitement par lots sont recensées ci-dessous.
- Les dossiers de sortie peuvent être spécifiés pour les transformations.

Organiser les ressources pour un accès rapide

Les ressources de dossier et de fichier peuvent être organisées dans des structures d'arborescence, vous fournissant un aperçu clair des différents dossiers et fichiers, et vous permettant d'accéder rapidement tout fichier dans un projet. Double-cliquez simplement dans un fichier dans la fenêtre Projet pour l'ouvrir. Vous pouvez rapidement ajouter des fichiers et des dossiers à un projet et supprimer des fichiers et dossiers non désirés. Si vous souhaitez travailler avec un autre projet, fermez le projet ouvert actuellement dans la fenêtre Projet et ouvrez le projet requis.

Traitement par lots

Les commandes pour le traitement par lot des fichiers dans un dossier, qu'il s'agisse du dossier de projet de niveau supérieur ou un dossier à tout autre niveau, sont **disponibles dans le menu contextuel de ce dossier** (obtenu en cliquant avec la touche de droite sur le dossier). Les étapes du traitement par lots sont les suivantes :

1. Définir les fichiers à utiliser pour la validation ou la transformation dans le dialogue Propriétés de ce dossier.
2. Spécifiez le dossier dans lequel la sortie des transformations doit être enregistrée. Si aucun dossier de sortie n'est spécifié pour un dossier, c'est le dossier de sortie du dossier ancêtre suivant dans l'arborescence de l'arbre qui sera utilisé.
3. Utiliser les commandes dans le menu contextuel pour l'exécution par lots. Si vous utilisez les commandes correspondantes dans les menus **XML**, **DTD/Schema** ou **XSL/XQuery** , la commande sera exécutée uniquement dans le document actif de la fenêtre principale, et non uniquement dans le dossier de projet de la fenêtre Projet.

Les commandes suivantes dans le menu contextuel d'un projet (niveau supérieur ou autre) sont disponibles pour un traitement par lots :

- *Vérification de bonne formation* : Si une erreur est détectée au cours de l'exécution par lots, elle est rapportée dans le fenêtre Messages.
- *Validation* : Si une erreur est détectée au cours de l'exécution par lots, elle est rapportée dans le fenêtre Messages.
- *Transformations* : Les sorties de transformation sont enregistrées dans le dossier spécifié en tant que le dossier de sortie dans le dialogue Propriétés de ce dossier. Si aucun dossier n'est spécifié, le dossier de sortie du dossier de projet de l'ancêtre suivant est utilisé. Si aucun dossier de projet ancêtre n'est défini, une fenêtre de document est ouverte et les résultats de transformation sont affichés successivement dans la fenêtre de document. Une transformation XSL-FO transforme un document XML ou un document FO en un PDF.
- *Générer un DTD/Schéma XML* : Avant de générer les schémas, vous êtes invité à spécifier un dossier de sortie. Les fichiers de schéma générés sont enregistrés dans ce dossier et sont affichés dans des fenêtres séparées dans la GUI.

Note : afin d'exécuter des commandes de lots, utilisez le menu contextuel du dossier concerné dans la fenêtre Projet. N'utilisez pas les commandes dans les menus XML, DTD/Schéma ou XSL/XQuery. Ces commandes seront exécutées dans le document actif dans la fenêtre Principale.

Validation et XSLT/XQuery avec RaptorXML Server

Les commandes de menu contextuel dans le dossier de projet vous permettent d'utiliser RaptorXML Server pour une validation XML haute performante et des transformations XSLT/XQuery. Voir la section RaptorXML Server pour plus d'informations.

© 2016-2022 Altova GmbH

22 RaptorXML(+XBRL) Server

Si Altova RaptorXML(+XBRL) Server (ci-après aussi nommé RaptorXML Server, RaptorXML ou Raptor en bref) est installé et détient une licence sur votre réseau et si votre installation XMLSpy y a accès, alors vous pouvez utiliser RaptorXML Server pour valider des documents XML et XBRL*, et pour exécuter des transformations XSLT et XQuery. Vous pouvez valider le document actif ou tous les documents contenus dans un dossier de projet XMLSpy. Les résultats de validation sont affichés dans la fenêtre Messages de la GUI.

Dans XMLSpy, vous pouvez (i) valider des documents ou (ii) exécuter des transformations XSLT/XQuery ou (iii) exécuter des documents XULE (ou des ensembles de documents) dans une instance XBRL. Un des avantages principaux d'une utilisation de Raptor est que vous pouvez configurer des validations individuelles ou des exécutions par le biais d'une large sélection d'options de validation. De plus, vous pouvez stocker un ensemble d'options Raptor en tant que "configuration" dans XMLSpy, puis sélectionner une de vos configurations définies pour une validation Raptor particulière. L'utilisation de Raptor peut s'avérer avantageuse lorsque vous souhaitez valider des quantités importantes de collections de données.

Note : La performance réelle dépend du nombre de cœurs de processeurs de PC utilisés par RaptorXML Server pour la validation : plus le nombre de cœurs utilisé est élevé, plus le traitement sera rapide.

***Note :** Il y a deux éditions de Raptor : *RaptorXML Server* (pour les validations XML) et *RaptorXML+XBRL Server* (pour les validations XML et XBRL). Si vous souhaitez valider des documents XBRL, vous devez utiliser RaptorXML+XBRL Server. Pour plus d'informations concernant RaptorXML(+XBRL) Server, veuillez consulter le site Internet Altova et les manuels d'utilisateur : *RaptorXML Server* et *RaptorXML+XBRL Server*.

Note: RaptorXML Server ne peut pas être utilisé avec des proxys HTTP parce que ceux-ci ne prennent pas en charge les mises à niveau websocket. Si vous devez faire face à ce problème, ajouter l'hôte RaptorXML Server à la liste proxy-ignore.

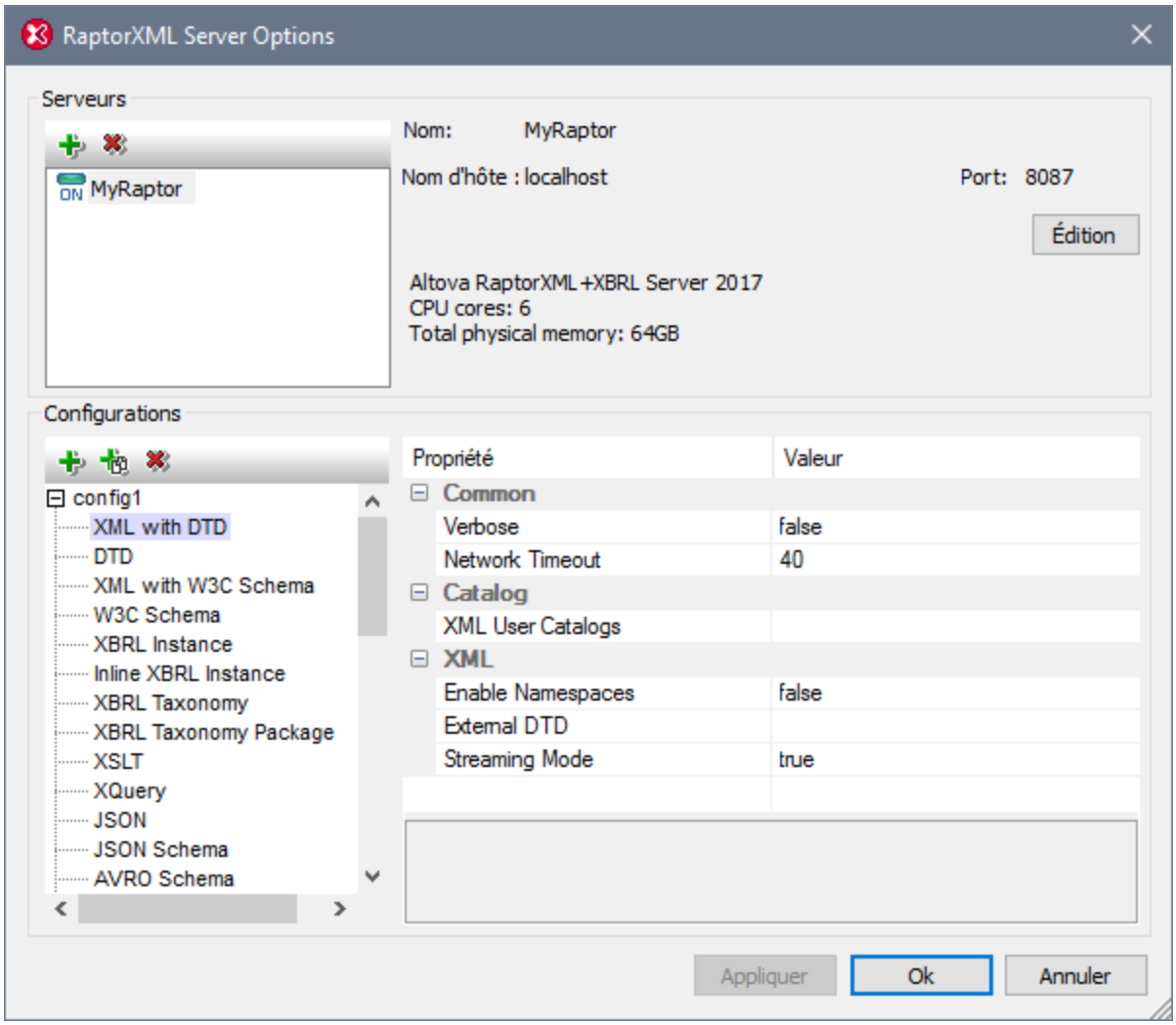
Comment valider ou transformer en utilisant RaptorXML Server

Pour valider un document XML ou XBRL avec RaptorXML Server, ou pour exécuter une transformation XSLT ou XQuery, XMLSpy doit savoir quel RaptorXML Server utiliser, comment accéder à ce serveur et quelles options passer à Raptor pour la validation. Cette information est gérée dans XMLSpy comme suit :

1. *En ajoutant un serveur au pool de serveurs Raptor*. Dans cette étape, les RaptorXML Servers sont ajoutés à un pool, et les informations d'accès de chaque serveur sont stockées dans XMLSpy. Chaque serveur est identifié par un nom.
2. *En définissant des configurations pour chaque serveur*. Une configuration est un ensemble d'options de validations de Raptor. Chaque serveur peut avoir des configurations multiples. Pour une validation, vous sélectionnez une configuration qui devient la configuration active.
3. *En sélectionnant une configuration de serveur avec laquelle valider*. Un serveur et une de ses configurations est sélectionnée pour être la configuration active. Celle-ci est utilisée pour toutes les validations suivantes qui utilisent Raptor.
4. *Valider ou exécuter la transformation XSLT/XQuery avec Raptor*.

22.1 Ajouter des serveurs et des configurations de serveur

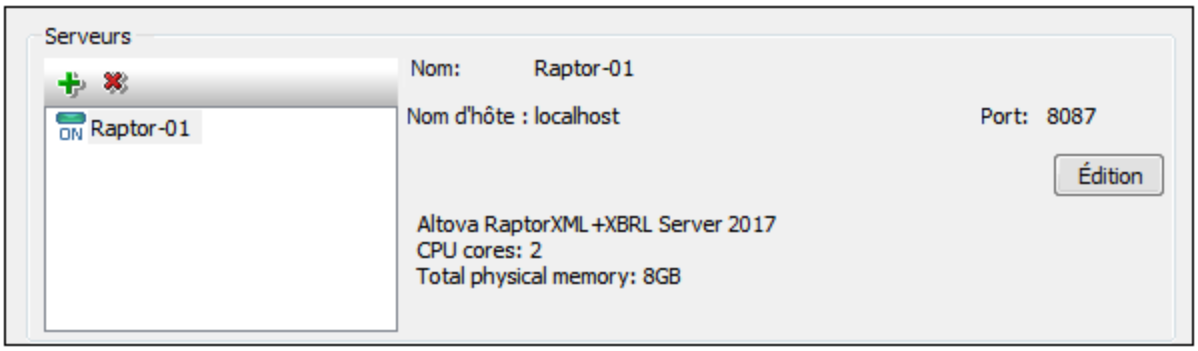
Dans le dialogue Options de RaptorXML Server (*capture d'écran ci-dessous,* **Outils | Gérer Raptor Servers**), vous pouvez ajouter plusieurs Raptor Servers au pool de Raptor Servers disponibles, puis vous pouvez définir plusieurs configurations pour chaque serveur. Les serveurs ajoutés, avec les configurations que vous définissez pour chacun d'entre eux, apparaîtront dans le sous-menu **Outils | Raptor Servers et Configurations**. Dans ce sous-menu, vous pouvez sélectionner la configuration de serveur que vous souhaitez utiliser pour une validation Raptor.



Ajouter un Raptor Server

Dans le volet *Serveurs* du panneau (*capture d'écran ci-dessous*), cliquer sur l'icône **Ajouter serveur**, puis saisir le nom avec lequel vous souhaitez identifier le serveur Raptor, le nom du réseau de la machine sur lequel Raptor est installé (nom d'hôte), et le port du serveur Raptor. Cliquer sur **OK** pour enregistrer les paramètres.

© 2016-2022 Altova GmbH



- *Nom :* une chaîne de votre choix. Elle est utilisée dans XMLSpy pour identifier un RaptorXML Server particulier.
- *Nom de l'hôte :* le nom ou l'adresse IP de la machine de réseau sur laquelle le Raptor server est installé. Le traitement sera plus rapide si vous utilisez une adresse IP au lieu d'un nom d'hôte. L'adresse IP correspondant à `localhost` (la machine locale) est `127.0.0.1`.
- *Port :* le port permettant d'accéder au serveur Raptor. Ce port est spécifié dans le fichier de configuration de Raptor (appelé `server_config.xml`). Le port doit être fixé et connu de manière à ce que les requêtes puissent être adressées correctement au service. Pour plus d'informations concernant le fichier de configuration Raptor, voir les manuels d'utilisateur : *RaptorXML Server* et *RaptorXML+XBRL Server*.

Une fois avoir saisi les informations de serveur, cliquer sur **OK**. Le nom de serveur que vous avez saisi apparaît dans la liste de serveur (à gauche du volet). Une icône verte apparaît à côté du nom du serveur, indiquant que le serveur Raptor a été lancé et est en marche. Les détails du serveur sont affichés dans le volet (*capture d'écran ci-dessus*). Une icône rouge indique que le serveur est hors ligne. Si le serveur ne peut pas être trouvé, un message d'erreur s'affiche.

Note : le serveur Raptor doit être en marche lorsque le serveur est ajouté. Cela est nécessaire pour que XMLSpy puisse obtenir des informations concernant le serveur et les stocker. Si, une fois que le serveur a été ajouté, le serveur est hors ligne ou ne peut pas être trouvé, ces situations sont indiquées, respectivement, par une icône rouge ou un message d'erreur.

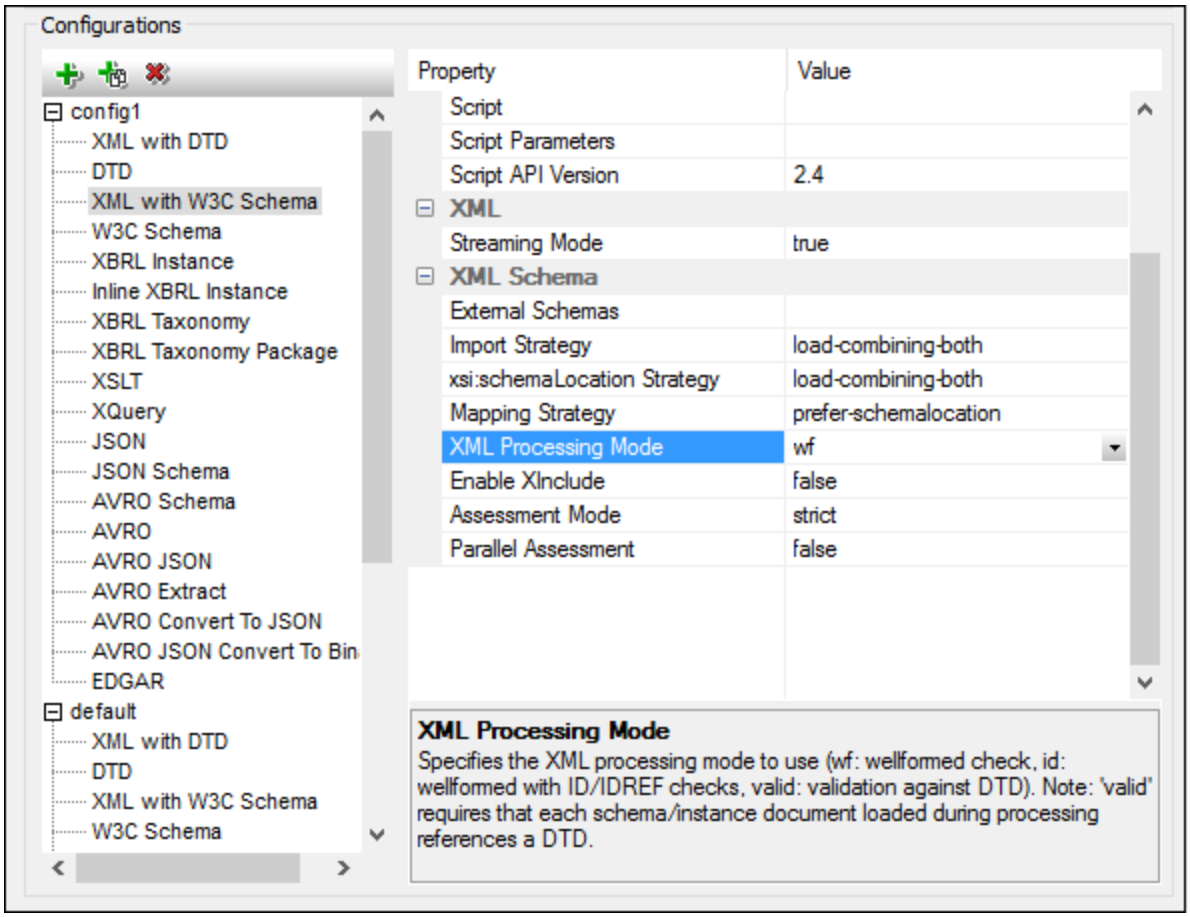
Pour éditer un nom de serveur, un nom d'hôte ou un port, sélectionner le serveur dans le volet de gauche, cliquer sur la touche **Édition**, et, dans le dialogue qui apparaît, éditer les informations que vous souhaitez modifier. Pour supprimer un serveur du pool, sélectionner le serveur et cliquer sur l'icône **Supprimer le serveur sélectionné**.

Configurations de Serveur

Une configuration est un ensemble d'options de validation RaptorXML. Lorsqu'un serveur est ajouté, il sera doté d'une configuration nommée `default`. Il s'agit d'un ensemble d'options RaptorXML configurées sur leurs valeurs par défaut. Vous pouvez ajouter de nouvelles configurations qui contiennent d'autres valeurs d'options. Une fois avoir défini plusieurs configurations de serveur, vous pouvez sélectionner une configuration qui sera la configuration active. Il s'agit de la configuration qui sera utilisée lorsque la commande **Valider sur le serveur** sera exécutée.

Le volet *Configurations* présente deux parties : (i) un volet de gauche, qui montre les configurations et les types, chacune contenant une liste des document-type pouvant être validés ; (ii) un panneau de droite qui affiche les options de validation pour le document-type sélectionné dans le panneau de gauche ; en bas du

volet de droite, vous trouverez une description de l'option sélectionnée (*capture d'écran ci-dessous*).



Ajouter une configuration

Dans le panneau *Configurations* du dialogue Options de RaptorXML Server (*capture d'écran ci-dessus*), cliquer sur **Ajouter une configuration**. Une nouvelle configuration est ajoutée avec des valeurs d'option par défaut. Vous pouvez aussi créer une nouvelle configuration en cliquant sur **Copier la configuration sélectionnée**. Cela permet de créer une nouvelle configuration avec des valeurs d'option qui sont les mêmes que celle de la configuration copiée. Les nouvelles configurations sont créées avec les noms par défaut du type `config<X>`; vous pouvez éditer le nom d'une configuration en double-cliquant dessus et en saisissant le nouveau nom. Vous pouvez ensuite éditer une des valeurs d'option de la configuration.

Éditer les valeurs d'options d'une configuration

Tout d'abord, sélectionner le document-type dans le volet de gauche. Cela permet d'afficher les options de validation du document-type sélectionné dans le volet de droite. Pour éditer la valeur d'une option, choisir une des possibilités suivantes (dépendant du type de la valeur d'option) :

- Si la valeur peut être une dans un ensemble de valeurs prédéfinies, sélectionner la valeur que vous souhaitez depuis la liste de choix de la colonne de valeur de cette option.
- Si la valeur n'est pas limitée, cliquer dans la valeur d'option enregistrée et saisir la valeur que vous souhaitez.
- Si la valeur est un chemin de fichier, outre la possibilité de saisir la valeur, vous pouvez aussi parcourir

 © 2016-2022 Altova GmbH

le fichier que vous souhaitez en utilisant la touche **Parcourir** dans la colonne de valeur de l'option.

Si vous sélectionnez une option, sa description est affichée dans une fenêtre en bas du panneau de droite. Pour des descriptions plus détaillées de chaque option, voir les chapitres de l'interface de ligne de commande des manuels RaptorXML Server et RaptorXML(+XBRL) Server.

Supprimer une configuration
Dans le volet de gauche, choisir la configuration à supprimer et cliquer sur **Supprimer la configuration sélectionnée**.

XMLSpy dans Visual Studio et Eclipse

Lorsque XMLSpy est intégré dans Visual Studio et Eclipse, la configuration active dans ces IDE sera celle qui est actuellement configurée en tant que la configuration active dans la version autonome de XMLSpy.

22.2 Valider avec RaptorXML Server

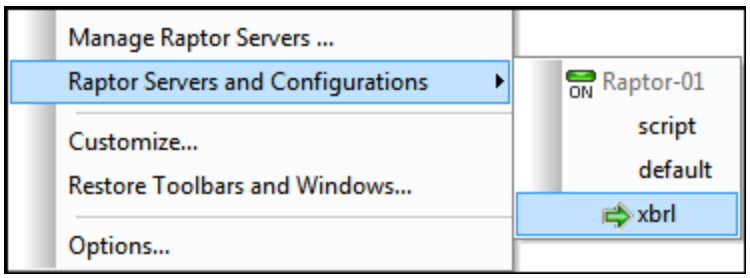
RaptorXML Server vous permet de valider des documents XML et XBRL*. Une validation nécessite deux étapes :

- Choisir le serveur et la configuration de serveur à utiliser pour la validation
- Exécuter la validation (en utilisant une des commandes **Valider sur le serveur** ; *voir ci-dessous*)

***Note :** Il y a deux éditions de Raptor : *RaptorXML Server* (pour les validations XML) et *RaptorXML+XBRL Server* (pour les validations XML et XBRL). Si vous souhaitez valider des documents XBRL, vous devez utiliser RaptorXML+XBRL Server. Pour plus d'informations concernant RaptorXML(+XBRL) Server, veuillez consulter le site Internet Altova et les manuels d'utilisateur : *RaptorXML Server* et *RaptorXML+XBRL Server*.

Sélectionner la configuration de serveur à utiliser

Si vous avez défini plusieurs configurations sur plusieurs serveurs, vous pouvez sélectionner un serveur et une de ses configurations en tant que la configuration active. Celle-ci sera utilisée pour les validations suivantes. En plaçant le curseur sur la commande **Outils | Serveurs Raptor et Configurations** (*capture d'écran ci-dessous*), un sous-menu apparaît qui contient tous les serveurs ajoutés, avec la configuration de chacun d'entre eux. Sélectionner la configuration de serveur que vous souhaitez pour rendre la configuration active. Dans la capture d'écran ci-dessous, la configuration `xbrl` du serveur nommé `Raptor-01` a été sélectionnée en tant que la configuration active (indiquée par la flèche verte).



Valider avec RaptorXML Server

Vous pouvez valider des documents XML et XBRL en utilisant les moteurs de validation de XMLSpy ou en utilisant RaptorXML Server. Pour valider avec RaptorXML Server, choisir une des possibilités suivantes :

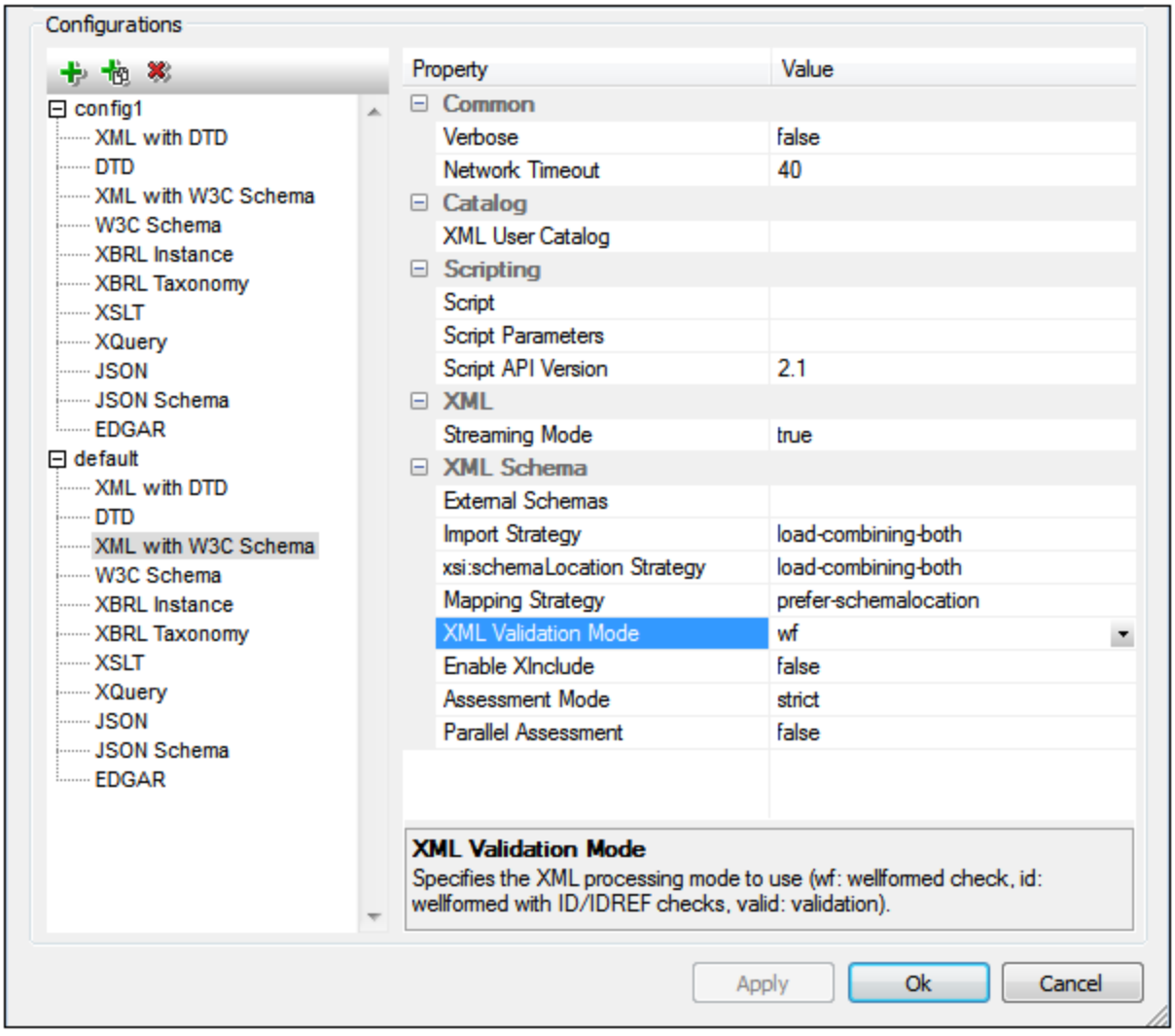
- Cliquer sur l'icône de barre d'outils **Valider sur le serveur**
- Sélectionner la commande **XML | Valider XML sur le serveur (haute-performance) (Ctrl+F8)**
- Dans l'assistant à la saisie Projet, cliquer avec la touche de droite sur le projet, un dossier ou un fichier et sélectionner **Valider XML sur le serveur (haute-performance)** pour valider les données XML ou XBRL dans l'objet sélectionné.

Note : La validation Raptor est disponible dans le Mode Texte, le Mode Schéma, le Mode XBRL et le Mode Grille.

© 2016-2022 Altova GmbH

22.3 Options de validation

Cette section est organisée par le type du document en cours de validation (voir le panneau de gauche de la capture d'écran ci-dessous). Par exemple, *XML avec Schéma W3C* valide un document XML par rapport à un Schéma XML W3C. Lorsqu'un type de validation est sélectionné dans le panneau de gauche, les options de validation de RaptorXML Server disponibles pour ce type de validation sont affichées dans le panneau de droite. Ces options sont organisées en groupes, comme *Scriptage* et *Schéma XML* (*voir capture d'écran ci-dessous*). Veuillez noter que tous les groupes affichés dans la capture d'écran (comme les groupes XBRL) sont disponibles dans l'édition Professional.



Les sous-sections de cette section contiennent des liens vers les descriptions des options de validation RaptorXML Server respectives.

22.3.1 Options communes

Les options qui sont communes à tous les types de validation.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

22.3.2 XML avec DTD

Options de validation de données XML par rapport à un DTD.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- XML

  Activer les espaces de noms
  DTD externe
  Mode Streaming

22.3.3 DTD

Options de validation de DTD.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- Scriptage

© 2016-2022 Altova GmbH

Script
Paramètres de script
Version API de script

22.3.4 XML avec Schéma W3C

Options pour valider les données XML par rapport au schéma XML.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- Scriptage

  Script
  Paramètres de script
  Version API de script

- XML

  Mode Streaming

- Schéma XML

  Schémas externes (xsd)
  Stratégie Import (schema-imports)
  Stratégie xsi:schemaLocation (schemalocation-hints)
  Stratégie Mappage (schema-mapping)
  Mode de traitement XML (xml-mode)
  Activer XInclude (xinclude)
  Mode assignation
  Assignation parallèle
  Mode de contrôle Restriction de type complexe
  Mode de traitement XML pour les Schémas
  Rapporter disparité d'espace de noms d'importation en tant qu'avertissement

22.3.5 Schéma W3C

Options de validation des schémas XML.

- Commun

Info Limite
Version détaillée
Timeout du réseau

- Catalogue

Catalogue de l'utilisateur XML

- Scriptage

Script
Paramètres de script
Version API de script

- Schéma XML

Stratégie Import (schema-imports)
Stratégie xsi:schemaLocation (schemalocation-hints)
Stratégie Mappage (schema-mapping)
Activer XInclude (xinclude)
Mode de traitement XML pour les schémas
Mode de contrôle Restriction de type complexe
Rapporter disparité d'espace de noms d'importation en tant qu'avertissement

22.3.6 Instance Inline XBRL

Options for validating Inline XBRL documents.

- Commun

Info Limite
Version détaillée
Timeout du réseau

- Catalogue

Catalogue de l'utilisateur XML

- Scriptage

Script
Paramètres de script
Version API de script

- XBRL

Activer l'extension de dimensions (dimensions)

© 2016-2022 Altova GmbH

Activer l'extension d'énumérations extensible (extensible-enumerations)
Précharger les schémas de spéc XBRL (preload-xbrl-schemas)
Paquets de taxonomie
Fichier de config packages taxonomie
Traiter les divergences XBRL en tant qu'erreurs (treat-inconsistencies-as-errors)
Fichier UTR
Statut UTR pris en charge
Point d'entrée DTS supplémentaire
Stratégie de Transformation URI (dans les documents de Sortie)
Rapporter disparités Summation-Item
Rapporter disparités Essence-Alias
Rapporter disparités Requires-Element
Activer Extension de libellé préféré générique
Activer extension de liens génériques
Dé-dupliquer
Rapporter duplicatas
Rapporter sévérité duplicatas

- Formule XBRL

Activer l'extension de formule (formula)
Activer l'extension de sévérité d'assertion (assertion-severity)
Précharger les schémas de spéc de formule (preload-formula-schemas)
Rapporter des évaluations d'assertion non satisfaites
Langue de message de validation (message-lang)
Rôle de Message de validation (message-role)
Fichier Formulas-to-Ignore
Fichier Formulas-to-Process
Fichier Assertions-to-Ignore
Fichier Assertions-to-Process
Formulas-to-Ignore
Assertions-to-Ignore
Valider sortie de formule
Activer optimisations de formule

- Table XBRL

Activer extension de table (table)
Précharger les schémas de spéc de table (preload-table-schemas)
Espace de noms de Linkbase de table
Ordre de Table AspectNode

- XBRL XULE

XULE
Taille de pile XULE
Liaisons d'espace de noms instance XULE
Règles XULE à traiter
Rapporter évaluations règle XULE
Fichier de sortie XULE

22.3.7 Instance XBRL

Options de validation des documents d'instance XBRL.

- Commun
  - Info Limite
  - Version détaillée
  - Timeout du réseau
- Catalogue
  - Catalogue de l'utilisateur XML
- Scriptage
  - Script
  - Paramètres de script
  - Version API de script
- Schéma XML
  - Stratégie Import (schema-imports)
  - Stratégie xsi:schemaLocation (schemalocation-hints)
  - Stratégie Mappage (schema-mapping)
  - Activer XInclude (xinclude)
  - Assignation parallèle
  - Mode de contrôle Restriction de type complexe
  - Rapporter disparité d'espace de noms d'importation en tant qu'avertissement
- XBRL
  - Activer l'extension de dimensions (dimensions)
  - Activer l'extension d'énumérations extensible (extensible-enumerations)
  - Activer les extensions de registre d'unité (utr)
  - Précharger les schémas de spéc XBRL (preload-xbrl-schemas)
  - Paquets de taxonomie
  - Valider uniquement DTS référencés (validate-dts-only)
  - Traiter les disparités XBRL en tant qu'erreurs (treat-inconsistencies-as-errors)
- Formule XBRL
  - Activer l'extension de formule (formula)
  - Activer l'extension de sévérité d'assertion (assertion-severity)
  - Précharger les schémas de spéc de formule (preload-formula-schemas)
  - Rapporter des évaluations d'assertion non satisfaites
  - Langue de message de validation (message-lang)
  - Rôle de Message de validation (message-role)
  - Formulas-to-Ignore File
  - Fichier Formulas-to-Process
  - Fichier Assertions-to-Ignore
  - Fichier Assertions-to-Process
  - Formulas-to-Ignore
  - Assertions-to-Ignore

© 2016-2022 Altova GmbH

Valider sortie de Formula
Activer optimisations de formule

- Table XBRL

Activer extension de table (table)
Précharger les schémas de spéc de table (preload-table-schemas)
Espace de noms de Linkbase de table
Ordre de Table AspectNode

- XBRL XULE

XULE
Taille de pile XULE
Liaisons d'espace de noms instance XULE
Règles XULE à traiter
Rapporter évaluations règle XULE
Fichier de sortie XULE

22.3.8 Taxonomie XBRL

Options de validation des taxonomies XBRL.

- Commun

Info Limite
Version détaillée
Timeout du réseau

- Catalogue

Catalogue de l'utilisateur XML

- Scriptage

Script
Paramètres de script
Version API de script

- Schéma XML

Stratégie Import (schema-imports)
Stratégie xsi:schemaLocation (schemalocation-hints)
Stratégie Mappage (schema-mapping)
Activer XInclude (xinclude)
Mode de contrôle Restriction de type complexe
Rapporter disparité d'espace de noms d'importation en tant qu'avertissement

- XBRL

Activer l'extension de dimensions (dimensions)

Activer l'extension d'énumérations extensible (extensible-enumerations)
Précharger les schémas de spéc XBRL (preload-xbrl-schemas)
Paquets de taxonomie
Fichier de config packages taxonomie
Traiter les disparités XBRL en tant qu'erreurs (treat-inconsistencies-as-errors)
Valider uniquement DTS référencés (validate-dts-only)
Traiter les disparités XBRL en tant qu'erreurs (treat-inconsistencies-as-errors)
Activer extension de liens génériques

- Formule XBRL

Activer l'extension de formule (formula)
Activer l'extension de sévérité d'assertion (assertion-severity)
Précharger les schémas de spéc de formule (preload-formula-schemas)

- Table XBRL

Activer extension de table (table)
Précharger les schémas de spéc de table (preload-table-schemas)
Espace de noms de Linkbase de table
Ordre de Table AspectNode

22.3.9 XBRL Taxonomy Package

Options for validating XBRL taxonomy packages.

- Commun

Info Limite
Version détaillée
Timeout du réseau

- Catalogue

Catalogue de l'utilisateur XML

- Scriptage

Script
Paramètres de script
Version API de script

- Schéma XML

Stratégie Import (schema-imports)
Stratégie xsi:schemaLocation (schemalocation-hints)
Stratégie Mappage (schema-mapping)

© 2016-2022 Altova GmbH

22.3.10 XBRL Versioning Report

Options for validating XBRL Versioning.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- Schéma XML

  Stratégie Import (schema-imports)
  Stratégie xsi:schemaLocation (schemalocation-hints)
  Stratégie Mappage (schema-mapping)
  Activer XInclude (xinclude)

22.3.11 XSLT

Options de validation des documents XSLT.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- Schéma XML

  Stratégie Import (schema-imports)
  Stratégie xsi:schemaLocation (schemalocation-hints)
  Stratégie Mappage (schema-mapping)
  Mode de traitement XML (xml-mode)
  Activer XInclude (xinclude)

- Extension Java

  Désactiver les extensions Java (javaext-disable)
  Emplacement d'extension du code-barres (javaext-barcode-location)

- Extensions Graphique

Désactiver les extensions de graphique (chartext-disable)

- Extensions .NET

  Désactiver les extensions .NET (dotnetext-disable)

- Options générales XEngines

  Charger XML avec PSVI (load-xml-with-psvi)

- XSLT

  Version de moteur XSLT (xslt-version)
  Mode Modèle
  Point d'entrée de modèle

22.3.12 XQuery

Options de validation des documents XQuery.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- Schéma XML

  Stratégie Import (schema-imports)
  Stratégie xsi:schemaLocation (schemalocation-hints)
  Stratégie Mappage (schema-mapping)
  Mode de traitement XML (xml-mode)
  Activer XInclude (xinclude)

- Extension Java

  Désactiver les extensions Java (javaext-disable)
  Emplacement d'extension du code-barres (javaext-barcode-location)

- Extensions Graphique

  Désactiver les extensions de graphique (chartext-disable)

- Extensions .NET

  Désactiver les extensions .NET (dotnetext-disable)

- Options générales XEngines

© 2016-2022 Altova GmbH

Charger XML avec PSVI (load-xml-with-psvi)

- XQuery

  Version de moteur XQuery (xquery-version)
  Omettre Déclaration XML

22.3.13 JSON

Options de validation des documents (instance) JSON.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- JSON

  Chemin de schéma JSON

22.3.14 JSON Schema

Options de validation des documents de Schéma JSON.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

22.3.15 AVRO

Options for validating a data block in one or more Avro binary files against the respective Avro schemas in each binary file.

- Commun

Info Limite
Version détaillée
Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

22.3.16 AVRO JSON

Options for validating a JSON document against an AVRO schema.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

22.3.17 AVRO Schema

Options for validating one or more Avro schema documents against the Avro schema specification.

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

22.3.18 EDGAR

EDGAR (Electronic Data Gathering, Analysis, and Retrieval) est un système qui effectue une collection, validation et un indexage automatisé de déclarations financières déposées par des entreprises auprès de la

© 2016-2022 Altova GmbH

United States SEC (Securities and Exchange Commission). En validant avec EDGAR, Raptor valide le document d'instance XBRL à l'aide d'un script EDGAR interne. Vous pouvez définir les options supplémentaires suivantes :

- Paramètres de script EDGAR

  Le script EDGAR effectue des contrôles supplémentaires tels que prescrit dans le manuel EDGAR Filing Manual Volume II: EDGAR Filing. Le script permet de spécifier les paramètres de script supplémentaires suivants :

`CIK`	Le CIK du déclarant
`submissionType`	Le type de soumission EDGAR, par exemple : `'10-K'`
`cikList`	Une liste des CIK, chacun séparé par une virgule : `','`
`cikNameList`	Une liste des noms de déclarants officiels pour chaque CIK dans `cikList`, séparé par `'|Edgar|'`
`forceUtrValidation`	Configuré sur `true` pour forcer l'activation de la validation UTR
`edbody-url`	Le chemin vers le `edbody.dtd` utilisé pour valider des fragments intégrés HTML
`edgar-taxonomies-url`	Le chemin vers le `edgartaxonomies.xml`, qui contient une liste de fichiers de taxonomie dont la référence est autorisée depuis la taxonomie d'extension de l'entreprise

- Commun

  Info Limite
  Version détaillée
  Timeout du réseau

- Catalogue

  Catalogue de l'utilisateur XML

- Schéma XML

  Stratégie Import (schema-imports)
  Stratégie xsi:schemaLocation (schemalocation-hints)
  Stratégie Mappage (schema-mapping)
  Activer XInclude (xinclude)
  Assignation parallèle
  Mode de contrôle Restriction de type complexe
  Rapporter disparité d'espace de noms d'importation en tant qu'avertissement

- XBRL

  Activer l'extension de dimensions (dimensions)
  Activer l'extension d'énumérations extensible (extensible-enumerations)
  Précharger les schémas de spéc XBRL (preload-xbrl-schemas)
  Paquets de taxonomie
  Fichier de config packages taxonomie
  Traiter les divergences XBRL en tant qu'erreurs (treat-inconsistencies-as-errors)
  Fichier UTR
  Statut UTR pris en charge

Point d'entrée DTS supplémentaire
Stratégie de Transformation URI (dans les documents de Sortie)
Rapporter disparités Summation-Item
Rapporter disparités Essence-Alias
Rapporter disparités Requires-Element
Activer Extension de libellé préféré générique
Activer extension de liens génériques
Dé-dupliquer
Rapporter duplicatas
Rapporter sévérité duplicatas

- Formule XBRL

Activer l'extension de formule (formula)
Activer l'extension de sévérité d'assertion (assertion-severity)
Précharger les schémas de spéc de formule (preload-formula-schemas)
Rapporter des évaluations d'assertion non satisfaites
Langue de message de validation (message-lang)
Rôle de Message de validation (message-role)
Fichier Formulas-to-Ignore
Fichier Formulas-to-Process
Fichier Assertions-to-Ignore
Fichier Assertions-to-Process
Formulas-to-Ignore
Assertions-to-Ignore
Valider sortie de formule
Activer optimisations de formule

- Table XBRL

Activer extension de table (table)
Précharger les schémas de spéc de table (preload-table-schemas)
Espace de noms de Linkbase de table
Ordre de Table AspectNode

- XBRL XULE

XULE
Taille de pile XULE
Liaisons d'espace de noms instance XULE
Règles XULE à traiter
Rapporter évaluations règle XULE
Fichier de sortie XULE

© 2016-2022 Altova GmbH

22.4 XSLT et XQuery avec RaptorXML Server

Vous pouvez utiliser RaptorXML Server pour exécuter des (i) transformations XSLT (ii) et des mises à jour XQuery ou des exécutions sur des documents XML. Ces actions sont disponibles uniquement via Projets, et comprennent trois étapes :

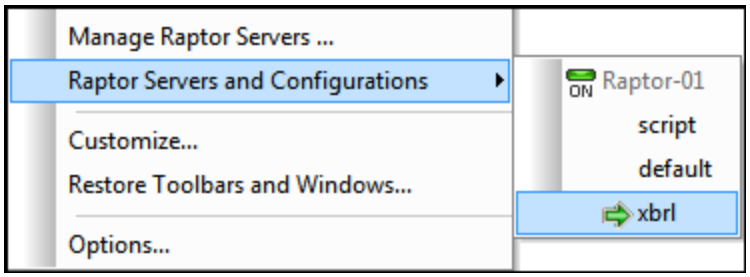
- Sélectionner le serveur et la configuration de serveur pour utiliser pour la tâche.
- Configurer le dossier de projet et spécifier les fichiers XSLT/XQuery à utiliser (dans le dialogue Propriétés de projet). Les fichiers XSLT/XQuery qui sont assignés dans le dialogue Propriétés de projet d'un dossier sont les fichiers qui seront utilisés pour les transformations XSLT et XQuery de tous les fichiers XML dans ce dossier de projet. Vous ne pouvez pas attribuer de fichiers XSLT/XQuery pour des fichiers individuels XML dans un dossier de projet ; les fichiers XSLT/XQuery peuvent uniquement être assignés à un dossier complet.
- Exécuter la transformation XSLT ou la mise à jour/transformation XQuery.

Note : si le document XSLT ou XQuery utilise des fonctions d'extension Java ou des fonctions d'extension .NET, les chemins de fichiers sont utilisés pour localiser les fichiers JAR (Java) ou les fichiers assembly externes (non enregistrés) (.NET). Cela signifie que si le même document XSLT/XQuery est utilisé pour les transformations/exécutions via XMLSpy et en tant que RaptorXML Server, les chemins de fichier contenus menant aux fichiers JAR et/ou les fichiers assembly doivent localiser ces fichiers correctement.

Note : si RaptorXML Server se trouve sur la même machine que XMLSpy, vous devriez, pour une performance idéale, spécifier que les paramètres de serveur **`server.unrestricted-filesystem-access`** ont une valeur `true`. Pour plus d'informations, voir la documentation du fichier de configuration RaptorXML Server.

Sélectionner la configuration de serveur à utiliser

Si vous avez défini plusieurs configurations sur plusieurs serveurs, vous pouvez sélectionner un serveur et une de ses configurations en tant que la configuration active. Celle-ci sera utilisée pour les validations suivantes. En plaçant le curseur sur la commande **Outils | Serveurs Raptor et Configurations** (*capture d'écran ci-dessous*), un sous-menu apparaît qui contient tous les serveurs ajoutés, avec la configuration de chacun d'entre eux. Sélectionner la configuration de serveur que vous souhaitez pour rendre la configuration active. Dans la capture d'écran ci-dessous, la configuration `xbrl` du serveur nommé `Raptor-01` a été sélectionnée en tant que la configuration active (indiquée par la flèche verte).



Exécuter une transformation XSLT

Vous pouvez effectuer une transformation XSLT en utilisant les moteurs XSLT de XMLSpy ou en utilisant RaptorXML Server. Pour effectuer des transformations XSLT avec RaptorXML Server, procéder comme suit :

- Cliquer avec la touche de droite sur le dossier de projet où les fichiers XML à transformer sont situés. Ce dossier peut être le dossier de projet entier ou un dossier individuel se trouvant n'importe où dans la hiérarchie du projet
- Dans le menu qui apparaît, choisir la commande **XSL Transformation sur serveur (haute performance)**

Note : vous ne pouvez pas attribuer de fichiers XSLT/XQuery pour des fichiers XML individuels dans un dossier de projet ; les fichiers XSLT/XQuery peuvent uniquement être attribuées pour un dossier complet. Voir début de la section.

Pour plus d'informations, voir les sections XSLT et Transformation XSLT.

Exécuter une mise à jour/exécution XQuery

Vous pouvez effectuer une mise à jour/transformation XQuery en utilisant les moteurs XQuery de XMLSpy ou en utilisant RaptorXML Server. Pour effectuer une mise à jour/transformation XQuery avec RaptorXML Server, procéder comme suit :

- Cliquer avec la touche de droite sur le dossier de projet où les fichiers XQuery ou XML à mettre à jour ou à exécuter (respectivement) se trouvent. Ce dossier peut être le dossier de projet entier ou un dossier individuel se trouvant n'importe où dans la hiérarchie du projet.
- Dans le menu qui apparaît, choisir la commande **XQuery/Mettre à jour exécution sur serveur (haute performance)**

Note : vous ne pouvez pas attribuer de fichiers XSLT/XQuery pour des fichiers XML individuels dans un dossier de projet ; les fichiers XSLT/XQuery peuvent uniquement être attribuées pour un dossier complet. Voir début de la section.

Pour plus d'informations, voir les sections XQuery et XQuery/ Exécution de mise à jour.

© 2016-2022 Altova GmbH

23 Comparaisons Fichier/Répertoire

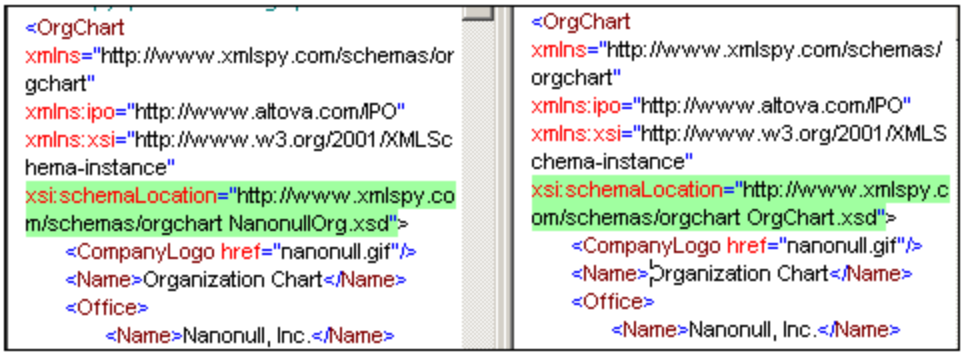
XMLSpy propose des fonctions de Comparaison des fichiers et de comparaison des répertoires liées l'une à l'autre. Lancer les comparaisons de fichiers et les comparaisons de répertoires avec les commandes **Comparer fichier ouvert avec** et **Comparer répertoires** dans le menu **Outils**, respectivement. Les options de comparaison pour les comparaisons de fichiers peuvent être définies dans le dialogue Paramètres qui est accessible en cliquant sur la commande **Comparer options** dans le menu **Outils**.

Chacune de ces commandes est décrite en détails dans la section Référence de l'utilisateur. Dans les sous-sections de cette section, nous proposons un aperçu des mécanismes des Comparaisons de fichiers et Comparaisons de répertoires.

23.1 Comparaisons de fichiers

La fonction de Comparaisons de fichiers vous permet de comparer de fichier actif avec un autre fichier, qui a été sélectionné par le biais du dialogue Ouvrir fichier ou par le biais d'une ressource globale. Les points suivants vous confèrent un aperçu du mécanisme. Pour plus de détails, voir la section Référence de l'utilisateur.

- Les paramètres actuellement définis dans le dialogue Options Comparer lorsqu'une session Comparer les fichiers est lancée, sont les paramètres qui seront actifs pour cette session.
- Vous pouvez choisir de comparer les fichiers en tant que fichiers XML (où la structure du document est aussi évaluée) ou en tant que fichiers de texte. Ce choix est effectué en sélectionnant, dans le dialogue Paramètres, soit (i) le Mode Grille, soit le Mode Texte (case Comparison textuelle seulement décochée) pour les comparaisons XML, ou (ii) Mode Texte (case Comparison textuelle seulement cochée) pour les comparaisons de texte.
- Les deux fichiers apparaissent dans deux volets adjacents dans le mode sélectionné (Mode Grille ou Mode Texte) et les différences sont marquées dans les deux fichiers (*capture d'écran ci-dessous*).



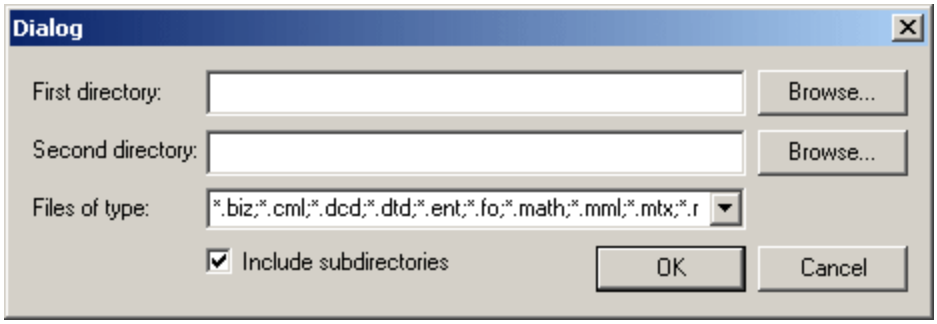
Une fenêtre de contrôle Comparer les fichiers apparaît, vous permettant de parcourir les différences et de les fusionner.

Le dialogue Paramètres contient plusieurs options pour spécifier quels aspects des documents XML doivent être considérés pour la comparaison, et quels aspects doivent être ignorés. Pour plus de détails, voir la section Options Comparer dans la Référence de l'utilisateur.

© 2016-2022 Altova GmbH

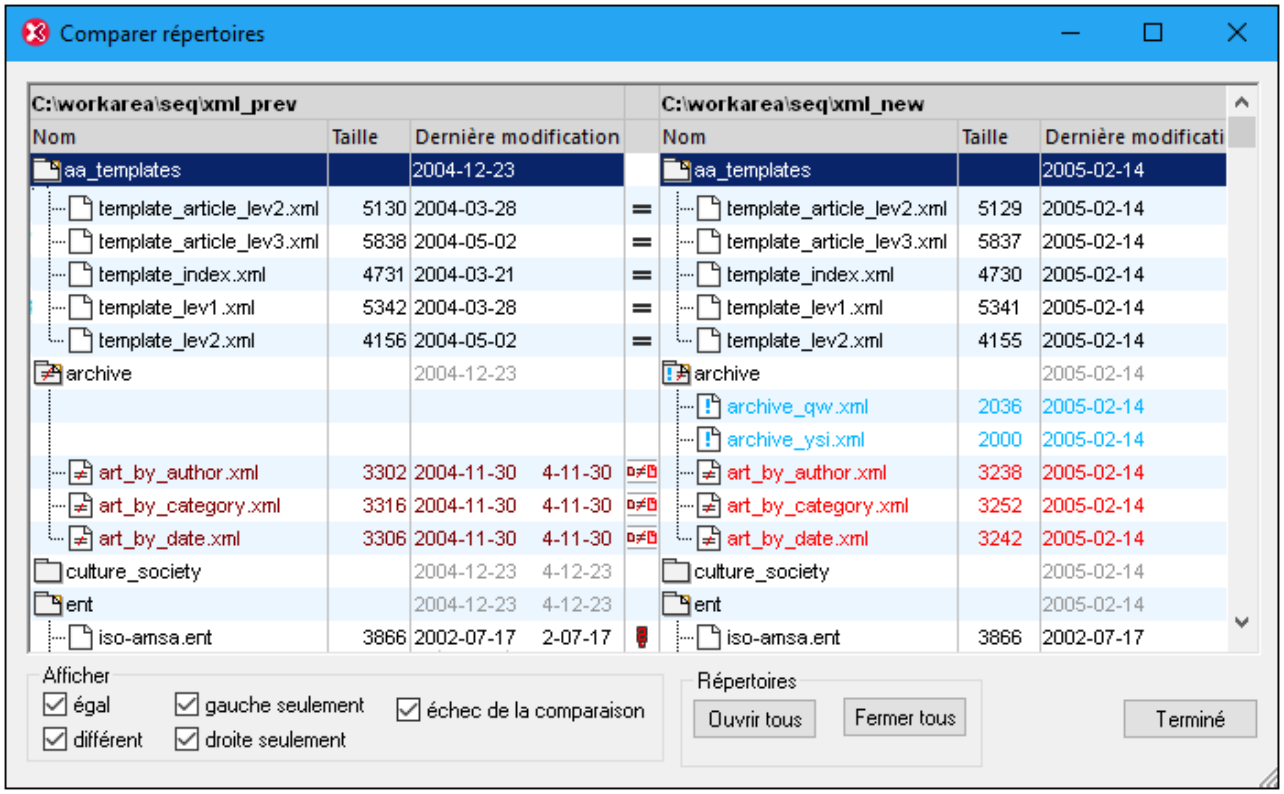
23.2 Comparaisons de répertoires

La fonction de Comparaison de répertoires vous permet de comparer les répertoires, que vous pouvez choisir dans des dialogues séparés Rechercher un dossier. Vous pouvez aussi choisir si vous voulez comparer des sous-répertoires, et quel type de fichier doit être pris en compte pour la comparaison de répertoire.



Les archives zip peuvent être incluses dans la comparaison en incluant l'extension de fichier zip dans la liste des types de fichier à évaluer.

Les répertoires sont comparés pour indiquer des fichiers manquants et si les fichiers portant le même nom sont différents ou pas. Les comparaisons entre les fichiers sont basées sur les paramètres dans le dialogue Paramètres. Les résultats de la comparaison de répertoires sont affichés dans une fenêtre séparée (*capture d'écran ci-dessous*).



Pour plus de détails concernant la lecture des symboles et la gestion du mode dans la fenêtre Comparer les répertoires, voir la description de la commande **Comparer les répertoires** dans la Référence de l'utilisateur. Vous pouvez ensuite double-cliquer sur une ligne de fichier pour lancer une comparaison de fichiers.

© 2016-2022 Altova GmbH

24 Contrôle de code source

La prise en charge du contrôle de code source dans XMLSpy est disponible avec le Microsoft Source Control Plug-in API (anciennement appelé MSSCCI API), versions 1.1, 1.2 et 1.3. Cela vous permet d'exécuter les commandes de contrôle de source comme par exemple "Archiver" ou "Extraire" directement depuis XMLSpy vers n'importe quel système de contrôle de source qui permet à des clients natifs ou tiers de s'y connecter par le biais de la Microsoft Source Control Plug-in API.

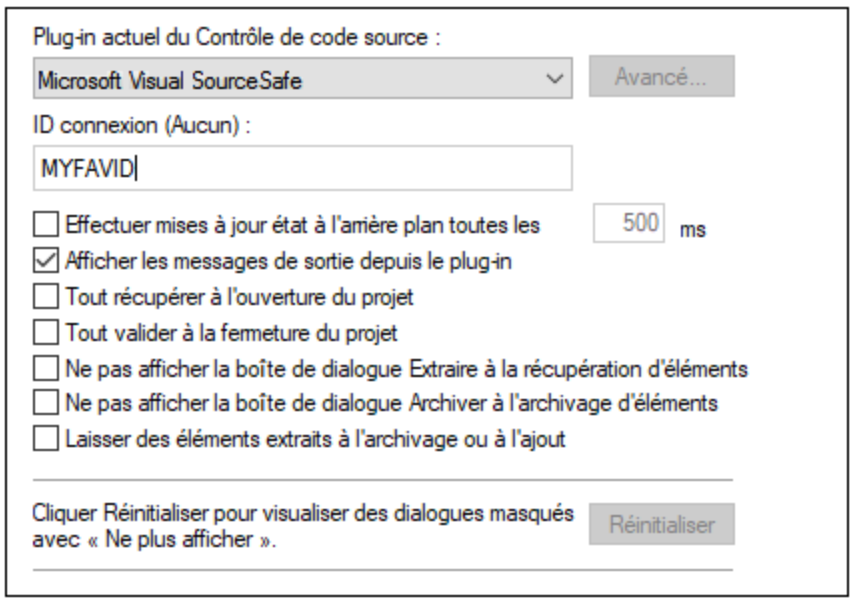
Vous pouvez utiliser, en guise de votre fournisseur de contrôle de source, tout plug-in commercial ou non commercial qui prend en charge la Microsoft Source Control Plug-in API, et qui peut se connecter à un système de contrôle de version compatible. En ce qui concerne la liste des systèmes de contrôle de source et des plug-ins testés par Altova, voir Systèmes de contrôle de la source pris en charge.

Installer et configurer le fournisseur de contrôle de source

Pour visionner les fournisseurs de contrôle de source disponibles sur votre système, procéder comme suit :

1. Dans le menu **Outils**, cliquer sur **Options**.
2. Cliquer dans l'onglet **Contrôle de code source**.

Tout plug-in de contrôle de source compatible avec la Microsoft Source Code Control Plug-in API est affiché dans la liste déroulante **Plug-in actuel du contrôle de code source**.



Si un plug-in compatible ne peut pas être trouvé sur votre système, le message suivant s'affichera :

```
"L'inscription des fournisseurs de contrôle de code source installés n'a pas été trouvée ou est incomplète."
```

Certains systèmes de contrôle de source peuvent ne pas installer automatiquement le plug-in de contrôle de source. Dans ce cas, vous devrez l'installer séparément. Pour plus d'instructions, veuillez vous référer à la documentation du système de contrôle de source respectif. Un (fournisseur) de plug-in compatible avec la

Microsoft Source Code Control Plug-in API devra être enregistré sous l'entrée de registre suivante sur votre système d'exploitation :

```
HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders
```

Une fois correctement installé, le plug-in devient automatiquement disponible dans la liste des plug-ins disponibles pour XMLSpy.

Accéder aux commandes de contrôle de source

Les commandes reliées au contrôle de source sont disponibles dans le menu **Projet | Contrôle de code source**.

Ressource / Problèmes de vitesse

Des bases de données de contrôle de source très importantes peuvent introduire une pénalité de vitesse/ressource lors de l'exécution automatique des mises à jour en arrière-plan.

Vous pourrez accélérer votre système en désactivant (ou en augmentant l'intervalle de) l'option **Effectuer des mises à jour d'état en arrière-plan toutes les ... secondes** dans l'onglet **Contrôle de source** accessible depuis **Outils | Options**.

Note : La version **64-bit** de votre application Altova prend automatiquement en charge n'importe lequel des programmes de contrôle de source 32-bit contenu dans cette documentation. Si vous utilisez une application Altova 64-bit avec un programme de contrôle de source 32-bit, l'option **des mises à jour d'état en arrière-plan toutes les ... secondes** est automatiquement grisée et ne peut pas être sélectionnée.

Différencier avec Altova DiffDog

Vous pouvez configurer de nombreux systèmes de contrôle de source (y compris Git et TortoiseSVN) pour qu'ils puissent utiliser Altova DiffDog en tant que leur outils de différenciation. Pour plus d'information concernant DiffDog, voir https://www.altova.com/diffdog.html. Pour consulter une documentation DiffDog, voir https://www.altova.com/documentation.html.

© 2016-2022 Altova GmbH

24.1 Configurer le contrôle de source

Le mécanisme permettant de configurer le contrôle de source et de placer les fichiers dans un projet XMLSpy se trouvant sous contrôle de source est le suivant :

1. Si cela n'a pas déjà été fait, installer le système de contrôle de source (voir Systèmes de contrôle de source pris en charge) et configurer la base de contrôle de source (archivage) dans lequel vous souhaitez enregistrer votre travail.
2. Créer un dossier de poste de travail local qui contiendra les fichiers de travail que vous souhaitez placer sous contrôle de source. Le dossier qui contient tous vos dossiers et fichiers de poste de travail est appelé le dossier local et le chemin menant au dossier local est appelé le chemin local. Ce dossier local sera lié à un dossier particulier dans l'archivage.
3. Dans votre application Altova, créer un dossier de projet d'application auquel vous devez ajouter les fichiers que vous souhaitez placer sous contrôle de source. Cette organisation de fichiers dans un projet d'application est abstrait. Les fichiers d'un projet référencent les fichiers physiques enregistrés localement, de préférence dans un dossier (avec des sous-dossiers si requis) pour chaque projet.
4. Dans la base de données du contrôle de source du système (aussi appelé contrôle de source ou archivage), un dossier est créé qui est lié au dossier local. Ce dossier (appelé le dossier lié) copiera la structure du dossier local afin que tous les fichiers devant être placés sous contrôle de source se situent correctement en terme d'hiérarchie dans le dossier lié. Le dossier lié est généralement créé lorsque vous ajoutez un fichier ou un projet d'application au contrôle de source pour la première fois. Voir la section, Projet d'application, pour plus d'informations concernant la structure du dossier de l'archivage.
5. Les fichiers de projet sont ajoutés au contrôle de source avec la commande **Projet | Contrôle de source | Ajouter au Contrôle de source**. Lorsque vous ajoutez un projet ou un fichier dans un projet pour la première fois au contrôle de source, les liaisons et la structure de dossier correcte sera créée dans l'archivage.
6. Les actions de contrôle de source, comme l'archivage et l'extraction de fichiers, et la suppression de fichiers depuis le contrôle de source peuvent être effectués depuis la commande dans le sous-menu **Projet | Contrôle de source**. Ces commandes sont décrites dans la section du menu Projet de la Référence de l'utilisateur.

Note : Si vous souhaitez changer le fournisseur de contrôle de source actuel, vous pouvez vous appuyer sur deux méthodes : (i) par le biais des options Contrôle de source (**Outils | Options | Contrôle de source**), ou (ii) dans le dialogue Changer le Contrôle de source (**Projet | Contrôle de source | Changer le Contrôle de source**).

24.2 Systèmes de Contrôle de source pris en charge

La liste ci-dessous montre les Source Control Servers (SCSs) pris en charge par XMLSpy, avec leurs Clients de Contrôle de source (Source Control Clients -SCC) respectifs. La liste est organisée alphabétiquement par SCS. Veuillez noter les éléments suivants :

- Altova a mis en place la Microsoft Source Control Plug-in API (versions 1.1, 1.2 et 1.3) dans XMLSpy, et a testé la prise en charge des pilotes recensés et des systèmes de contrôle de révision. Il est prévu que XMLSpy poursuivra la prise en charge de ces produits, si et quand ils sont mis à jours.
- Les clients du Contrôle de code source qui ne sont pas recensés ci-dessous, mais qui mettent en place la Microsoft Source Control Plug-in API devraient aussi fonctionner avec XMLSpy.

Système de contrôle de source	Clients de Contrôle de source
AccuRev 4.7.0 Windows	AccuBridge pour Microsoft SCC 2008.2
Bazaar 1.9 Windows	Aigenta Unified SCC 1.0.6
Borland StarTeam 2008	Borland StarTeam Cross-Platform Client 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10 (Server)	Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)
Collabnet Subversion 1.5.4	• Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 version 1.6.3.1 • TamTam SVN SCC 1.2.24
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere pour VSS 5.3.2 Standard/Professional Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
Git	Plug-in PushOK GIT SCC (voir Source Control with Git)
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5 (2.5.03.2382)	Aigenta Unified SCC 1.0.6
March-Hare CVS Suite 2008	• Jalindi Igloo 1.0.3 • March-Hare CVS Suite Client 2008 (3321) • PushOK CVS SCC NT 2.1.2.5 • PushOK CVS SCC x64 version 2.2.0.4 • TamTam CVS SCC 1.2.40
Mercurial 1.0.2 pour Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 avec CTP	Microsoft SourceSafe 2005 avec CTP

© 2016-2022 Altova GmbH

Système de contrôle de source	Clients de Contrôle de source
Microsoft Visual Studio Team System 2008/2010 Team Foundation Server	Microsoft Team Foundation Server 2008/2010 MSSCCI Provider
Perforce 2008 P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.1.35	QSC Team Coherence Client 7.2.1.35
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM Client/Server pour Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 pour Win32 Server	Serena Dimensions 10.1.3 pour Win32 Client
Softimage Alienbrain Server 8.1.0.7300	Softimage Alienbrain Essentials/Advanced Client 8.1.0.7300
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite Server 4.2.0	SourceGear SourceOffsite Client 4.2.0 (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6	• Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 version 1.6.3.1 • TamTam SVN SCC 1.2.24

24.3 Dossier Poste de travail local

Les fichiers avec lesquels vous allez travailler doivent être enregistrés dans une hiérarchie à l'intérieur d'un dossier de poste de travail local (*voir diagramme ci-dessous*).

```
Local Workspace Folder
|
|-- MyProject.spp
|-- QuickStart
|   |-- QuickStart.css
|   |-- QuickStart.xml
|   |-- QuickStart.xsd
|-- Grouping
|   |-- Persons
|   |   |-- Persons.xml
```

Le fichier de projet d'application (fichier `.spp`) sera généralement situé directement dans le dossier du poste de travail local (*voir diagramme ci-dessus*).

Si un ou plusieurs fichiers de ce dossier (de poste de travail) sont placés sous contrôle de source, la structure du dossier du poste de travail local sera partiellement ou entièrement reproduite dans l'archivage. Par exemple, si le fichier `Persons.xml` provenant du dossier local affiché ci-dessus est placé sour contrôle de source, le chemin y menant depuis l'archivage sera :

```
[RepositoryFolder]/MyProject/Grouping/Persons/Persons.xml
```

Le dossier `MyProject` dans le dossier d'archivage est lié au dossier local. Généralement, il portera le nom du projet, mais vous pouvez lui donner n'importe quel nom.

Si tout le projet d'application est placé sous le contrôle de source (en choisissant le nom du projet dans la fenêtre Projets et en le plaçant sous contrôle de source), alors l'ensemble de la structure du dossier local est recréée dans l'archivage.

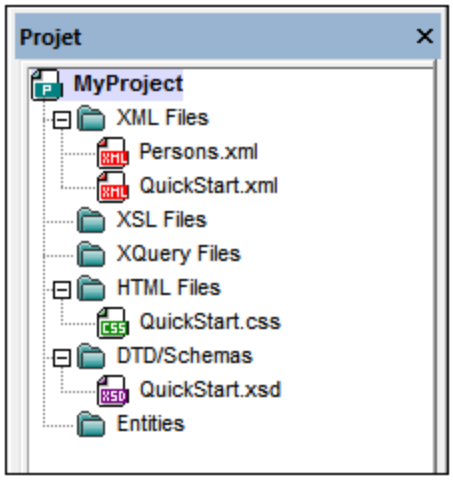
Note : Les fichiers provenant de l'extérieur du dossier du poste de travail local peuvent être ajoutés au projet d'application. Mais le fait de pouvoir placer ce type de fichier sous contrôle de source dépend du système de contrôle de source que vous utilisez. Certains systèmes de contrôle de source peuvent avoir des difficultés à placer un fichier depuis l'extérieur du dossier local dans l'archivage. C'est pourquoi nous recommandons que tous les fichiers de projet que vous souhaitez placer sous contrôle de source soient placés dans le dossier du poste de travail local.

© 2016-2022 Altova GmbH

24.4 Projet d'application

Créer ou charger le projet d'application Altova que vous souhaitez placer sous contrôle de source. Si vous souhaitez placer un seul fichier sous contrôle de source, ce fichier devra être intégré dans un projet, étant donné qu'un contrôle de source peut uniquement être accédé par un projet.

Par exemple, veuillez considérer un projet dans l'application XMLSpy d'Altova. Les propriétés de projet sont enregistrées dans un fichier `.spp`. Dans l'application, le projet est affiché dans la fenêtre Projet de l'application (*voir capture d'écran ci-dessous*). Le projet dans la capture d'écran ci-dessous est nommé `MyProject` et les propriétés du projet sont sauvegardées dans le fichier `MyProject.spp`.



Vous pouvez placer tout le projet (tous les fichiers dans le projet) ou bien seulement certains fichiers de projet sous contrôle de source. **Seuls des fichiers se trouvant dans le projet peuvent être placés sous contrôle de source.** Donc si vous devez ajouter des fichiers au projet avant de pouvoir les placer sous contrôle de source. Le fichier de projet (fichier `.spp`) sera placé automatiquement sous contrôle de source dès qu'un fichier dans le projet est placé sous contrôle de source.

L'ensemble du projet, ou bien un ou plusieurs fichiers du projet, est placé sous contrôle de source par le biais de la commande **Projet | Contrôle de source | Ajouter le Contrôle de source** (*voir section suivante*).

Veuillez noter, néanmoins, que la structure du dossier de l'archivage ne correspond pas à la structure de dossier du projet (*capture d'écran ci-dessus*) mais à la structure du dossier de poste de travail local (*voir diagramme de dossier ci-dessous*). Dans le diagramme ci-dessous, veuillez noter que le dossier `MyProject` dans l'archivage a une structure de dossier correspondant à celle du dossier de poste de travail local. Veuillez noter que le dossier lié se produit dans le dossier d'archivage.

```
Local Workspace Folder                Archivage
|                                     |
|-- MyProject.spp                     |-- MyProject (lié au Poste de travail local)
|-- QuickStart                        ||-- MyProject.spp
|   |-- QuickStart.css                ||-- QuickStart
|   |-- QuickStart.xml                ||   |-- QuickStart.css
|   |-- QuickStart.xsd                ||   |-- QuickStart.xml
```

```
|-- Grouping                    ||   |-- QuickStart.xsd
|   |-- Persons                 ||-- Grouping
|   |   |-- Persons.xml         ||   |-- Persons
                                ||   |   |-- Persons.xml
```

Note : un projet d'application peut contenir des dossiers de projet (vert) et des dossiers externes (jaune). Seuls les fichiers contenus dans les dossiers de projet (vert) peuvent être placés sous contrôle de source. Les fichiers contenus dans les dossiers externes (jaune) ne peuvent pas être placés sous contrôle de source.

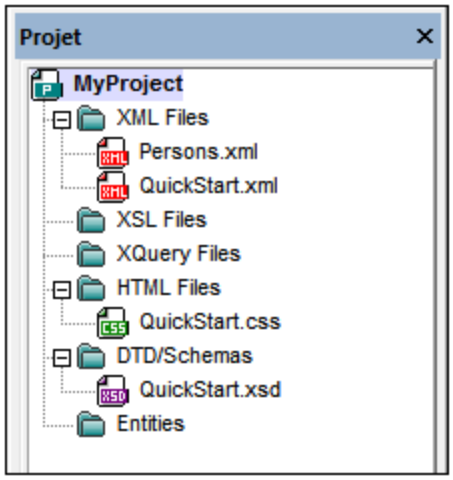
Note : les fichiers provenant d'en dehors du dossier de poste de travail local peuvent être ajoutés au projet d'application. Mais que vous placiez ce type de fichier sous contrôle de source dépend du système de contrôle de source que vous utilisez. Certains systèmes de contrôle de source peuvent avoir des difficultés à placer un fichier dans l'archivage depuis l'extérieur du dossier local. C'est pourquoi nous recommandons que tous les fichiers de projet que vous souhaitez placer sous contrôle de source soient situés dans le dossier de poste de travail local.

© 2016-2022 Altova GmbH

24.5 Ajouter au contrôle de source

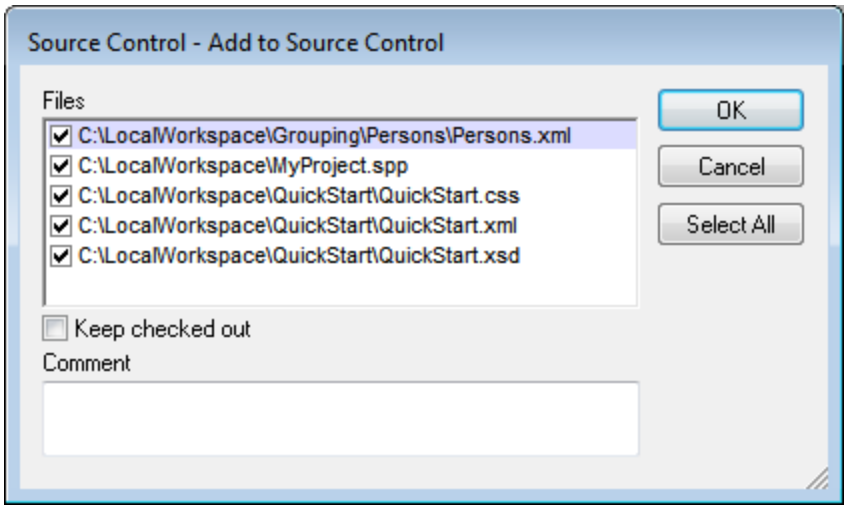
L'ajout du projet au contrôle de source créera automatiquement les liaisons correctes et la structure d'archivage avant d'ajouter le fichier de projet (fichier `.spp`) ou les fichiers individuels au contrôle de source. Ajouter le projet au contrôle de source comme suit.

Choisir le projet dans la fenêtre Projet (`MyProject` dans la capture d'écran ci-dessous) de manière à ce qu'il soit marqué (*comme dans la capture d'écran ci-dessous*). En alternative, choisir un seul fichier ou sélectionner plusieurs fichiers en cliquant dessus tout en maintenant la touche **Ctrl** appuyée. L'ajout d'un seul fichier au contrôle de source ajoutera aussi automatiquement le fichier de projet (fichier `.spp`) au contrôle de source.



Ensuite, choisir la commande de menu **Projet | Contrôle de source | Ajouter au Contrôle de source**. Cela entraîne l'ouverture des dialogues de connexion et de configuration du système de contrôle de source actuellement sélectionné. (Vous pouvez changer le système de contrôle de source par le biais du dialogue Changer Contrôle de source (**Projet | Contrôle de source | Changer Contrôle de source**).)

Suivre les instructions du système de contrôle de source pour établir la connexion et la configuration. Une fois cette étape achevée, tous les fichiers sélectionnés pour l'ajout plus le fichier de projet (fichier `.spp`) sont affichés dans un dialogue Ajouter au contrôle de source (*capture d'écran ci-dessous*). Choisir les fichiers que vous souhaitez ajouter et cliquer sur **OK**.



Les fichiers seront ajoutés à l'archivage et peuvent être archivés ou extraits selon que la case *Garder ces fichiers extraits* a été cochée ou pas.

Notes de configuration

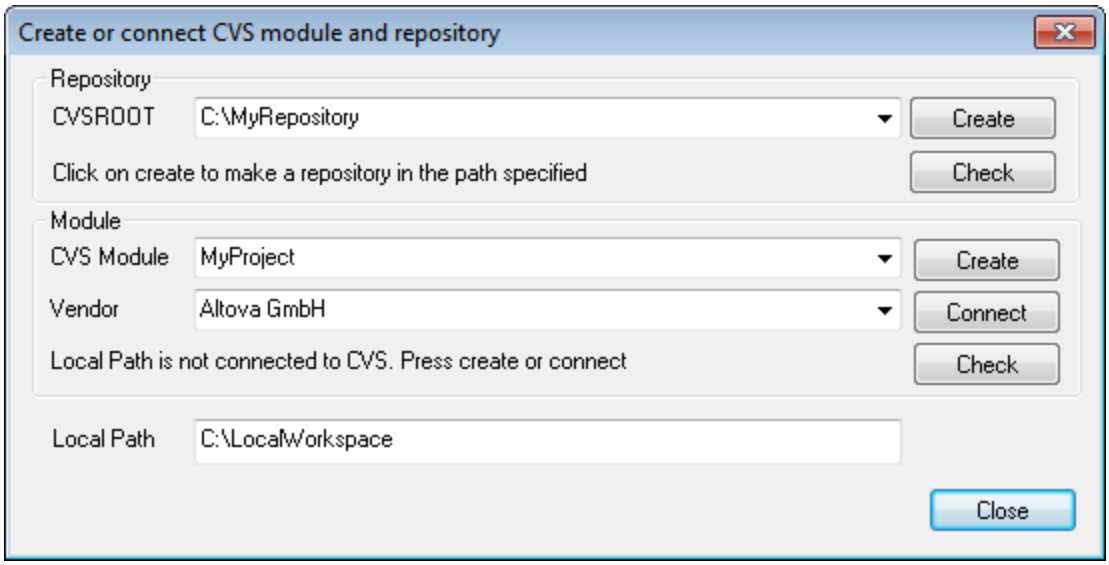
Vous serez éventuellement invité à créer un dossier dans l'archivage pour le projet s'il n'a pas déjà été créé. Si vous êtes invité, créez le dossier. Le dossier de poste de travail local sera lié à ce dossier créé dans l'archivage (*voir les diagrammes ci-dessous*).

```
Local Workspace Folder
|
|-- MyProject.spp
|-- QuickStart
|   |-- QuickStart.css
|   |-- QuickStart.xml
|   |-- QuickStart.xsd
|-- Grouping
|   |-- Persons
|   |   |-- Persons.xml
```

```
Archivage
|
|-- MyProject (lié au poste de travail local)
| |-- MyProject.spp
| |-- QuickStart
| |   |-- QuickStart.css
| |   |-- QuickStart.xml
| |   |-- QuickStart.xsd
| |-- Grouping
| |   |-- Persons
| |   |   |-- Persons.xml
```

Le dialogue de configuration de Jalindi Igloo est affiché ci-dessous. Le champ CVSROOT est le chemin menant au dossier de l'archivage.

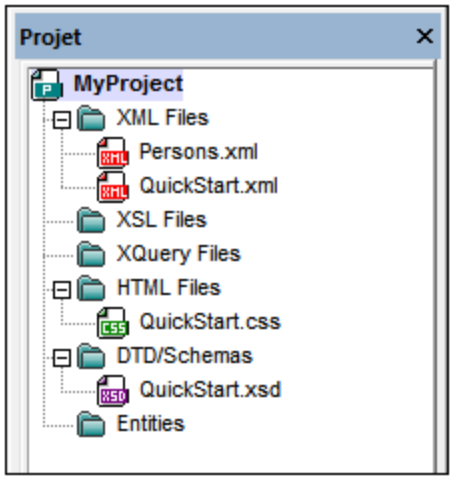
© 2016-2022 Altova GmbH



Dans la capture d'écran ci-dessus, le chemin local situe le dossier de poste de travail local qui correspond au module CVS, `MyProject`, et il y est lié.

24.6 Travailler avec le contrôle de source

Afin de travailler avec le contrôle de source, sélectionner le projet, un dossier de projet, ou un fichier de projet dans la fenêtre Projet (*capture d'écran ci-dessous*) puis choisir la commande que vous souhaitez dans le menu **Projet | Contrôle de source**. Les commandes **Archiver** et **Extraire** sont disponibles en tant que commandes de menu contextuel des items de la fenêtre Projet.



Dans cette section, nous décrivons en détail les fonctions principales du contrôle de source :

- Ajouter à, Supprimer du contrôle de source
- Extraire, Archiver
- Obtenir les fichiers en lecture seule
- Copier et partager depuis le contrôle de source
- Modifier le contrôle de source

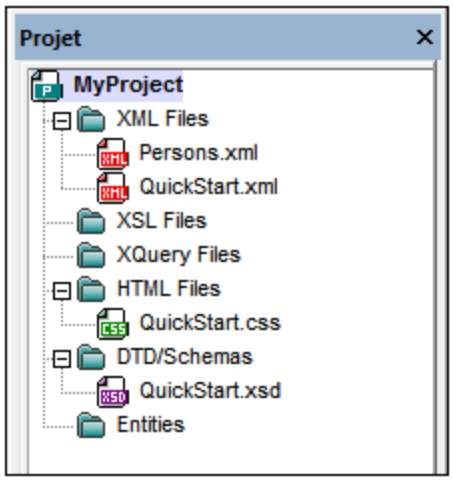
Les commandes supplémentaires dans le menu **Projet | Contrôle de source** sont décrites dans la section Référence menu du manuel. En ce qui concerne les informations spécifiques à un système de contrôle de source particulier, veuillez consulter la documentation d'utilisation de ce système.

24.6.1 Ajouter à, Supprimer du contrôle de source

Ajouter

Une fois qu'un projet a été ajouté au contrôle de source, vous pouvez placer les fichiers sous contrôle de source soit individuellement soit en groupe. On parle aussi d'ajouter les fichiers au contrôle de source. Choisir le fichier dans la fenêtre Projet et cliquer sur la commande **Projet | Contrôle de source | Ajouter au Contrôle de source**. Pour sélectionner des fichiers multiples, maintenir la touche **Ctrl** appuyée tout en cliquant sur les fichiers que vous souhaitez ajouter. Exécuter la commande sur un dossier de projet (vert) (*voir capture d'écran ci-dessous*) pour ajouter tous les fichiers dans le dossier et ses sous-dossiers au contrôle de source.

© 2016-2022 Altova GmbH



Lorsque les fichiers sont ajoutés au contrôle de source, la hiérarchie du dossier local est répliquée dans l'archivage (ce n'est pas la hiérarchie du dossier de projet qui est répliquée). Donc, si un fichier se trouve dans un sous-dossiers à x niveaux de profondeurs dans le dossier local, le dossier parent du fichier et tous les anciens dossiers ancêtres sont créés automatiquement dans l'archivage.

Lorsque le premier fichier d'un projet est ajouté au contrôle de source, les liaisons correctes sont créées dans l'archivage et le fichier de projet (fichier `.spp`) est ajouté automatiquement. Pour plus de détails, voir la section Ajouter au contrôle de source.

Symboles du contrôle de source

Les fichiers et le dossier de projet affichent certains symboles, dont la signification est indiquée ci-dessous.

	Archivé. Disponible pour une extraction.
	Extrait par un autre utilisateur. Non-disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

Supprimer

Pour supprimer un fichier du contrôle de source, choisir le fichier et cliquer sur la commande **Projet | Contrôle de source | Supprimer du contrôle de source**. Vous pouvez aussi supprimer : (i) les fichiers contenus dans un dossier de projet en exécutant la commande dans le dossier, et (ii) le projet complet en exécutant la commande dans le projet.

24.6.2 Extraire, Archiver

Une fois qu'un fichier de projet a été placé sous contrôle de source, il peut être extrait ou archivé en sélectionnant le fichier (dans la fenêtre Projet) et en cliquant sur la commande respective dans le menu **Projet | Contrôle de source** : **Extraire** et **Archiver**.

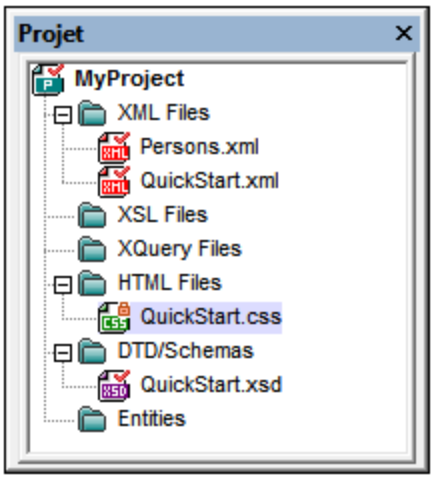
Lorsqu'un fichier est extrait, une copie de l'archivage est placée dans le dossier local. Un fichier qui est extrait peut être édité. Si un fichier qui se trouve sous contrôle de source n'a pas été extrait, il ne peut pas être édité.

Une fois que le fichier a été édité, les modifications peuvent être enregistrées dans l'archivage en archivant le fichier. Même si le fichier n'est pas enregistré dans l'application, son archivage permettra d'enregistrer les changements dans l'archivage. Un symbole de coche ou de cadenas dans l'icône de la fenêtre Projet indique si un fichier est extrait ou pas.

Les fichiers et le dossier de projet affichent certains symboles, dont la signification est indiquée ci-dessous.

	Archivé. Disponible pour une extraction.
	Extrait par un autre utilisateur. Non-disponible pour une extraction
	Extrait localement. Peut être édité et archivé.

Le fait de sélectionner le projet ou un dossier dans le projet permet de sélectionner tous les fichiers dans l'objet sélectionné. Afin de sélectionner plusieurs objets (fichiers et dossiers), Appuyer sur la touche **Ctrl** tout en cliquant sur les objets. La capture d'écran ci-dessous montre un projet qui a été extrait. Le fichier `QuickStart.css` a ensuite été archivé.



Enregistrer et rejeter les modifications d'édition

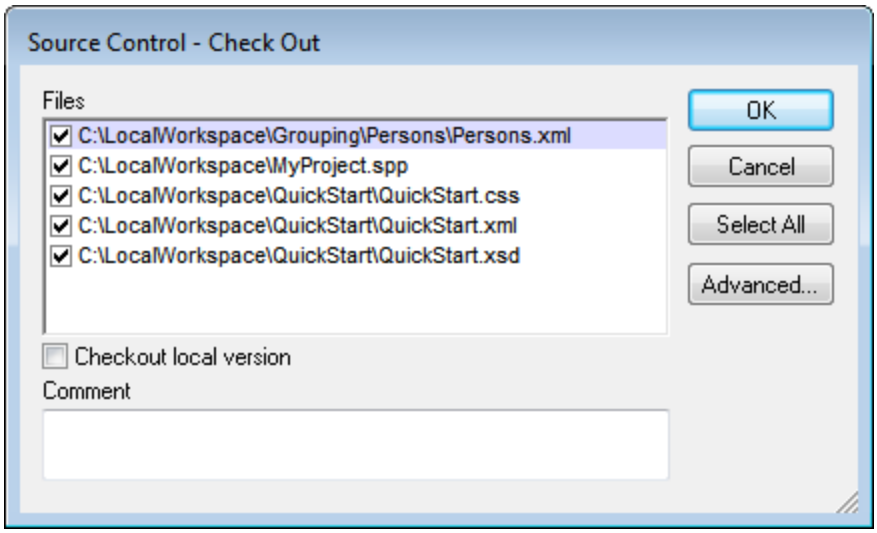
Veuillez noter que lorsque vous archivez un fichier, vous pouvez choisir de laisser le fichier extrait. Cela vous permet d'enregistrer les changements d'édition dans l'archivage tout en gardant le fichier extrait, ce qui peut être utile si vous souhaitez enregistrer régulièrement les modifications d'édition dans l'archivage puis continuer l'édition.

Si vous avez extrait un fichier et que vous y avez apporté des modifications, et que vous souhaitez ensuite rejeter ces changements, vous pouvez retourner à la version du document enregistrée dans l'archivage en choisissant la commande **Projet | Contrôle de source | Annuler extraction**.

Extraire

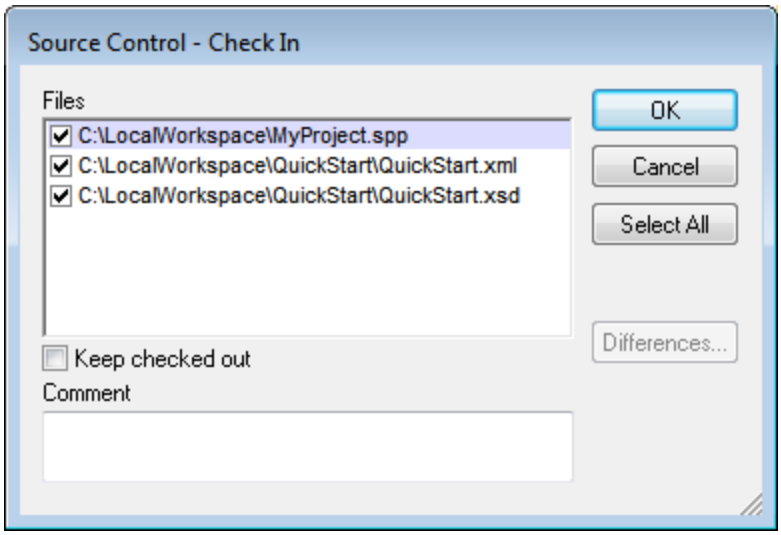
Le dialogue Extraire (*capture d'écran ci-dessous*) vous permet de : (i) sélectionner les fichiers à extraire, et (ii) sélectionner si vous souhaitez extraire la version d'archivage ou la version locale.

© 2016-2022 Altova GmbH



Archiver

Le dialogue Archiver (*capture d'écran ci-dessous*) vous permet de : (i) sélectionner les fichiers à archiver, et (ii) si vous le souhaitez, de garder le fichier extrait.



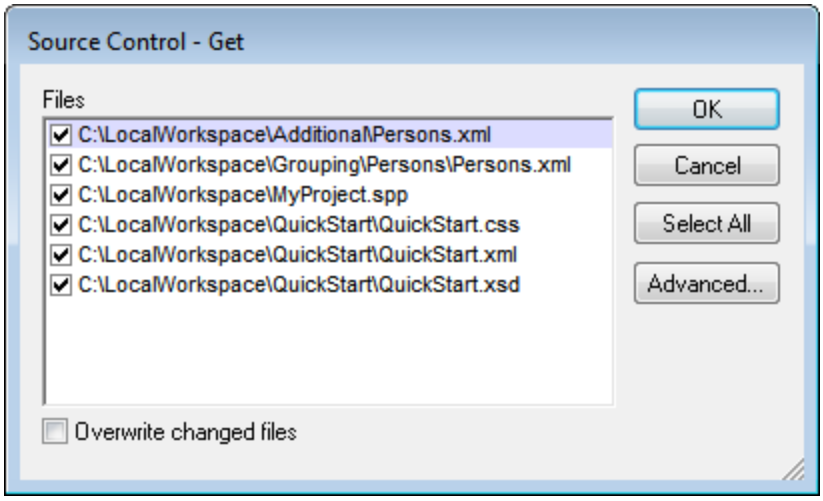
Note : Dans les deux dialogues (Extraire et Archiver), des fichiers multiples apparaissent si l'objet sélectionné (projet ou dossier/s de projet) contient plusieurs fichiers.

24.6.3 Obtenir les fichiers en lecture seule

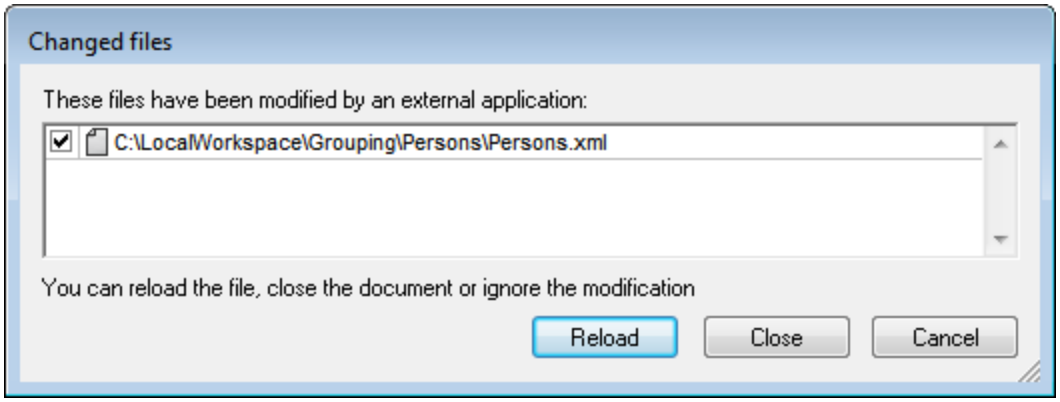
La commande **Obtenir** (dans le menu **Projet | Contrôle de source**) extrait les fichier de l'archivage en tant que fichiers en lecture seule. (Pour pouvoir éditer un fichier, vous devez l'extraire.) Le dialogue Obtenir rassemble les fichiers dans l'objet (projet ou dossier) dans lequel la commande **Obtenir** a été exécutée (*voir*

capture d'écran ci-dessous). Vous pouvez choisir les fichiers à sélectionner en les cochant dans la liste du dialogue Obtenir.

Note : La commande **Obtenir dossiers** vous permet de sélectionner des sous-dossiers individuels dans l'archivage si cela est autorisé par votre système de contrôle de source.



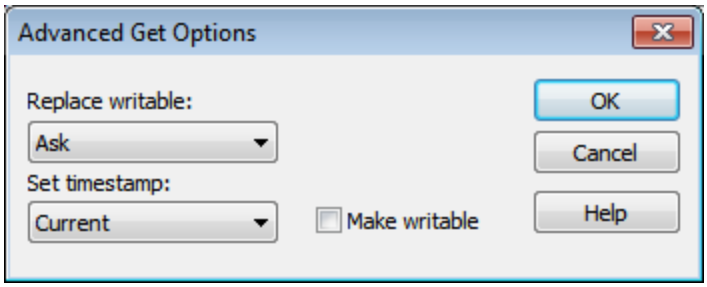
Vous pouvez choisir d'écraser les fichiers extraits modifiés en cochant cette option en bas du dialogue Obtenir. En cliquant sur **OK**, les fichiers seront écrasés. Si un des fichiers écrasé est actuellement ouvert, un dialogue s'affichera (*capture d'écran ci-dessous*) vous demandant si vous souhaitez recharger le fichier (bouton **Recharger**), fermer le fichier (**Fermer**), ou garder la vue actuelle du fichier (**Annuler**).



Options Obtenir avancées

Le dialogue Options | Obtenir avancées (*capture d'écran ci-dessous*) est accessible par le biais de la touche **Avancé** dans le dialogue Obtenir (*voir première capture d'écran dans cette section*).

© 2016-2022 Altova GmbH



Ici, vous pouvez définir les options pour (i) remplacer les fichiers accessibles en écriture qui sont extraits, (ii) l'horodatage, et (iii) si la propriété en lecture seule du fichier extrait doit être changé de manière à ce qu'il puisse être accédé en écriture.

Obtenir la dernière version

La commande **Obtenir la dernière version** (dans le menu **Projet | Contrôle de source**) extrait et place la dernière version du contrôle de source du/des fichier/s sélectionné/s dans le répertoire de travail. Les fichiers sont extraits en tant que fichiers en lecture seule et ne sont pas extraits. Cette commande fonctionne comme la commande **Obtenir** (*voir ci-dessus*), mais n'affiche pas le dialogue Obtenir.

Si les fichiers sélectionnés sont actuellement extraits, l'action entreprise dépendra de la manière dont votre système de contrôle de source gère ce genre de situation. Généralement, le système de contrôle de source vous demandera si vous souhaitez remplacer, fusionner ou laisser le fichier extrait tel qu'il est.

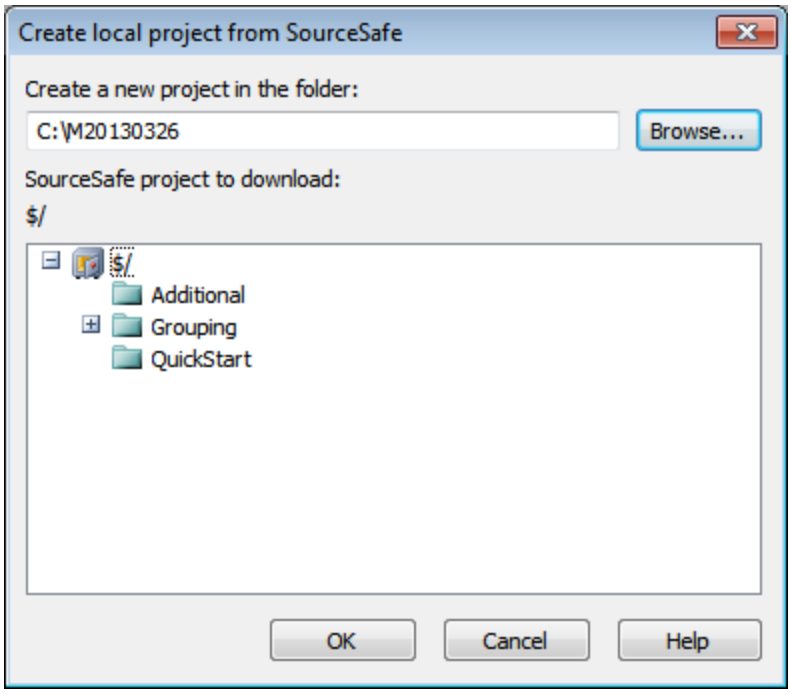
Note : Cette commande est récursive lorsqu'elle est exécutée dans un dossier, elle touche tous les fichiers se trouvant en dessous du fichier actuel dans la hiérarchie du dossier.

24.6.4 Copier et partager depuis le contrôle de source

La commande **Ouvrir depuis le contrôle de source** crée un nouveau projet d'application depuis un projet se trouvant sous contrôle de source.

Créer le nouveau projet comme suit :

1. Selon le système de contrôle de source utilisé, il peut s'avérer nécessaire, avant de créer un nouveau projet depuis le contrôle de source, de vous assurer qu'aucun autre fichier provenant du projet se trouvant sous contrôle de source n'est extrait.
2. Il n'est pas nécessaire d'ouvrir un projet dans l'application, mais vous pouvez le faire.
3. Choisir la commande **Projet | Contrôle de source | Ouvrir depuis le contrôle de source**.
4. Le système de contrôle de source actuellement configuré affichera ses dialogues de vérification et de connexion. Effectuez la connexion au dossier lié dans l'archivage que vous souhaitez copier.
5. Dans le dialogue qui s'affiche (*capture d'écran ci-dessous*), chercher le dossier local vers lequel les contenus du dossier lié dans l'archivage (auquel vous venez de vous connecter) doit être copié. Dans la capture d'écran ci-dessous, le dossier lié est appelé `MyProject` et est représenté par le signe `$` ; le dossier local est `C:\M20130326`.



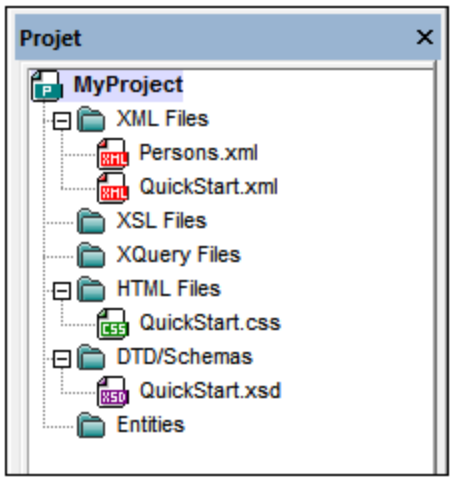
6. Cliquer sur **OK**. Les contenus du dossier lié (`MyProject`) seront copiés dans le dossier local `C:\M20130326`., et un dialogue s'ouvrira vous demandant de choisir le fichier de projet (fichier `.spp`) qui doit être créé en tant que le nouveau projet.
7. Choisir le fichier `.spp` qui aura été copié vers le dossier local. Dans notre exemple, il s'agira de `MyProject.spp` situé dans le dossier `C:\M20130326`. Un nouveau projet nommé `MyProject` sera créé dans l'application et sera affiché dans la fenêtre Projet. Les fichiers du projet se trouvent dans le dossier `C:\M20130326`.

Partager depuis le contrôle de source

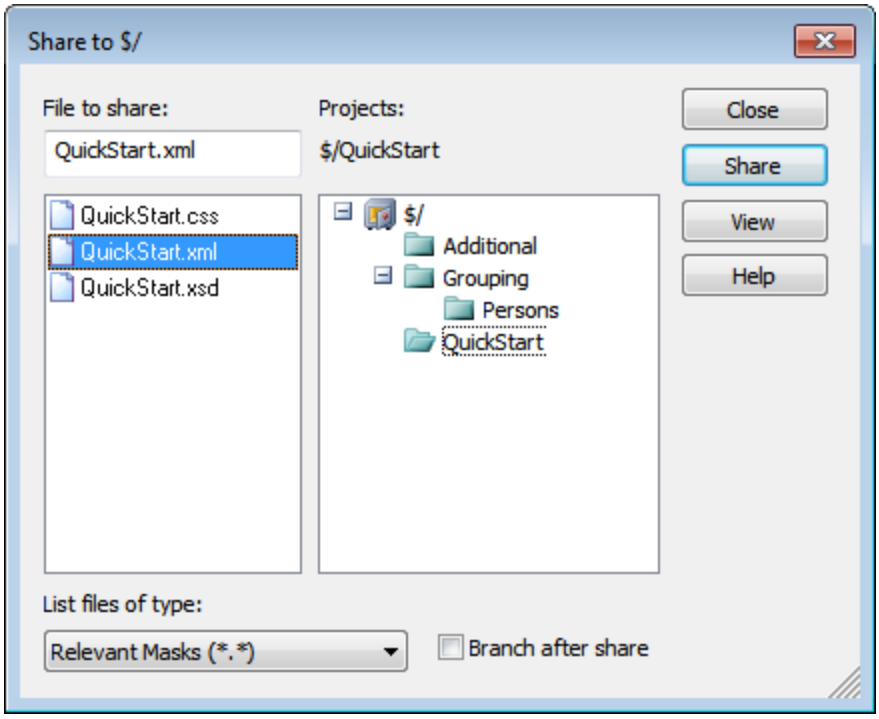
La commande **Partager depuis le Contrôle de source** est prise en charge lorsque le système de contrôle de source utilisé prend en charge des partages. Vous pouvez partager un fichier de manière à ce qu'il soit disponible dans plusieurs emplacements locaux. Un changement effectué dans un de ces fichiers locaux se répercutera dans toutes les autres versions "partagées".

Dans la fenêtre Projet de l'application, veuillez tout d'abord sélectionner le projet (*marqué dans la capture d'écran ci-dessous*). Puis cliquer sur **Partager depuis le Contrôle de source**.

© 2016-2022 Altova GmbH



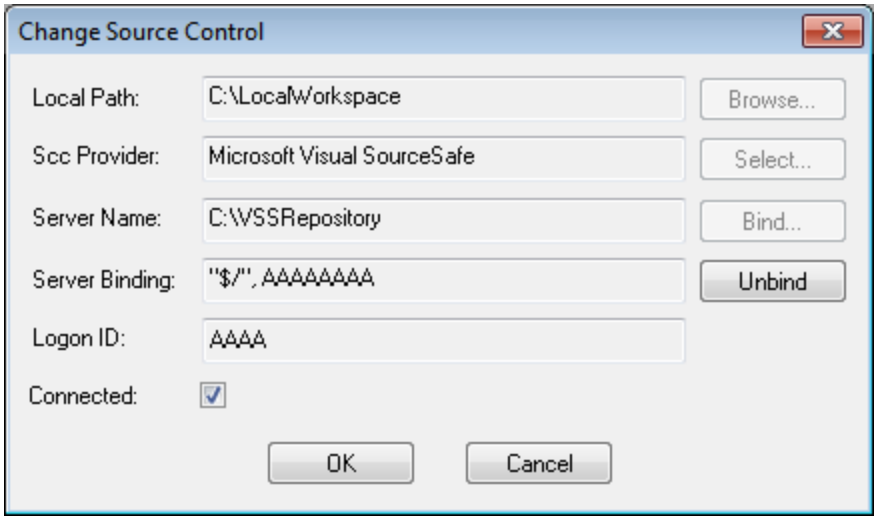
Le dialogue Partager avec [Dossier] (*capture d'écran ci-dessous*) s'ouvre.



Afin de sélectionner les fichiers à partager, choisissez tout d'abord, dans le volet de droite de l'arborescence de projet (*voir capture d'écran ci-dessus*), le dossier dans lequel les fichiers se trouvent. Les fichiers contenus dans le dossier choisi sont affichés dans le volet de gauche. Sélectionner le fichier que vous souhaitez partager (plusieurs fichiers en appuyant sur la touche **Ctrl** et en cliquant sur les fichiers que vous souhaitez partager). Les fichiers sélectionnés seront affichés dans le champ de saisie *Fichiers à partager* (en haut à gauche). Les fichiers disparaissent du volet de gauche. Cliquer sur **Partager** et puis **Fermer** pour copier les fichiers sélectionnés dans le dossier de partage local. Lorsque vous cliquez sur **Fermer**, les fichiers à partager seront copiés dans l'emplacement local sélectionné.

Le dossier de partage est noté dans le nom du dialogue Partager avec [Dossier]. Dans la capture d'écran ci-dessus, il s'agit du dossier local (puisque le signe `$` est le dossier dans l'archivage auquel le dossier local est

lié). Vous pouvez consulter et configurer le dossier de partage dans le dialogue Modifier le Contrôle de source (*capture d'écran ci-dessous*, **Modifier le Contrôle de source**) en changeant le chemin local et la liaison de serveur.



Pour plus de détails concernant le partage avec votre système de contrôle de source, voir la documentation de l'utilisateur du système de contrôle de source.

24.6.5 Modifier le contrôle de source

Les paramètres du contrôle de source peuvent être modifiés avec deux commandes dans le menu **Projet | Contrôle de source** :

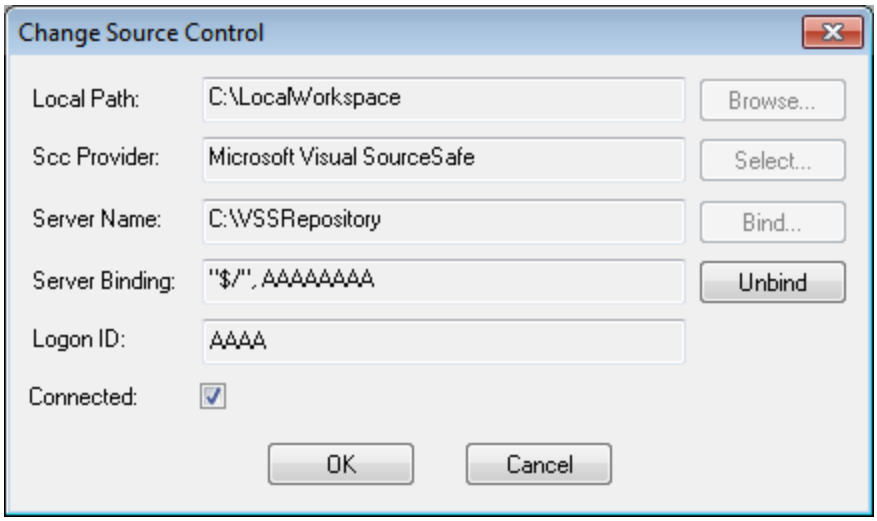
- **Gestionnaire de Contrôle de source**, qui ouvre l'application du système de contrôle de source et vous permet de configurer les bases de données et les liaisons.
- **Modifier Contrôle de source**, qui permet d'ouvrir le dialogue Modifier le contrôle de source, dans lequel vous pouvez modifier le système de contrôle de source utilisé par l'application Altova et la liaison actuelle. Ce dialogue est décrit ci-dessous.

La liaison actuelle est ce que le projet d'application actif utilisera pour se connecter à la base de données du contrôle de source. La liaison actuelle est correcte lorsque le fichier de projet d'application (fichier `.spp`) se trouve dans le dossier local et que le dossier lié dans l'archivage se trouve à l'endroit où les fichiers de ce projet sont stockés. Généralement, le dossier lié et sa sous--structure correspondra au dossier du poste de travail local et à sa sous-structure.

Dans le dialogue Modifier le Contrôle de source (*capture d'écran ci-dessous*), vous pouvez modifier le système de contrôle de source (*Fournisseur SCC*), le dossier local (*Chemin local*), et la liaison d'archivage (*Nom de serveur* et *Liaison de serveur*).

Ce n'est qu'après avoir défait la liaison actuelle que les paramètres peuvent être changés. Défaire la liaison actuelle avec la touche **Annuler**. Tous les paramètres sont maintenant éditables.

© 2016-2022 Altova GmbH



Modifier les paramètres de contrôle de source comme suit :

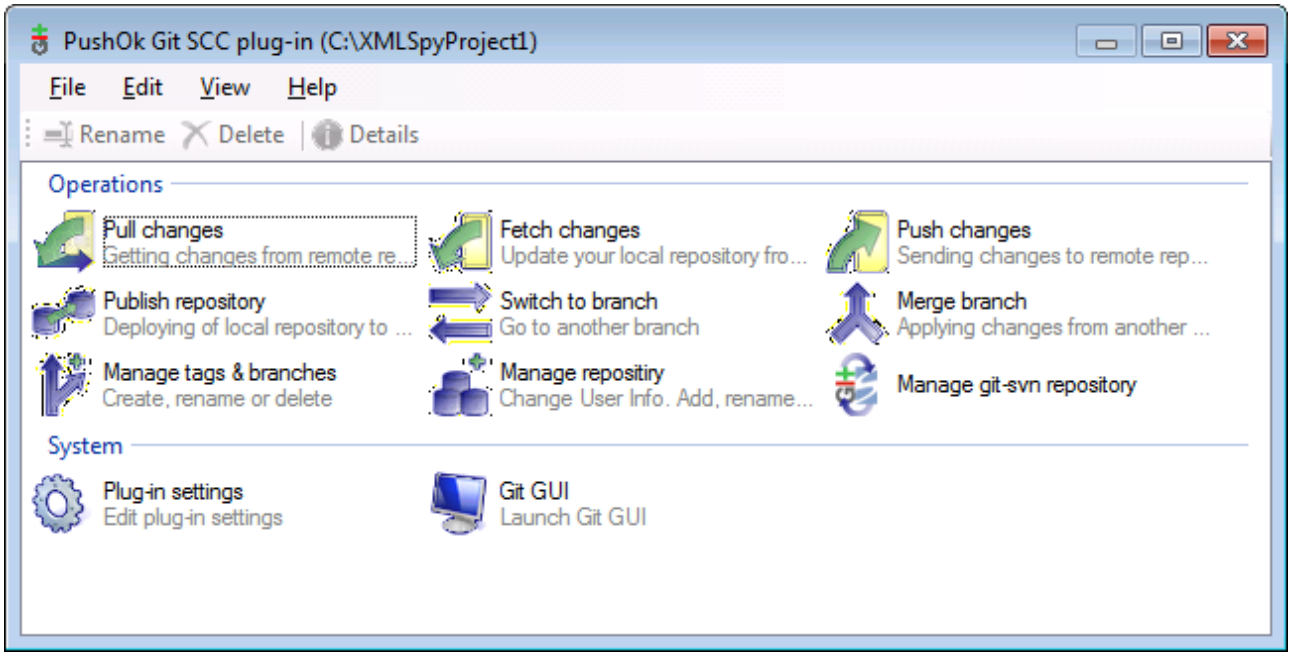
1. Utiliser la touche **Chercher** pour parcourir le dossier local et la touche **Sélectionner** pour choisir parmi les systèmes de contrôle de source installés.
2. Ensuite, vous pourrez lier le dossier local à une base de donnée d'archivage. Cliquer sur la touche **Lier**. Le dialogue de connexion de votre système de contrôle de source s'affiche.
3. Si vous avez saisi une *ID de connexion*, cela sera transféré au système de contrôle de source ; sinon, vous devrez saisir vos détails d'enregistrement dans le dialogue de connexion.
4. Choisir la base de données dans l'archivage que vous souhaitez lier à ce dossier local. Ce paramètre peut être réparti sur plusieurs dialogues.
5. Une fois que le paramètre a été créé, cliquer sur **OK** dans le dialogue Modification du Contrôle de source.

24.7 Contrôle de source avec Git

La prise en charge de Git en tant que système de contrôle de source dans XMLSpy est disponible par le biais d'un plugin tiers appelé **plugin GIT SCC** (http://www.pushok.com/software/git.html).

> Au moment de la rédaction de cette documentation, le **plugin GIT SCC** est disponible pour une utilisation expérimentale. L'enregistrement avec le éditeur du plugin est exigée pour pouvoir utiliser le plugin.

Le plugin GIT SCC vous permet de travailler avec un archivage Git avec les commandes disponibles dans le menu **Projet | Contrôle de source** de XMLSpy. Veuillez noter que les commandes dans le menu **Projet | Contrôle de source** de XMLSpy sont fournies par Microsoft Source Control Plug-in API (MSSCCI API), qui aborde une philosophie de design différente de Git. C'est ainsi que le plugin fait office d'intermédiaire entre des fonctions de type "Visual Source Safe" et des fonctions Git. D'un côté, cela signifie qu'une commande comme **Obtenir la dernière version** peut ne pas être applicable avec Git. D'un autre côté, il existe des nouvelles actions spécifiques à Git, qui sont disponibles dans le dialogue "Gestionnaire de Contrôle de source" fourni par le plugin (sous le menu **Projet | Contrôle de source | Gestionnaire du Contrôle de source** de XMLSpy).



Le dialogue Gestionnaire de Contrôle de source

Les autres commandes que vous utiliserez fréquemment sont disponibles directement dans le menu **Projet | Contrôle de source**.

Les sections suivantes décrivent la configuration initiale du plugin, et le flux de travail de base :

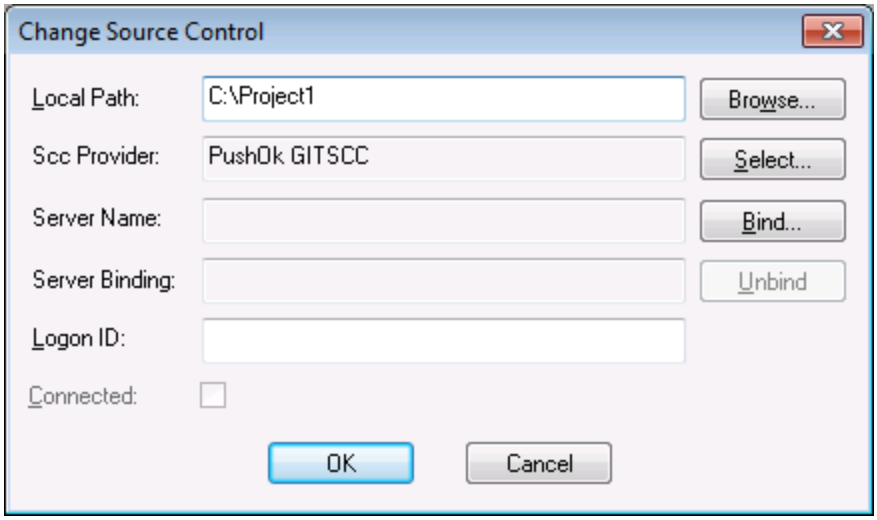
- Activer le Contrôle de source avec le plugin Git SCC
- Ajouter un projet au Contrôle de source avec Git
- Cloner un projet depuis le Contrôle de source avec Git

© 2016-2022 Altova GmbH

24.7.1 Activer le Contrôle de source avec le plugin Git SCC

Pour activer le contrôle de source Git avec XMLSpy, le **plug-in PushOK GIT SCC** tiers doit être installé, enregistré et sélectionné en tant que fournisseur de contrôle de source, comme suit :

1. Télécharger le fichier d'installation de plugin depuis le site web de l'éditeur (http://www.pushok.com), l'exécuter et suivre les instructions d'installation.
2. Dans le menu **Projet** de XMLSpy, cliquer sur **Changer contrôle de source**, et vérifier que **PushOk GITSCC** a été sélectionné en tant que fournisseur de contrôle de source. Si vous ne voyez pas **Push Ok GITSCC** dans la liste des fournisseurs, il est probable que l'installation du plugin n'a pas réussi. Dans ce cas, consultez la documentation de l'éditeur pour une trouver une solution.



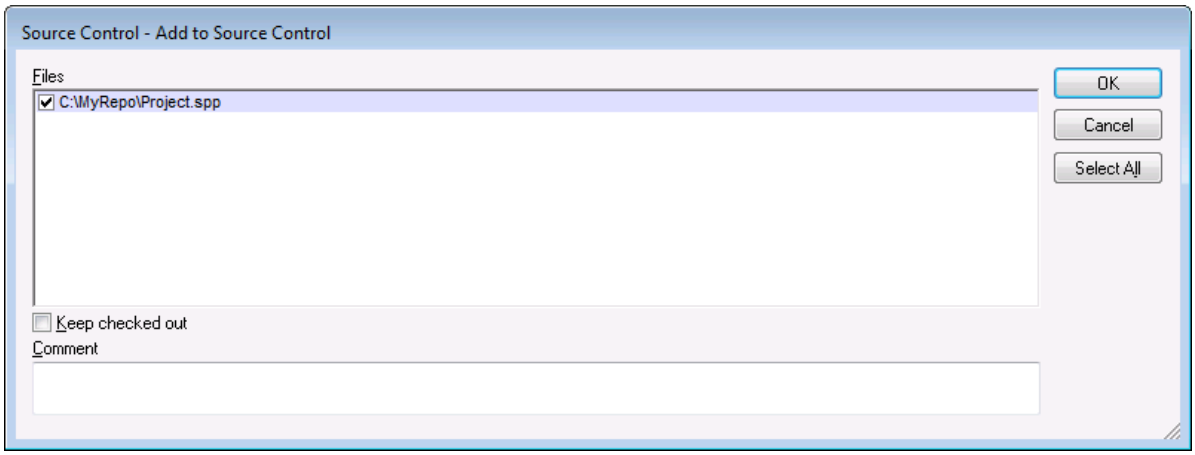
3. Lorsqu'un dialogue vous invite à enregistrer le plugin, cliquer sur **Enregistrement** et suivre les étapes de l'assistant pour terminer le processus d'enregistrement.

24.7.2 Ajouter un projet au Contrôle de source avec Git

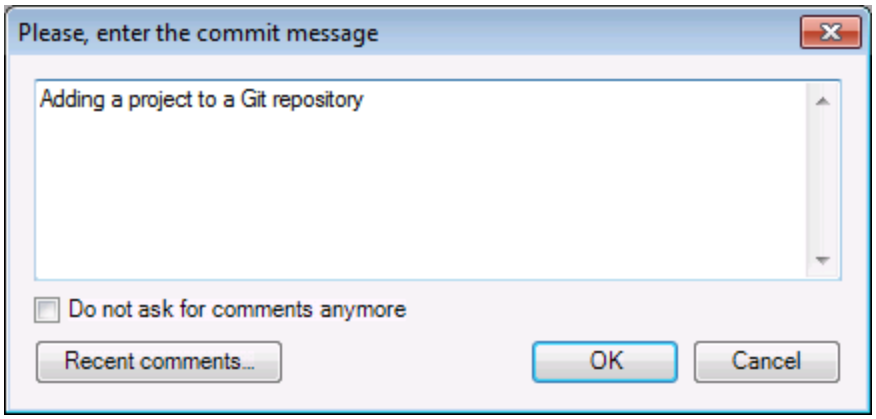
Vous pouvez enregistrer des projets XMLSpy en tant qu'archivages Git. La structure des fichiers ou des dossiers que vous ajoutez au projet correspondraient à la structure de l'archivage Git.

Pour ajouter un projet au contrôle de source Git :

1. Veuillez vous assurer que le **Plug-in PushOK GIT SCC** est configuré en tant que le fournisseur de contrôle de source (voir Activer le contrôle de source Git avec le Plugin GIT SCC).
2. Créer un nouveau projet en utilisant la commande de menu **Projet | Créer Projet**.
3. Enregistrer le projet dans un dossier local, par exemple `C:\MyRepo\Project.spp`
4. Dans le menu **Projet**, sous **Contrôle de source**, cliquer **Ajouter au Contrôle de source**.

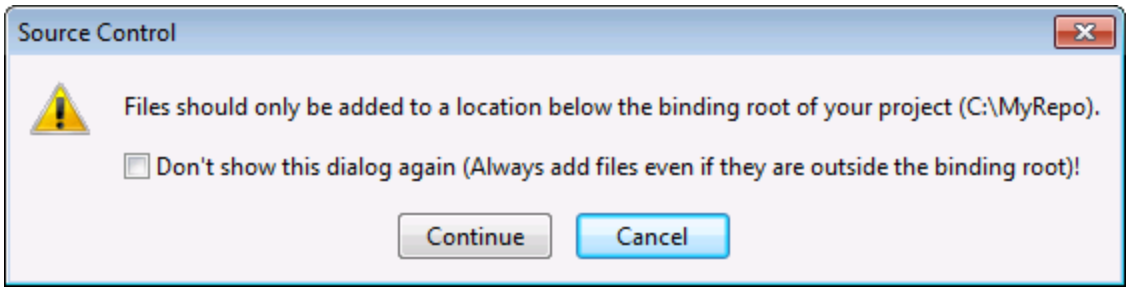


5. Cliquer sur **OK**.



6. Saisir le texte de votre message de validation et cliquer sur **OK**.

Vous pouvez maintenant commencer à ajouter des fichiers et des dossiers à votre projet. Veuillez noter que tous les fichiers de projet et les dossiers doivent se trouver sous un dossier racine du projet. Par exemple, si le projet a été créé dans le dossier `C:\MyRepo`, alors seuls les fichiers se trouvant sous `C:\MyRepo` devraient être ajoutés au projet. Sinon, si vous essayez d'ajouter à votre projet des fichiers qui se trouvent en dehors du dossier racine de projet, un message d'avertissement s'affichera :

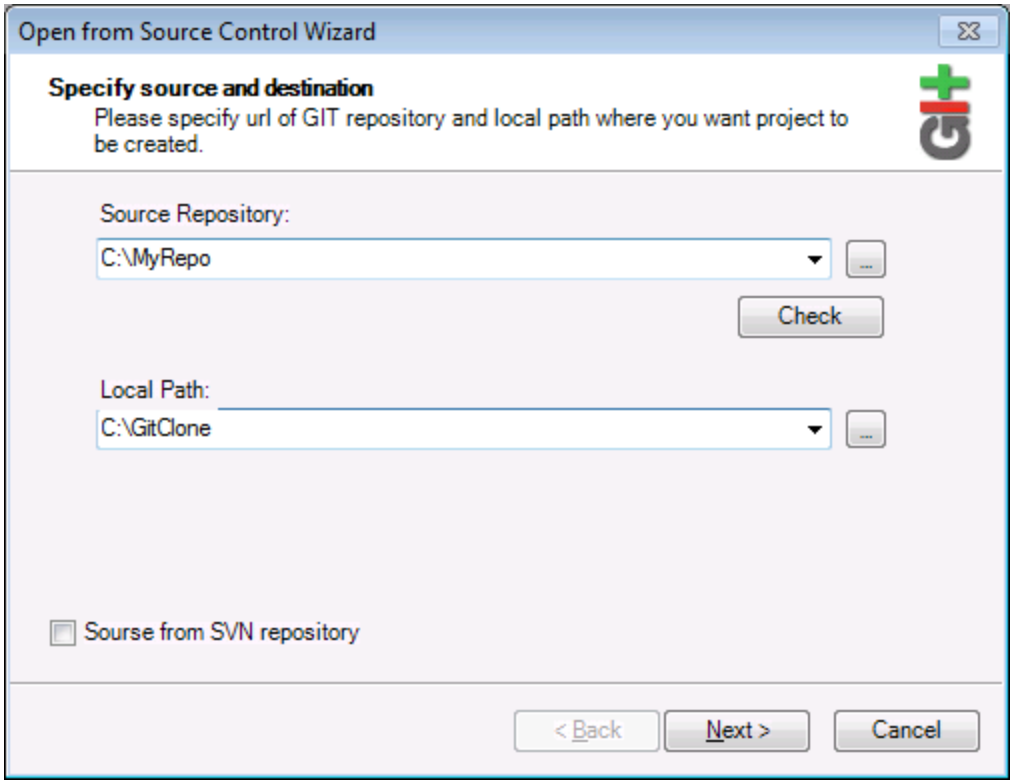


 © 2016-2022 Altova GmbH

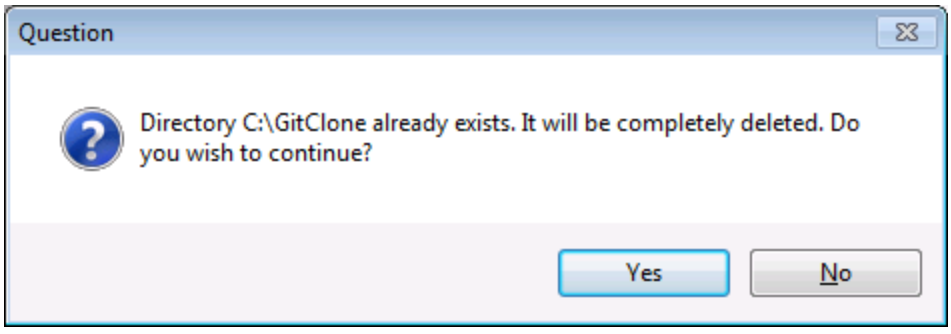
24.7.3 Cloner un projet depuis le Contrôle de source avec Git

Les projets qui ont précédemment été ajoutés au Contrôle de source Git (voir Ajouter un projet au contrôle de source Git) peuvent être ouverts depuis l'archivage Git comme suit :

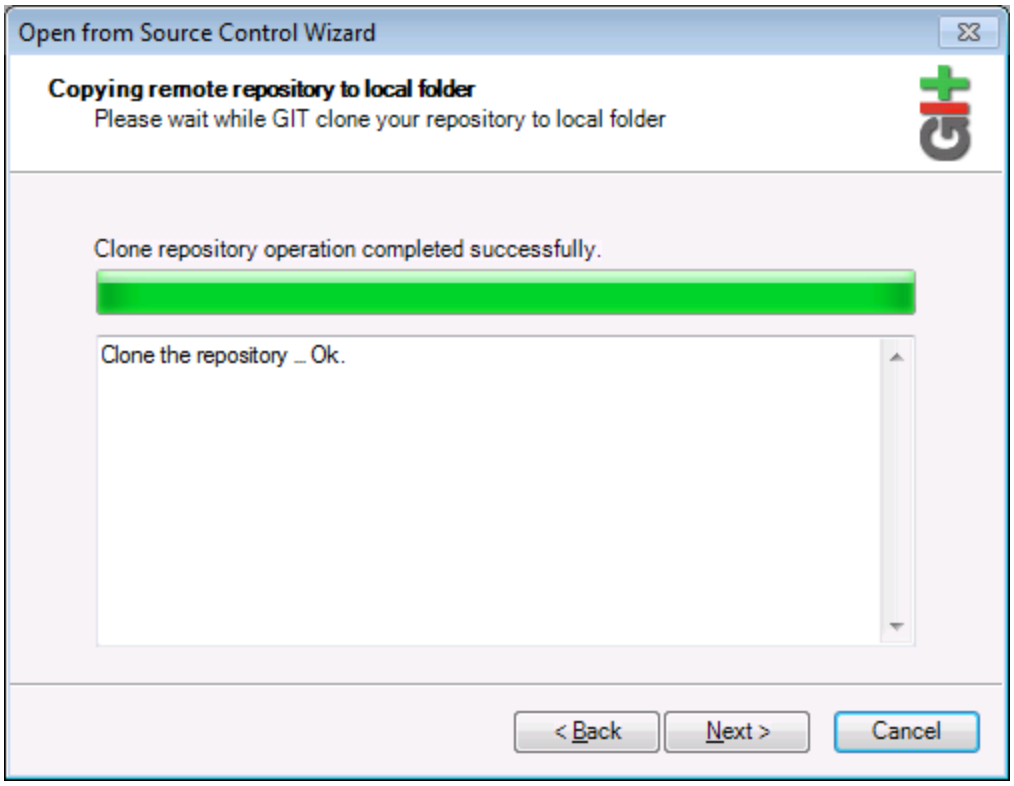
1. Veuillez vous assurer que le **Plugin PushOK GIT SCC** est configuré en tant que le fournisseur de contrôle de source (voir Activer le contrôle de source Git avec le Plugin GIT SCC).
2. Dans le menu **Projet**, cliquer sur **Contrôle de source | Ouvrir depuis le Contrôle de source**.
3. Saisir le chemin ou l'URL de l'archivage de la source. Cliquer sur **Contrôler** pour vérifier la validité du chemin ou de l'URL.



4. Sous **Chemin local**, saisir le chemin vers le dossier local où vous souhaitez que le projet soit créé et cliquer sur **Suivant**. Si le dossier local existe (même s'il est vide), le dialogue suivant s'ouvre :



5. Cliquer **Oui** pour confirmer, puis cliquer sur **Suivant**.



6. Suivez les étapes restantes nécessaires pour votre cas de figure particulier.
7. Une fois l'assistant terminé, un dialogue Chercher apparaît, vous demandant d'ouvrir le fichier de projet XMLSpy (*.spp). Choisir le fichier de projet pour charger les contenus du projet dans XMLSpy.

© 2016-2022 Altova GmbH

25 XMLSpy dans Visual Studio

XMLSpy peut être intégré dans les versions 2012/2013/2015/2017/2019/2022 de Microsoft Visual Studio IDE. Cela permet d'obtenir le meilleur des deux mondes et d'intégrer des fonctions d'édition XML avancées avec l'environnement de développement avancé de Visual Studio.

Dans cette section, nous décrivons :

- Le processus d'installation général et l'intégration du plugin XMLSpy dans Visual Studio.
- Les Différences entre la version Visual Studio et la version autonome.
- Les débogueurs de XMLSpy dans Visual Studio.

25.1 Installer le plugin XMLSpy

Pour installer le Plug-in XMLSpy pour Visual Studio, suivez les étapes suivantes :

1. Installer Microsoft Visual Studio 2012/2013/2015/2017/2019/2022. Veuillez noter qu'à partir de Visual Studio 2022, Visual Studio est mis à disposition uniquement comme application 64-bit.
2. Installez XMLSpy(Enterprise ou Professional Edition). Si vous avez installé Visual Studio 2022+, alors vous devez installer la version 64-bit de XMLSpy.
3. Téléchargez et exécutez le pack d'intégration XMLSpy pour Microsoft Visual Studio. Ce pack est disponible sur la page de téléchargement de XMLSpy (Enterprise et Professional Editions) sur www.altova.com

Une fois que le pack d'intégration a été installé, vous pourrez utiliser XMLSpy dans l'environnement de Visual Studio.

Important

Vous devez utiliser le pack d'intégration correspondant à votre version de XMLSpy (la version actuelle est 2023). La package d'intégration n'est pas spécifique à l'édition et peut donc être utilisé pour les deux éditions Enterprise et Professional.

© 2016-2022 Altova GmbH

25.2 Différences avec XMLSpy Standalone

Cette section recense les manières dont les versions de Visual Studio diffèrent des versions standalone de XMLSpy. La liste commence avec des fonctions qui ne sont pas prises en charge dans la version de Visual Studio et poursuit avec une liste des autres différences entre la version Visual Studio et la version standalone.

- Fonctions non prises en charge dans Visual Studio
- Menus XMLSpy supplémentaires dans Visual Studio
- Assistants à la saisie dans Visual Studio
- Même fonction, commande différente
- Commandes XMLSpy en tant que commandes Visual Studio

Fonctions non prises en charge dans Visual Studio

Les fonctions XMLSpy suivantes ne sont pas disponibles dans Visual Studio :

- L'environnement Scripting (**Outils | Options XMLSpy | Scripting**) n'est pas pris en charge actuellement. Les icônes de barre d'outils qui ont été créées pour exécuter les scripts ne seront donc pas affichées.
- Les icônes de statut du texte de Authentic View ne sont pas pris en charge.
- La fenêtre de navigateur séparée (une option dans la section **Tools | Options | Mode**) n'est pas prise en charge. Cela signifie que les Modes Texte et Navigateur sont toujours dans la même fenêtre.
- Toutes les fonctions de Contrôle de Source.
- Toutes les fonctions de comparaison (disponibles dans le menu **Outils** de la version standalone).

Menus XMLSpy supplémentaires dans Visual Studio

Les commandes suivantes sont spécifiques à XMLSpy dans Visual Studio :

- **Mode | Fenêtre Outils XMLSpy**
- **Mode | Mode XMLSpy**
- **XMLSpy** (inclut les items de menu des Ressources gloables et la possibilité de basculer les thèmes XMLSpy)
- **Outils | Options XMLSPY**

Note : dans Visual Studio 2019 et plus élevé, la fonctionnalité de XMLSpy peut être accédée dans le menu **Extensions** de Visual Studio. Dans des versions antérieures de Visual Studio, les fonctions de XMLSpy sont disponibles dans des menus de niveau supérieur de Visual Studio.

Assistants à la saisie (fenêtres Outils dans Visual Studio)

Les assistants à la saisie de XMLSpy sont disponibles en tant que fenêtres Outils dans Visual Studio. Veuillez noter les points suivants. (Pour une description des assistants à la saisie et de la GUI XMLSpy, voir la section, GUI et environnement.)

- Vous pouvez glisser des fenêtres d'assistants à la saisie à n'importe quel endroit de l'environnement de développement.
- Cliquer avec la touche de droite sur un onglet d'assistant à la saisie pour personnaliser encore plus votre interface. Les options de configuration de l'assistant à la saisie sont les suivantes : ancrable, dissimuler, flottant, et auto-dissimuler.

Même fonction, commande différente

Certaines fonctions de XMLSpy sont disponibles dans Visual Studio sous des commandes nommées différemment. Il s'agit de :

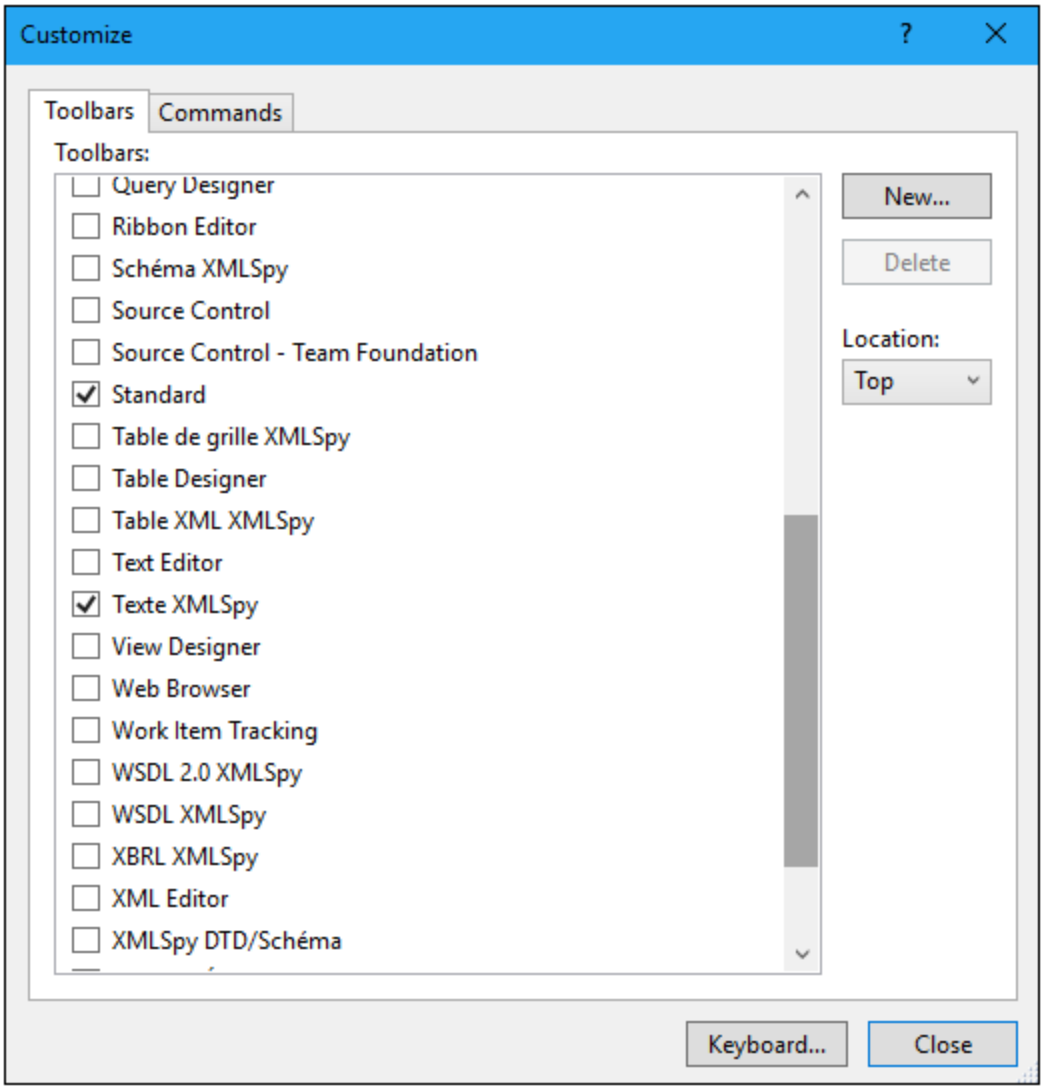
XMLSpy	Visual Studio	Fonction
Fichier | Ouvrir | Passer à l'URL	Fichier | Ouvrir | Site web	Ouvre le fichier depuis l'URL
Basculer vers URL | Enregistrer	Fichier | Enregistrer le fichier XMLSpy dans l'URL	Enregistre le fichier dans l'URL

commandes XMLSpy en tant que commandes Visual Studio

Certaines commandes XMLSpy sont présentes en tant que commandes Visual Studio dans la GUI Visual Studio. Il s'agit de :

- ***Annuler, Rétablir :*** Ces commandes Visual Studio touchent toutes les actions dans l'environnement de développement Visual Studio.
- ***Projets :*** Les projets XMLSpy sont gérés comme projets de Visual Studio.
- ***Personnaliser les barres d'outils, Personnaliser les commandes :*** Les onglets Barres d'outils et Commandes (*voir la capture d'écran ci-dessous*) contenues dans le dialogue Personnaliser (**Outils | Personnaliser**) contiennent les commandes Visual Studio et les commandes XMLSpy.

© 2016-2022 Altova GmbH



- ***Modes :*** Dans le menu **Mode**, les deux commandes, **XMLSpy Tool Windows** et **XMLSpy View** contiennent des options pour activer les fenêtres de l'assistant de saisie et les autres barres latérales, basculent les les modes d'édition, et activent et désactivent certains guides d'édition.
- ***Aide XMLSpy*** : Ce menu XMLSpy apparaît en tant que sous-menu dans le menu **Aide** de Visual Studio.

25.3 Débogueurs de XMLSpy dans Visual Studio

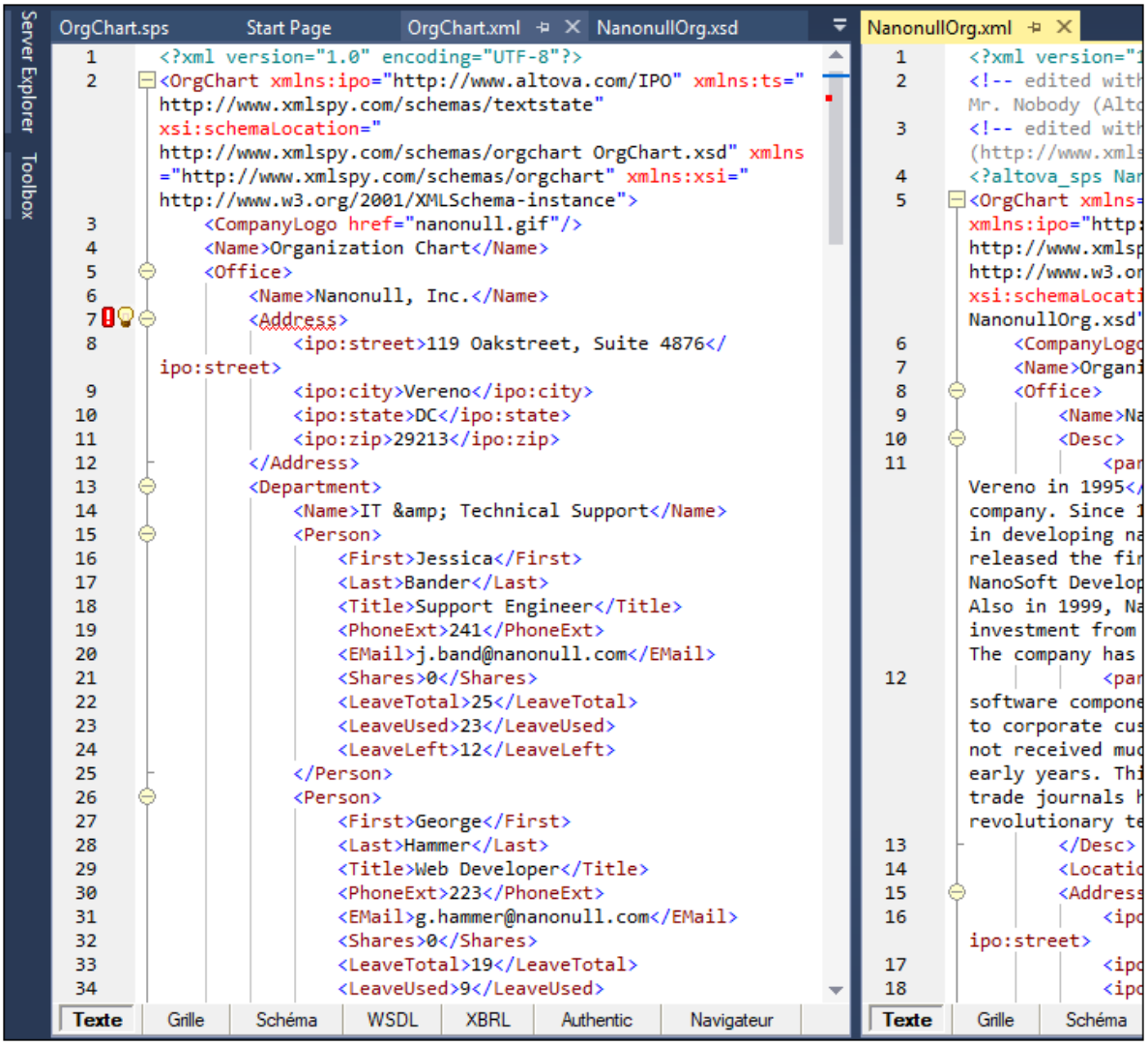
XMLSpy contient un débogueur XSLT/XQuery (*éditions Enterprise et Professional*) et un débogueur SOAP (*édition Enterprise*). Un processus de débogueur implique l'affichage d'un ou de plusieurs fichiers (par exemple des fichiers de sortie XML, XSLT et XSLT), qui sont tous affichés dans Visual Studio en tant qu'un seul groupe à onglets. Pour que le débogage soit plus simple à suivre, vous pouvez créer un ou plusieurs groupes à onglets dans Visual Studio. Procéder comme suit :

1. Cliquer sur l'onglet que vous souhaitez séparer du groupe unique à onglet, puis le glisser et le déposer quelque part dans l'onglet actif actuel. Un menu s'ouvre qui permet de définir le type d'onglet que vous souhaitez créer.



2. Choisir **New Vertical Tab Group**. Cela crée un nouvel onglet contenant uniquement l'onglet sélectionné (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH


Server Explorer
Toolbox
OrgChart.sps
Start Page
OrgChart.xml
NanonullOrg.xsd
NanonullOrg.xml
Texte
Grille
Schéma
WSDL
XBRL
Authentic
Navigateur

26 XMLSpy dans Eclipse

Eclipse est un framework open source qui intègre différents types d'applications fournies sous la forme de plugins. Le Package d'intégration XMLSpy pour Eclipse vous permet d'intégrer et accéder la fonctionnalité de XMLSpy dans la Plateforme Eclipse pour Windows. Les versions prises en charge d'Eclipse sont : 2022-09, 2022-06, 2022-03, 2021-12.

Dans cette section, nous allons décrire les points suivants :

- Comment installer le Package d'installation XMLSpy pour Eclipse et intégrer XMLSpy dans Eclipse
- La perspective XMLSpy dans Eclipse
- Autres Points d'entrée XMLSpy dans Eclipse
- Perspectives de débogage XMLSpy

Note: Les fonctions de Contrôle de code source, qui sont disponibles dans la version standalone de XMLSpy, ne sont pas prises en charge dans la version Eclipse.

© 2016-2022 Altova GmbH

26.1 Install the Integration Package for Eclipse (2)

Prérequis

- Eclipse 2022-09, 2022-06, 2022-03, 2021-12 (http://www.eclipse.org) exploitation 64-bit.
- Java Runtime Environment (JRE) ou Java Development Kit (JDK) pour la plateforme 64-bit.
- XMLSpy Enterprise ou Professional Edition 64-bit.

Note : tous les prérequis cités ci-dessus doivent être dotés de la plateforme 64-bit. L'intégration avec d'autres plateformes 32-bit n'est plus prise en charge, bien qu'elle puisse encore fonctionner.

Une fois que les pérequis ci-dessus sont en place, vous pouvez installer le Package d'intégration (64-bit) XMLSpy pour intégrer XMLSpy dans Eclipse. L'intégration peut être effectuée au cours de l'installation du Package d'intégration ou manuellement depuis Eclipse une fois que le Package d'intégration a été installé. Le Package d'intégration XMLSpy est disponible pour être téléchargé à l'adresse https://www.altova.com/components/download.

Note : Eclipse doit être fermé pendant que vous installez ou désinstallez le Package d'intégrationXMLSpy.

Intégrez XMLSpy lors de l'installation du Package d'intégration

Vous pouvez intégrez XMLSpy dans Eclipse lors de l'installation du Package d'intégration XMLSpy. Pour ce faire, suivez les étapes suivantes :

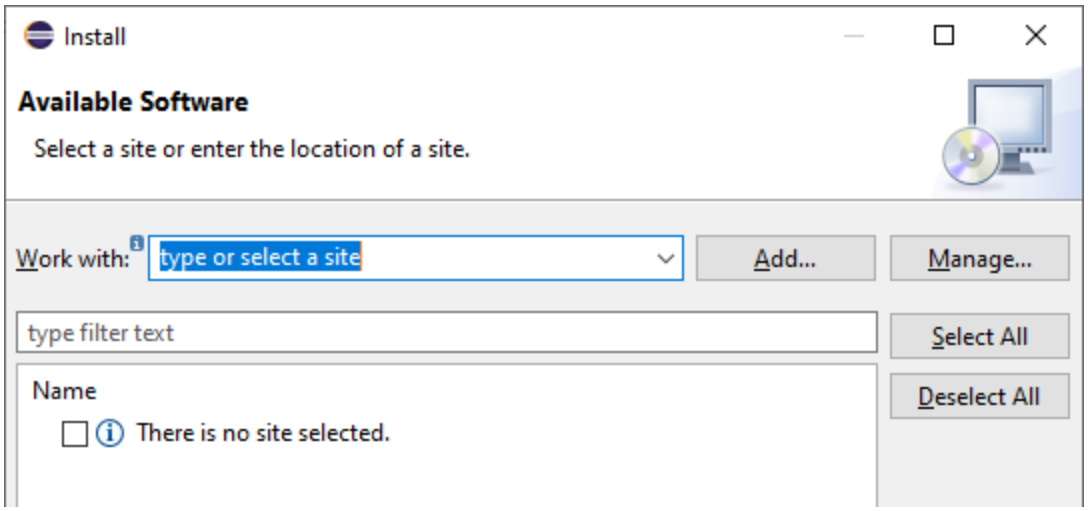
1. Exécutez le Package d'intégration XMLSpy pour lancer l'assistant d'installation.
2. Suivez les étapes initiales de l'installation avec l'assistant d'installation.
3. Dans l'étape d'intégration, sélectionnez *Permettre à l'assistant d'intégrer Altova XMLSpy plug-in dans Eclipse*, et cherchez le répertoire dans lequel le fichier exécutable pour Eclipse (`eclipse.exe`) se trouve.
4. Cliquez sur **Suivant** et terminez l'installation.

La perspective XMLSpy et les menus seront disponibles dans Eclipse la prochaine fois que vous le démarrez.

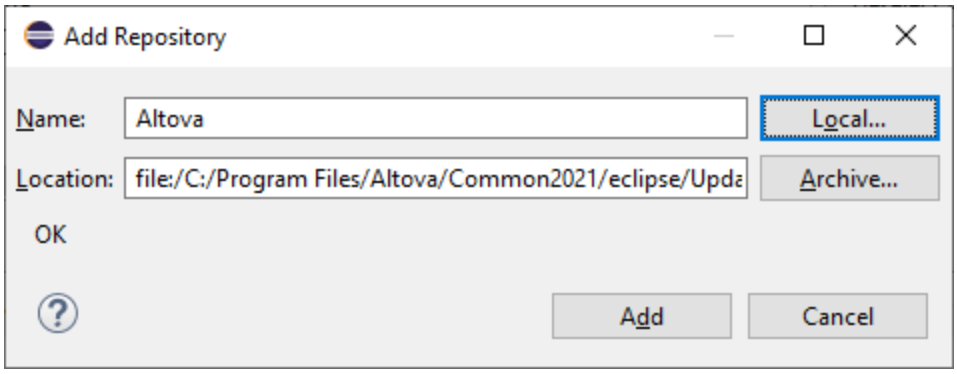
Intégrer XMLSpy dans Eclipse manuellement

Une fois que vous avez installé le Package d'intégration XMLSpy, vous pouvez intégrer manuellement XMLSpy dans Eclipse comme suit:

1. Dans Eclipse, sélectionnez la commande de menu **Aide | Installer nouveau logiciel**.
2. Dans la boîte de dialogue Installer, cliquez sur **Ajouter**.



3. Dans la boîte de dialogue Ajouter Référentiel, cliquez sur **Local**. Recherchez le dossier `C:\Program Files\Altova\Common2023\eclipse\UpdateSite` et sélectionnez-le. Donner un nom au site (comme « Altova »).



4. Répétez les étapes 2 à 3 ci-dessus, en sélectionnant cette fois-ci le dossier `C:\Program Files\Altova\<% APPNAMESHORT%>\eclipse\UpdateSite` et en donnant un nom comme « Altova » XMLSpy".
5. Dans la boîte de dialogue Installer, sélectionnez *Uniquement sites locaux*. Puis, sélectionnez le dossier « catégorie Altova » et cliquez sur **Suivant**.
6. Revérifiez les éléments à installer puis cliquez sur **Suivant** pour procéder.
7. Pour accepter le contrat de licence, sélectionnez la case à cocher respective.
8. Cliquez sur **Terminer** et terminez l'installation.

Note : Si vous avez des problèmes avec le plug-in (icônes manquantes, par exemple), lancez Eclipse depuis la ligne de commande avec l'indicateur `-clean`.

© 2016-2022 Altova GmbH

26.2 XMLSpy Perspective in Eclipse

Dans Eclipse, une perspective est un affichage GUI qui est configuré avec la fonctionnalité d'une application spécifique. Lorsque XMLSpy est intégré dans Eclipse, une perspective XMLSpy par défaut est créée automatiquement. Cette perspective est une GUI qui inclut les éléments de GUI de XMLSpy : ses modes d'édition, ses menus, ses assistants à la saisie, et d'autres barres latérales.

Lorsqu'un fichier dont le type de fichier est associé avec XMLSpy est ouvert, ce fichier peut être édité dans la perspective XMLSpy. De même, un fichier d'un autre type de fichier peut être ouvert dans une autre perspective dans Eclipse. En outre, pour tout fichier actif, vous pouvez sauter la perspective, vous permettant donc d'éditer ou de traiter ce fichier dans un autre environnement.

Deux avantages principaux se présentent donc pour les perspectives :

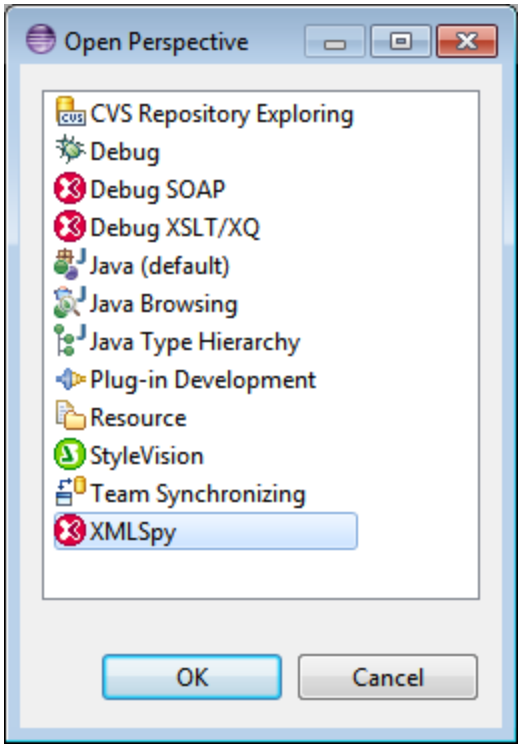
1. la possibilité de changer rapidement l'environnement de travail du fichier actif et
2. La possibilité de passer entre les fichiers sans devoir ouvrir un nouvel environnement de développement (l'environnement associé est disponible dans une perspective)

Le travail avec la perspective XMLSpy implique les éléments suivants :

- Passer à la perspective XMLSpy.
- Définir les préférences pour la perspective XMLSpy.
- Personnaliser la perspective XMLSpy.

Passer à la perspective XMLSpy

Dans Eclipse, sélectionnez la commande **Window | Perspective | Open Perspective | Other**. Dans le dialogue qui s'ouvre (*capture d'écran ci-dessous*), sélectionnez **XMLSpy** et cliquez sur **Ouvrir**.



La fenêtre vide ou le document actif aura maintenant la perspective XMLSpy. Voici comment l'utilisateur change de perspective par le menu. Pour accéder à une perspective plus rapidement depuis une autre perspective, la perspective requise peut être recensée dans le sous-menu **Open Perspective** au-dessus de l'item **Other**. Ce paramètre est dans le dialogue de personnalisation (*voir plus bas ci-dessous*).

Les perspectives peuvent aussi être changées lorsqu'un fichier est ouvert ou est rendu actif. La perspective de l'application associée avec un type de fichier d'un fichier sera ouvert automatiquement lorsque ce fichier est ouvert pour la première fois. Avant de changer de perspective, un dialogue apparaît vous demandant si vous souhaitez associer automatiquement la perspective par défaut avec ce type de fichier. Vérifiez l'option *Do Not Ask Again* si vous souhaitez associer la perspective avec un filetype sans devoir y être invité chaque fois qu'un fichier de ce filetype est ouvert, puis cliquez sur **OK**.

Les préférences pour la perspective XMLSpy

Les préférences d'une perspective comprennent : (i) un paramètre pour changer automatiquement la perspective lorsqu'un fichier d'un type de fichier associé est ouvert (*voir ci-dessus*), et (ii) des options pour inclure ou exclure des barres d'outils XMLSpy individuelles et (iii) accéder aux options XMLSpy. Pour accéder au dialogue Préférences, sélectionnez la commande **Window | Preferences**. Dans la liste des perspectives du panneau de gauche, sélectionnez XMLSpy, puis sélectionnez les préférences requises. Terminez en cliquant sur **OK**.

Personnalisez la perspective XMLSpy

Les options de personnalisation vous permettent de déterminer quels raccourcis et commandes sont inclus dans la perspective. Pour accéder au dialogue Personnaliser la Perspective, transformez la perspective en perspective active et sélectionnez la commande **Window | Perspective | Customize Perspective**.

- Dans les onglets *Toolbar Visibility* et *Menu Visibility*, vous pourrez spécifier quelles barres d'outils et

 © 2016-2022 Altova GmbH

menu doivent être affichés.
- Dans l'onglet *Action Set Availability*, vous pouvez ajouter des groupes de commandes à leurs menu parents et à la barre d'outils. Si vous souhaitez activer un groupe de commande, cochez sa case.
- Dans l'onglet *Shortcuts* du dialogue Customize Perspective, vous pouvez configurer des raccourcis pour les sous-menus. Sélectionnez le sous-menu dans la liste déroulante des sous-menus. Puis sélectionnez une catégorie de raccourcis, et cochez les raccourcis que vous souhaitez inclure à la perspective.

Cliquer sur **Appliquer et Fermer** pour terminer la personnalisation et pour que les modifications prennent effet.

26.3 Other XMLSpy Entry Points in Eclipse

Outre la perspective XMLSpy, deux autres points d'entrée dans Eclipse peuvent être utilisés pour accéder la fonctionnalité XMLSpy :

- menu XMLSpy
- barre d'outils XMLSpy

Menu XMLSpy dans Eclipse

Le menu **XMLSpy** d'Eclipse contient les commandes XMLSpy qui fournissent la fonctionnalité XMLSpy. Ces commandes apparaissent dans de nombreux menus de la version autonome de XMLSpy.

En bas de ce menu vous trouverez les commandes qui définissent le thème de la perspective XMLSpy dans Eclipse.

Barre d'outils XMLSpy dans Eclipse

La barre d'outils XMLSpy dans Eclipse (*voir la capture d'écran ci-dessous*) contient deux boutons.



Ces boutons font la chose suivante :

- Ouvrir l'Aide XMLSpy
- Fournit un accès aux commandes XMLSpy (comme alternative pour y accéder depuis le menu **XMLSpy**, *voir ci-dessus*).

Note : les commandes de la barre d'outils me sont pas prises en charge. Si vous avez défini une commande de barre d'outils dans XMLSpy qui exécute une commande ou un script, alors cette commande de barre d'outils ne sera pas disponible dans le plug-in.

Formats de fichier XMLSpy et comportement des modes Eclipse

Lorsque certains types de fichiers reconnus par XMLSpy sont actifs (en focus) dans Eclipse, les modes *Éléments*, *Attributs*, et *Entités* apparaîssent avec le nom qui est explicite pour ce format. Par exemple, lorsqu'un fichier `.css` est actif, le mode Éléments porte les nom *CSS Outline*. La table ci-dessous illustre comment les noms des modes changent selon le fichier actif :

Ce format de fichier actif	Le mode Éléments devient...	Le mode Attributs devient...	Le mode Propriétés devient...
`.css`	CSS Outline	Propriétés CSS	Éléments HTML
`.xquery, .xq`	Mots-clés XQuery	Variables XQuery	Fonctions XQuery
`.xsd`	Composants	Détails	Facettes

© 2016-2022 Altova GmbH

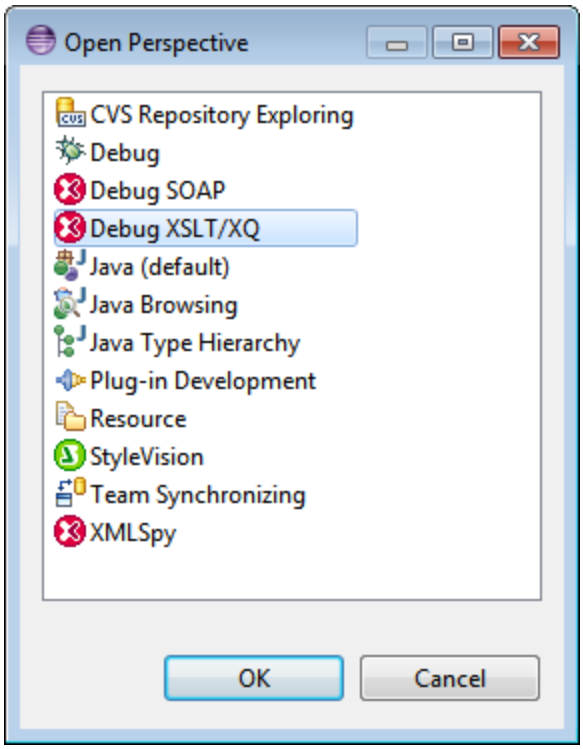
Si vous fermez un de ces modes, vous pourrez le rétablir ultérieurement en utilisant la commande de menu **Window | Show | View**. Veuillez noter, néanmoins, que les modes sont affichés dans ce menu avec leur nom générique (c.-à-d., *Éléments*, *Attributs*, et *Entités*). Ainsi, par exemple, pour restaurer le mode *CSS Outline*, vous devrez sélectionner **Show | View | Elements**.

En alternative, réinitialisez la perspective XMLSpy à ses valeurs par défaut, à en cliquant **Window | Reset Perspective**.

26.4 XMLSpy's Debugger Perspectives

Il existe deux débogueurs dans l'édition Enterprise de XMLSpy (XSLT/XQuery et SOAP), et un débogueur dans l'édition Professional de XMLSpy (XSLT/XQuery). Les perspectives pour ces débogueurs sont disponibles dans Eclipse conformément à l'édition XMLSpy actuellement installée.

Pour changer de perspective de débogage, sélectionnez la commande **Window | Open Perspective | Other**. Dans le dialogue qui s'affiche (*capture d'écran ci-dessous*), choisir le débogueur (par ex., Debug XSLT/XQ), et cliquez sur **OK**.



La fenêtre vide du document actif présentera maintenant la perspective du débogueur sélectionné. Voici comment l'utilisateur change de perspective par le menu. Pour accéder à une perspective plus rapidement depuis une autre perspective, la perspective requise peut être recensée dans le sous-menu **Open Perspective** au-dessus de l'item **Other** ; ce paramètre se trouve dans le dialogue de personnalisation.

Pour une description de l'utilisation des débogueurs, consultez les sections respectives dans cette documentation : XSLT et XQuery, et WSDL et SOAP.

© 2016-2022 Altova GmbH

26.5 Points de contact XMLSpy dans Eclipse

Les points d'entrée suivants dans Eclipse peuvent être utilisés pour accéder à la fonctionnalité de XMLSpy :

- Perspective XMLSpy, qui procure les fonctions de GUI de XMLSpy au sein de la GUI d'Eclipse.
- Menu et barre outils XMLSpy

Perspective XMLSpy

Dans Eclipse, une perspective est un mode GUI configuré avec une fonctionnalité de différentes applications. Lorsque le plugin XMLSpy pour Eclipse est intégré dans Eclipse, une perspective XMLSpy par défaut est créée automatiquement. Cette perspective est une GUI qui inclut les éléments de GUI de XMLSpy : ses modes d'édition, ses menus, ses assistants à la saisie, et d'autres barres latérales.

Lorsqu'un fichier dont le type de fichier est associé avec XMLSpy est ouvert (`.xml`, par exemple), ce fichier peut être édité dans la perspective XMLSpy. De même, un fichier d'un autre type de fichier peut être ouvert dans une autre perspective dans Eclipse. En outre, pour tout fichier actif, vous pouvez sauter la perspective, vous permettant donc d'éditer ou de traiter ce fichier dans un autre environnement. Deux avantages principaux se présentent donc pour les perspectives :

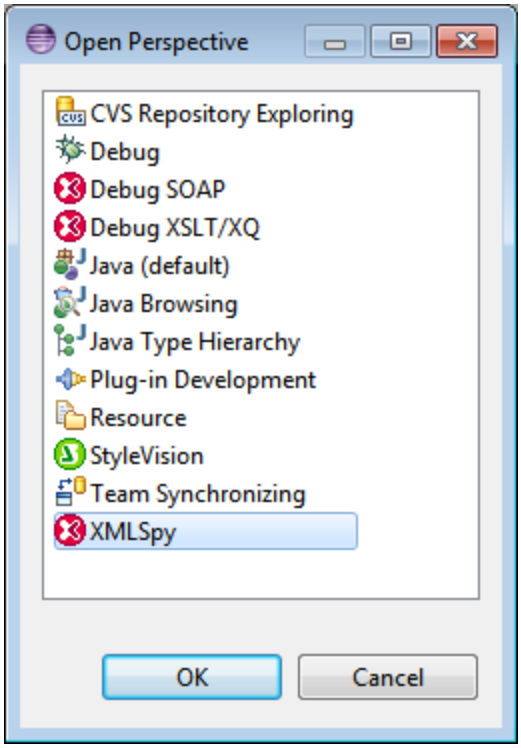
1. la possibilité de changer rapidement l'environnement de travail du fichier actif et
2. la possibilité de passer entre les fichiers sans devoir ouvrir un nouvel environnement de développement (l'environnement associé est disponible dans une perspective)

Le travail avec la perspective XMLSpy implique les éléments suivants :

- Passer à la perspective XMLSpy.
- Définir les préférences pour la perspective XMLSpy.
- Personnaliser la perspective XMLSpy.

Passer à la perspective XMLSpy

Dans Eclipse, sélectionnez la commande **Window | Perspective | Open Perspective | Other**. Dans le dialogue qui s'ouvre (*capture d'écran ci-dessous*), sélectionnez **XMLSpy** et cliquez sur **Ouvrir**.



La fenêtre vide ou le document actif aura maintenant la perspective XMLSpy. Voici comment l'utilisateur change de perspective par le menu. Pour accéder à une perspective plus rapidement depuis une autre perspective, la perspective souhaitée peut être incluse dans le sous-menu **Open Perspective**, situé au-dessus de l'item **Other** ; ce paramètre se trouve dans le dialogue de personnalisation (*voir ci-dessous*).

Les perspectives peuvent aussi être changées lorsqu'un fichier est ouvert ou est rendu actif. La perspective de l'application associée avec un type de fichier d'un fichier sera ouvert automatiquement lorsque ce fichier est ouvert pour la première fois. Avant de changer de perspective, un dialogue apparaît vous demandant si vous souhaitez associer automatiquement la perspective par défaut avec ce type de fichier. Vérifiez l'option *Do Not Ask Again* si vous souhaitez associer la perspective avec un filetype sans devoir y être invité chaque fois qu'un fichier de ce filetype est ouvert, puis cliquez sur **OK**.

Définir les préférences pour la perspective XMLSpy

Les préférences d'une perspective comprennent : (i) un paramètre pour changer automatiquement la perspective lorsqu'un fichier d'un type de fichier associé est ouvert (*voir ci-dessus*), et (ii) des options pour inclure ou exclure des barres d'outils XMLSpy individuelles. Pour accéder au dialogue Préférences, sélectionnez la commande **Window | Preferences**. Dans la liste des perspectives du panneau de gauche, sélectionnez XMLSpy, puis sélectionnez les préférences requises. Terminez en cliquant sur **OK**.

Personnalisez la perspective XMLSpy

Les options de personnalisation vous permettent de déterminer quels raccourcis et commandes sont inclus dans la perspective. Pour accéder au dialogue Customize Perspective d'une perspective (*la capture d'écran ci-dessous montre le dialogue pour la perspective XMLSpy*), rendre la perspective active (dans ce cas, la perspective XMLSpy), et sélectionnez la commande **Window | Perspective | Customize Perspective**. Dans les onglets « Tool Bar Visibility » et « Menu Visibility », vous pourrez spécifier quelles barres d'outils et

 © 2016-2022 Altova GmbH

menu doivent être affichés. Dans l'onglet « Action Set Availability », vous pouvez ajouter des groupes de commandes à leurs menu parents et à la barre d'outils. Si vous souhaitez activer un groupe de commande, cochez sa case. Dans l'onglet « Shortcuts » du dialogue « Customize Perspective », vous pouvez configurer des raccourcis pour les sous-menus. Sélectionnez le sous-menu dans la liste de choix « Submenus ». Puis sélectionnez une catégorie de raccourcis, et cochez les raccourcis que vous souhaitez inclure à la perspective. Cliquer sur **Appliquer & Fermer** pour terminer la personnalisation et pour que les modifications prennent effet.

Menu et barre outils XMLSpy

Le menu **XMLSpy** contient des commandes qui sont pertinentes pour XMLSpy. Dans la version standalone de XMLSpy, certaines de ces commandes se trouvent dans le menu **Fichier**. La barre outils XMLSpy contient les boutons suivants (*capture d'écran ci-dessous*).



De gauche à droite, ils servent à : (i) ouvrir l'aide de XMLSpy, et (ii) à accéder aux commandes XMLSpy (en tant qu'alternative d'accès depuis le menu **XMLSpy**).

Note : les commandes de la barre d'outils me sont pas prises en charge. Si vous avez défini une commande de barre d'outils dans XMLSpy qui exécute une commande ou un script, alors cette commande de barre d'outils ne sera pas disponible dans le plug-in.

Formats de fichier XMLSpy et comportement des modes Eclipse

Lorsque certains types de fichiers reconnus par XMLSpy sont actifs (en focus) dans Eclipse, les modes « Éléments », « Attributs » et « Entités » apparaîssent avec le nom qui est explicite pour ce format. Par exemple, lorsqu'un fichier .css est actif, le mode « Éléments » apparaîtra portant le nom « CSS Outline ». La table ci-dessous illustre comment les noms des modes changent selon le fichier actif :

Lorsque ce format de fichier est actif...	**Le mode « Éléments » devient ...**	**Le mode « Attributs » devient...**	**Le mode « Propriétés » devient...**
.css	CSS Outline	Propriétés CSS	Éléments HTML
.xquery, .xq	Mots-clés XQuery	Variables XQuery	Fonctions XQuery
.xsd	Composants	Détails	Facettes

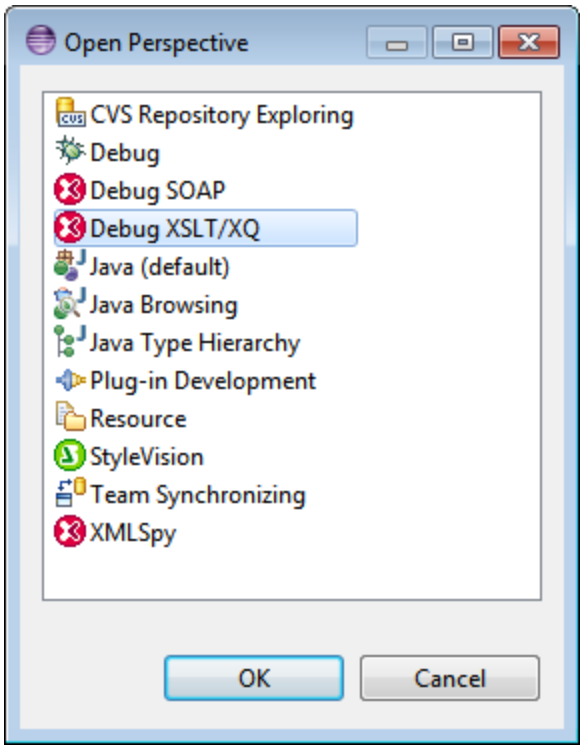
Si vous fermez un de ces modes, vous pourrez le rétablir ultérieurement en utilisant la commande de menu **Window | Show | View**. Veuillez noter, néanmoins, que les modes sont affichés dans ce menu avec leur nom générique (c.-à-d., « Éléments », « Attributs » et « Entités »). Ainsi, par exemple, pour restaurer le mode « CSS Outline », vous devrez sélectionner **Show | View | Elements**.

En alternative, réinitialisez la perspective XMLSpy à ses valeurs par défaut, à partir de **Window | Reset Perspective**.

26.6 Perspectives de débogeur de XMLSpy

Il existe deux débogueurs dans l'édition Enterprise de XMLSpy (XSLT/XQuery et SOAP), et un débogueur dans l'édition Professional de XMLSpy (XSLT/XQuery). Les perspectives pour ces débogueurs sont disponibles dans Eclipse conformément à l'édition XMLSpy actuellement installée.

Pour changer de perspective de débogueur, sélectionner la commande **Window | Open Perspective | Other**. Dans le dialogue qui s'affiche (*capture d'écran ci-dessous*), choisir le débogueur (par ex., Debug XSLT/XQ), et cliquer sur **OK**.



La fenêtre vide du document actif présentera maintenant la perspective du débogueur sélectionné. Voici comment l'utilisateur passe d'une perspective à une autre avec le menu. Pour accéder à une perspective plus rapidement depuis une autre perspective, la perspective requise peut être recensée dans le sous-menu **Open Perspective** au-dessus de l'item **Other** ; Ce paramètre se trouve dans le dialogue de personnalisation .

Pour une description de l'utilisation des débogueurs, consulter les sections respectives dans cette documentation : XSLT et XQuery, et WSDL et SOAP.

© 2016-2022 Altova GmbH

27 Générateur de code

XMLSpy contient un générateur de code built-in qui peut générer des fichiers de classe Java, C++ ou C# depuis des schémas XML. Le code généré consiste en des bibliothèques de schema wrapper de type fort qui vous permettent de créer des applications de logiciel qui traitent des données XML. Les bibliothèques de schema wrapper vous permettent de travailler avec des données XML de manière programmatique, utilisant des types générés depuis le schéma, sans trop se soucier de l'API XML sous-jacent.

L'éditeur de schéma XMLSpy convient en tant qu'outil de prototypage et de modelage de logiciel, permettant aux applications XML d'être rapidement prototypées à un niveau élevé dans le Schéma XML et ensuite automatiquement générées. Les changements à un modèle de contenu de Schéma XML peuvent immédiatement être rapprochés avec une mise en œuvre du logiciel, simplement en exécutant une nouvelle fois le générateur de code.

Le code généré prend en charge les opérations suivantes :

- Lire les fichiers XML dans un Document Object Model (DOM) en représentation en mémoire
- Écrire des fichiers XML depuis une représentation DOM de retour à un fichier de système
- Convertir les strings vers des arborescences DOM XML et vice-versa.

Le code généré est exprimé en langages de programmation C++, Java ou C#.

Langage cible	C++	C#	Java
Environnements de développement	Microsoft Visual Studio 2013, 2015, 2017, 2019, 2022	Microsoft Visual Studio 2013, 2015, 2017, 2019, 2022 Frameworks cibles : • .NET Framework • .NET Core 3.1 • .NET 5.0 • .NET 6.0	Java SE JDK 8 ou 11 (y compris OpenJDK) Eclipse 4.4 ou ultérieur Apache Ant (build.xml file)
Mises en œuvre XML DOM	MSXML 6.0 Apache Xerces 3	System.Xml	JAXP

C++

Vous pouvez configurer si la sortie générée C++ doit utiliser MSXML 6.0 ou Apache Xerces 3. XMLSpy génère le projet complet (.vcproj) et les fichiers de solution (.sln) pour toutes les versions prises en charge de Visual Studio (voir la table ci-dessus)Le code généré prend en charge MFC optionnellement.

Prérequis :

1. Pour compiler le code C++ généré, Windows SDK doit être installé sur votre ordinateur.
2. Pour utiliser Xerces 3 pour C++, vous allez devoir l'installer et le construire par les instructions suivantes depuis http://xerces.apache.org/xerces-c/. Assurez-vous d'ajouter la variable d'environnement XERCES3 qui pointe vers le répertoire dans lequel Xerces a été installé, par exemple : `C:\xerces-c-`

`3.2.2`. Aussi, la variable d'environnement PATH doit inclure le chemin où se trouvent les binaires Xerces, par exemple : `%XERCES3%\bin`.

3. Lorsque vous construisez un code C++ pour Visual Studio et utilisez une bibliothèque Xerces précompilée pour Visual C++, un paramètre de compilateur doit être changé dans tous les projets de la solution :

   a) Sélectionnez tous les projets dans le Solution Explorer.
   b) Dans le menu **Projet**, cliquez sur **Propriétés**.
   c) Cliquez sur **Propriétés de configuration | C/C++ | Langage**.
   d) Dans la liste de configurations, sélectionnez *Toutes les Configurations*.
   e) Changer **Treat wchar_t as Built-in Type** à **No (/Zc:wchar_t-)**.

C#

Le code C# généré peut être utilisé depuis tout langage de programmation .NET capable, tel que VB.NET, Managed C++, ou J#. Les fichiers de projet peuvent être générés pour toutes les versions prises en charge de Visual Studio (voir la table ci-dessus).

Java

La sortie Java générée est écrite par rapport à l'API Java pour Traitement XML (JAXP) et inclut un fichier Ant build et des fichiers de projet pour des versions prises en charge de Java et Eclipse (voir la table ci-dessus).

Sortie générée

Le dossier de destination désigné pour le code généré inclut également toutes les bibliothèques et tous les fichiers requis pour manipuler les fichiers XML de manière programmatique, à savoir :

- Les bibliothèques standard d'Altova
- Les bibliothèques Schema wrapper
- Une application test vide avec un code source échantillon. Le squelette de l'application test est une application compilable qui appelle une méthode d'Exemple() vide. Vous pouvez ajouter votre code test à cette méthode pour des tests rapides et facile de votre bibliothèque nouvellement générée.

Modèles de générateur de code

Le code généré prend en charge la personnalisation par le biais d'un langage de modèle appelé SPL (Spy Programming Language). SPL donne le contrôle total du mappage de XML Schema built-in data-types vers les types de données primitifs d'un langage de programmation particulier. Ceci vous permet de construire vos propres modèles en vue d'automatiser la génération de presque tout autre format, par exemple : EJB, fichiers WSDL, scripts SQL, ASP ou le code WML.

© 2016-2022 Altova GmbH

27.1 Les nouveautés...

Version 2022 R2

- Prise en charge de Visual Studio 2022.
- Prise en charge de . NET 6.0.

Version 2021 R2

- Lorsque vous générez le code C#, vous pouvez choisir .NET Core 3,1, NET 5.0 ou .NET 6.0 en tant qu'infrastructures cibles depuis les options de génération de code (ceci s'ajoute à la prise en charge existante pour les projets .NET Framework).

Version 2020 R2

- Le code généré pour des bibliothèques de wrapper de schéma XML propose maintenant un plus grand contrôle sur les espaces de noms d'élément et des préfixes. De nouvelles méthodes sont disponibles pour déclarer ou contourner des espaces de noms pour un élément, ou pour apposer un élément avec un espace de noms préfixé. Voir Exemple : Purchase Order.

Version 2020

- Prise en charge de Visual Studio 2019
- Fin de la prise en charge de Visual Studio 2008

Version 2018

- Prise en charge étendue à Microsoft Visual Studio 2013, 2015, 2017
- Fin de la prise en charge de Visual Studio 2005 et Xerces 2.x

Version 2014

- Retrait de l'option de mode de compatibilité pour la génération de code

Version 2011

- Contient des corrections de bogues et des améliorations

Version 2010 R3

- Prise en charge de Microsoft Visual Studio 2010
- Prise en charge de MSXML 6.0 dans du code C++ généré
- Prise en charge de cibles 64-bit pour les projets C++ et C#

Version 2010 R2

- Génération de code C++ pour la plateforme Linux

Version 2010

- Des facettes d'énumération provenant de schémas XML sont maintenant disponibles en tant que constantes symboliques dans les classes générées (en utilisant les modèles 2007r3)

Version 2009 sp1

- Prise en charge d'Apache Xerces version 3.x (les anciennes versions commençant à partir de Xerces 2.6.x sont toujours prises en charge)

Version 2008 R2

- La prise en charge de la génération de fichiers de projet Visual Studio 2008 pour C# et C++ a été ajoutée

Version 2008

- Les nouveaux modèles 2007 R3-style SPL ont encore été améliorés :
  - Il est maintenant possible de supprimer des éléments uniques
  - L'accès aux métadonnées de schéma (par ex. noms d'élément, facettes, énumérations, occurrence, etc.) est possible
  - Les types complexes dérivés par extension sont maintenant générés en tant que classes dérivées

Version 2007 R3

Le Générateur de code a été reconçu pour la version 2007 release 3 dans le but de simplifier l'utilisation du code généré, réduire le volume de code et augmenter la performance.

- Gestion des documents XML et des nœuds avec une propriété explicite pour éviter les pertes de mémoire et permettre un multi-threading
- Nouvelle syntaxe pour éviter les collisions de nom
- Nouveaux types de données pour une utilisation plus simple et une meilleure performance (types natifs si possible, nouveau null handling, ...)
- Les attributs ne sont plus générés en tant que collections
- Le contenu d'élément simple est maintenant aussi traité comme un attribut spécial, pour une plus grande homogénéité
- Nouveau modèle d'objet interne (important pour les modèles SPL personnalisés)
- Mode de compatibilité pour générer du code dans le style des releases plus anciens

© 2016-2022 Altova GmbH

27.2 Générer du code depuis des Schémas XML ou des DTDs

Avec le générateur de code XMLSpy, vous pouvez générer du code de programmation C#, C++, ou Java depuis des schémas XML ou des DTDs. Les bibliothèques de schema wrapper générées peuvent ensuite être intégrées dans votre application personnalisée pour lire, modifier ou écrire des documents XML par le biais d'un programme.

Générer du code de programme

1. Ouvrir le schéma pour lequel vous souhaitez générer du code de source.
2. Choisir l'item de menu **DTD/Schéma | Générer du code de programme**.
3. Dans le volet Choisir modèle du dialogue qui s'ouvre, définir les options de générateur de code.
4. Cliquer sur **OK**. Le dialogue *Chercher des dossiers* apparaît.
5. Choisir le dossier cible et cliquer sur **OK**.
6. Vous êtes invité à ouvrir le projet récemment créé dans Microsoft Visual Studio. Cliquer sur **Yes**. Si le code de Java est produit, vous êtes invité à ouvrir le répertoire de sortie correspondant.

Lorsque XMLSpy génère du code depuis un Schéma XML ou DTD, les bibliothèques suivantes sont créées :

C++ ou C#	Java	Objectif
Altova	`com.altova`	Bibliothèque de base contenant un support de runtime commun, à l'identique pour chaque schéma.
AltovaXML	`com.altova.xml`	Bibliothèque de base contenant un support de runtime pour XML, à l'identique pour chaque schéma.
YourSchema	*com.YourSchema*	Une bibliothèque contenant des déclarations générées depuis le schéma d'entrée, nommé comme le fichier de schéma ou le DTD. Cette Bibliothèque est un wrapper DOM (W3C Document Object Model) qui vous permet de lire, modifier et créer des documents XML en toute sécurité. Toutes les données contenues dans le DOM, et il existe des méthodes pour extraire des données depuis le DOM, et pour mettre à jour et créer des données dans le DOM. Le code C++ généré prend en charge soit Microsoft MSXML soit Apache Xerces 3. La syntaxe pour utiliser le code généré est généralement semblable pour les deux implémentations DOM, sauf pour quelques légères différences (par exemple, Xerces prend en charge plus de fonctions surchargées). Le code C# généré utilise la bibliothèque standard .NET .XML en guise de mise en place DOM sous-jacente. Le code Java généré utilise JAXP (Java API pour traitement XML) en tant que l'interface DOM sous-jacente.

YourSchemaTest	*com.YourSchemaTest*	Le code généré contient également un squelette d'application de test nommé d'après votre schéma (par exemple, *YourSchemaTest*). Ceci est une application compilable qui appelle une méthode Example() vide. Vous pouvez ajouter votre code de texte dans cette méthode pour un test rapide et simple de votre bibliothèque récemment générée.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir Intégrer des Bibliothèques Schema Wrapper).

Génération de nom et espaces de noms

XMLSpy génère des classes correspondant à tous les éléments déclarés ou les types complexes qui redéfinissent tout type complexe dans votre Schéma XML, préservant la dérivation de classe telle que définie par les extensions des types complexes dans votre Schéma XML. Dans le cas de schémas complexes qui importent des composants de schéma depuis plusieurs espaces de noms, XMLSpy préserve cette information en générant les espaces de noms C# ou C++ ou les packages Java appropriés.

Généralement, le générateur de code tente de préserver les noms pour les espaces de noms générés, les classes et les membres provenant du schéma XML original. Les caractères qui ne sont pas valides dans les identifieurs dans le langage cible sont remplacés par "_". Les noms qui se heurtent avec d'autres noms ou des mots réservés sont rendus uniques en ajoutant un nombre. La génération de nom peut être influée en changeant les paramètres par défaut dans le modèle SPL.

Les espaces de noms provenant du Schéma XML sont convertis en packages dans Java ou des espaces de noms dans du code C# ou C++, en utilisant le préfixe d'espace de noms depuis le schéma en tant qu'espace de noms de code. La bibliothèque complète est contenue dans un package ou dans un espace de noms dérivé depuis le nom de fichier, pour que vous puissiez utiliser plusieurs bibliothèques générées dans un programme sans conflits de noms.

Types de données

Schéma XML dispose d'un modèle de type de données plus élaboré que Java, C# ou C++. Le générateur de code convertit les types de Schéma XML dans des types primitifs spécifiques au langage, ou dans les classes fournies avec la bibliothèque Altova. Les types complexes et les types dérivés définis dans le schéma sont convertis en classes dans la bibliothèque générée. Les facettes d'énumération provenant de types simples sont converties dans des constantes symboliques.

Le mappage des types simples peut être configuré dans le modèle SPL, voir Référence SPL.

Si vos fichiers d'instance XML utilisent des types de schéma liés à l'heure et la durée, ceux-ci sont convertis en classes natives Altova dans le code généré. Pour plus d'informations concernant les classes de bibliothèque, voir :

© 2016-2022 Altova GmbH

- Référence aux classes générées (C++)
- Référence aux classes générées (C#)
- Référence aux classes générées (Java)

Pour plus d'informations concernant la conversion de type et d'autres détails applicables dans chaque langage, voir :

- À propos des Bibliothèques de Schema Wrapper (C++)
- À propos des Schema Wrapper Libraries (C#)
- À propos des Schema Wrapper Libraries (Java)

Gestion de mémoire

Une arborescence DOM est constituée de nœuds, qui sont toujours possédées par un document DOM spécifique - même si le nœud ne fait pas actuellement partie du document. Toutes les classes générées sont des références aux nœuds DOM qu'elles représentent, par des valeurs. Cela signifie que l'attribution d'une instance d'une classe générée ne copie pas la valeur, elle crée uniquement une référence supplémentaire dans la même donnée.

Prise en charge de Schéma XML

Les structures de Schéma XML suivant sont traduites dans du code :

a) Espace des noms XML

b) Types simple :

- Types de schéma XML Built-in
- Types simples dérivés par extension
- Types simples dérivés par restriction
- Facettes
- Enumérations
- Motifs

c) Types complexes :

- Nœud anyType Built-in
- Types complexes définis par l'utilisateur
- Dérivé par extension : Mappé vers les classes dérivées
- Dérivé par restriction
- Contenu complexe
- Contenu simple
- Contenu mixte

Les fonctions de Schéma XML avancées suivantes ne sont pas prises en charge (ou pas entièrement prises en charge) dans les classes de wrapper générées :

- Caractères génériques : `xs:any` et `xs:anyAttribute`
- Modèles de contenu (séquence, choix, tout). Le compositeur de niveau supérieur est disponible dans SPL, mais n'est pas mis en place par des classes générées.
- Les valeurs par défaut et fixées pour les attributs. Celles-ci sont disponibles dans SPL, mais ne sont pas configurées ou mises en place par des classes générées.

- Les attributs `xsi:type`, les types abstraits. Lorsque vous souhaitez écrire l'attribut xsi:type, utiliser la méthode `SetXsiType()` des classes générées.
- Types d'union : toutes les combinaisons ne sont pas prises en charge.
- Les groupes de substitution sont partiellement pris en charge (résolus en tant que "choice").
- Attribut `nillable="true"` et `xsi:nil`
- Contraintes d'unicité
- Contraintes d'identité (`key` et `keyref`)

27.2.1 À propos des Schema Wrapper Libraries (C++)

Types de caractère

Le code C++ généré peut être compilé avec ou sans prise en charge Unicode. Selon ce paramètre, les types `string_type` et `tstring` seront tous les deux définis en tant que `std::string` ou `std::wstring`, consistant en des caractères larges ou étroits. Pour utiliser les caractères Unicode dans votre fichier XML qui ne sont pas représentables avec l'ensemble de caractères 8-bit actuels, la prise en charge Unicode doit être activée. Prêtez une attention particulière aux macros `_T()` . Ce macro garantit que les constantes de string sont stockées correctement, que vous compiliez pour les programmes Unicode ou non-Unicode.

Types de données

Le mappage par défaut des types Schéma XML dans les types de données C++ est :

Schéma XML	C++	Remarques
`xs:string`	`string_type`	string_type est défini en tant que std::string ou std:wstring
`xs:boolean`	`bool`	
`xs:decimal`	`double`	C++ n'a pas de type décimal, donc double est utilisé.
`xs:float, xs:double`	`double`	
`xs:integer`	`__int64`	xs:integer a une plage illimitée, mappée sur __int64 pour des raisons d'efficacité.
`xs:nonNegativeInteger`	`unsigned __int64`	voir ci-dessus
`xs:int`	`int`	
`xs:unsignedInt`	`unsigned int`	
`xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth`	`altova::DateTime`	
`xs:duration`	`altova::Duration`	
`xs:hexBinary and xs:base64Binary`	`std::vector<unsigned char>`	L'encodage et le décodage des données binaires est effectué

© 2016-2022 Altova GmbH

Schéma XML	C++	Remarques
		automatiquement.
xs:anySimpleType	string_type	

Tous les types de Schéma XML qui ne sont pas contenus dans la liste sont des types dérivés, et mappés sur le même C++ que leur type de base respectif.

Classes générées

Pour chaque type dans le schéma, une classe est générée qui contient un membre pour chaque attribut et élément du type. Les membres sont nommés comme les attributs ou éléments dans le schéma original (en cas d'une collision éventuelle, un nombre est ajouté). Pour les types simples, des opérateurs d'attribution et de conversion sont générés. Pour des types simples avec des facettes d'énumération, les méthodes `GetEnumerationValue()` et `SetEnumerationValue(int)` peuvent être utilisés ensemble avec des constantes générées pour chaque valeur d'énumération. De plus, la méthode `StaticInfo()` permet l'accès à l'information de schéma en tant qu'un des types suivants :

`altova::meta::SimpleType`
`altova::meta::ComplexType`

Les classes générées depuis des types complexes comprennent la méthode `SetXsiType()`, qui vous permet de définir l'attribut `xsi:type` du type. Cette méthode est utile si vous souhaitez créer des éléments d'instance XML d'un type dérivé.

Outre les classes pour les types déclarés dans le schéma XML, une classe de document (identifiée par "CDoc", voir ci-dessous) est générée. Elle contient tous les éléments root possibles, en tant que membres, ainsi que plusieurs autres méthodes. Pour plus d'informations concernant la classe, voir `[YourSchema]::[CDoc]` .

Note: Le véritable nom de la classe dépend du nom du schéma .xsd.

Pour chaque attribut ou élément membre d'un type de schéma, une nouvelle classe est générée. Pour plus d'informations concernant ces classes, voir :

`[YourSchema]::MemberAttribute`
`[YourSchema]::MemberElement`

Note: Les véritables noms de la classe dépendent du nom de l'attribut ou de l'élément membre du schéma.

Voir aussi Exemple : Utiliser les bibliothèques de Schema Wrapper.

Gestion d'erreur

Les erreurs sont rapportées par exceptions. Les classes d'exception suivantes sont définies dans l'espace de noms `altova`:

Classe	Classe de Base	Description
Erreur	std::logic_error	Erreur de logique de programme interne (indépendant des données

		d'entrée).
`Exception`	`std::runtime_error`	Classe de Base pour les erreurs de runtime.
`InvalidArgumentsException`	`Exception`	Une méthode a été appelée avec des valeurs d'argument invalide.
`ConversionException`	`Exception`	Exception lancée lorsqu'une conversion de type échoue.
`StringParseException`	`ConversionException`	Une valeur dans l'espace lexical ne peut pas être converti dans un espace de valeur.
`ValueNotRepresentableException`	`ConversionException`	Une valeur dans l'espace de valeur ne peut pas être converti dans un espace lexical.
`OutOfRangeException`	`ConversionException`	Une valeur de source ne peut pas être représentée dans un domaine cible.
`InvalidOperationException`	`Exception`	Une opération a été tentée qui n'est pas valide dans le contexte donné.
`DataSourceUnavailableException`	`Exception`	Un problème est survenu pendant le chargement d'une instance XML.
`DataTargetUnavailableException`	`Exception`	Un problème est survenu pendant l'enregistrement d'une instance XML.

Toutes les classes d'exception contiennent un texte de message et un pointeur dans une exception intérieure possible.

Méthode	Objectif
`string_type message()`	Retourne une description textuelle de l'exception.
`std::exception inner()`	Retourne l'exception qui a causé cette exception, si disponible, ou NULL.

Accéder à l'information de schéma

La bibliothèque générée permet d'accéder à des informations de schéma statique par le biais des classes suivantes. `Toutes les méthodes sont déclarées en tant que` const. Les méthodes qui retournent une de ces classes des métadonnées retournent un objet NULL si la propriété respective n'existe pas.

`altova::meta::Attribute`
`altova::meta::ComplexType`
`altova::meta::Element`
`altova::meta::SimpleType`

© 2016-2022 Altova GmbH

27.2.2 À propos des Schema Wrapper Libraries (C#)

Le mappage par défaut des types Schéma XML dans les types de données C# est le suivant.

Schéma XML	C#	Remarques
xs:string	string	
xs:boolean	bool	
xs:decimal	decimal	xs:decimal dispose une plage et une précision illimitée, mappé vers les décimales pour des raisons d'efficacité.
xs:float, xs:double	double	
xs:long	long	
xs:unsignedLong	ulong	
xs:int	int	
xs:unsignedInt	uint	
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	Altova.Types.DateTime	
xs:duration	Altova.Types.Duration	
xs:hexBinary and xs:base64Binary	byte[]	L'encodage et le décodage des données binaires est effectué automatiquement.
xs:anySimpleType	string	

Tous les types de Schéma XML qui ne sont pas contenus dans la liste sont des types dérivés, et mappés sur le même C# que leur type de base respectif.

Classes générées

Pour chaque type dans le schéma, une classe est générée qui contient un membre pour chaque attribut et élément du type. Les membres sont nommés comme les attributs ou éléments dans le schéma original (en cas d'une collision éventuelle, un nombre est ajouté). Pour les types simples, des opérateurs d'attribution et de conversion sont générés. Pour des types simples avec des facettes d'énumération, les méthodes `GetEnumerationValue()` et `SetEnumerationValue(int)` peuvent être utilisés ensemble avec des constantes générées pour chaque valeur d'énumération. De plus, la méthode `StaticInfo()` permet l'accès à l'information de schéma en tant qu'un des types suivants :

Altova.Xml.Meta.SimpleType
Altova.Xml.Meta.ComplexType

Les classes générées depuis des types complexes comprennent la méthode `SetXsiType()`, qui vous permet de définir l'attribut `xsi:type` du type. Cette méthode est utile si vous souhaitez créer des éléments d'instance XML d'un type dérivé.

Outre les classes pour les types déclarés dans le schéma XML, une classe de document (identifiée par "Doc", voir ci-dessous) est générée. Elle contient tous les éléments root possibles, en tant que membres, ainsi que plusieurs autres méthodes. Pour plus d'informations concernant la classe, voir `[YourSchema].[Doc]`.

Note: Le véritable nom de la classe dépend du nom du schéma .xsd.

Pour chaque attribut ou élément membre d'un type de schéma, une nouvelle classe est générée. Pour plus d'informations concernant ces classes, voir :

`[YourSchemaType].MemberAttribute`
`[YourSchemaType].MemberElement`

Note: Les véritables noms de la classe dépendent du nom de l'attribut ou de l'élément membre du schéma.

Gestion d'erreur

Les erreurs sont rapportées par exceptions. Les classes d'exception suivantes sont définies dans l'espace de noms Altova :

Classe	Classe de Base	Description
`ConversionException`	`Exception`	Exception lancée lorsqu'une conversion de type échoue
`StringParseException`	`ConversionException`	Une valeur dans l'espace lexical ne peut pas être converti dans un espace de valeur.
`DataSourceUnavailableException`	`System.Exception`	Un problème est survenu pendant le chargement d'une instance XML.
`DataTargetUnavailableException`	`System.Exception`	Un problème est survenu pendant l'enregistrement d'une instance XML.

NET suivantes sont utilisées communément :

Classe	Description
`System.Exception`	Classe de Base pour les erreurs de runtime
`System.ArgumentException`	Une méthode a été appelée avec des valeurs d'argument invalide, ou bien une conversion de type a échoué.
`System.FormatException`	Une valeur dans l'espace lexical ne peut pas être converti dans un espace de valeur.
`System.InvalidCastException`	Une valeur ne peut pas être converie dans un autre type.

© 2016-2022 Altova GmbH

Classe	Description
System.OverflowException	Une valeur de source ne peut pas être représentée dans un domaine cible.

Accéder à l'information de schéma

La bibliothèque générée permet d'accéder à des informations de schéma statique par le biais des classes suivantes :

Altova.Xml.Meta.Attribute
Altova.Xml.Meta.ComplexType
Altova.Xml.Meta.Element
Altova.Xml.Meta.SimpleType

Les propriétés qui retournent une de ces classes des métadonnées retournent null si la propriété respective n'existe pas.

27.2.3 À propos des Schema Wrapper Libraries (Java)

Le mappage par défaut des types Schéma XML dans les types de données Java est le suivant :

Schéma XML	Java	Remarques
xs:string	String	
xs:boolean	booléenne	
xs:decimal	java.math.BigDecimal	
xs:float, xs:double	double	
xs:integer	java.math.BigInteger	
xs:long	long	
xs:unsignedLong	java.math.BigInteger	Java ne dispose pas de types non signés.
xs:int	int	
xs:unsignedInt	long	Java ne dispose pas de types non signés.
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	com.altova.types.DateTime	
xs:duration	com.altova.types.Duration	
xs:hexBinary and xs:base64Binary	byte[]	L'encodage et le décodage des données binaires est effectué

Schéma XML	Java	Remarques
		automatiquement.
`xs:anySimpleType`	`string`	

Tous les types de Schéma XML qui ne sont pas contenus dans la liste sont des types dérivés, et mappés sur le même type Java que leur type de base respectif.

Classes générées

Pour chaque type dans le schéma, une classe est générée qui contient un membre pour chaque attribut et élément du type. Les membres sont nommés comme les attributs ou éléments dans le schéma original (en cas d'une collision éventuelle, un nombre est ajouté). Pour les types simples, des opérateurs d'attribution et de conversion sont générés. Pour des types simples avec des facettes d'énumération, les méthodes `GetEnumerationValue()` et `SetEnumerationValue(int)` peuvent être utilisés ensemble avec des constantes générées pour chaque valeur d'énumération. De plus, la méthode `StaticInfo()` permet l'accès à l'information de schéma en tant qu'un des types suivants :

`com.altova.xml.meta.SimpleType`
`com.altova.xml.meta.ComplexType`

Les classes générées depuis des types complexes comprennent la méthode `SetXsiType()`, qui vous permet de définir l'attribut `xsi:type` du type. Cette méthode est utile si vous souhaitez créer des éléments d'instance XML d'un type dérivé.

Outre les classes pour les types déclarés dans le schéma XML, une classe de document (identifiée par "Doc", voir ci-dessous) est générée. Elle contient tous les éléments root possibles, en tant que membres, ainsi que plusieurs autres méthodes. Pour plus d'informations concernant la classe, voir `com.[YourSchema].[Doc]`.

Note: Le véritable nom de la classe dépend du nom du schéma .xsd.

Pour chaque attribut ou élément membre d'un type de schéma, une nouvelle classe est générée. Pour plus d'informations concernant ces classes, voir :

`com.[YourSchema].[YourSchemaType].MemberAttribute`
`com.[YourSchema].[YourSchemaType].MemberElement`

Note: Les véritables noms de la classe dépendent du nom de l'attribut ou de l'élément membre du schéma.

Gestion d'erreur

Les erreurs sont rapportées par exceptions. Les classes d'exception suivantes sont définies dans l'espace de noms com.altova:

Classe	Classe de Base	Description
`SourceInstanceUnvailableException`	Exception	Un problème est survenu pendant le chargement d'une instance XML.

© 2016-2022 Altova GmbH

Classe	Classe de Base	Description
`TargetInstanceUnavailableException`	Exception	Un problème est survenu pendant l'enregistrement d'une instance XML.

De plus, les exceptions Java suivantes sont utilisées communément :

Classe	Description
`java.lang.Error`	Erreur de logique de programme interne (indépendant des données d'entrée)
`java.lang.Exception`	Classe de Base pour les erreurs de runtime
`java.lang.IllegalArgumentsException`	Une méthode a été appelée avec des valeurs d'argument invalide, ou bien une conversion de type a échoué.
`java.lang.ArithmeticException`	Exception lancée lorsqu'une conversion de type numérique échoue.

Accéder à l'information de schéma

La bibliothèque générée permet d'accéder à des informations de schéma statique par le biais des classes suivantes :

`com.altova.xml.meta.Attribute`
`com.altova.xml.meta.ComplexType`
`com.altova.xml.meta.Element`
`com.altova.xml.meta.SimpleType`

Les propriétés qui retournent une de ces classes des métadonnées retournent null si la propriété respective n'existe pas.

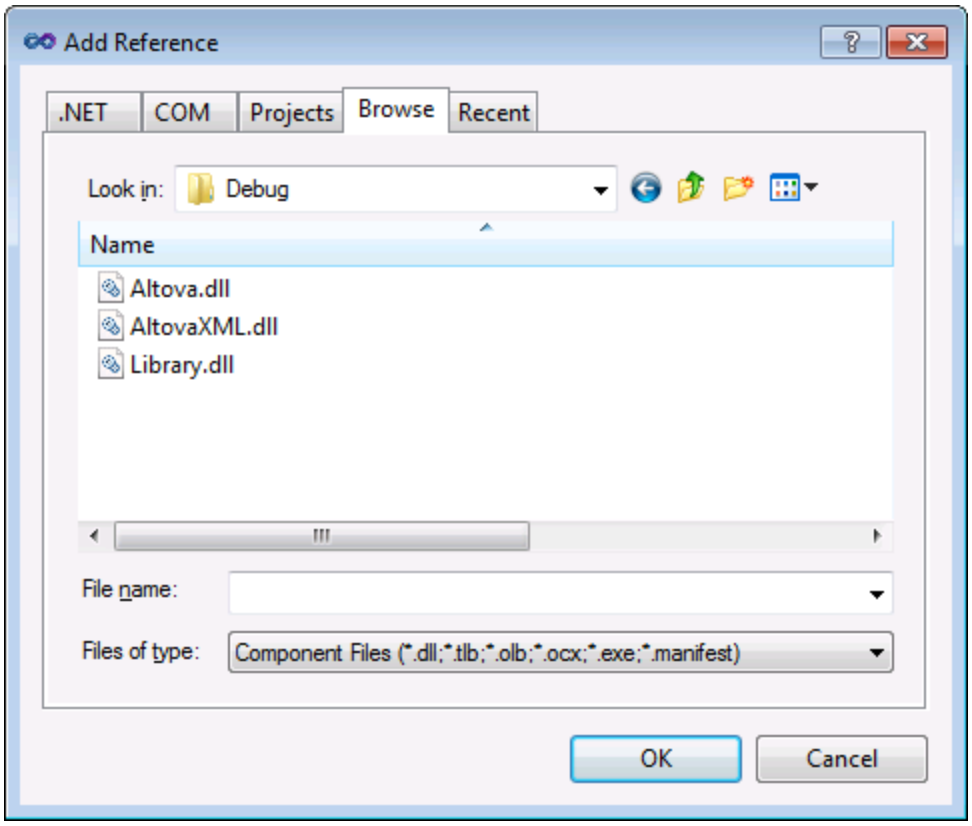
27.2.4 Intégrer des Bibliothèques Schema Wrapper

Pour utiliser les bibliothèques Altova dans votre projet personnalisé, veuillez vous référer aux bibliothèques depuis votre projet ou les inclure dans votre projet, comme indiqué ci-dessous pour chaque langage.

C#

Pour intégrer les bibliothèques Altova dans un projet C# existant :

1. Une fois que XMLSpy génère du code depuis un schéma (par exemple, **YourSchema.xsd**), construire la solution générée **YourSchema.sln** dans Visual Studio. Cette solution se trouve dans un dossier de projet avec le même nom en tant que le schéma.
2. Cliquer avec la touche de droite dans le projet existant dans Visual Studio, et sélectionner **Ajouter Référence**.
3. Dans l'onglet Chercher, naviguer dans les bibliothèques suivantes : **Altova.dll**, **AltovaXML.dll** et **YourSchema.dll** situés dans le répertoire de sortie des projets générés (par exemple, **bin\Debug**).



C++

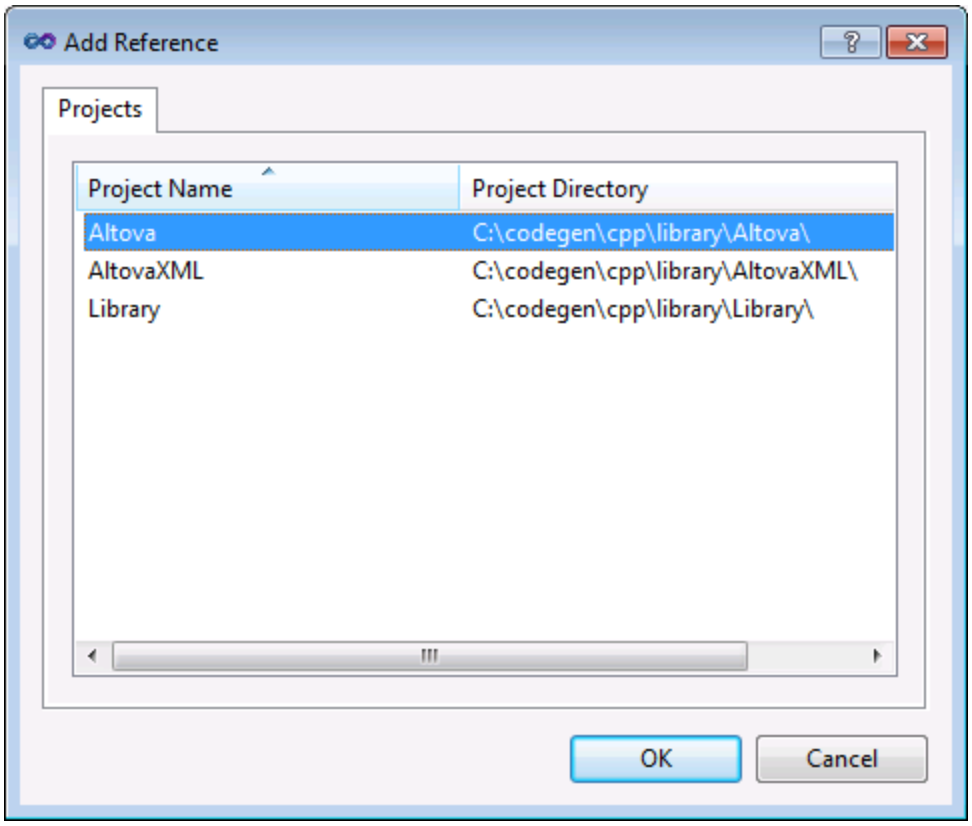
Le moyen le plus simple d'intégrer les bibliothèques dans un projet C++ existant est d'ajouter les fichiers de projet généré dans votre solution. Par exemple, assumons que vous avez généré du code depuis une schéma appelé **Library.xsd** et sélectionné **c:\codegen\cpp\library** en tant que répertoire cible. Les bibliothèques générés dans ce cas sont disponibles sous :

- c:\codegen\cpp\library\Altova.vcxproj
- c:\codegen\cpp\library\AltovaXML\AltovaXML.vcxproj
- c:\codegen\cpp\library\Library.vcxproj

Tout d'abord, ouvrir la solution **c:\codegen\cpp\library\Library.sln** générée et l'intégrer dans Visual Studio.

Ensuite ouvrir votre solution Visual Studio existante (dans Visual Studio 2010, dans cet exemple), cliquer avec la touche de droite, choisir **Add | Existing Project**, et ajouter les fichiers de projet recensés ci-dessus, un par un. Soyez patient tant que Visual Studio parse les fichiers. Ensuite cliquer avec la touche de droite sur votre projet et choisir **Propriétés**. Dans le dialogue Pages de propriété, sélectionner **Common Properties | Framework and References**, puis cliquer sur **Add New Reference**. Ensuite, sélectionner et ajouter chacun des projets suivants : *Altova*, *AltovaXML* et *Library*.

© 2016-2022 Altova GmbH



Voir aussi la documentation MSDN pour utiliser les fonctions depuis une bibliothèque personnalisée, comme applicables à votre version de Visual Studio, par exemple :

- Si vous avez choisi de générer des bibliothèques statiques, voir https://msdn.microsoft.com/en-us/library/ms235627(v=vs.100).aspx
- Si vous avez choisi de générer des bibliothèques dynamique, voir https://msdn.microsoft.com/en-us/library/ms235636(v=vs.100).aspx

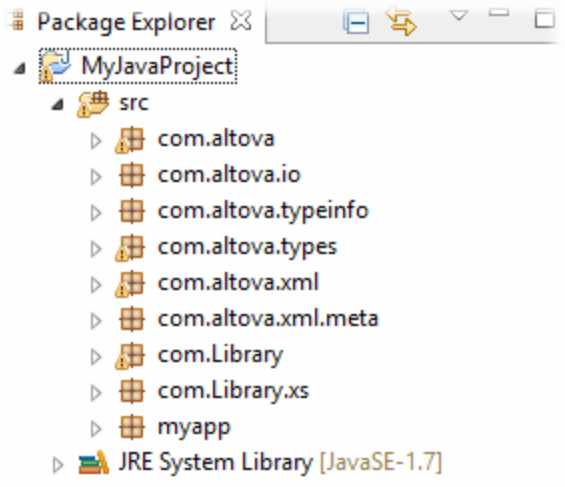
L'option de générer des bibliothèques statiques ou dynamiques est disponible dans les options de la génération de code (voir Options de Générateur de Code).

Java

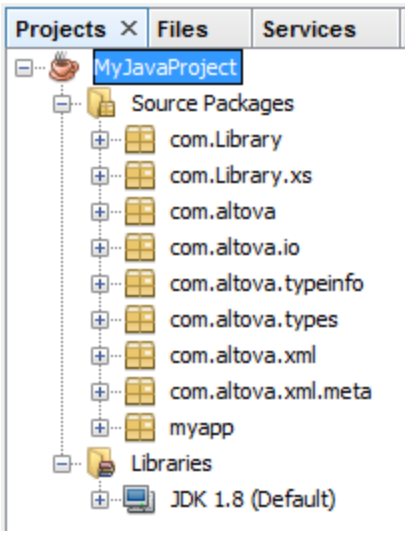
Un des moyens d'intégrer les packages Altova dans vos projets Java est de copier le répertoire **com** du code généré vers le répertoire qui stocke les packages de source de votre projet Java (par exemple, **C:\Workspace\MyJavaProject\src**). Par exemple, nous partons du principe de vous avez généré du code dans **c:\codegen\java\library**. Les classes Altova générées dans ce cas sont disponibles sous **c:\codegen\java\library\com**.

Une fois avoir copié les bibliothèques, actualiser le projet. Pour actualiser le projet dans Eclipse, le sélectionner dans le Package Explorer, et appuyer sur **F5**. Pour actualiser le projet dans NetBeans IDE 8.0, sélectionner la commande de menu **Source | Scan for External Changes**.

Une fois que vous avez effectué l'opération copy, les packages Altova sont disponibles dans le Package Explorer (dans le cas d'Eclipse), ou sous "Source Packages" dans le volet Projets (dans le cas de NetBeans IDE).



Packages Altova dans Eclipse 4.4



Packages Altova dans NetBeans IDE 8.0.2

27.2.5 Exemple: Bibliothèques de livres

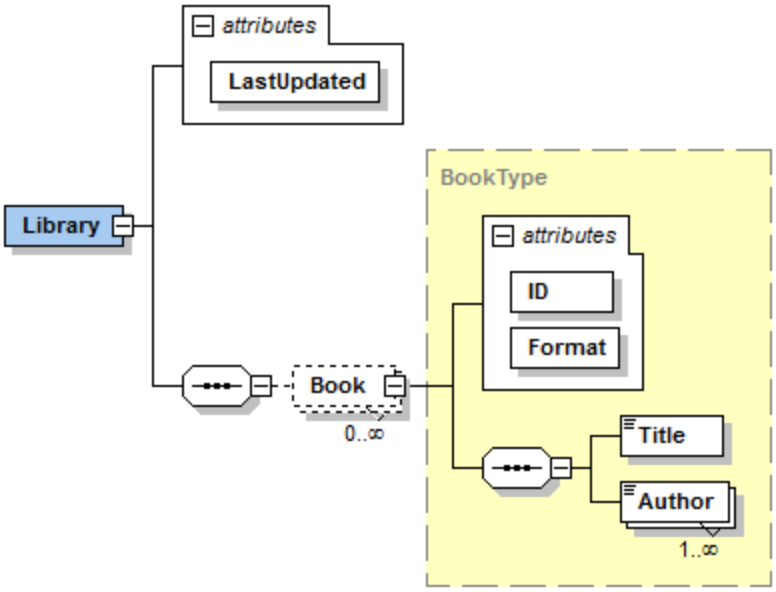
Cet exemple illustre comment utiliser les bibliothèques de schema wrapper générées afin d'écrire ou de lire des documents XML conforme en terme de programmation au schéma. Avant d'utiliser le code d'échantillon, veuillez prendre le temps de comprendre la structure du schéma d'exemple inclus.

Le schéma utilisé dans cet exemple décrit une bibliothèque de livres. La définition complète du schéma est affichée ci-dessous. Enregistrer cet extrait de code en tant que **Library.xsd** si vous souhaitez obtenir les mêmes résultats que dans cet exemple. Vous aurez besoin de ce schéma pour générer les bibliothèques de code utilisées dans cet exemple.

 © 2016-2022 Altova GmbH

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.nanonull.com/LibrarySample"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.nanonull.com/LibrarySample" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="Library">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Book" type="BookType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="LastUpdated" type="xs:dateTime"/>
      </xs:complexType>
   </xs:element>
   <xs:complexType name="BookType">
      <xs:sequence>
         <xs:element name="Title" type="xs:string"/>
         <xs:element name="Author" type="xs:string" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
      <xs:attribute name="Format" type="BookFormatType" use="required"/>
   </xs:complexType>
   <xs:complexType name="DictionaryType">
      <xs:complexContent>
         <xs:extension base="BookType">
            <xs:sequence>
               <xs:element name="FromLang" type="xs:string"/>
               <xs:element name="ToLang" type="xs:string"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:simpleType name="BookFormatType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Hardcover"/>
         <xs:enumeration value="Paperback"/>
         <xs:enumeration value="Audiobook"/>
         <xs:enumeration value="E-book"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

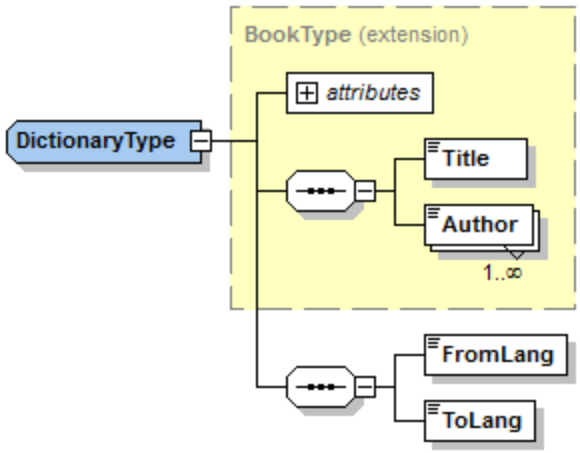
Library est un élément racide d'un `complexType` qui peut être représenté graphiquement comme suit dans le mode Schéma de XMLSpy:



Comme indiqué ci-dessus, la bibliothèque contient un attribut **LastUpdated** (défini en tant que `xs:dateTime`), et stocke une séquence de livres. Chaque livre est un `xs:complexType` et possède deux attributs : un **ID** (défini en tant que `xs:integer`), et un **Format**. Le format de chaque livre peut être hardcover, paperback, audiobook, ou e-book. Dans le schéma, **Format** est défini en tant que `xs:simpleType` qui utilise une énumération des valeurs mentionnées ci-dessus.

Chaque livre présente aussi un élément **Title** (défini en tant que `xs:string`), et un ou plusieurs **Author** éléments (définis en tant que `xs:string`).

La bibliothèque peut aussi contenir des livres qui sont des dictionnaires. Dictionaries ont le type `DictionaryType`, qui est dérivé par extension depuis `BookType`. En d'autres termes, un dictionnaire hérite de tous les attributs et éléments d'un Book (livre), plus deux éléments supplémentaires : **FromLang** et **ToLang**, comme illustré ci-dessous.



 © 2016-2022 Altova GmbH

Les éléments **FromLang** et **ToLang** stockent le langage de source et de destination du dictionnaire.

Un fichier d'instance XML valide conformément au schéma ci-dessus peut donc ressembler à l'exemple affiché dans l'exemple ci-dessous (s'il se trouve dans le même répertoire que le fichier de schéma) :

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
 <Title>English-German Dictionary</Title>
 <Author>John Doe</Author>
 <FromLang>English</FromLang>
 <ToLang>German</ToLang>
 </Book>
</Library>
```

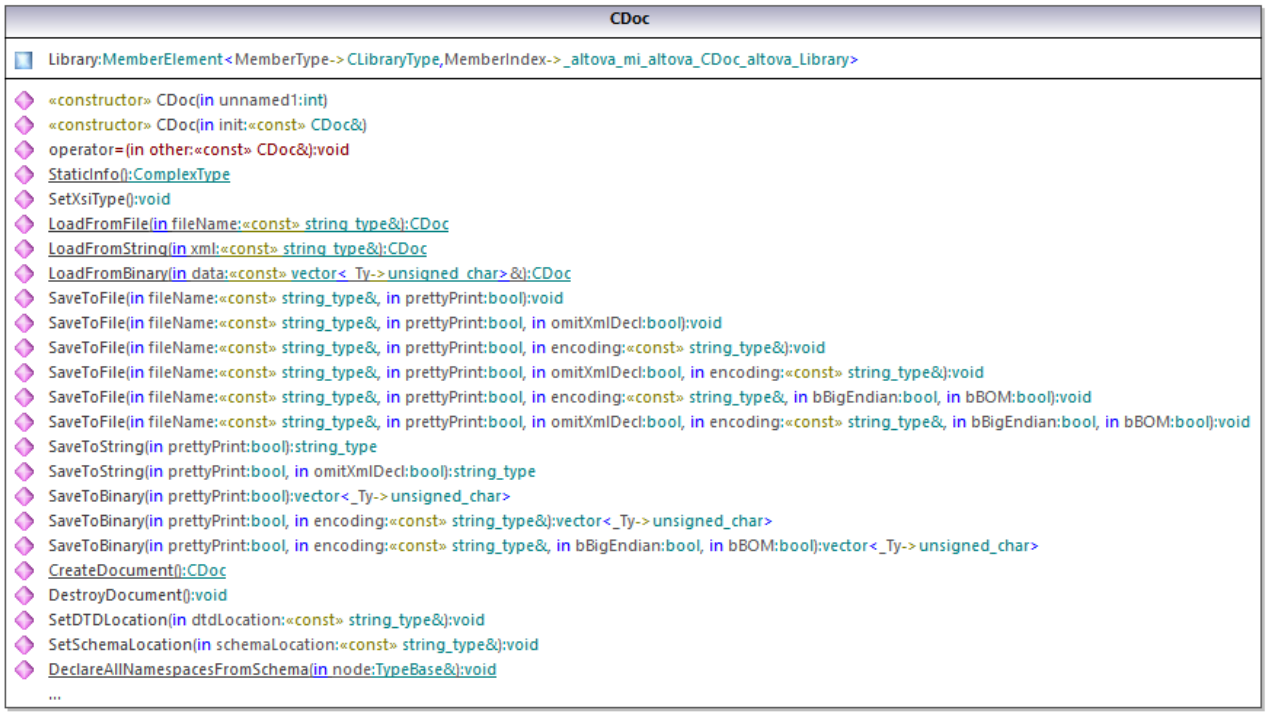
La rubrique suivante illustre comment lire ou écrire dans un tel fichier par le biais d'un programme. Pour commencer, générer le code de schema wrapper depuis le schéma ci-dessus, en utilisant les étapes décrites dans Générer du code depuis des Schémas XML ou des DTD.

27.2.5.1 Lire et écrire des documents XML (C++)

Après avoir généré du code depuis le schéma d'exemple, une application C++ test est créée, avec plusieurs bibliothèques Altova de support.

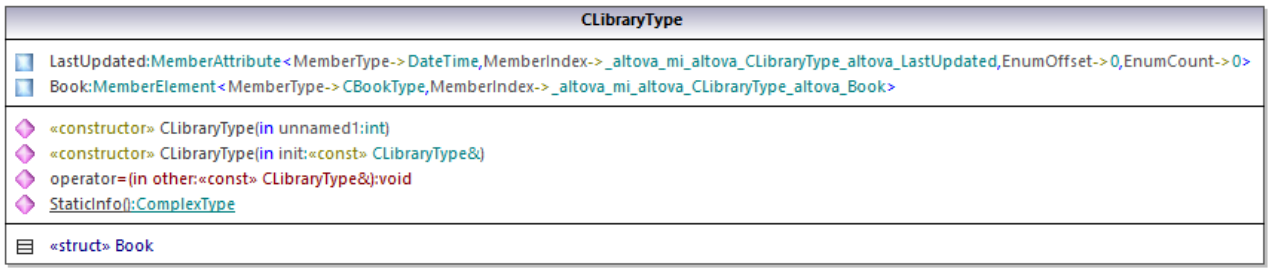
À propos des bibliothèques C++ générées

La classe centrale du code généré est la classe `CDoc` qui représente le document XML. Une telle classe est générée pour tous les schémas et son nom dépend du nom de fichier de schéma. Comme indiqué dans le diagramme, cette classe fournit des méthodes pour charger les documents depuis des fichiers, des streams binaires ou des strings (ou pour enregistrer des documents dans des fichiers, des streams, des strings). Pour consulter une liste de tous les membres exposés par cette classe, voir la référence de classe (`[YourSchema]::[CDoc]`).



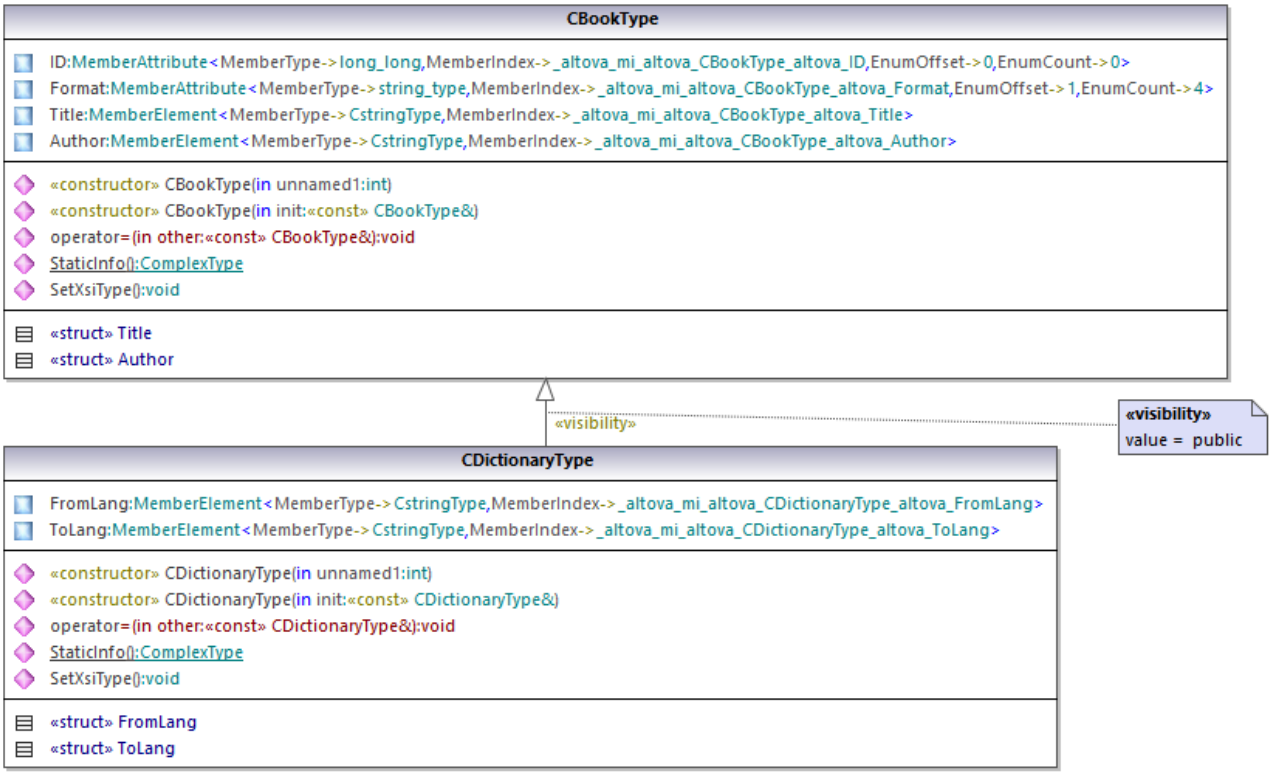
Le champ `Library` de la classe `CDoc` représente la racine réelle du document. **Library** est un élément dans le fichier XML, donc dans le code C++ il a une classe de modèle en tant que type (`MemberElement`). Le modèle de classe expose des méthodes et des propriétés pour interagir avec l'élément **Library** . En général, chaque attribut et chaque élément d'un type dans le schéma est saisi dans le code généré avec les classes de modèle `MemberAttribute` et `MemberElement`, respectivement. Pour plus d'informations, voir la référence de classe `[YourSchema]::MemberAttribute` et `[YourSchema]::MemberElement` .

La classe `CLibraryType` est générée depuis le type complexe **LibraryType** dans le schéma. Veuillez noter que la classe `CLibraryType` contient deux champs : `Book` et `LastUpdated`. Conformément à la logique déjà mentionnée ci-dessus, ces champs correspondent à l'élément **Book** et à l'attribut **LastUpdated** dans le schéma, et vous permettent de manipuler par le biais d'un programme (apposer, supprimer, etc.) des éléments et des attributs dans le document d'instance XML.



Puisque le `DictionaryType` est un type complexe dérivé depuis **BookType** dans le schéma, cette relation se retrouve aussi dans les classes générées. Comme illustré dans le diagramme, la classe `CDictionaryType` hérite la classe `CBookType`.

 © 2016-2022 Altova GmbH



Si votre schéma XML définit des types simples en tant qu'énumérations, les valeurs énumérées deviennent disponibles en tant que valeurs `enum` dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un `enum` membre de la classe `CBookFormatType`.

Écrire un document XML

1. Ouvrir la solution **LibraryTest.sln** dans Visual Studio générée depuis le schéma Library mentionné plus tôt dans cet exemple.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir Intégrer des Bibliothèques Schema Wrapper).

2. Dans Solution Explorer, ouvrir le fichier **LibraryTest.cpp**, et éditer la méthode `Example()` comme indiqué ci-dessous.

```
#include <ctime> // required to get current time
using namespace Doc; // required to work with Altova libraries
```

```
void Example()
{
   // Create a new, empty XML document
   CDoc libDoc = CDoc::CreateDocument();

   // Create the root element <Library> and add it to the document
   CLibraryType lib = libDoc.Library.append();

   // Get current time and set the "LastUpdated" attribute using Altova classes
   time_t t = time(NULL);
   struct tm * now = localtime( & t );
   altova::DateTime dt = altova::DateTime(now->tm_year + 1900, now->tm_mon + 1, now-
>tm_mday, now->tm_hour, now->tm_min, now->tm_sec);
   lib.LastUpdated = dt;

   // Create a new <Book> and add it to the library
   CBookType book = lib.Book.append();

   // Set the "ID" attribute of the book
   book.ID = 1;

   // Set the "Format" attribute of the <Book> using an enumeration constant
   book.Format.SetEnumerationValue( CBookFormatType::k_Paperback );

   // Add the <Title> and <Author> elements, and set values
   book.Title.append() = _T("The XML Spy Handbook");
   book.Author.append() = _T("Altova");

   // Append a dictionary (book of derived type) and populate its attributes and elements
   CDictionaryType dictionary = CDictionaryType(lib.Book.append().GetNode());
   dictionary.ID = 2;
   dictionary.Format.SetEnumerationValue( CBookFormatType::k_E_book);
   dictionary.Title.append() = _T("English-German Dictionary");
   dictionary.Author.append() = _T("John Doe");
   dictionary.FromLang.append() = _T("English");
   dictionary.ToLang.append() = _T("German");

   // Since dictionary a derived type, set the xsi:type attribute of the book element
   dictionary.SetXsiType();

   // Optionally, set the schema location
   libDoc.SetSchemaLocation(_T("Library.xsd"));

   // Save the XML document to a file with default encoding (UTF-8),
   // "true" causes the file to be pretty-printed.
   libDoc.SaveToFile(_T("GeneratedLibrary.xml"), true);

   // Destroy the document
   libDoc.DestroyDocument();
}
```

3. Appuyer sur **F5** pour lancer le débogage. Si le code a été exécuté avec succès, un fichier **GeneratedLibrary.xml** est créé dans le répertoire de sortie de solution.

© 2016-2022 Altova GmbH

Lire un document XML

1. Ouvrir la solution **LibraryTest.sln** dans Visual Studio.
2. Enregistrer le code ci-dessous en tant que **Library1.xml** dans un répertoire qui peut être lu par le code de programme (par exemple, le même répertoire que **LibraryTest.sln**).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
 <Title>English-German Dictionary</Title>
 <Author>John Doe</Author>
 <FromLang>English</FromLang>
 <ToLang>German</ToLang>
 </Book>
</Library>
```

3. Dans Solution Explorer, ouvrir le fichier **LibraryTest.cpp**, et éditer la méthode `Example()` comme indiqué ci-dessous.

```
using namespace Doc;
void Example()
{
   // Load XML document
   CDoc libDoc = CDoc::LoadFromFile(_T("Library1.xml"));

   // Get the first (and only) root element <Library>
   CLibraryType lib = libDoc.Library.first();

   // Check whether an element exists:
   if (!lib.Book.exists())
   {
      tcout << "This library is empty." << std::endl;
      return;
   }

   // iteration: for each <Book>...
   for (Iterator<CBookType> itBook = lib.Book.all(); itBook; ++itBook)
   {
      // output values of ISBN attribute and (first and only) title element
      tcout << "ID: " << itBook->ID << std::endl;
      tcout << "Title: " << tstring(itBook->Title.first()) << std::endl;

      // read and compare an enumeration value
```

```
      if (itBook->Format.GetEnumerationValue() == CBookFormatType::k_Paperback)
         tcout << "This is a paperback book." << std::endl;

      // for each <Author>...
      for (CBookType::Author::iterator itAuthor = itBook->Author.all(); itAuthor; +
+itAuthor)
         tcout << "Author: " << tstring(itAuthor) << std::endl;

      // alternative: use count and index
      for (unsigned int j = 0; j < itBook->Author.count(); ++j)
         tcout << "Author: " << tstring(itBook->Author[j]) << std::endl;
   }

   // Destroy the document
   libDoc.DestroyDocument();
}
```

4. Press **F5** to start debugging.

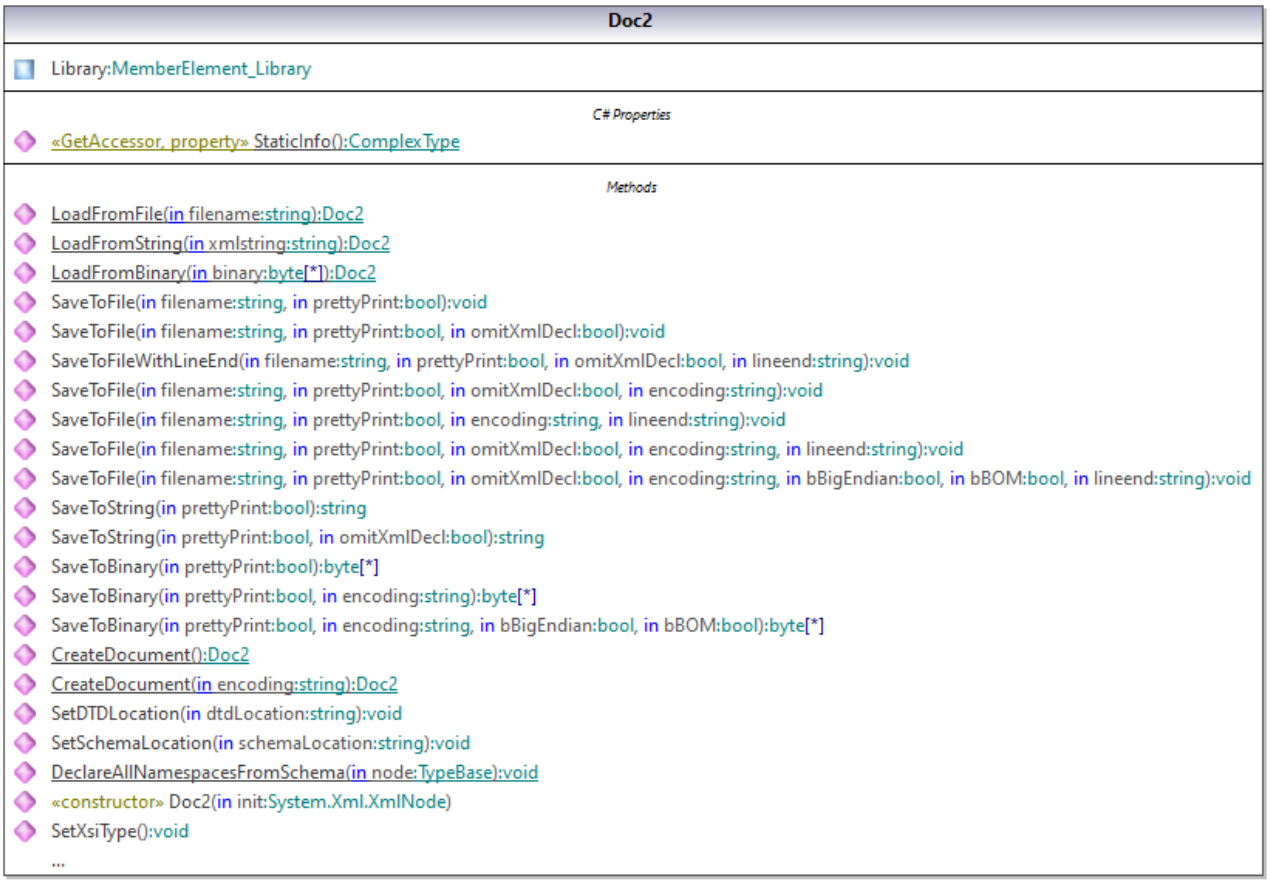
27.2.5.2 Lire et écrire des documents XML (C#)

Après avoir généré du code depuis le schéma de bibliothèque (voir Schéma d'exemple), une application C# test est créée, avec plusieurs bibliothèques Altova.

À propos des bibliothèques C# générées

La classe centrale du code généré est la classe `Doc2`qui représente le document XML. Une telle classe est générée pour tous les schémas et son nom dépend du nom de fichier de schéma. Veuillez noter que cette classe est appelée `Doc2` pour éviter un conflit possible avec le nom d'espace de noms. Comme indiqué dans le diagramme, cette classe fournit des méthodes pour charger les documents depuis des fichiers, des streams binaires ou des strings (ou pour enregistrer des documents dans des fichiers, des streams, des strings). Pour consulter une description de cette classe, voir la référence de classe ([YourSchema].[Doc]).

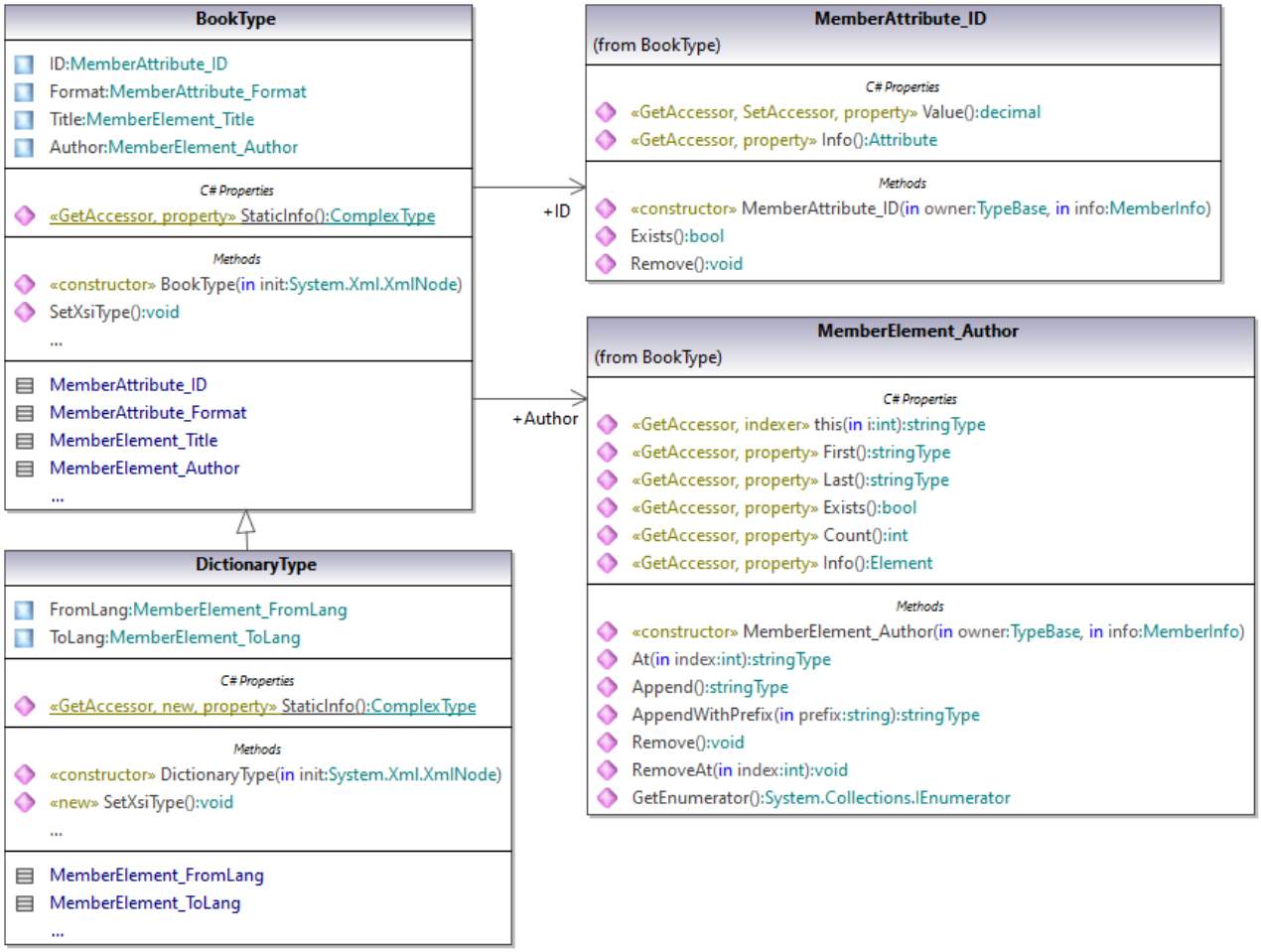
© 2016-2022 Altova GmbH



Le membre `Library` de la classe `Doc2` représente la véritable racine du document.

Conformément aux règles de génération de code mentionnées dans À propos des Schema Wrapper Libraries (C#), les membres de classes sont générés pour chaque attribut et pour chaque élément d'un type. Dans le code généré, le nom de ce type de classes de membre est préfixé avec `MemberAttribute_` et `MemberElement_`, respectivement. Des exemples de telles classes sont `MemberAttribute_ID` et `MemberElement_Author`, générés depuis l'élément **Author** et l'attribut **ID** d'un livre, respectivement (dans le diagramme ci-dessous, ils s'agit de classes imbriquées sous `BookType`). Ces classes vous permettent de manipuler à l'aide d'un programme les éléments et les attributs correspondants dans l'instance du document XML (par exemple, apposer, supprimer, définir une valeur, etc). Pour plus d'informations, voir la référence de classe `[YourSchemaType].MemberAttribute` et `[YourSchemaType].MemberElement` .

Puisque le **DictionaryType** est un type complexe dérivé depuis **BookType** dans le schéma, cette relation se retrouve aussi dans les classes générées. Comme illustré dans le diagramme, la classe `DictionaryType` hérite la classe `BookType`.



Si votre schéma XML définit des types simples en tant qu'énumérations, les valeurs énumérées deviennent disponibles en tant que valeurs `Enum` dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un `Enum` membre de la classe BookFormatType.

Écrire un document XML

1. Ouvrir la solution **LibraryTest.sln** dans Visual Studio généré depuis le schéma Library mentionné plus tôt dans cet exemple.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir Intégrer des Bibliothèques Schema Wrapper).

© 2016-2022 Altova GmbH

2. Dans Solution Explorer, ouvrir le fichier **LibraryTest.cs**, et éditer la méthode `Example()` comme indiqué ci-dessous.

```
protected static void Example()
{
    // Create a new XML document
    Doc2 doc = Doc2.CreateDocument();
    // Append the root element
    LibraryType root = doc.Library.Append();

    // Create the generation date using Altova DateTime class
    Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
    // Append the date to the root
    root.LastUpdated.Value = dt;

    // Add a new book
    BookType book = root.Book.Append();
    // Set the value of the ID attribute
    book.ID.Value = 1;
    // Set the format of the book (enumeration)
    book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
    // Set the Title and Author elements
    book.Title.Append().Value = "The XMLSpy Handbook";
    book.Author.Append().Value = "Altova";

    // Append a dictionary (book of derived type) and populate its attributes and
elements
    DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
    dictionary.ID.Value = 2;
    dictionary.Title.Append().Value = "English-German Dictionary";
    dictionary.Format.EnumerationValue = BookFormatType.EnumValues.eE_book;
    dictionary.Author.Append().Value = "John Doe";
    dictionary.FromLang.Append().Value = "English";
    dictionary.ToLang.Append().Value = "German";
    // Since it's a derived type, make sure to set the xsi:type attribute of the
book element
    dictionary.SetXsiType();

    // Optionally, set the schema location (adjust the path if
    // your schema is not in the same folder as the generated instance file)
    doc.SetSchemaLocation("Library.xsd");

    // Save the XML document with the "pretty print" option enabled
    doc.SaveToFile("GeneratedLibrary.xml", true);
}
```

3. Appuyer sur **F5** pour lancer le débogage. Si le code a été exécuté avec succès, un fichier **GeneratedLibrary.xml** est créé dans le répertoire de sortie de solution (généralement, **bin/Debug**).

Lire un document XML

1. Ouvrir la solution **LibraryTest.sln** dans Visual Studio.

2. Enregistrer le code-ci-dessous en tant que **Library.xml** dans le répertoire de sortie du projet (par défaut, **bin/Debug**). C'est le fichier qui sera lu par le code du programme.

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
 <Title>English-German Dictionary</Title>
 <Author>John Doe</Author>
 <FromLang>English</FromLang>
 <ToLang>German</ToLang>
 </Book>
</Library>
```

3. Dans Solution Explorer, ouvrir le fichier **LibraryTest.cs**, et éditer la méthode `Example()` comme indiqué ci-dessous.

```
protected static void Example()
{
    // Load the XML file
    Doc2 doc = Doc2.LoadFromFile("Library.xml");
    // Get the root element
    LibraryType root = doc.Library.First;

    // Read the library generation date
    Altova.Types.DateTime dt = root.LastUpdated.Value;
    string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
    Console.WriteLine("The library generation date is: " + dt_as_string);

    // Iteration: for each <Book>...
    foreach (BookType book in root.Book)
    {
      // Output values of ID attribute and (first and only) title element
      Console.WriteLine("ID:    " + book.ID.Value);
      Console.WriteLine("Title: " + book.Title.First.Value);

      // Read and compare an enumeration value
      if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
         Console.WriteLine("This is a paperback book.");

      // Iteration: for each <Author>
      foreach (xs.stringType author in book.Author)
         Console.WriteLine("Author: " + author.Value);

      // Determine if this book is of derived type
```


© 2016-2022 Altova GmbH

```
      if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
         // Find the value of the xsi:type attribute
         string xsiTypeValue =
book.Node.Attributes.GetNamedItem("xsi:type").Value;
         // Get the namespace URI and the lookup prefix of this namespace
         string namespaceUri = book.Node.NamespaceURI;
         string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);

         // if this book has DictionaryType
         if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
         {
            // output additional fields
            DictionaryType dictionary = new DictionaryType(book.Node);
            Console.WriteLine("Language from: " +
dictionary.FromLang.First.Value);
            Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
         }
         else
         {
            throw new Exception("Unexpected book type");
         }
      }
   }

   Console.ReadLine();
}
```

4. Appuyer sur **F5** pour lancer le débogage. Si le code a été exécuté avec succès, **Library.xml** sera lu par le code de programme, et ses contenus seront affichés en tant que sortie de console.

Lire et écrire des éléments et des attributs

Les valeurs des éléments et des attributs peut être accédé en utilisant la propriété `Value` de la classe d'élément ou d'attribut du membre généré, par exemple :

```
// Output values of ID attribute and (first and only) title element
Console.WriteLine("ID:    " + book.ID.Value);
Console.WriteLine("Title: " + book.Title.First.Value);
```

Pour obtenir la valeur de l'élément **Title** dans cet exemple particulier, nous avons aussi utilisé la méthode `First()`, étant donné que c'est le premier (et seul) élément **Title** d'un livre. Dans les cas où vous devez choisir un élément spécifique depuis une liste par index, utiliser la méthode `At()`.

La classe générée pour chaque élément de membre d'un type met en place l'interface standard `System.Collections.IEnumerable`. Cela permet de boucler dans plusieurs éléments du même type. Dans cet exemple particulier, vous pouvez boucler à travers tous les livres d'un objet Bibliothèque comme suit :

```
// Iteration: for each <Book>...
foreach (BookType book in root.Book)
```

```
{
    // your code here...
}
```

Pour ajouter un nouvel élément, utiliser la méthode `Append()`. Par exemple, le code suivant appose l'élément racine au document :

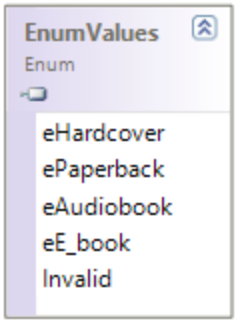
```
// Append the root element to the library
LibraryType root = doc.Library.Append();
```

Vous pouvez définir la valeur d'un attribut (comme ID dans cet exemple) comme suit :

```
// Set the value of the ID attribute
book.ID.Value = 1;
```

Lire et écrire des valeurs d'énumération

Si votre schéma XML définit des types simples en tant qu'énumérations, les valeurs énumérées deviennent disponibles en tant que valeurs `Enum` dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un `Enum`.



Pour attribuer des valeurs d'énumération à un objet, utiliser du code comme celui ci-dessous :

```
// Set the format of the book (enumeration)
book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
```

Vous pouvez lire ces valeurs d'énumération provenant de documents d'instance XML comme suit :

```
// Read and compare an enumeration value
if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
Console.WriteLine("This is a paperback book.");
```

Si une condition "if" n'est pas suffisante, créer un interrupteur pour déterminer chaque valeur d'énumération et traiter comme requis.

© 2016-2022 Altova GmbH

Travailler avec des types xs:dateTime et xs:duration

Si le schéma à partir duquel vous avez généré du code utilise les types heure et durée comme `xs:dateTime`, ou `xs:duration`, ils sont convertis en classes natives Altova dans un code généré. C'est pourquoi, pour écrire une valeur date ou durée vers le document XML, procéder comme suit :

1. Construit un objet `Altova.Types.DateTime` ou `Altova.Types.Duration` (soit depuis `System.DateTime`, ou en utilisant des parties comme des heures et des minutes, voir `Altova.Types.DateTime` et `Altova.Types.Duration` pour plus d'informations).
2. Définir l'objet en tant que la valeur de l'élément ou de l'attribut requis, par exemple :

```
// Create the library generation date using Altova DateTime class
Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
// Append the date to the root
root.LastUpdated.Value = dt;
```

Pour lire une date ou une durée depuis un document XML, procéder comme suit :

1. Déclarer la valeur d'élément (ou d'attribut) comme objet `Altova.Types.DateTime` ou `Altova.Types.Duration` .
2. Formate l'élément ou l'attribut requis, par exemple :

```
// Read the library generation date
Altova.Types.DateTime dt = root.LastUpdated.Value;
string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
Console.WriteLine("The library generation date is: " + dt_as_string);
```

Pour plus d'informations, voir la référence de classe `Altova.Types.DateTime` et `Altova.Types.Duration` .

Travailler avec des types dérivés

Si votre schéma XML définit des types dérivés, vous pouvez préserver la dérivation de type dans des documents XML que vous créez ou chargez par le biais d'un programme. Prendre le schéma utilisé dans cet exemple, l'extrait de code suivant illustre comment créer un nouveau livre de type dérivé`DictionaryType`:

```
// Append a dictionary (book of derived type) and populate its attributes and elements
DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
dictionary.ID.Value = 2;
dictionary.Title.Append().Value = "English-German Dictionary";
dictionary.Author.Append().Value = "John Doe";
dictionary.FromLanguage.Append().Value = "English";
dictionary.ToLanguage.Append().Value = "German";

// Since it's a derived type, make sure to set the xsi:type attribute of the book element
dictionary.SetXsiType();
```

Veuillez noter qu'il est important de définir l'attribut `xsi:type` du livre récemment créé. Cela garantit que le type de livre sera interprété correctement par le schéma losrque le document XML est validé.

Lorsque vous chargez des données depuis un document XML, l'extrait de code suivant montre comment identifier un livre de type dérivé `DictionaryType` dans l'instance XML chargée. Premièrement, le code trouve la valeur de l'attribut `xsi:type` du nœud de livre. Si l'espace de noms URI de ce nœud est `http://www.nanonull.com/LibrarySample` et si le préfixe et type de lookup URI correspond à la valeur de l'attribut `xsi:type`, alors il s'agit d'un dictionnaire :

```
      // Determine if this book is of derived type
      if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
          // Find the value of the xsi:type attribute
          string xsiTypeValue = book.Node.Attributes.GetNamedItem("xsi:type").Value;
          // Get the namespace URI and the lookup prefix of this namespace
          string namespaceUri = book.Node.NamespaceURI;
          string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);

          // if this book has DictionaryType
          if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
          {
             // output additional fields
             DictionaryType dictionary = new DictionaryType(book.Node);
             Console.WriteLine("Language from: " + dictionary.FromLang.First.Value);
            Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
          }
          else
          {
              throw new Exception("Unexpected book type");
          }
      }
```

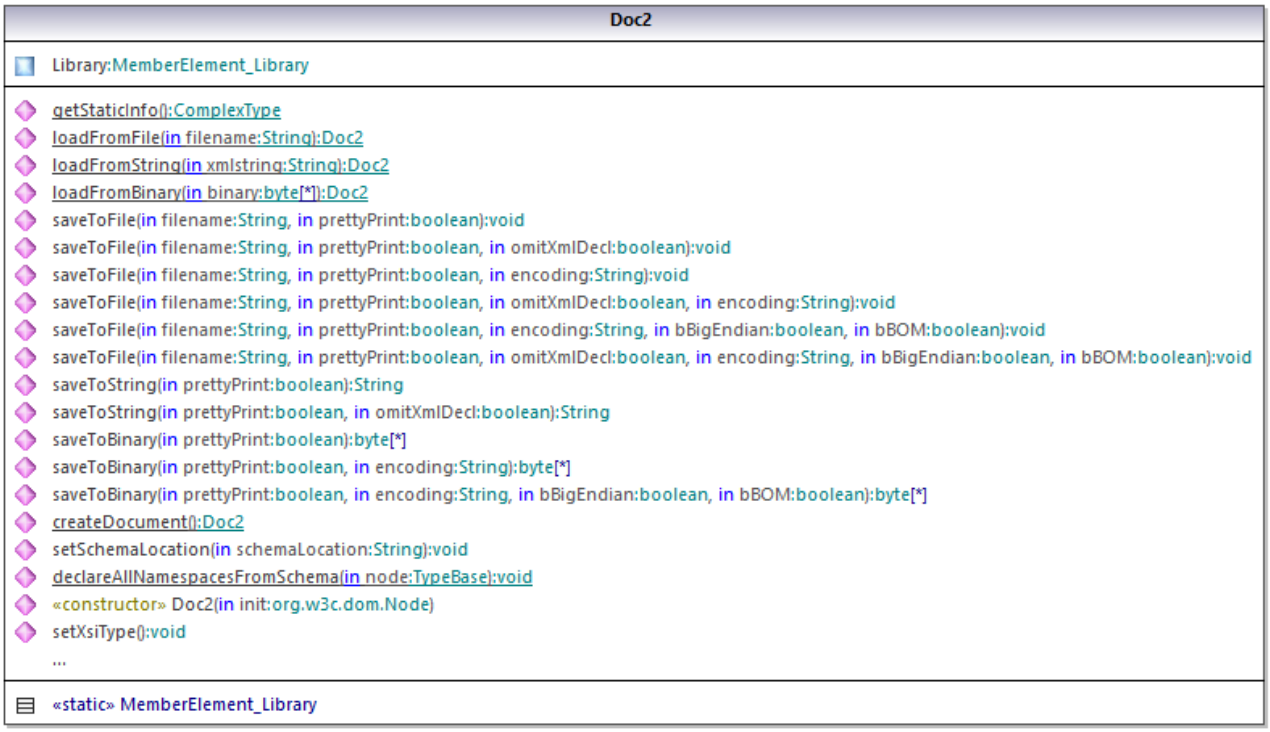
27.2.5.3 Lire et écrire des documents XML (Java)

Après avoir généré du code depuis le schéma de bibliothèque (voir Schéma d'exemple), un projet de text Java est créé, avec plusieurs bibliothèques Altova.

À propos des bibliothèques Java générées

La classe centrale du code généré est la classe `Doc2` qui représente le document XML. Une telle classe est générée pour tous les schémas et son nom dépend du nom de fichier de schéma. Veuillez noter que cette classe est appelée `Doc2` pour éviter un conflit possible avec le nom d'espace de noms. Comme indiqué dans le diagramme, cette classe fournit des méthodes pour charger les documents depuis des fichiers, des streams binaires ou des strings (ou pour enregistrer des documents dans des fichiers, des streams, des strings). Pour consulter une description de cette classe, voir la référence de classe `com.[YourSchema].[Doc]`.

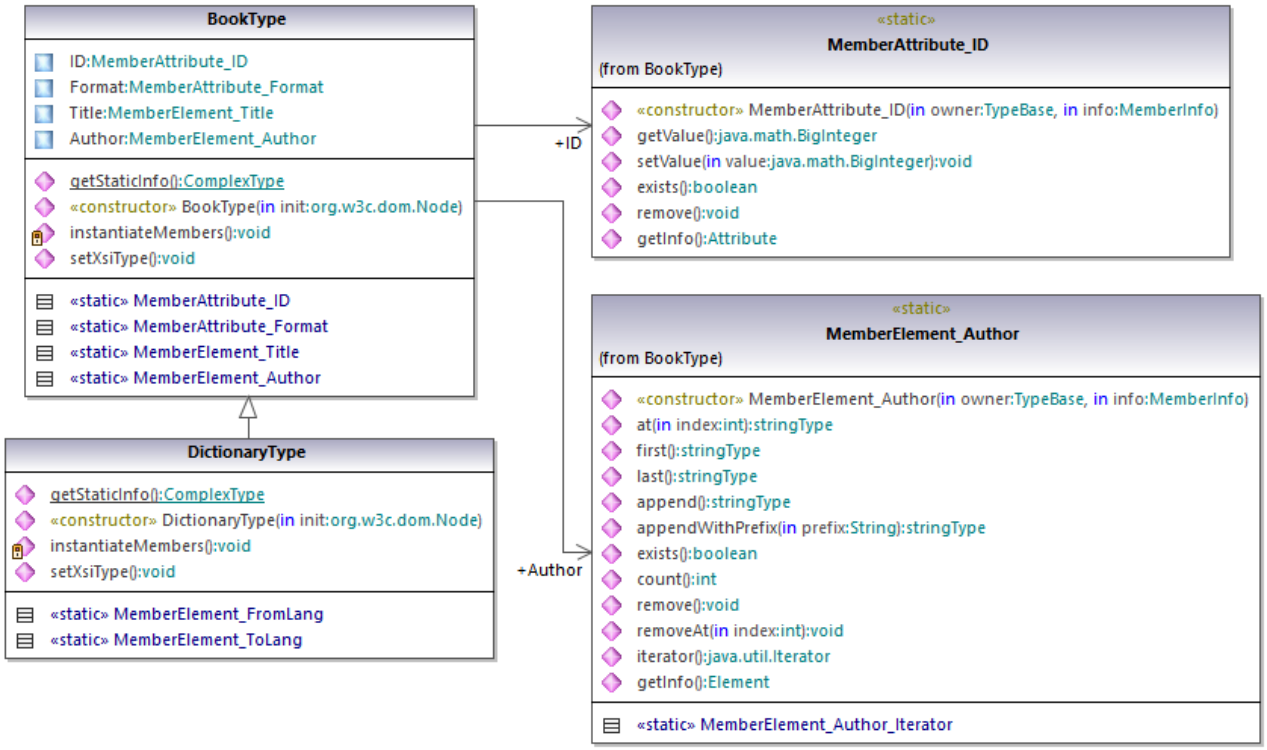
© 2016-2022 Altova GmbH



Le membre `Library` de la classe `Doc2` représente la véritable racine du document.

Conformément aux règles de génération de code mentionnées dans À propos du code Java généré, les membres de classes sont générés pour chaque attribut et pour chaque élément d'un type. Dans le code généré, le nom de ce type de classes de membre est préfixé avec `MemberAttribute_` et `MemberElement_`, respectivement. Dans le diagramme ci-dessus, des exemples de ce type de classes sont `MemberAttribute_ID` et `MemberElement_Author`, générés depuis l'élément **Author** et l'attribut **ID** d'un livre, respectivement. Ces classes vous permettent de manipuler à l'aide d'un programme les éléments et les attributs correspondants dans l'instance du document XML (par exemple, apposer, supprimer, définir une valeur, etc). Pour plus d'informations, voir la référence de classe `com.[YourSchema].[YourSchemaType].MemberAttribute` et `com.[YourSchema].[YourSchemaType].MemberElement` .

Puisque le **DictionaryType** est un type complexe dérivé depuis **BookType** dans le schéma, cette relation se retrouve aussi dans les classes générées. Comme illustré dans le diagramme, la classe `DictionaryType` hérite la classe `BookType`.



Si votre schéma XML définit des types simples en tant qu'énumerations, les valeurs énumérées deviennent disponibles en tant que valeurs `Enum` dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un `Enum` membre de la classe BookFormatType.

Écrire un document XML

1. Dans le menu **Fichier** d'Eclipse, cliquer sur **Import**, choisir **Existing Projects into Workspace**, et cliquer sur **Next**.
2. À côté de **Select root directory**, cliquer sur **Browse**, choisir le répertoire dans lequel vous souhaitez générer le code Java, et cliquer sur **Finish**.
3. Dans Eclipse Package Explorer, agrandir le package **com.LibraryTest** et ouvrir le fichier **LibraryTest.java**.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir Intégrer des Bibliothèques Schema Wrapper).

4. Éditer la méthode `Example()` comme indiqué ci-dessous.

 © 2016-2022 Altova GmbH

```
protected static void example() throws Exception {
      // create a new, empty XML document
      Doc2 libDoc = Doc2.createDocument();

      // create the root element <Library> and add it to the document
      LibraryType lib = libDoc.Library.append();

      // set the "LastUpdated" attribute
      com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
      lib.LastUpdated.setValue(dt);

      // create a new <Book> and populate its elements and attributes
      BookType book = lib.Book.append();
      book.ID.setValue(java.math.BigInteger.valueOf(1));
      book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
      book.Title.append().setValue("The XML Spy Handbook");
      book.Author.append().setValue("Altova");

      // create a dictionary (book of derived type)  and populate its elements and
attributes
      DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
      dict.ID.setValue(java.math.BigInteger.valueOf(2));
      dict.Title.append().setValue("English-German Dictionary");
      dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
      dict.Author.append().setValue("John Doe");
      dict.FromLang.append().setValue("English");
      dict.ToLang.append().setValue("German");
      dict.setXsiType();

      // set the schema location (this is optional)
      libDoc.setSchemaLocation("Library.xsd");

      // save the XML document to a file with default encoding (UTF-8). "true" causes the
file to be pretty-printed.
      libDoc.saveToFile("Library1.xml", true);
 }
```

5. Construire le projet Java et l'exécuter. Si le code a été exécuté avec succès, un fichier **Library1.xml** est créé dans le répertoire de projet.

Lire un document XML

1. Dans le menu **Fichier** d'Eclipse, cliquer sur **Import**, choisir **Existing Projects into Workspace**, et cliquer sur **Next**.
2. À côté de **Select root directory**, cliquer sur **Browse**, choisir le répertoire dans lequel vous souhaitez générer le code Java, et cliquer sur **Finish**.
3. Enregistrer le code ci-dessous en tant que **Library1.xml** dans un répertoire local (vous devrez vous référer au chemin du fichier **Library1.xml** depuis l'échantillon de code ci-dessous).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
```

```
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
 <Title>English-German Dictionary</Title>
 <Author>John Doe</Author>
 <FromLang>English</FromLang>
 <ToLang>German</ToLang>
 </Book>
</Library>
```

4. Dans Eclipse Package Explorer, agrandir le package **com.LibraryTest** et ouvrir le fichier **LibraryTest.java**.
5. Éditer la méthode `Example()` comme indiqué ci-dessous.

```
protected static void example() throws Exception {
     // load XML document from a path, make sure to adjust the path as necessary
     Doc2 libDoc = Doc2.loadFromFile("Library1.xml");

     // get the first (and only) root element <Library>
     LibraryType lib = libDoc.Library.first();

     // check whether an element exists:
     if (!lib.Book.exists()) {
        System.out.println("This library is empty.");
        return;
     }

     // read a DateTime schema type
     com.altova.types.DateTime dt = lib.LastUpdated.getValue();
     System.out.println("The library was last updated on: " + dt.toDateString());

     // iteration: for each <Book>...
     for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
        BookType book = (BookType) itBook.next();
        // output values of ID attribute and (first and only) title element
        System.out.println("ID: " + book.ID.getValue());
        System.out.println("Title: " + book.Title.first().getValue());

        // read and compare an enumeration value
        if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
           System.out.println("This is a paperback book.");

        // for each <Author>...
        for (java.util.Iterator itAuthor = book.Author.iterator(); itAuthor
              .hasNext();)
           System.out.println("Author: " + ((com.Doc.xs.stringType)
itAuthor.next()).getValue());
```


© 2016-2022 Altova GmbH

```
        // find the derived type of this book
        // by looking at the value of the xsi:type attribute, using DOM
        org.w3c.dom.Node bookNode = book.getNode();
        if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
           // Get the value of the xsi:type attribute
           String xsiTypeValue =
bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();

           // Get the namespace URI and lookup prefix of this namespace
           String namespaceUri = bookNode.getNamespaceURI();
           String lookupPrefix = bookNode.lookupPrefix(namespaceUri);

           // If xsi:type matches the namespace URI and type of the book node
           if (namespaceUri == "http://www.nanonull.com/LibrarySample"
                 && (  xsiTypeValue.equals(lookupPrefix + ":DictionaryType" )))    {
              // ...then this is a book of derived type (dictionary)
              DictionaryType dictionary = new DictionaryType(   book.getNode());
              // output the value of the "FromLang" and "ToLang" elements
              System.out.println("From language: " +
dictionary.FromLang.first().getValue());
              System.out.println("To language: " + dictionary.ToLang.first().getValue());
           }
           else
           {
              // throw an error
              throw new java.lang.Error("This book has an unknown type.");
           }
        }
     }
   }
```

6. Construire le projet Java et l'exécuter. Si le code est exécuté avec succès, **Library1.xml** sera lu par le code de programme, et ses contenus seront affichés dans le mode Console.

Lire et écrire des éléments et des attributs

Les valeurs des éléments et des attributs peut être accédé en utilisant la méthode `getValue()` de la classe d'élément ou d'attribut du membre généré, par exemple :

```
// output values of ID attribute and (first and only) title element
System.out.println("ID: " + book.ID.getValue());
System.out.println("Title: " + book.Title.first().getValue());
```

Pour obtenir la valeur de l'élément **Title** dans cet exemple particulier, nous avons aussi utilisé la méthode `first()`, étant donné que c'est le premier (et seul) élément **Title** d'un livre. Dans les cas où vous devez choisir un élément spécifique depuis une liste par index, utiliser la méthode `at()`.

Pour itérer dans plusieurs éléments, utiliser soit une itération basée sur l'index ou `java.util.Iterator`. Par exemple, vous pouvez itérer dans les livres d'une bibliothèque comme suit :

```
// index-based iteration
for (int j = 0; j < lib.Book.count(); ++j ) {
   // your code here
}

// alternative iteration using java.util.Iterator
for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
      // your code here
   }
```

Pour ajouter un nouvel élément, utiliser la méthode `Append()`. Par exemple, le code suivant appose un élément de racine vide**Library** dans le document :

```
// create the root element <Library> and add it to the document
LibraryType lib = libDoc.Library3.append();
```

Une fois qu'un élément est apposé, vous pouvez définir la valeur d'un de ses éléments ou un attribut en utilisant la méthode setValue().

```
// set the value of the Title element
book.Title.append().setValue("The XML Spy Handbook");
// set the value of the ID attribute
book.ID.setValue(java.math.BigInteger.valueOf(1));
```

Lire et écrire des valeurs d'énumération

Si votre schéma XML définit des types simples en tant qu'énumerations, les valeurs énumérées deviennent disponibles en tant que valeurs `Enum` dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un `Enum` (voir le diagramme de classe BookFormatType `ci-dessus)`. Pour attribuer des valeurs d'énumération à un objet, utiliser du code comme celui ci-dessous :

```
// set an enumeration value
book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
```

Vous pouvez lire ces valeurs d'énumération provenant de documents d'instance XML comme suit :

```
// read an enumeration value
if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
 System.out.println("This is a paperback book."
```

Si une condition "if" n'est pas suffisante, créer un interrupteur pour déterminer chaque valeur d'énumération et traiter comme requis.

Travailler avec des types xs:dateTime et xs:duration

Si le schéma à partir duquel vous avez généré du code utilise les types heure et durée comme `xs:dateTime`, ou `xs:duration`, ils sont convertis en classes natives Altova dans un code généré. C'est pourquoi, pour écrire une valeur date ou durée vers le document XML, procéder comme suit :

© 2016-2022 Altova GmbH

1. Construit un objet com.altova.types.DateTime ou com.altova.types.Duration .
2. Définir l'objet en tant que la valeur de l'élément ou de l'attribut requis, par exemple :

```
// set the value of an attribute of DateTime type
com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
lib.LastUpdated.setValue(dt);
```

Pour lire une date ou une durée depuis un document XML :

1. Déclarer la valeur d'élément (ou d'attribut) comme objet com.altova.types.DateTime ou com.altova.types.Duration .
2. Formate l'élément ou l'attribut requis, par exemple :

```
// read a DateTime type
com.altova.types.DateTime dt = lib.LastUpdated.getValue();
 System.out.println("The library was last updated on: " + dt.toDateString());
```

Pour plus d'informations, voir la référence de classe com.altova.types.DateTime et com.altova.types.Duration .

Travailler avec des types dérivés

Si votre schéma XML définit des types dérivés, vous pouvez préserver la dérivation de type dans des documents XML que vous créez ou chargez par le biais d'un programme. Prendre le schéma utilisé dans cet exemple, l'extrait de code suivant illustre comment créer un nouveau livre de type dérivéDictionaryType:

```
// create a dictionary (book of derived type) and populate its elements and attributes
DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
dict.ID.setValue(java.math.BigInteger.valueOf(2));
dict.Title.append().setValue("English-German Dictionary");
dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
dict.Author.append().setValue("John Doe");
dict.FromLang.append().setValue("English");
dict.ToLang.append().setValue("German");
dict.setXsiType();
```

Veuillez noter qu'il est important de définir l'attribut xsi:type du livre récemment créé. Cela garantit que le type de livre sera interprété correctement par le schéma losrque le document XML est validé.

Lorsque vous chargez des données depuis un document XML, l'extrait de code suivant montre comment identifier un livre de type dérivé DictionaryType dans l'instance XML chargée. Premièrement, le code trouve la valeur de l'attribut xsi:type du nœud de livre. Si l'espace de noms URI de ce nœud est http://www.nanonull.com/LibrarySample et si le préfixe et type de lookup URI correspond à la valeur de l'attribut xsi:type, alors il s'agit d'un dictionnaire :

```
        // find the derived type of this book
 // by looking at the value of the xsi:type attribute, using DOM
 org.w3c.dom.Node bookNode = book.getNode();
 if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
```

```
// Get the value of the xsi:type attribute
String xsiTypeValue = bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();

// Get the namespace URI and lookup prefix of the book node
String namespaceUri = bookNode.getNamespaceURI();
String lookupPrefix = bookNode.lookupPrefix(namespaceUri);

// If xsi:type matches the namespace URI and type of the book node
if (namespaceUri == "http://www.nanonull.com/LibrarySample"
&& ( xsiTypeValue.equals(lookupPrefix + ":DictionaryType" ))) {
// ...then this is a book of derived type (dictionary)
DictionaryType dictionary = new DictionaryType( book.getNode());
// output the value of the "FromLang" and "ToLang" elements
System.out.println("From language: " + dictionary.FromLang.first().getValue());
System.out.println("To language: " + dictionary.ToLang.first().getValue());
}
else
{
// throw an error
throw new java.lang.Error("This book has an unknown type.");
}
}
```

27.2.6 Exemple: Purchase Order

Cet exemple illustre comment travailler avec le code de programme généré depuis un schéma XML "main" qui importe d'autres schémas. Chacun des schémas importés a un espace de noms cible différent. L'objectif est de créer par programme un document XML dans lequel tous les éléments sont préfixés conformément à leur espace de noms. Plus précisément, le document XML créé depuis votre code C++, C# ou Java doit ressembler à l'exemple ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
<p:Purchase xsi:schemaLocation="http://NamespaceTest.com/Purchase Main.xsd"
            xmlns:p="http://NamespaceTest.com/Purchase"
            xmlns:o="http://NamespaceTest.com/OrderTypes"
            xmlns:c="http://NamespaceTest.com/CustomerTypes"
            xmlns:cmn="http://NamespaceTest.com/CommonTypes"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <p:OrderDetail>
      <o:Item>
         <o:ProductName>Lawnmower</o:ProductName>
         <o:Quantity>1</o:Quantity>
         <o:UnitPrice>148.42</o:UnitPrice>
      </o:Item>
   </p:OrderDetail>
   <p:PaymentMethod>VISA</p:PaymentMethod>
   <p:CustomerDetails>
      <c:Name>Alice Smith</c:Name>
      <c:DeliveryAddress>
         <cmn:Line1>123 Maple Street</cmn:Line1>
```

 © 2016-2022 Altova GmbH

```
            <cmn:Line2>Mill Valley</cmn:Line2>
        </c:DeliveryAddress>
        <c:BillingAddress>
            <cmn:Line1>8 Oak Avenue</cmn:Line1>
            <cmn:Line2>Old Town</cmn:Line2>
        </c:BillingAddress>
    </p:CustomerDetails>
</p:Purchase>
```

Le schéma principal utilisé dans ces exemple est appelé **Main.xsd**. Comme indiqué dans l'extrait de code ci-dessous, il importe trois autres schémas : **CommonTypes.xsd**, **CustomerTypes.xsd** et **OrderTypes.xsd**. Pour obtenir les mêmes résultats que dans cet exemple, enregistrer tous les extraits de code ci-dessous dans des fichiers et utiliser les mêmes noms de fichier que ci-dessus. Veuillez noter que le schéma mappe chacun des préfixes **ord**, **pur**, **cmn**, et **cust** vers des espaces de noms (Order types, Purchase types, Common types, et Customer types, respectivement). Cela signifie que, dans le code généré, les classes correspondant à Orders, Purchases, Customers, etc, seront disponibles sous leur espace de noms respectif.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/Purchase"
      xmlns:ord="http://NamespaceTest.com/OrderTypes"
      xmlns:pur="http://NamespaceTest.com/Purchase"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      xmlns:cust="http://NamespaceTest.com/CustomerTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:import schemaLocation="CustomerTypes.xsd"
namespace="http://NamespaceTest.com/CustomerTypes" />
    <xs:import schemaLocation="OrderTypes.xsd"
namespace="http://NamespaceTest.com/OrderTypes" />
    <xs:element name="Purchase">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="OrderDetail" type="ord:OrderType" />
                <xs:element name="PaymentMethod" type="cmn:PaymentMethodType" />
                <xs:element ref="pur:CustomerDetails" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="CustomerDetails" type="cust:CustomerType" />
</xs:schema>
```

Main.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
   <xs:complexType name="AddressType">
      <xs:sequence>
```

```
        <xs:element name="Line1" type="xs:string"/>
        <xs:element name="Line2" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="PriceType">
      <xs:restriction base="xs:decimal">
         <xs:fractionDigits value="2"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="PaymentMethodType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="VISA"/>
         <xs:enumeration value="MasterCard"/>
         <xs:enumeration value="Cash"/>
         <xs:enumeration value="AMEX"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

CommonTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/CustomerTypes"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:complexType name="CustomerType">
        <xs:sequence>
            <xs:element name="Name" type="xs:string" />
            <xs:element name="DeliveryAddress" type="cmn:AddressType" />
            <xs:element name="BillingAddress" type="cmn:AddressType" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

CustomerTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://NamespaceTest.com/OrderTypes"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:complexType name="OrderType">
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="Item">
                <xs:complexType>
                    <xs:sequence>
```


© 2016-2022 Altova GmbH

```
                        <xs:element name="ProductName" type="xs:string" />
                        <xs:element name="Quantity" type="xs:int" />
                        <xs:element name="UnitPrice" type="cmn:PriceType" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

OrderTypes.xsd

Pour terminer cet exemple, suivre les étapes suivantes :

1. Enregistrer tous les schémas depuis les extraits de code ci-dessus dans les fichiers sur le disque, en vous assurant que vous préservez les noms de fichier indiqués.
2. Générer le code de schema wrapper depuis le schéma **Main.xsd** ci-dessus, en utilisant les étapes décrites dans Générer du code depuis des Schémas XML ou des DTD. Une fois cette étape achevée , vous aurez généré un programme compilable dans le langage de votre choix (C++, C#, ou Java).
3. Ajouter du code dans votre programme C++, C# ou Java depuis un des extraits de code d'exemple, le cas échéant :

   - Espaces de noms XML et Préfixes (C++)
   - Espaces de noms XML et Préfixes (C#)
   - Espaces de noms XML et Préfixes (Java)

27.2.6.1 Espaces de noms XML et Préfixes (C++)

Après avoir généré du code depuis le schéma d'exemple, une application C++ test est créée, avec plusieurs bibliothèques Altova de support. Rappelez-vous que le schéma d'exemple (**Main.xsd**) a plusieurs déclarations d'espace de noms. Par conséquent, le code généré comprend des espaces de noms qui correspondent aux alias d'espace de noms (préfixes) provenant du schéma : `Main::ord`, `Main::pur`, `Main::cmn` et `Main::cust`.

En général, afin de contrôler les espaces de noms XML et les préfixes avec l'aide des bibliothèques de schéma, vous disposez des méthodes suivantes :

- DeclareAllNamespacesFromSchema(). Appeler cette méthode si vous souhaitez déclarer les mêmes espaces de noms dans votre instance XML que dans le schéma. Sinon, si vous souhaitez utiliser d'autres espaces de noms que ceux utilisés dans cet exemple, vous devriez utiliser `DeclareNamespace()`. La méthode `DeclareAllNamespacesFromSchema()` n'est pas utilisée dans cet exemple parce que nous souhaitons utiliser spécifiquement des éléments XML avec des préfixes qui sont légèrement différents de ceux déclarés dans le schéma.
- DeclareNamespace(). Appeler cette méthode pour créer ou contourner l'attribut de préfixe d'espace de noms existant sur un élément. L'élément doit déjà être créé en utilisant soit la méthode `append()` soit `appendWithPrefix()`, comme illustré ci-dessous.
- appendWithPrefix(). Utiliser cette méthode pour apposer un élément d'instance avec un préfixe spécifique. Pour créer l'instance XML illustrée dans cet exemple, il a suffit d'appeler cette méthode pour l'élément root uniquement. Tous les autres éléments ont été apposés en utilisant uniquement append(), et leurs préfixes ont été ajoutés automatiquement sur la base de leurs espaces de noms, conformément aux règles ci-dessus.

L'extrait de code ci-dessous vous montre comment créer un document XML avec plusieurs déclarations d'espace de noms et des noms d'élément préfixés. En particulier, il génère une instance Purchase Order comme illustré dans l'Example : Purchase Order. Plus important encore, à des fins d'illustration, certains préfixes sont contournés dans l'instance XML (c'est à dire qu'ils ne sont tout à fait identiques que ceux déclarés dans le schéma).

```
void Example()
{
   // Create the XML document and append the root element
   Main::pur::CMain doc = Main::pur::CMain::CreateDocument();
   Main::pur::CPurchaseType purchase = doc.Purchase.appendWithPrefix(_T("p"));

   // Set schema location
   doc.SetSchemaLocation(_T("Main.xsd"));

   // Declare namespaces on root element
   purchase.DeclareNamespace(_T("o"), _T("http://NamespaceTest.com/OrderTypes"));
   purchase.DeclareNamespace(_T("c"), _T("http://NamespaceTest.com/CustomerTypes"));
   purchase.DeclareNamespace(_T("cmn"), _T("http://NamespaceTest.com/CommonTypes"));

   // Append the OrderDetail element
   Main::ord::COrderType order = purchase.OrderDetail.append();
   Main::ord::CItemType item = order.Item.append();
   item.ProductName.append() = _T("Lawnmower");
   item.Quantity.append() = 1;
   item.UnitPrice.append() = 148.42;

   // Append the PaymentMethod element
   Main::cmn::CPaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
   paymentMethod.SetEnumerationValue(Main::cmn::CPaymentMethodTypeType::k_VISA);

   // Append the CustomerDetails element
   Main::cust::CCustomerType customer = purchase.CustomerDetails.append();
   customer.Name.append() = _T("Alice Smith");
   Main::cmn::CAddressType deliveryAddress = customer.DeliveryAddress.append();
   deliveryAddress.Line1.append() = _T("123 Maple Street");
   deliveryAddress.Line2.append() = _T("Mill Valley");
   Main::cmn::CAddressType billingAddress = customer.BillingAddress.append();
   billingAddress.Line1.append() = _T("8 Oak Avenue");
   billingAddress.Line2.append() = _T("Old Town");

   // Save to file and release object from memory
   doc.SaveToFile(_T("Main1.xml"), true);
   doc.DestroyDocument();
}
```

27.2.6.2 Espaces de noms XML et Préfixes (C#)

Après avoir généré du code depuis le schéma de bibliothèque (voir Schéma d'exemple), une application C# test est créée, avec plusieurs bibliothèques Altova. Rappelez-vous que le schéma d'exemple (**Main.xsd**) a plusieurs déclarations d'espace de noms. Par conséquent, le code généré comprend des espaces de noms qui

© 2016-2022 Altova GmbH

correspondent aux alias d'espace de noms (préfixes) provenant du schéma : `Main.ord`, `Main.pur`, `Main.cmn`, et `Main.cust`.

En général, afin de contrôler les espaces de noms XML et les préfixes avec l'aide des bibliothèques de schéma, vous disposez des méthodes suivantes :

- DeclareAllNamespacesFromSchema(). Appeler cette méthode si vous souhaitez déclarer les mêmes espaces de noms dans votre instance XML que dans le schéma. Sinon, si vous souhaitez utiliser d'autres espaces de noms que ceux utilisés dans cet exemple, vous devriez utiliser `DeclareNamespace()`. La méthode `DeclareAllNamespacesFromSchema()` n'est pas utilisée dans cet exemple parce que nous souhaitons utiliser spécifiquement des éléments XML avec des préfixes qui sont légèrement différents de ceux déclarés dans le schéma.
- DeclareNamespace(). Appeler cette méthode pour créer ou contourner l'attribut de préfixe d'espace de noms existant sur un élément. L'élément doit déjà être créé en utilisant soit la méthode `Append()` soit `AppendWithPrefix()`, comme illustré ci-dessous.
- AppendWithPrefix(). Utiliser cette méthode pour apposer un élément d'instance avec un préfixe spécifique. Pour créer l'instance XML illustrée dans cet exemple, il a suffit d'appeler cette méthode pour l'élément root uniquement. Tous les autres éléments ont été apposés en utilisant uniquement Append(), et leurs préfixes ont été ajoutés automatiquement sur la base de leurs espaces de noms, conformément aux règles ci-dessus.

L'extrait de code ci-dessous vous montre comment créer un document XML avec plusieurs déclarations d'espace de noms et des noms d'élément préfixés. En particulier, il génère une instance Purchase Order comme illustré dans l'Example : Purchase Order. Plus important encore, à des fins d'illustration, certains préfixes sont contournés dans l'instance XML (c'est à dire qu'ils ne sont tout à fait identiques que ceux déclarés dans le schéma).

```
protected static void Example()
{
   // Create the XML document and append the root element
   pur.Main2 doc = pur.Main2.CreateDocument();
   pur.PurchaseType purchase = doc.Purchase.AppendWithPrefix("p");

   // Set schema location
   doc.SetSchemaLocation(@"Main.xsd");

   // Declare namespaces on root element
   purchase.DeclareNamespace("o", "http://NamespaceTest.com/OrderTypes");
   purchase.DeclareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
   purchase.DeclareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");

   // Append the OrderDetail element
   ord.OrderType order = purchase.OrderDetail.Append();
   ord.ItemType item = order.Item.Append();
   item.ProductName.Append().Value = "Lawnmower";
   item.Quantity.Append().Value = 1;
   item.UnitPrice.Append().Value = 148.42M;

   // Append the PaymentMethod element
   cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.Append();
   paymentMethod.EnumerationValue = cmn.PaymentMethodTypeType.EnumValues.eVISA;

   // Append the CustomerDetails element
```

```
   cust.CustomerType customer = purchase.CustomerDetails.Append();
   customer.Name.Append().Value = "Alice Smith";
   cmn.AddressType deliveryAddress = customer.DeliveryAddress.Append();
   deliveryAddress.Line1.Append().Value = "123 Maple Street";
   deliveryAddress.Line2.Append().Value = "Mill Valley";
   cmn.AddressType billingAddress = customer.BillingAddress.Append();
   billingAddress.Line1.Append().Value = "8 Oak Avenue";
   billingAddress.Line2.Append().Value = "Old Town";

   // Save to file
   doc.SaveToFile("PurchaseOrder.xml", true);
}
```

27.2.6.3 Espaces de noms XML et Préfixes (Java)

Après avoir généré du code depuis le schéma de bibliothèque (voir Schéma d'exemple), une application Java de test est créée, avec plusieurs bibliothèques Altova. Rappelez-vous que le schéma d'exemple (**Main.xsd**) a plusieurs déclarations d'espace de noms. Par conséquent, le code généré comprend des espaces de noms qui correspondent aux alias d'espace de noms (préfixes) provenant du schéma : `com.Main.ord`, `com.Main.pur`, `com.Main.cmn`, et `com.Main.cust`.

En général, afin de contrôler les espaces de noms XML et les préfixes avec l'aide des bibliothèques de schéma, vous disposez des méthodes suivantes :

- declareAllNamespacesFromSchema(). Appeler cette méthode si vous souhaitez déclarer les mêmes espaces de noms dans votre instance XML que dans le schéma. Sinon, si vous souhaitez utiliser d'autres espaces de noms que ceux utilisés dans cet exemple, vous devriez utiliser `declareNamespace()`. La méthode `declareAllNamespacesFromSchema()` n'est pas utilisée dans cet exemple parce que nous souhaitons utiliser spécifiquement des éléments XML avec des préfixes qui sont légèrement différents de ceux déclarés dans le schéma.
- declareNamespace(). Appeler cette méthode pour créer ou contourner l'attribut de préfixe d'espace de noms existant sur un élément. L'élément doit déjà être créé en utilisant soit la méthode `append()` soit `appendWithPrefix()`, comme illustré ci-dessous.
- appendWithPrefix(). Utiliser cette méthode pour apposer un élément d'instance avec un préfixe spécifique. Pour créer l'instance XML illustrée dans cet exemple, il a suffit d'appeler cette méthode pour l'élément root uniquement. Tous les autres éléments ont été apposés en utilisant uniquement append(), et leurs préfixes ont été ajoutés automatiquement sur la base de leurs espaces de noms, conformément aux règles ci-dessus.

L'extrait de code ci-dessous vous montre comment créer un document XML avec plusieurs déclarations d'espace de noms et des noms d'élément préfixés. En particulier, il génère une instance Purchase Order comme illustré dans l'Example : Purchase Order. Plus important encore, à des fins d'illustration, certains préfixes sont contournés dans l'instance XML (c'est à dire qu'ils ne sont tout à fait identiques que ceux déclarés dans le schéma).

```
protected static void example() throws Exception {
   // Create the XML document and append the root element
   com.Main.pur.Main2 doc = com.Main.pur.Main2.createDocument();
   com.Main.pur.PurchaseType purchase = doc.Purchase.appendWithPrefix("p");
```

```
    // Set schema location
    doc.setSchemaLocation("Main.xsd");

    // Declare namespaces on root element
    purchase.declareNamespace("o", "http://NamespaceTest.com/OrderTypes");
    purchase.declareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
    purchase.declareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");

    // Append the OrderDetail element
    com.Main.ord.OrderType order = purchase.OrderDetail.append();
    com.Main.ord.ItemType item = order.Item.append();
    item.ProductName.append().setValue("Lawnmower");
    item.Quantity.append().setValue(1);
    java.math.BigDecimal price = new java.math.BigDecimal("148.42");
    item.UnitPrice.append().setValue(price);

    // Append the PaymentMethod element
    com.Main.cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
    paymentMethod.setEnumerationValue(com.Main.cmn.PaymentMethodTypeType.EVISA);

    // Append the CustomerDetails element
    com.Main.cust.CustomerType customer = purchase.CustomerDetails.append();
    customer.Name.append().setValue("Alice Smith");
    com.Main.cmn.AddressType deliveryAddress = customer.DeliveryAddress.append();
    deliveryAddress.Line1.append().setValue("123 Maple Street");
    deliveryAddress.Line2.append().setValue("Mill Valley");
    com.Main.cmn.AddressType billingAddress = customer.BillingAddress.append();
    billingAddress.Line1.append().setValue("8 Oak Avenue");
    billingAddress.Line2.append().setValue("Old Town");

    // Save to file
    doc.saveToFile("PurchaseOrder.xml", true);
}
```

27.3 Référence aux classes générées (C++)

Ce chapitre contient une description des classes C++ avec XMLSpy depuis un Schéma XML ou DTD (voir Générer du code depuis des Schémas XML ou DTDs). Vous pouvez intégrer ces classes dans votre code pour lire, modifier et écrire des documentation XML.

Note: Le code généré peut inclure d'autres classes de support, qui ne sont pas recensées ici et qui sont sujettes à des modifications.

27.3.1 altova::DateTime

Cette classe vous permet de traiter des attributs XML ou des éléments qui ont des types de date et d'heure, comme `xs:dateTime`.

Constructeurs

Nom	Description
`DateTime()`	Initialise une nouvelle instance de la classe `DateTime` à 12:00:00 minuit, le 1er janvier, 0001.
`DateTime(__int64 value, short timezone)`	Initialise une nouvelle instance de la classe `DateTime`. Le paramètre `value` représente le nombre de tics (intervales de 100 nanosecondes) qui se sont écoulés depuis 12:00:00 minuit, le 1er janvier, 0001.
`DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second)`	Initialise une nouvelle instance de la classe `DateTime` à l'année, mois, jour, heure, minute et seconde fournis en tant qu'argument.
`DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second, short timezone)`	Initialise une nouvelle instance de la classe `DateTime` à l'année, mois, jour, heure, minute, seconde et fuseau horaire fournis en tant qu'argument. Le fuseau horaire est exprimé en minutes et peut être positif ou négatif. Par exemple, le fuseau horaire "UTC-01:00" est exprimé en tant que "-60".

Méthodes

Nom	Description
`unsigned char Day() const`	Retourne le jour du mois de l'objet `DateTime` actuel. Les valeurs de retour s'étendent de 1 à 31.
`int DayOfYear() const`	Retourne le jour de l'année de l'objet `DateTime` actuel. Les valeurs de retour s'étendent de 1 à 366.
`bool HasTimezone() const`	Retourne Booléenne **true** si l'objet `DateTime` actuel présente un fuseau horaire défini ; **false** sinon.

© 2016-2022 Altova GmbH

Nom	Description
`unsigned char Hour() const`	Retourne l'heure de l'objet `DateTime` actuel. Les valeurs de retour s'étendent de 0 à 23.
`static bool IsLeapYear(int year)`	Retourne Booléenne **true** si l'année de la classe `DateTime` est une année bissextile ; **false** sinon.
`unsigned char Minute() const`	Retourne la minute de l'objet `DateTime` actuel. Les valeurs de retour s'étendent de 0 à 59.
`unsigned char Month() const`	Retourne le mois de l'objet `DateTime` actuel. Les valeurs de retour s'étendent de 1 à 12.
`__int64 NormalizedValue() const`	Retourne la valeur de l'objet `DateTime` exprimé en tant que Coordinated Universal Time (UTC).
`double Second() const`	Retourne la seconde de l'objet `DateTime` actuel. Les valeurs de retour s'étendent de 0 à 59.
`void SetTimezone(short tz)`	Définit le fuseau horaire de l'objet `DateTime` actuel à la valeur du fuseau horaire fourni en tant qu'argument. L'argument **tz** est exprimé en minutes et peut êtr positif ou négatif.
`short Timezone() const`	Retourne le fuseau horaire, en minutes de l'objet `DateTime` actuel. Avant d'utiliser cette méthode, veuillez vous assurer que l'objet comporte réellement un fuseau horaire, en appelan la méthode `HasTimezone()`.
`__int64 Value() const`	Retourne la valeur de l'objet `DateTime` object, exprimé dans le nombre de tics (intervalles de 100 nanosecondes) qui se sont écoulés depuis 12:00:00 minuit, le 1er janvier, 0001.
`int Weekday() const`	Retourne le jour de la semaine de l'objet `DateTime` actuel, en tant qu'un entier. Les valeurs s'étendent de 0 à 6, 0 étant lundi (ISO-8601).
`int Weeknumber() const`	Retourne le nombre de semaines dans l'année de l'objet `DateTime` actuel. Les valeurs de retour sont conformes à ISO-8601.
`int WeekOfMonth() const`	Retourne le nombre de semaines dans le mois de l'objet `DateTime` actuel. Les valeurs de retour sont conformes à ISO-8601.
`int Year() const`	Retoune l'année de l'objet `DateTime` actuel.

Exemple

```
void Example()
{
   // initialize a new DateTime instance to 12:00:00 midnight, January 1st, 0001
   altova::DateTime dt1 = altova::DateTime();

   // initialize a new DateTime instance using the year, month, day, hour, minute, and
```

```
second
   altova::DateTime dt2 = altova::DateTime(2015, 11, 10, 9, 8, 7);

   // initialize a new DateTime instance using the year, month, day, hour, minute,
second, and UTC +01:00 timezone
    altova::DateTime dt = altova::DateTime(2015, 11, 22, 13, 53, 7, 60);

   // Get the value of this DateTime object
   std::cout << "The number of ticks of the DateTime object is: " << dt.Value() <<
std::endl;

   // Get the year
   cout << "The year is: " << dt.Year() << endl;
   // Get the month
   cout << "The month is: " << (int)dt.Month() << endl;
   // Get the day of the month
   cout << "The day of the month is: " << (int) dt.Day() << endl;
   // Get the day of the year
   cout << "The day of the year is: " << dt.DayOfYear() << endl;
   // Get the hour
   cout << "The hour is: " << (int) dt.Hour() << endl;
   // Get the minute
   cout << "The minute is: " << (int) dt.Minute() << endl;
   // Get the second
   cout << "The second is: " << dt.Second() << endl;
   // Get the weekday
   cout << "The weekday is: " << dt.Weekday() << endl;
   // Get the week number
   cout << "The week of year is: " << dt.Weeknumber() << endl;
   // Get the week in month
   cout << "The week of month is: " << dt.WeekOfMonth() << endl;

   // Check whether a DateTime instance has a timezone
   if (dt.HasTimezone() == TRUE)
   {
      // output the value of the Timezone
      cout <<  "The timezone is: " << dt.Timezone() << endl;
   }
   else
   {
      cout <<  "No timezone has been defined." << endl;
   }

   // Construct a DateTime object with a timezone UTC+01:00 (Vienna)
   altova::DateTime vienna_dt = DateTime(2015, 11, 23, 14, 30, 59, +60);
   // Output the result in readable format
   cout << "The Vienna time: "
       << (int) vienna_dt.Month()
       << "-" << (int) vienna_dt.Day()
       << " " << (int) vienna_dt.Hour()
       << ":" << (int) vienna_dt.Minute()
       << ":" << (int) vienna_dt.Second()
       << endl;
```


© 2016-2022 Altova GmbH

```
   // Convert the value to UTC time
   DateTime utc_dt = DateTime(vienna_dt.NormalizedValue());
   // Output the result in readable format
   cout << "The UTC time:    "
      << (int) utc_dt.Month()
      << "-" << (int) utc_dt.Day()
      << " " << (int) utc_dt.Hour()
      << ":" << (int) utc_dt.Minute()
      << ":" << (int) utc_dt.Second()
      << endl;

   // Check if a year is a leap year
   int year = 2016;
   if( altova::DateTime::IsLeapYear(year) )
   { cout << year << " is a leap year" << endl; }
   else
   { cout << year << " is not a leap year" << endl; }
}
```

27.3.2 altova::Duration

Cette classe vous permet de traiter des attributs XML ou des éléments de type `xs: duration` .

Constructeurs

Nom	Description
`Duration()`	Initialise une nouvelle instance de la classe `Duration` dans une valeur vide.
`Duration(const DayTimeDuration& dt)`	Initialise une nouvelle instance de la classe `Duration` dans une durée définie par l'argument `dt` (voir altova::DayTimeDuration).
`Duration(const YearMonthDuration& ym)`	Initialise une nouvelle instance de la classe `Duration` dans la durée définie par l'argument `ym` (voir altova::YearMonthDuration).
`Duration(const YearMonthDuration& ym, const DayTimeDuration& dt)`	Initialise une nouvelle instance de la classe `Duration` dans les durées définies par les arguments `dt` et `ym` (voir altova::YearMonthDuration et altova::DayTimeDuration).

Méthodes

Nom	Description
`int Days() const`	Retourne le nombre de jours dans l'instance `Duration` actuelle.
`DayTimeDuration DayTime() const`	Retourne le jour et la durée de temps dans l'instance `Duration` actuelle, exprimé en tant qu'un objet `DayTimeDuration` (voir altova::DayTimeDuration).

Nom	Description
`int Hours() const`	Retourne le nombre d'heures dans l'instance `Duration` actuelle.
`bool IsNegative() const`	Retourne Boléenne **true** si l'instance `Duration` actuelle est négative.
`bool IsPositive() const`	Retourne Boléenne **true** si l'instance `Duration` actuelle est positive.
`int Minutes() const`	Retourne le nombre de minutes dans l'instance `Duration` actuelle.
`int Months() const`	Retourne le nombre de mois dans l'instance `Duration` actuelle.
`double Seconds() const`	Retourne le nombre de secondes dans l'instance `Duration` actuelle.
`YearMonthDuration YearMonth() const`	Retourne la durée d'année et de mois dans l'instance `Duration` actuelle, exprimée en tant qu'un objet `YearMonthDuration` (voir altova::YearMonthDuration).
`int Years() const`	Retourne le nombre des années dans l'instance `Duration` actuelle.

Exemple

L'extrait de code suivant illustre la création d'un nouvel objet `Duration`, ainsi que la lecture de valeurs en provenant.

```
void ExampleDuration()
{
   // Create an empty Duration object
   altova::Duration empty_duration = altova::Duration();

   // Create a Duration object using an existing duration value
   altova::Duration duration1 = altova::Duration(empty_duration);

   // Create a YearMonth duration of six years and five months
   altova::YearMonthDuration yrduration = altova::YearMonthDuration(6, 5);

   // Create a DayTime duration of four days, three hours, two minutes, and one second
   altova::DayTimeDuration dtduration = altova::DayTimeDuration(4, 3, 2, 1);

   // Create a Duration object by combining the two previously created durations
   altova::Duration duration = altova::Duration(yrduration, dtduration);

   // Get the number of years in this Duration instance
   cout << "Years:  " << duration.Years() << endl;

   // Get the number of months in this Duration instance
   cout << "Months: " << duration.Months() << endl;

   // Get the number of days in this Duration instance
   cout << "Days:   " << duration.Days() << endl;

   // Get the number of hours in this Duration instance
```


© 2016-2022 Altova GmbH

```
   cout << "Hours:  " << duration.Hours() << endl;

   // Get the number of hours in this Duration instance
   cout << "Minutes: " << duration.Minutes() << endl;

   // Get the number of seconds in this Duration instance
   cout << "Seconds: " << duration.Seconds() << endl;
}
```

27.3.3 altova::DayTimeDuration

Cette classe vous permet de traiter des types de durée de schéma XML qui consistent en une partie jour et une partie heure.

Constructeurs

Nom	Description
`DayTimeDuration()`	Initialise une nouvelle instance de la classe `DayTimeDuration` dans une valeur vide.
`DayTimeDuration(int days, int hours, int minutes, double seconds)`	Initialise une nouvelle instance de la classe `DayTimeDuration` dans le nombre de jours, heures, minutes, secondes fournis en tant qu'argument.
`explicit DayTimeDuration(__int64 value)`	Initialise une nouvelle instance de la classe `DayTimeDuration` dans une durée qui consiste en autant de tics (intervalles de 100 nanosecondes) que fournis dans l'argument **value**.

Méthodes

Nom	Description
`int Days() const`	Retourne le nombre de jours dans l'instance `DayTimeDuration` actuelle.
`int Hours() const`	Retourne le nombre d'heures dans l'instance `DayTimeDuration` actuelle.
`bool IsNegative() const`	Retourne Booléenne **true** si l'instance `DayTimeDuration` actuelle est négative.
`bool IsPositive() const`	Retourne Booléenne **true** si l'instance `DayTimeDuration` actuelle est positive.
`int Minutes() const`	Retourne le nombre de minutes dans l'instance `DayTimeDuration` actuelle.
`double Seconds() const`	Retourne le nombre de secondes dans l'instance `DayTimeDuration` actuelle.

`__int64 Value() const`	Retourne la valeur (en tics) de l'instance`DayTimeDuration` actuelle.

27.3.4 altova::YearMonthDuration

Cette classe vous permet de traiter des types de durée de schéma XML qui consistent en une partie année et mois.

Constructeurs

Nom	Description
`YearMonthDuration()`	Initialise une nouvelle instance de la classe `YearMonthDuration` dans une valeur vide.
`YearMonthDuration(int years, int months)`	Initialise une nouvelle instance de la classe `YearMonthDuration` dans le nombre des années et des mois fournis dans les arguments **années** et **mois**.
`explicit YearMonthDuration(int value)`	Initialise une nouvelle instance de la classe `YearMonthDuration` dans une durée qui consiste en autant de tics (intervalles de 100 nanosecondes) comme fournir dans l'argument **value**.

Méthodes

Nom	Description
`bool IsNegative() const`	Retourne Booléennen **true** si l'instance `YearMonthDuration` actuelle est négative.
`bool IsPositive() const`	Retourne Booléennen **true** si l'instance `YearMonthDuration` actuelle est positive.
`int Months() const`	Retourne le nombre de mois dans l'instance `YearMonthDuration` actuelle.
`int Value() const`	Retourne la valeur (en tics) de l'instance `YearMonthDuration` actuelle.
`int Years()`	Retourne le nombre d'années dans l'instance `YearMonthDuration` actuelle.

27.3.5 altova::meta::Attribute

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des attributs. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un attribut dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un attribut particulier défini dans le schéma XML.

© 2016-2022 Altova GmbH

Méthodes

Nom	Description
`SimpleType GetDataType()`	Retourne le type du contenu d'attribut.
`string_type GetLocalName()`	Retourne le nom local de l'attribut.
`string_type GetNamespaceURI()`	Retourne l'URI d'espace de noms de l'attribut.
`bool IsRequired()`	Retourne true si l'attribut est requis.

Opérateurs

Nom	Description
`bool operator()`	Retourne true si ce n'est pas l'attribut NULL.
`bool operator!()`	Retourne true s'il s'agit de l'attribut NULL.

27.3.6 altova::meta::ComplexType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types complexes. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types complexes dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type complexes particulier défini dans le schéma XML.

Méthodes

Nom	Description
`Attribute FindAttribute(const char_type* localName, const char_type* namespaceURI)`	Trouve l'attribut avec le nom local spécifié et l'URI d'espace de noms
`Element FindElement(const char_type* localName, const char_type* namespaceURI)`	Trouve l'élément avec le nom local spécifié et l'URI d'espace de noms.
`std::vector<Attribute> GetAttributes()`	Retourne une liste de tous les attributs.
`ComplexType GetBaseType()`	Retourne le type de base de ce type.
`SimpleType GetContentType()`	Retourne le type simple du contenu.
`std::vector<Element> GetElements()`	Retourne une liste de tous les éléments.
`string_type GetLocalName()`	Retourne le nom local du type.

Nom	Description
`string_type GetNamespaceURI()`	Retourne l'URI d'espace de noms du type.

Opérateurs

Nom	Description
`bool operator()`	Retourne true si ce n'est pas le ComplexType NULL.
`bool operator!()`	Retourne true s'il s'agit du ComplexType NULL.

27.3.7 altova::meta::Element

Cette classe vous permet d'accéder à des informations concernant des classes générées depuis des éléments de schéma. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un élément dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un élément particulier défini dans le schéma XML.

Méthodes

Nom	Description
`ComplexType GetDataType()`	Retourne le type de l'élément. Noter qu'il s'agit toujours d'un type de complexe même s'il est déclaré en tant que simple dans le schéma original. Utiliser GetContentType() de l'objet retourné pour obtenir le type de contenu simple.
`string_type GetLocalName()`	Retourne le nom local de l'élément.
`unsigned int GetMaxOccurs()`	Retourne la valeur maxOccurs définie dans le schéma.
`unsigned int GetMinOccurs()`	Retoune la valeur minOccurs définie dans le schéma.
`string_type GetNamespaceURI()`	Retourne l'URI d'espace de noms de l'élément .

Opérateurs

Nom	Description
`bool operator()`	Retourne true si ce n'est pas l'élément NULL.
`bool operator!()`	Retourne true s'il s'agit de l'élément NULL.

27.3.8 altova::meta::SimpleType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types simples. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à

© 2016-2022 Altova GmbH

propos des instances particulières des types simples dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type simple particulier défini dans le schéma XML.

Méthodes

Nom	Description
`SimpleType GetBaseType()`	Retourne le type de base de ce type.
`std::vector<string_type> GetEnumerations()`	Retourne une liste de toutes les facettes d'énumération.
`unsigned int GetFractionDigits()`	Retourne la valeur de cette facette.
`unsigned int GetLength()`	Retourne la valeur de cette facette.
`string_type GetLocalName()`	Retourne le nom local du type.
`string_type GetMaxExclusive()`	Retourne la valeur de cette facette.
`string_type GetMaxInclusive()`	Retourne la valeur de cette facette.
`unsigned int GetMaxLength()`	Retourne la valeur de cette facette.
`string_type GetMinExclusive()`	Retourne la valeur de cette facette.
`string_type GetMinInclusive()`	Retourne la valeur de cette facette.
`unsigned int GetMinLength()`	Retourne la valeur de cette facette.
`string_type GetNamespaceURI()`	Retourne l'URI d'espace de noms du type.
`std::vector<string_type> GetPatterns()`	Retourne une liste de toutes les facettes de motif.
`unsigned int GetTotalDigits()`	Retourne la valeur de cette facette.
`WhitespaceType GetWhitespace()`	Retourne la valeur de la facette d'espace blanc, qui est un de : • Whitespace_Unknown • Whitespace_Preserve • Whitespace_Replace • Whitespace_Collapse

Opérateurs

Nom	Description
`bool operator()`	Retourne true si ce n'est pas de SimpleType NULL.
`bool operator!()`	Retourne true s'il s'agit du SimpleType NULL.

27.3.9 [YourSchema]::[CDoc]

Lorsque le code est généré depuis un schéma XML, le code généré fournit une classe de document avec le même nom que le schéma. Cette classe contient tous les éléments root possibles en tant que membres, ainsi que les méthodes suivantes. Veuillez noter que, dans les noms de méthode ci-dessous, "CDoc" représente le nom de la classe de document générée elle-même.

Méthodes

Nom	Description
`static CDoc CreateDocument()`	Crée un nouveau document XML vide. Doit être publié en utilisant DestroyDocument().
`static void DeclareAllNamespacesFromSchema(ElementType& node)`	Déclare tous les espaces de noms depuis le Schéma XML dans l'élément fournit en dans qu'argument (généralement, l'élément racine XML). L'appel de cette méthode est utile si votre schéma a plusieurs déclarations d'espace de noms, chacune mappée dans un préfixe, et que vous souhaitez les déclarer tous dans l'élément fourni en tant qu'argument.
`void DestroyDocument()`	Détruit un document. Toutes les références au document et à ses nœuds sont invalidées. Cela doit être appelé lorsque vous terminez de travailler avec un document.
`static CDoc LoadFromBinary(const std:vector<unsigned char>& xml)`	Charge un document XML depuis un byte array.
`static CDoc LoadFromFile(const string_type& fileName)`	Charge un document XML depuis un fichier.
`static CDoc LoadFromString(const string_type& xml)`	Charge un document XML depuis un string.
`std::vector<unsigned char> SaveToBinary(bool prettyPrint)`	Enregistre un document XML sous un byte array. Lorsque défini sur true, l'argument `prettyPrint` reformate le document XML pour une meilleure lecture.
`std::vector<unsigned char> SaveToBinary(bool prettyPrint, const string_type & encoding)`	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
`std::vector<unsigned char> SaveToBinary(bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM)`	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
`void SaveToFile(const string_type & fileName, bool prettyPrint)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
`void SaveToFile(const string_type & fileName, bool omitXmlDecl)`	Enregistre un document XML sous un fichier. Si l'argument `omitXmlDecl` est défini pour être true, la déclaration XML ne sera pas rédigée.

© 2016-2022 Altova GmbH

Nom	Description
`void SaveToFile(const string_type & fileName, bool omitXmlDecl, const string_type & encoding)`	Enregistre un document XML sous un fichier avec l'encodage spécifié. Si l'argument `omitXmlDecl` est défini pour être true, la déclaration XML ne sera pas rédigée.
`void SaveToFile(const string_type & fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, bool bBigEndian, bool bBOM)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
`void SaveToFile(const string_type & fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, bool bBigEndian, bool bBOM, const string_type & lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et la fin de ligne spécifiée. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode. Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir Options de Générateur de Code).
`void SaveToFile(const string_type& fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, const string_type & lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et la fin de ligne spécifiée. Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir Options de Générateur de Code).
`void SaveToFile(const string_type & fileName, bool prettyPrint, const string_type & encoding)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
`void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
`void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM, const string_type & lineend)`	Enregistre un document XML sous un fichier, avec l'encodage spécifié et la fin de ligne spécifiée. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode. Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir Options de Générateur de Code).
`void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, const string_type & lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et la fin de ligne spécifiée. Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir Options de Générateur de Code).

Nom	Description
`string_type SaveToString(bool prettyPrint)`	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel.
`string_type SaveToString(bool prettyPrint, bool omitXmlDecl)`	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel. Si l'argument `omitXmlDecl` est défini pour être true, la déclaration XML ne sera pas rédigée.
`void SetDTDLocation(const string_type & dtdLocation)`	Ajoute une déclaration DOCTYPE avec l'ID de système spécifiée. Un élément root doit déjà exister. Cette méthode n'est pas prise en charge pour MSXML, puisqu'il n'est pas possible d'ajouter une déclaration DOCTYPE dans un document dans la mémoire.
`void SetSchemaLocation(const string_type & schemaLocation)`	Ajoute un attribut xsi:schemaLocation ou xsi:noNamespaceSchemaLocation à l'élément root. Un élément root doit déjà exister.

27.3.10 [YourSchema]::[ElementType]

Cette classe propose des méthodes pour manipuler des éléments XML depuis votre schéma. Des méthodes de cette classe peuvent être appelées sur des éléments, pas sur le document XML lui-même. Noter qu'il n'est pas nécessaire d'instancier la classe directement pour appeler des méthodes de cette classe. Tout élément créé en utilisant les méthodes **`append()`** ou **`appendWithPrefix()`** est de type `[ElementType]`.

Méthodes

Nom	Description

`void DeclareNamespace(const string_type prefix, const string_type nsURI)`	Cette méthode prend deux arguments qui sont tous les deux de type string : le préfixe et l'URI d'espace de noms que vous souhaitez utiliser. Le préfixe fournit en tant qu'argument sera mappé dans la valeur URI d'espace de noms fournie en tant qu'argument. Si le préfixe fourni en tant qu'argument est vide, la méthode crée ou contourne la déclaration d'espace de noms par défaut dans l'élément.
	
	Par exemple, assumons que le document XML a un élément XML appelé "purchase". Si vous appelez
	
	`purchase.DeclareNamespace(_T("ord"), _T("http://OrderTypes"));`
	
	Alors le document XML devient
	
	`<purchase xmlns:ord="http://OrderTypes" />`
	
	Un autre exemple ; si vous appelez :

 © 2016-2022 Altova GmbH

Nom	Description

	`purchase.DeclareNamespace(_T(""), _T("http://OrderTypes"));`
	
	Alors le document XML devient
	
	`<purchase xmlns="http://OrderTypes" />`
	
	Note : l'espace de noms déclaré est utilisé lorsque les éléments ou les attributs enfants ultérieurs, conformément aux règles suivantes :
	
	1. si l'espace de noms enfant est le défaut, utiliser le préfixe vide.
	2. si l'espace de noms enfant est égal à l'espace de noms parent, utiliser le préfixe parent.
	3. Sinon, chercher le préfixe le plus proche en commençant du parent vers le haut, en utilisant l'algorithme lookup décrit dans la section "B.2: Namespace Prefix Lookup" at https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html.
	4. Si aucun préfixe n'a été trouvé pour l'espace de noms élément, utiliser un préfixe vide.

27.3.11 [YourSchema]::MemberAttribute

Lorsque le code est généré depuis un schéma XML, une classe telle que celle-ci est créée pour chaque attribut de membre d'un type.

Méthodes

Nom	Description
`bool exists()`	Retourne true si l'attribut existe.
`int GetEnumerationValue()`	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou "Invalid" si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.
`altova::meta::Attribute info()`	Retourne un objet pour l'information de schéma de requête (voir `altova::meta::Attribute`).
`void remove()`	Supprime l'attribut depuis son élément parent.

Nom	Description
`void SetEnumerationValue(int)`	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.

27.3.12 [YourSchema]::MemberElement

Lorsque le code est généré depuis un schéma XML, une classe telle que celle-ci est créée pour chaque élément de membre d'un type. Dans les descriptions ci-dessous, "MemberType" représente le nom de l'élément de membre lui-même.

Méthodes

Nom	Description
`Iterator<MemberType> all()`	Retourne un objet pour itérer des instances de l'élément de membre.
`MemberType append()`	Crée un nouvel élément et l'ajoute à son parent.
`MemberType appendWithPrefix(string_type prefix)`	Crée un nouvel élément ayant le préfixe fournit en tant qu'argument, et l'appose à son parent. Pour consulter un exemple, voir Exemple : Purchase Order.
`unsigned int count()`	Retourne le décompte des éléments.
`int GetEnumerationValue()`	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou **Invalid** si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.
`bool exists()`	Retourne true si au moins un élément existe.
`MemberType first()`	Retourne la première instance de l'élément de membre.
`MemberType operator[](unsigned int index)`	Retourne l'élément de membre spécifié par l'index.
`altova::meta::Element info()`	Retourne un objet pour l'information de schéma de requête (voir `altova::meta::Element`).
`MemberType last()`	Retourne la dernière instance de l'élément de membre.
`void remove()`	Supprime toutes les occurrences de l'élément depuis son parent.
`void removeAT(unsigned int index)`	Supprime l'occurrence de l'élément spécifié par l'index.
`void SetEnumerationValue(int)`	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.

© 2016-2022 Altova GmbH

27.4 Référence aux classes générées (C#)

Ce chapitre contient une description des classes C# générées avec XMLSpy epuis un schéma XML ou DTD (voir Générer du code depuis des Schémas XML ou DTDs). Vous pouvez intégrer ces classes dans votre code pour lire, modifier et écrire des documentation XML.

Note: Le code généré peut inclure d'autres classes de support, qui ne sont pas recensées ici et qui sont sujettes à des modifications.

27.4.1 Altova.Types.DateTime

Cette classe vous permet de traiter des attributs XML ou des éléments qui ont des types de date et d'heure, comme `xs:dateTime`.

Constructeurs

	Nom	Description
	`DateTime(DateTime obj)`	Initialise une nouvelle instance de la classe `DateTime` dans l'objet `DateTime` fourni en tant qu'argument.
	`DateTime(System.DateTime newvalue)`	Initialise une nouvelle instance de la classe `DateTime` dans l'objet `System.DateTime` fourni en tant qu'argument.
	`DateTime(int year, int month, int day, int hour, int minute, double second, int offsetTZ)`	Initialise une nouvelle instance de la classe `DateTime` à l'offset d'année, mois, jour, heure, minute, seconde et fuseau horaire fournis en tant qu'arguments.
	`DateTime(int year, int month, int day, int hour, int minute, double second)`	Initialise une nouvelle instance de la classe `DateTime` à l'année, mois, jour, heure, minute et seconde fournis en tant qu'arguments.
	`DateTime(int year, int month, int day)`	Initialise une nouvelle instance de la classe `DateTime` à l'année, mois et jour fournis en tant qu'argument.

Propriétés

	Nom	Description
	`bool HasTimezone`	Obtient une valeur Booléenne qui indique si `DateTime` a un fuseau horaire.
	`static DateTime Now`	Obtient un objet `DateTime` qui est défini sur la date et l'heure actuelles sur cet ordinateur.
	`short TimezoneOffset`	Obtient ou définit l'offset de fuseau horaire, en minutes de l'objet `DateTime`.

	Nom	Description
	`System.DateTime Value`	Obtient ou définit l'offset de l'objet `DateTime` en tant qu'une valeur `System.DateTime`.

Méthodes

	Nom	Description
	`int CompareTo(object obj)`	La classe `DateTime` met en place l'interface `IComparable`. Cette méthode compare l'instance actuelle de `DateTime` avec un autre objet et retourne un entier qui indique si l'instance actuelle précèdes, suit, ou se produit dans la même position dans l'ordre de tri que l'autre objet. Voir aussi https://msdn.microsoft.com/en-us/library/system.icomparable.compareto(v=vs.110).aspx
	`override bool Equals(object obj)`	Retourne **true** si l'objet spécifié est égal à l'objet actuel ; **false** sinon.
	`System.DateTime GetDateTime(bool correctTZ)`	Retourne un objet `System.DateTime` depuis l'instance `Altova.Types.DateTime`. L'argument booléen `correctTZ` spécifie si le temps de l'objet retourné doit être ajusté conformément au fuseau horaire de l'instance actuel `Altova.Types.DateTime`.
	`override int GetHashCode()`	Retourne le code hash de l'instance actuelle.
	`int GetWeekOfMonth()`	Retourne le nombre de semaines dans le mois sous forme d'un entier.

	`static DateTime Parse( string s )`	Crée un objet `DateTime` depuis le string fournit en tant qu'argument. Par exemple, les valeurs de string d'échantillon suivantes sont converties avec succès dans un objet `DateTime` :
		
		`2015-01-01T23:23:23`
		`01/01/2015`
		`2015-11`
		`23:23:23`
		
		Une exception est soulevée si le string ne peut pas être converti en un objet `DateTime`.
		
		Veuillez noter que cette méthode est statique et qu'elle peut uniquement être appelée dans la classe `Altova.Types.DateTime` elle-même, pas dans une instance de la classe.

	`static DateTime Parse(string s, DateTimeFormat format)`	Crée un objet `DateTime` depuis un string, en utilisant le format fourni en tant qu'argument. Pour la liste des formats possibles, voir Altova.Types.DateTimeFormat. Une exception est soulevée si le string ne peut pas être converti en un objet `DateTime`.

© 2016-2022 Altova GmbH

	Nom	Description
		Veuillez noter que cette méthode est statique et qu'elle peut uniquement être appelée dans la classe `Altova.Types.DateTime` elle-même, pas dans une instance de la classe.
	`override string ToString()`	Convertit l'objet `DateTime` en un string.
	`string ToString(DateTimeFormat format)`	Convertit l'objet `DateTime` en un string, utiliser le format fournit en tant qu'argument. Pour la liste des formats possibles, voir Altova.Types.DateTimeFormat.

Opérateurs

Nom	Description
`!=`	Détermine si `DateTime` a n'est pas égal à `DateTime` b.
`<`	Détermine si `DateTime` a est inférieur à `DateTime` b.
`<=`	Détermine si `DateTime` a est inférieur ou égal à `DateTime` b.
`==`	Détermine si `DateTime` a est égal à `DateTime` b.
`>`	Détermine si `DateTime` a est supérieur à `DateTime` b.
`>=`	Détermine si `DateTime` a est supérieur ou égal à `DateTime` b.

Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

```
utiliser Altova.Types;
```

L'extrait de code suivant illustre plusieurs moyens de créer des objets `DateTime` :

```
protected static void DateTimeExample1()
{
   // Create a DateTime object from the current system time
   Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
   Console.WriteLine("The current time is: " + dt.ToString());

   // Create an Altova DateTime object from parts (no timezone)
   Altova.Types.DateTime dt1 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33);
   Console.WriteLine("My custom time is : " + dt1.ToString());

   // Create an Altova DateTime object from parts (with UTC+60 minutes timezone)
   Altova.Types.DateTime dt2 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33, 60);
```

```
   Console.WriteLine("My custom time with timezone is : " + dt2.ToString());

   // Create an Altova DateTime object by parsing a string
   Altova.Types.DateTime dt3 = Altova.Types.DateTime.Parse("2015-01-01T23:23:23");
   Console.WriteLine("Time created from string: " + dt3.ToString());

   // Create an Altova DateTime object by parsing a string formatted as schema date
   Altova.Types.DateTime dt4 = Altova.Types.DateTime.Parse("2015-01-01",
DateTimeFormat.W3_date);
   Console.WriteLine("Time created from string formatted as schema date: " +
dt4.ToString());
}
```

The following code listing illustrates various ways to format `DateTime` objects:

```
protected static void DateTimeExample2()
{
   // Create a DateTime object from the current system time
   Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);

   // Output the unformatted DateTime
   Console.WriteLine("Unformatted time: " + dt.ToString());

   // Output this DateTime formatted using various formats
   Console.WriteLine("S_DateTime:       " + dt.ToString(DateTimeFormat.S_DateTime));
   Console.WriteLine("S_Days:           " + dt.ToString(DateTimeFormat.S_Days));
   Console.WriteLine("S_Seconds:        " + dt.ToString(DateTimeFormat.S_Seconds));
   Console.WriteLine("W3_date:          " + dt.ToString(DateTimeFormat.W3_date));
   Console.WriteLine("W3_dateTime:      " + dt.ToString(DateTimeFormat.W3_dateTime));
   Console.WriteLine("W3_gDay:          " + dt.ToString(DateTimeFormat.W3_gDay));
   Console.WriteLine("W3_gMonth:        " + dt.ToString(DateTimeFormat.W3_gMonth));
   Console.WriteLine("W3_gMonthDay:     " + dt.ToString(DateTimeFormat.W3_gMonthDay));
   Console.WriteLine("W3_gYear:         " + dt.ToString(DateTimeFormat.W3_gYear));
   Console.WriteLine("W3_gYearMonth:    " + dt.ToString(DateTimeFormat.W3_gYearMonth));
   Console.WriteLine("W3_time:          " + dt.ToString(DateTimeFormat.W3_time));
}
```

27.4.2 Altova.Types.DateTimeFormat

Le type `DateTimeFormat` enum comporte les valeurs constantes suivantes :

Valeur	Description	Exemple
S_DateTime	Formate la valeur en tant que dateTime standard, avec une précision d'un dix millionième de seconde, y compris le fuseau horaire.	`2015-11-12 12:19:03.9019132+01:00`
S_Days	Formate la valeur en tant que nombre de jours passés depuis l'époque UNIX.	`735913,631897345108796296 3`

© 2016-2022 Altova GmbH

Valeur	Description	Exemple
S_Seconds	Formate la valeur en tant que le nombre de seconde écoulées depuis l'époque UNIX, avec une précision d'un dix millionième d'une seconde.	`63582937678,0769062`
W3_date	Formate la valeur en tant que date de schéma.	`12/11/2015`
W3_dateTime	Formate la valeur en tant que dateTime de schéma.	`2015-11-12T15:12:14.5194251`
W3_gDay	Formate la valeur en tant que gDay de schéma.	`---12` (en partant du principe que la date est le 12 du mois)
W3_gMonth	Formate la valeur en tant que gMonth de schéma.	`--11` (en partant du principe que le mois est novembre)
W3_gMonthDay	Formate la valeur en tant que gMonthDay de schéma.	`--11-12` (en partant du principe que la date est le 12 novembre)
W3_gYear	Formate la valeur en tant que gYear de schéma.	`2015` (en partant du principe que l'année est 2015)
W3_gYearMonth	Formate la valeur en tant que gYearMonth de schéma.	`2015-11` (en partant du principe que l'année est 2015 et que le mois est novembre)
W3_time	Formate la valeur en tant qu'heure de schéma, avec une précision d'un dix millionième d'une seconde.	`15:19:07.5582719`

27.4.3 Altova.Types.Duration

Cette classe vous permet de traiter des attributs XML ou des éléments de type `xs: duration` .

Constructeurs

	Nom	Description
	`Duration(Duration obj)`	Initialise une nouvelle instance de la classe `Duration` dans l'objet `Duration` fournie en tant qu'argument.
	`Duration(Duration newvalue)`	Initialise une nouvelle instance de la classe `Duration` dans l'objet `System.TimeSpan` fournie en tant qu'argument.

	Nom	Description
	`Duration(long ticks)`	Initialise une nouvelle instance de la classe `Duration` dans la le nombre de tics fournie en tant qu'argument.
	`Duration(int newyears, int newmonths, int days, int hours, int minutes, int seconds, double partseconds, bool bnegative)`	Initialise une nouvelle instance de la classe `Duration` dans une durée construite depuis des parties fournies en tant qu'argument.

Propriétés

	Nom	Description
	`int Months`	Obtient ou définit le nombre des mois dans l'instance actuelle de `Duration`.
	`System.TimeSpan Value`	Obtient ou définit la valeur (en tant que `System.TimeSpan`) de l'instance actuelle de `Duration`.
	`int Years`	Retourne le nombre des années dans l'instance actuelle de `Duration`.

Méthodes

	Nom	Description
	`override bool Equals(object other)`	Retourne **true** si l'objet spécifié est égal à l'objet actuel ; **false** sinon.
	`override int GetHashCode()`	Retourne le code hash de l'instance actuelle.
	`bool IsNegative()`	Retourne **true** si l'instance actuelle de `Duration` représente une durée négative.
	`static Duration Parse( string s, ParseType pt )`	Retourne un objet `Altova.Types.Duration` parsé depuis le string fourni en tant qu'argument, en utilisant le type de parsage fourni en tant qu'argument. Valeurs de type de parsage valides : **DURATION** Parse la durée en partant du principe que la partie de l'année, du mois, du jour et de la durée de temps existe. **YEARMONTH** Parse la durée en partant du principe que seules les parties d'année et de mois existent. **DAYTIME** Parse la durée en partant du principe que la partie du jour et d'heure existent. Veuillez noter que cette méthode est statique et peut uniquement être appelée dans la classe elle-même, pas dans une instance de la classe.

© 2016-2022 Altova GmbH

	Nom	Description
	`override string ToString()`	Convertit l'instance actuelle `Duration` en un string. Par exemple, une durée de 3 heures, 4 minutes et 5 secondes peut être convertie en "PT3H4M5S".
	`string ToYearMonthString()`	Convertit l'instance actuelle `Duration` en un string, en utilisant le type de parsage "Year and Month".

Opérateurs

Nom	Description
`!=`	Détermine si `Duration` a n'est pas égal à `Duration` b.
`==`	Détermine si `Duration` a est égal à `Duration` b.

Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

```
using Altova.Types;
```

L'extrait de code suivant illustre plusieurs moyens de créer des objets `Duration` :

```
protected static void DurationExample1()
{
   // Create a new time span of 3 hours, 4 minutes, and 5 seconds
   System.TimeSpan ts = new TimeSpan(3, 4, 5);
   // Create a Duration from the time span
   Duration dr = new Duration(ts);
   // The output is: PT3H4M5S
   Console.WriteLine("Duration created from TimeSpan: " + dr.ToString());

   // Create a negative Altova.Types.Duration from 6 years, 5 months, 4 days, 3 hours,
       // 2 minutes, 1 second, and .33 of a second
   Duration dr1 = new Duration(6, 5, 4, 3, 2, 1, .33, true);
   // The output is: -P6Y5M4DT3H2M1.33S
   Console.WriteLine("Duration created from parts: " + dr1.ToString());

   // Create a Duration from a string using the DAYTIME parse type
   Duration dr2 = Altova.Types.Duration.Parse("-P4DT3H2M1S", Duration.ParseType.DAYTIME);
   // The output is -P4DT3H2M1S
   Console.WriteLine("Duration created from string: " + dr2.ToString());

   // Create a duration from ticks
   Duration dr3 = new Duration(System.DateTime.UtcNow.Ticks);
   // Output the result
```

```
   Console.WriteLine("Duration created from ticks: " + dr3.ToString());
}
```

L'extrait de code suivant illustre comment obtenir des valeurs depuis des objets `Duration` :

```
protected static void DurationExample2()
{
   // Create a negative Altova.Types.Duration from 6 years, 5 months, 4 days, 3 hours,
       // 2 minutes, 1 second, and .33 of a second
   Duration dr = new Duration(6, 5, 4, 3, 2, 1, .33, true);
   // The output is: -P6Y5M4DT3H2M1.33S
   Console.WriteLine("The complete duration is: " + dr.ToString());

   // Get only the year and month part as string
   string dr1 = dr.ToYearMonthString();
   Console.WriteLine("The YEARMONTH part is: " + dr1);

   // Get the number of years in duration
   Console.WriteLine("Years: " + dr.Years);

   // Get the number of months in duration
   Console.WriteLine("Months: " + dr.Months);
}
```

27.4.4 Altova.Xml.Meta.Attribute

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des attributs. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un attribut dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un attribut particulier défini dans le schéma XML.

Propriétés

	Nom	Description
	`SimpleType DataType`	Retourne le type du contenu d'attribut.
	`string LocalName`	Retourne le nom local de l'attribut.
	`string NamespaceURI`	Retourne l'URI d'espace de noms de l'attribut.
	`XmlQualifiedName QualifiedName`	Retourne le nom qualifié de l'attribut.
	`bool Required()`	Retourne true si l'attribut est requis.

© 2016-2022 Altova GmbH

27.4.5 Altova.Xml.Meta.ComplexType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types complexes. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types complexes dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type complexes particulier défini dans le schéma XML.

Propriétés

	Nom	Description
	`Attribute[] Attributes`	Retourne une liste de tous les attributs.
	`ComplexType BaseType`	Retourne le type de base de ce type ou null si aucun type de base type n'existe.
	`SimpleType ContentType`	Retourne le type simple du contenu.
	`Element[] Elements`	Retourne une liste de tous les éléments.
	`string LocalName`	Retourne le nom local du type.
	`string NamespaceURI`	Retourne l'URI d'espace de noms du type.
	`XmlQualifiedName QualifiedName`	Retourne le nom qualifié de ce type.

Méthodes

	Nom	Description
	`ComplexType BaseType`	Retourne le type de base de ce type.
	`bool Equals(obj)`	Contrôle si deux objets d'info se réfèrent au même type, basé dans la comparaison du nom qualifié. Retourne true si le type a le même nom qualifié.
	`Attribute FindAttribute(string localName, string namespaceURI)`	Trouve l'attribut avec le nom local spécifié et l'URI d'espace de noms
	`Element FindElement(string localName, string namespaceURI)`	Trouve l'élément avec le nom local spécifié et l'URI d'espace de noms.

27.4.6 Altova.Xml.Meta.Element

Cette classe vous permet d'accéder à des informations concernant des classes générées depuis des éléments de schéma. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos

des instances particulières d'un élément dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un élément particulier défini dans le schéma XML.

Propriétés

	Nom	Description
	`ComplexType DataType`	Retourne le type de l'élément. Noter qu'il s'agit toujours d'un type de complexe même s'il est déclaré en tant que simple dans le schéma original. Utiliser la propriété `ContentType` de l'objet retourné pour obtenir le type de contenu simple.
	`string` `LocalName`	Retourne le nom local de l'élément.
	`int` `MaxOccurs`	Retourne la valeur `maxOccurs` définie dans le schéma.
	`int` `MinOccurs`	Retourne la valeur `minOccurs` définie dans le schéma.
	`string` `NamespaceURI`	Retourne l'URI d'espace de noms de l'élément .
	`XmlQualifiedName QualifiedName`	Retourne le nom qualifié de l'élément.

27.4.7 Altova.Xml.Meta.SimpleType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types simples. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types simples dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type simple particulier défini dans le schéma XML.

Propriétés

	Nom	Description
	`SimpleType BaseType`	Retourne le type de base de ce type.
	`string``[] Enumerations`	Retourne une liste de toutes les facettes d'énumération.
	`int` `FractionDigits`	Retourne la valeur de cette facette.
	`int` `Length`	Retourne la valeur de cette facette.
	`string` `LocalName`	Retourne le nom local du type.
	`string` `MaxExclusive`	Retourne la valeur de cette facette.
	`string` `MaxInclusive`	Retourne la valeur de cette facette.
	`int` `MaxLength`	Retourne la valeur de cette facette.
	`string` `MinExclusive`	Retourne la valeur de cette facette.

© 2016-2022 Altova GmbH

	Nom	Description
	string MinInclusive	Retourne la valeur de cette facette.
	int MinLength	Retourne la valeur de cette facette.
	string NamespaceURI	Retourne l'URI d'espace de noms du type.
	string[] Patterns	Retourne les facettes de motif, ou null si aucun motif n'est spécifié.
	XmlQualifiedName QualifiedName	Retourne le nom qualifié de ce type.
	int TotalDigits	Retourne la valeur de cette facette.
	WhitespaceType Whitespace	Retourne la facette de normalisation d'espace blanc.

27.4.8 [YourSchema].[Doc]

Lorsque le code est généré depuis un schéma XML, le code généré fournit une classe de document avec le même nom que le schéma. Cette classe contient tous les éléments root possibles en tant que membres, ainsi que les membres recensés ci-dessous. Veuillez noter que, dans les noms de méthode ci-dessous, "Doc" représente le nom de la classe de document générée elle-même.

Méthodes

	Nom	Description
	static Doc CreateDocument()	Crée un nouveau document XML vide.
	static Doc CreateDocument(**string** encoding)	Crée un nouveau document XML vide, avec l'encodage de type "encoding".
	static void DeclareAllNamespacesFromSchema(nœud **Altova.Xml.TypeBase** node)	Déclare tous les espaces de noms depuis le Schéma XML dans l'élément fournit en dans qu'argument (généralement, l'élément racine XML). L'appel de cette méthode est utile si votre schéma a plusieurs déclarations d'espace de noms, chacune mappée dans un préfixe, et que vous souhaitez les déclarer tous dans l'élément fourni en tant qu'argument.
	static Doc LoadFromBinary(**byte**[] binary)	Charge un document XML depuis un byte array.
	static Doc LoadFromFile(**string** filename)	Charge un document XML depuis un fichier.
	static Doc LoadFromString(**string** xmlstring)	Charge un document XML depuis un string.

	Nom	Description
	`byte[] SaveToBinary(bool prettyPrint)`	Enregistre un document XML sous un de byte, avec un formatage "pretty-print" optionnel.
	`byte[] SaveToBinary(bool prettyPrint, string encoding)`	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
	`byte[] SaveToBinary(bool prettyPrint, string encoding, bool bBigEndian, bool bBOM)`	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié, l'ordre de byte et BOM (Byte Order Mark).
	`void SaveToFile(string fileName, bool prettyPrint)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
	`void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void SaveToFile(string fileName, bool prettyPrint, string encoding, string lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et le(s) caractère(s) de fin de ligne.
	`void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding, string lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et le(s) caractère(s) de fin de ligne. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding, bool bBigEndian, bool bBOM, string lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié, l'ordre de byte, BOM (Byte Order Mark) et le(s) caractère(s) de fin de ligne. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void SaveToFileWithLineEnd(string fileName, bool prettyPrint, bool omitXmlDecl, string lineend)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel et le(s) caractère(s) de fin de ligne. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`string SaveToString(bool prettyPrint)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
	`string SaveToString(bool prettyPrint, bool omitXmlDecl)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.

© 2016-2022 Altova GmbH

	Nom	Description
	`void SetDTDLocation(string dtdLocation)`	Ajoute une déclaration DOCTYPE avec l'ID de système spécifiée. Un élément root doit déjà exister.
	`void SetSchemaLocation(string schemaLocation)`	Ajoute un attribut xsi:schemaLocation ou xsi:noNamespaceSchemaLocation à l'élément root. Un élément root doit déjà exister.

27.4.9 [YourSchema].[ElementType]

Cette classe propose des méthodes pour manipuler des éléments XML depuis votre schéma. Des méthodes de cette classe peuvent être appelées sur des éléments, pas sur le document XML lui-même. Noter qu'il n'est pas nécessaire d'instancier la classe directement pour appeler des méthodes de cette classe. Tout élément créé en utilisant les méthodes **`Append()`** ou **`AppendWithPrefix()`** est de type `[ElementType]`.

Méthodes

	Nom	Description

	`void DeclareNamespace(string prefix, string nsURI)`	Cette méthode prend deux arguments qui sont tous les deux de type string : le préfixe et l'URI d'espace de noms que vous souhaitez utiliser. Le préfixe fournit en tant qu'argument sera mappé dans la valeur URI d'espace de noms fournie en tant qu'argument. Si le préfixe fourni en tant qu'argument est vide, la méthode crée ou contourne la déclaration d'espace de noms par défaut dans l'élément.
		
		Par exemple, assumons que le document XML a un élément XML appelé "purchase". Si vous appelez
		
		`purchase.DeclareNamespace("ord", "http://OrderTypes");`
		
		Alors le document XML devient
		
		`<purchase xmlns:ord="http://OrderTypes" />`
		
		Un autre exemple ; si vous appelez :
		
		`purchase.DeclareNamespace("", "http://OrderTypes");`
		
		Alors le document XML devient
		
		`<purchase xmlns="http://OrderTypes" />`
		
		Note : l'espace de noms déclaré est utilisé lorsque les

	Nom	Description
		éléments ou les attributs enfants ultérieurs, conformément aux règles suivantes : 1. si l'espace de noms enfant est le défaut, utiliser le préfixe vide. 2. si l'espace de noms enfant est égal à l'espace de noms parent, utiliser le préfixe parent. 3. Sinon, chercher le préfixe le plus proche en commençant du parent vers le haut, en utilisant l'algorithme lookup décrit dans la section "B.2: Namespace Prefix Lookup" at https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html. 4. Si aucun préfixe n'a été trouvé pour l'espace de noms élément, utiliser un préfixe vide.

27.4.10 [YourSchemaType].MemberAttribute

Lorsque le code est généré depuis un schéma XML, une classe est créée pour chaque attribut membre d'un type. Dans les descriptions ci-dessous, "AttributeType" représente le type de l'attribut de membre lui-même.

Méthodes

	Nom	Description
	`bool Exists()`	Retourne true si l'attribut existe.
	`void Remove()`	Supprime l'attribut depuis son élément parent.

Propriétés

	Nom	Description
	**`int`** `EnumerationValue`	Généré pour des types d'énumération uniquement. Définit ou obtient la valeur d'attribut en utilisant une des constantes générée pour les valeurs possibles. Retourne **Invalid** si les valeurs ne correspondent pas à une des valeurs énumérées dans le schéma.
	`Altova.Xml.Meta.Attribute Info`	Retourne un objet pour l'information de schéma de requête (voir `Altova.Xml.Meta.Attribute`).
	`AttributeType Value`	Définit ou obtient la valeur d'attribut.

© 2016-2022 Altova GmbH

27.4.11 [YourSchemaType].MemberElement

Lorsque le code est généré depuis un schéma XML, une classe avec les membres suivants est créée pour chaque élément de membre d'un type. La classe met en place l'interface standard `System.Collections.IEnumerable`, donc il peut être utilisé avec l'instruction `foreach`.

Dans les descriptions ci-dessous, "MemberType" représente le type de l'élément de membre lui-même.

Méthodes

	Nom	Description
	`MemberType Append()`	Crée un nouvel élément et l'ajoute à son parent.
	`MemberType AppendWithPrefix(string prefix)`	Crée un nouvel élément ayant le préfixe fournit en tant qu'argument, et l'appose à son parent. Pour consulter un exemple, voir Exemple : Purchase Order.
	`MemberType At(int index)`	Retourne l'élément de membre spécifié par l'index.
	`System.Collections.IEnumerator GetEnumerator()`	Retourne un objet pour itérer des instances de l'élément de membre.
	`void Remove()`	Supprime toutes les occurrences de l'élément depuis son parent.
	`void RemoveAt(int index)`	Supprime l'occurrence de l'élément spécifié par l'index.

Propriétés

	Nom	Description
	`int Count`	Retourne le décompte des éléments.
	`int EnumerationValue`	Généré pour des types d'énumération uniquement. Définit ou obtient la valeur d'élément en utilisant une des constantes générée pour les valeurs possibles. Retourne **Invalid** si les valeurs ne correspondent pas à une des valeurs énumérées dans le schéma.
	`bool Exists`	Retourne true si au moins un élément existe.
	`MemberType First`	Retourne la première instance de l'élément de membre.
	`Altova.Xml.Meta.Element Info`	Retourne un objet pour l'information de schéma de requête (voir `Altova.Xml.Meta.Element`).
	`MemberType Last`	Retourne la dernière instance de l'élément de membre.
	`MemberType this[int index]`	Retourne l'élément de membre spécifié par l'index.

	Nom	Description
	`MemberType Value`	Définit ou obtient le contenu d'élément (uniquement généré si l'élément peut avoir un contenu simple ou mixte).

© 2016-2022 Altova GmbH

27.5 Référence aux classes générées (Java)

Ce chapitre contient une description des classes Java générées avec XMLSpy depuis un schéma XML ou DTD (voir Générer du code depuis des Schémas XML ou DTDs). Vous pouvez intégrer ces classes dans votre code pour lire, modifier et écrire des documentation XML.

Note: Le code généré peut inclure d'autres classes de support, qui ne sont pas recensées ici et qui sont sujettes à des modifications.

27.5.1 com.altova.types.DateTime

Cette classe vous permet de traiter des attributs XML ou des éléments qui ont des types de date et d'heure, comme `xs:dateTime`.

Constructeurs

	Nom	Description
C	`public DateTime()`	Initialise une nouvelle instance de la classe `DateTime` dans une valeur vide.
C	`public DateTime(DateTime newvalue)`	Initialise une nouvelle instance de la classe `DateTime` dans l'objet `DateTime` fourni en tant qu'argument.
C	`public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, int newoffsetTZ)`	Initialise une nouvelle instance de la classe `DateTime` à l'année, mois, jour, heure, minute, seconde, la partie fractionnaire de la seconde et fuseau horaire fournis en tant qu'argument. La partie fractionnaire de la seconde newpartsecond doit se situer entre 0 et 1. L'offset de fuseau horaire newoffsetTZ peut être soit positif soit négatif et s'exprime en minutes.
C	`public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond)`	Initialise une nouvelle instance de la classe `DateTime` dans l'année, le mois, le jour, l'heure, la minute, la seconde et la partie fractionnelle d'une seconde fournie en tant qu'arguments.
C	`public DateTime(int newyear, int newmonth, int newday)`	Initialise une nouvelle instance de la classe `DateTime` pour l'année, le mois et le jour fournis en tant qu'arguments.
C	`public DateTime(Calendar newvalue)`	Initialise une nouvelle instance de la classe `DateTime` dans la valeur `java.util.Calendar` fournie en tant qu'argument.

Méthodes

	Nom	Description
S	`static DateTime now()`	Retourne l'heure actuelle en tant qu'un objet `DateTime`.

	Nom	Description
S	**static** DateTime parse(String s)	Retourne un objet DateTime parsé depuis la valeur de string fournie en tant qu'argument. Par exemple, les valeurs de string d'échantillon suivantes sont converties avec succès dans un objet DateTime : 2015-11-24T12:54:47.969+01:00 2015-11-24T12:54:47 24/11/2015
	int getDay()	Retourne le jour de l'instance actuelle de DateTime.
	int getHour()	Retourne l'heure de l'instance actuelle de DateTime.
	int getMillisecond()	Retourne la milliseconde de l'instance actuelle de DateTime, en tant qu'une valeur d'entier.
	int getMinute()	Retourne la minute de l'instance actuelle DateTime.
	int getMonth()	Retourne le mois de l'instance actuelle DateTime.
	double getPartSecond()	Retourne la partie fractionnaire de la seconde de l'instance actuelle de DateTime, en tant que valeur **double**. La valeur de retour est supérieure à zéro et inférieure à un, par exemple : 0.313
	int getSecond()	Retourne la seconde de l'instance actuelle DateTime.
	int getTimezoneOffset()	Retourne l'offset de fuseau horaire, en minutes de l'instance actuelleDateTime. Par exemple, le fuseau horaire "UTC-01:00" serait retourné en tant que : -60
	Calendar getValue()	Retourne l'instance actuelle de DateTime en tant qu'une valeur java.util.Calendar.
	int getWeekday()	Retourne le jour dans la semaine de l'instance actuelle DateTime. Les valeurs s'étendent de 0 à 6, 0 étant lundi (ISO-8601).
	int getYear()	Retoune l'année de l'instance actuelle DateTime.
	int hasTimezone()	Retourne l'information concernant le fuseau horaire de l'instance actuelle DateTime. Les valeurs de retour possibles sont : CalendarBase.TZ_MISSING — Un offset de fuseau horaire n'est pas défini. CalendarBase.TZ_UTC — Le fuseau horaire est UTC. CalendarBase.TZ_OFFSET — Un offset de fuseau d'horaire a été défini.

© 2016-2022 Altova GmbH

	Nom	Description
	`void setDay( int nDay )`	Définit le jour de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`void setHasTimezone( int nHasTZ )`	Définit l'information de fuseau horaire de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument. Cette méthode peut être utilisée pour supprimer l'information de fuseau horaire ou définir le fuseau horaire sur UTC (Coordinated Universal Time). Les valeurs valides pour l'argument nHasTZ : `CalendarBase.TZ_MISSING` Configurer l'offset de fuseau horaire sur non-défini. `CalendarBase.TZ_UTC` Configurer le fuseau horaire sur UTC. `CalendarBase.TZ_OFFSET` Si l'objet actuel a un offset de fuseau horaire, ne pas le modifier.
	`void setHour( int nHour )`	Définit l'heure de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`void setMinute( int nMinute )`	Définit la minute de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`void setMonth( int nMonth )`	Définit le mois de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`void setPartSecond( double nPartSecond )`	Définit la partie fractionnaire de la seconde de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`void setSecond( int nSecond )`	Définit la seconde de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`void setTimezoneOffset( int nOffsetTZ )`	Définit l'offset de fuseau horaire de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument. La valeur nOffsetTZ doit être un entier (positif ou négatif) et doit être exprimée en minutes.
	`void setYear( int nYear )`	Définit l'année de l'instance actuelle `DateTime` à la valeur fournie en tant qu'argument.
	`String toString()`	Retourne la représentation de string de l'instance actuelle `DateTime`, par exemple : `2015-11-24T15:50:56.968+01:00`

Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

```
import com.altova.types.*;
```

L'extrait de code suivant illustre plusieurs moyens de créer des objets `DateTime` :

```
protected static void DateTimeExample1()
{
    // Initialize a new instance of the DateTime class to the current time
   DateTime dt = new DateTime(DateTime.now());
   System.out.println("DateTime created from current date and time: " + dt.toString());

   // Initialize a new instance of the DateTime class by supplying the parts
   DateTime dt1 = new DateTime(2015, 11, 23, 14, 30, 24, .459);
   System.out.println("DateTime from parts (no timezone): " + dt1.toString());

   // Initialize a new instance of the DateTime class by supplying the parts
   DateTime dt2 = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
   System.out.println("DateTime from parts (with negative timezone): " + dt2.toString());

   // Initialize a new instance of the DateTime class by parsing a string value
   DateTime dt3 = DateTime.parse("2015-11-24T12:54:47.969+01:00");
   System.out.println("DateTime parsed from string: " + dt3.toString());
}
```

The following code listing illustrates getting values from `DateTime` objects:

```
protected static void DateTimeExample2()
   {
      // Initialize a new instance of the DateTime class to the current time
      DateTime dt = new DateTime(DateTime.now());

      // Output the formatted year, month, and day of this DateTime instance
      String str1 = String.format("Year: %d; Month: %d; Day: %d;", dt.getYear(),
dt.getMonth(), dt.getDay());
      System.out.println(str1);

      // Output the formatted hour, minute, and second of this DateTime instance
      String str2 = String.format("Hour: %d; Minute: %d; Second: %d;", dt.getHour(),
dt.getMinute(), dt.getSecond());
      System.out.println(str2);

      // Return the timezone (in minutes) of this DateTime instance
      System.out.println("Timezone: " + dt.getTimezoneOffset());

      // Get the DateTime as a java.util.Calendar value
      java.util.Calendar dt_java = dt.getValue();
      System.out.println("" + dt_java.toString());

      // Return the day of week of this DateTime instance
      System.out.println("Weekday: " + dt.getWeekday());

      // Check whether the DateTime instance has a timezone defined
      switch(dt.hasTimezone())
      {
         case CalendarBase.TZ_MISSING:
```

 © 2016-2022 Altova GmbH

```
            System.out.println("No timezone.");
            break;
         case CalendarBase.TZ_UTC:
            System.out.println("The timezone is UTC.");
            break;
         case CalendarBase.TZ_OFFSET:
            System.out.println("This object has a timezone.");
            break;
         default:
            System.out.println("Unable to determine whether a timezone is defined.");
            break;
      }
   }
```

The following code listing illustrates changing the timezone offset of a `DateTime` object:

```
protected static void DateTimeExample3()
{
   // Create a new DateTime object with timezone -0100 UTC
   DateTime dt = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
   // Output the value before the change
   System.out.println("Before: " + dt.toString());
   // Change the offset to +0100 UTC
   dt.setTimezoneOffset(60);
   // Output the value after the change
   System.out.println("After:  " + dt.toString());
}
```

27.5.2 com.altova.types.Duration

Cette classe vous permet de traiter des attributs XML ou des éléments de type `xs: duration` .

Constructeurs

	Nom	Description
C	`Duration(Duration newvalue)`	Initialise une nouvelle instance de la classe `Duration` dans l'objet `Duration` fournie en tant qu'argument.
C	`Duration(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, boolean newisnegative)`	Initialise une nouvelle instance de la classe `Duration` dans une durée construite depuis des parties fournies en tant qu'argument.

Méthodes

	Nom	Description
S	`static Duration getFromDayTime( int newday, int newhour, int newminute, int newsecond, double newpartsecond )`	Retourne un objet `Duration` créé depuis le nombre de jours, heures, minutes, secondes et partie de seconde fractionnelle fourni en tant qu'argument.
S	`static Duration getFromYearMonth( int newyear, int newmonth )`	Retourne un objet `Duration` créé depuis le nombre d'années et de mois fourni en tant qu'argument.
S	`static Duration parse( String s )`	Retourne un objet `Duration` créé depuis le string fourni en tant qu'argument. Par exemple, le string -P1Y1M1DT1H1M1.333S peut être utilisé pour créer une durée négative d'une année, d'un mois, d'un jour, d'une heure, d'une minute, d'une seconde et des parties fractionnelles 0.333 d'une seconde. Pour créer une durée négative, apposer le signe moins (-) au string.
S	`static Duration parse( String s, ParseType pt )`	Retourne un objet `Duration` créé depuis le string fourni en tant qu'argument, en utilisant un format de parsage spécifique. Le format de parsage peut être un des suivants : **ParseType.DAYTIME** Peut être utilisé lorsque le string **s** consiste en un des éléments suivants : jours, heures, minutes, secondes, parties fractionnelles de la seconde, par exemple -P4DT4H4M4.774S.

© 2016-2022 Altova GmbH

	Nom	Description
		ParseType.DURATION Peut être utilisé lorsque le string **s** consiste en un des éléments suivants : années, mois, jours, heures, minutes, secondes, parties fractionnelles de la seconde, par exemple P1Y1M1DT1H1M1.333S. **ParseType.YEARMONTH** Peut être utilisé lorsque le string **s** consiste en un des éléments suivants : années, mois. Par exemple : P3Y2M.
	int getDay()	Retourne le nombre de jours dans l'instance Duration actuelle.
	long getDayTimeValue()	Retourne la valeur de jour et d'heure (en millisecondes) de l'instance Duration actuelle. Les années et le mois sont ignorés.
	int getHour()	Retourne le nombre d'heures dans l'instance Duration actuelle.
	int getMillisecond()	Retourne le nombre de millisecondes dans l'instance Duration actuelle.
	int getMinute()	Retourne le nombre de minutes dans l'instance Duration actuelle.
	int getMonth()	Retourne le nombre de mois dans l'instance Duration actuelle.
	double getPartSecond()	Retourne le nombre de parties de secondes fractionnelles dans l'instance Duration actuelle.
	int getSecond()	Retourne le nombre de secondes dans l'instance Duration actuelle.
	int getYear()	Retourne le nombre des années dans l'instance Duration actuelle.
	int getYearMonthValue()	Retourne la valeur (en mois) de l'année et du mois de l'instance Duration actuelle. Les jours, heures, secondes et millisecondes sont ignorés.
	boolean isNegative()	Retourne Boléenne **true** si l'instance Duration actuelle est négative.
	void setDayTimeValue(**long** l)	Définit la durée dans le nombre de millisecondes fournies en tant qu'argument, ne touchant que la partie jour et heure de la durée.
	void setNegative(**boolean** isnegative)	Convertit l'instance actuelle Duration en une durée négative.
	void setYearMonthValue(**int** l)	Définit la durée dans le nombre de mois fournis en tant qu'argument. Seule la partie années et mois de la durée est

	Nom	Description
		touchée.
	`String toString()`	Retourne la représentation de string de l'instance actuelle `Duration`, par exemple : `-P4DT4H4M4.774S`
	`String toYearMonthString()`	Retourne la représentation de string de la partie YearMonth de l'instance actuelle `Duration`, par exemple : `P1Y2M`

Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

```
import com.altova.types.*;
import com.altova.types.Duration.ParseType;
```

L'extrait de code suivant illustre plusieurs moyens de créer des objets `Duration` :

```
protected static void ExampleDuration()
{
   // Create a negative duration of 1 year, 1 month, 1 day, 1 hour, 1 minute, 1 second,
   // and 0.333 fractional second parts
   Duration dr = new Duration(1, 1, 1, 1, 1, 1, .333, true);

   // Create a duration from an existing Duration object
   Duration dr1 = new Duration(dr);

   // Create a duration of 4 days, 4 hours, 4 minutes, 4 seconds, .774 fractional second
parts
   Duration dr2 = Duration.getFromDayTime(4, 4, 4, 4, .774);

   // Create a duration of 3 years and 2 months
   Duration dr3 = Duration.getFromYearMonth(3, 2);

   // Create a duration from a string
   Duration dr4 = Duration.parse("-P4DT4H4M4.774S");

   // Create a duration from a string, using specific parse formats
   Duration dr5 = Duration.parse("-P1Y1M1DT1H1M1.333S", ParseType.DURATION);
   Duration dr6 = Duration.parse("P3Y2M", ParseType.YEARMONTH);
   Duration dr7 = Duration.parse("-P4DT4H4M4.774S", ParseType.DAYTIME);
}
```

L'extrait de code suivant illustre l'obtention et la configuration d'objets `Duration` :

© 2016-2022 Altova GmbH

```
protected static void DurationExample2()
{
   // Create a duration of 1 year, 2 month, 3 days, 4 hours, 5 minutes, 6 seconds,
       // and 333 milliseconds
   Duration dr = new Duration(1, 2, 3, 4, 5, 6, .333, false);
   // Output the number of days in this duration
   System.out.println(dr.getDay());

   // Create a positive duration of one year and 333 milliseconds
   Duration dr1 = new Duration(1, 0, 0, 0, 0, 0, .333, false);
   // Output the day and time value in milliseconds
   System.out.println(dr1.getDayTimeValue());

   // Create a positive duration of 1 year, 1 month, 1 day, 1 hour, 1 minute, 1 second,
       // and 333 milliseconds
   Duration dr2 = new Duration(1, 1, 1, 1, 1, 1, .333, false);
   // Output the year and month value in months
   System.out.println(dr2.getYearMonthValue());

   // Create a positive duration of 1 year and 1 month
   Duration dr3 = new Duration(1, 1, 0, 0, 0, 0, 0, false);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Set the DayTime part of duration to 1000 milliseconds
   dr3.setDayTimeValue(1000);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Set the YearMonth part of duration to 1 month
   dr3.setYearMonthValue(1);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Output the year and month part of the duration
   System.out.println("The YearMonth part of the duration is: " +
dr3.toYearMonthString());
}
```

27.5.3 com.altova.xml.meta.Attribute

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des attributs. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un attribut dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un attribut particulier défini dans le schéma XML.

Méthodes

	Nom	Description
●	`SimpleType getDataType()`	Retourne le type du contenu d'attribut.
●	`String getLocalName()`	Retourne le nom local de l'attribut.

	Nom	Description
	`String getNamespaceURI()`	Retourne l'URI d'espace de noms de l'attribut.
	`boolean` `isRequired()`	Retourne true si l'attribut est requis.

27.5.4 com.altova.xml.meta.ComplexType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types complexes. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types complexes dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type complexes particulier défini dans le schéma XML.

Méthodes

	Nom	Description
	`Attribute findAttribute(String localName, String namespaceURI)`	Trouve l'attribut avec le nom local spécifié et l'URI d'espace de noms
	`Element findElement(String localName, String namespaceURI)`	Trouve l'élément avec le nom local spécifié et l'URI d'espace de noms.
	`Attribute[] GetAttributes()`	Retourne une liste de tous les attributs.
	`ComplexType getBaseType()`	Retourne le type de base de ce type.
	`SimpleType getContentType()`	Retourne le type simple du contenu.
	`Element[] GetElements()`	Retourne une liste de tous les éléments.
	`String getLocalName()`	Retourne le nom local du type.
	`String getNamespaceURI()`	Retourne l'URI d'espace de noms du type.

27.5.5 com.altova.xml.meta.Element

Cette classe vous permet d'accéder à des informations concernant des classes générées depuis des éléments de schéma. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un élément dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un élément particulier défini dans le schéma XML.

© 2016-2022 Altova GmbH

Méthodes

	Nom	Description
	`ComplexType getDataType()`	Retourne le type de l'élément. Noter qu'il s'agit toujours d'un type de complexe même s'il est déclaré en tant que simple dans le schéma original. Utiliser `getContentType()` de l'objet retourné pour obtenir le type de contenu simple.
	`String getLocalName()`	Retourne le nom local de l'élément.
	`int getMaxOccurs()`	Retourne la valeur maxOccurs définie dans le schéma.
	`int getMinOccurs()`	Retoune la valeur minOccurs définie dans le schéma.
	`String getNamespaceURI()`	Retoune l'URI d'espace de noms de l'élément .

27.5.6 com.altova.xml.meta.SimpleType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types simples. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types simples dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type simple particulier défini dans le schéma XML.

Méthodes

	Nom	Description
	`SimpleType getBaseType()`	Retourne le type de base de ce type.
	`String[] getEnumerations()`	Retourne un array de toutes les facettes d'énumération.
	`int getFractionDigits()`	Retourne la valeur de cette facette.
	`int getLength()`	Retourne la valeur de cette facette.
	`String getLocalName()`	Retourne le nom local du type.
	`String getMaxExclusive()`	Retourne la valeur de cette facette.
	`String getMaxInclusive()`	Retourne la valeur de cette facette.
	`int getMaxLength()`	Retourne la valeur de cette facette.
	`String getMinExclusive()`	Retourne la valeur de cette facette.
	`String getMinInclusive()`	Retourne la valeur de cette facette.
	`int getMinLength()`	Retourne la valeur de cette facette.
	`String getNamespaceURI()`	Retourne l'URI d'espace de noms du type.

	Nom	Description
	`String[] getPatterns()`	Retoune un array de toutes les facettes de motif.
	`int getTotalDigits()`	Retourne la valeur de cette facette.
	`int getWhitespace()`	Retourne la valeur de la facette d'espace blanc, qui est un de : `com.altova.typeinfo.WhitespaceType.Whitespace_Unknown` `com.altova.typeinfo.WhitespaceType.Whitespace_Preserve` `com.altova.typeinfo.WhitespaceType.Whitespace_Replace` `com.altova.typeinfo.WhitespaceType.Whitespace_Collapse`

27.5.7 com.[YourSchema].[Doc]

Lorsque le code est généré depuis un schéma XML, le code généré fournit une classe de document avec le même nom que le schéma. Cette classe contient tous les éléments root possibles en tant que membres, ainsi que les membres recensés ci-dessous. Veuillez noter que, dans les noms de méthode ci-dessous, "Doc" représente le nom de la classe de document générée elle-même.

Méthodes

	Nom	Description
S	`static Doc createDocument()`	Crée un nouveau document XML vide.
S	`static void declareAllNamespacesFromSchema(com.altova.xml.ElementType node)`	Déclare tous les espaces de noms depuis le Schéma XML dans l'élément fournit en dans qu'argument (généralement, l'élément racine XML). L'appel de cette méthode est utile si votre schéma a plusieurs déclarations d'espace de noms, chacune mappée dans un préfixe, et que vous souhaitez les déclarer tous dans l'élément fourni en tant qu'argument.
S	`static Doc loadFromBinary(byte[] xml)`	Charge un document XML depuis un byte array.
S	`static Doc loadFromFile(String fileName)`	Charge un document XML depuis un fichier.
S	`static Doc loadFromString(String xml)`	Charge un document XML depuis un string.
	`byte[] saveToBinary(boolean prettyPrint)`	Enregistre un document XML sous un de byte, avec un formatage "pretty-print" optionnel.
	`byte[] saveToBinary(boolean prettyPrint, String encoding)`	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
	`byte[] saveToBinary(boolean prettyPrint, String encoding, boolean bigEndian, boolean writeBOM)`	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La

© 2016-2022 Altova GmbH

	Nom	Description
		marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
	`void saveToFile(String fileName, boolean prettyPrint)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
	`void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage UTF-8. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl, String encoding)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl, String encoding, boolean bBigEndian, boolean bBOM)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
	`void saveToFile(String fileName, boolean prettyPrint, String encoding)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
	`void saveToFile(String fileName, boolean prettyPrint, String encoding, boolean bBigEndian, boolean bBOM)`	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
	`String saveToString(boolean prettyPrint)`	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel.
	`String saveToString(boolean prettyPrint, boolean omitXmlDecl)`	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel. Si `omitXmlDecl` est true, la déclaration XML ne sera pas rédigée.
	`void setSchemaLocation(String schemaLocation)`	Ajoute un attribut `xsi:schemaLocation` ou `xsi:noNamespaceSchemaLocation` à l'élément root. Un élément root doit déjà exister.

27.5.8 com.[YourSchema].[ElementType]

Cette classe propose des méthodes pour manipuler des éléments XML depuis votre schéma. Des méthodes de cette classe peuvent être appelées sur des éléments, pas sur le document XML lui-même. Noter qu'il n'est pas nécessaire d'instancier la classe directement pour appeler des méthodes de cette classe. Tout élément créé en utilisant les méthodes **`append()`** ou **`appendWithPrefix()`** est de type `ElementType`.

Méthodes

	Nom	Description

	`void declareNamespace(String prefix, String nsURI)`	Cette méthode prend deux arguments qui sont tous les deux de type string : le préfixe et l'URI d'espace de noms que vous souhaitez utiliser. Le préfixe fournit en tant qu'argument sera mappé dans la valeur URI d'espace de noms fournie en tant qu'argument. Si le préfixe fourni en tant qu'argument est vide, la méthode crée ou contourne la déclaration d'espace de noms par défaut dans l'élément.
		
		Par exemple, assumons que le document XML a un élément XML appelé "purchase". Si vous appelez
		
		`purchase.declareNamespace("ord", "http://OrderTypes");`
		
		Alors le document XML devient
		
		`<purchase xmlns:ord="http://OrderTypes" />`
		
		Un autre exemple ; si vous appelez :
		
		`purchase.declareNamespace("", "http://OrderTypes");`
		
		Alors le document XML devient
		
		`<purchase xmlns="http://OrderTypes" />`
		
		Note : l'espace de noms déclaré est utilisé lorsque les éléments ou les attributs enfants ultérieurs, conformément aux règles suivantes :
		
		1. si l'espace de noms enfant est le défaut, utiliser le préfixe vide.
		2. si l'espace de noms enfant est égal à l'espace de noms parent, utiliser le préfixe parent.
		3. Sinon, chercher le préfixe le plus proche en commençant du parent vers le haut, en utilisant l'algorithme lookup décrit dans la section "B.2: Namespace Prefix Lookup" at https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html.
		4. Si aucun préfixe n'a été trouvé pour l'espace de noms élément, utiliser un préfixe vide.

© 2016-2022 Altova GmbH

27.5.9 com.[YourSchema].[YourSchemaType].MemberAttribute

Lorsque le code est généré depuis un schéma XML, une classe est créée pour chaque attribut membre d'un type. Dans les descriptions ci-dessous, "AttributeType" représente le type de l'attribut de membre lui-même.

Méthodes

	Nom	Description
	`boolean exists()`	Retourne true si l'attribut existe.
	`int getEnumerationValue()`	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou **Invalid** si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.
	`com.altova.xml.meta.Attribute getInfo()`	Retourne un objet pour l'information de schéma de requête (voir `com.altova.xml.meta.Attribute`).
	`AttributeType getValue()`	Définit ou obtient la valeur d'attribut.
	`void remove()`	Supprime l'attribut depuis son élément parent.
	`void setEnumerationValue(int)`	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.
	`void setValue(AttributeType value)`	Définit la valeur d'attribut.

27.5.10 com.[YourSchema].[YourSchemaType].MemberElement

Lorsque le code est généré depuis un schéma XML, une classe avec les membres suivants est créée pour chaque élément de membre d'un type. Dans les descriptions ci-dessous, "MemberType" représente le type de l'élément de membre lui-même.

Méthodes

	Nom	Description
	`MemberType append()`	Crée un nouvel élément et l'ajoute à son parent.
	`MemberType appendWithPrefix(String prefix)`	Crée un nouvel élément ayant le préfixe fournit en tant qu'argument, et l'appose à son parent. Pour consulter un exemple, voir Exemple : Purchase Order.
	`MemberType at(int index)`	Retourne l'instance de l'élément membre en tant que l'index spécifié.
	`int count()`	Retourne le décompte des éléments.

	Nom	Description
	`boolean exists()`	Retourne true si au moins un élément existe.
	`MemberType first()`	Retourne la première instance de l'élément de membre.
	`int getEnumerationValue()`	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou **Invalid** si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.
	`com.altova.xml.meta.Element getInfo()`	Retourne un objet pour l'information de schéma de requête (voir `com.altova.xml.meta.Element`).
	`MemberType getValue()`	Obtient le contenu de l'élément (uniquement généré si l'élément peut avoir un contenu simple ou mixte).
	`java.util.Iterator iterator()`	Retourne un objet pour itérer des instances de l'élément de membre.
	`MemberType last()`	Retourne la dernière instance de l'élément de membre.
	`void remove()`	Supprime toutes les occurrences de l'élément depuis son parent.
	`void removeAt(int index)`	Supprime l'occurrence de l'élément spécifié par l'index.
	`void setEnumerationValue(int index)`	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.
	`void setValue(MemberType value)`	Définit le contenu d'élément (uniquement généré si l'élément peut avoir un contenu simple ou mixte).

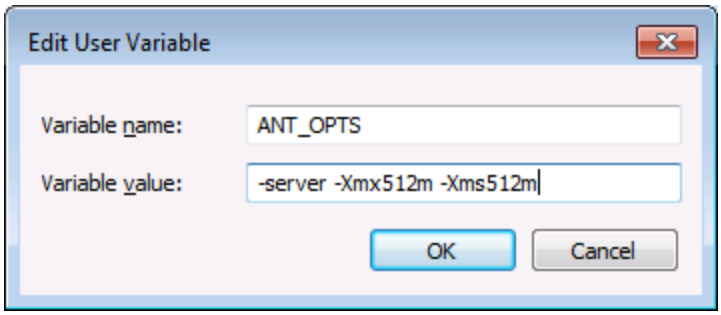
© 2016-2022 Altova GmbH

27.6 Astuces de Génération de Code

Résoudre les exceptions "Mémoire insuffisante" pendant la compilation Java

Schémas complexes Peuvent produire une grande quantité de code, qui peut entraîner une exception `java.lang.OutofMemory` pendant la compilation en utilisant Ant. Pour corriger cela :

- Ajouter la variable d'environnement `ANT_OPTS`, qui définit des options Ant spécifiques comme la mémoire à attribuer à l'ordinateur, et configurer sa valeur telle qu'indiqué ci-dessous.



- Pour vous assurer que le compileur et le code généré soient exécutés dans le même processus qu'Ant, changer l'attribut `fork`, dans **build.xml**, sur `false`.

Vous devrez éventuellement personnaliser les valeurs selon la quantité de mémoire disponible dans votre appareil et la taille du projet sur lequel vous travaillez. Pour plus de détails, voir votre documentation Java VM.

Lorsque vous exécutez la commande `ant jar`, vous recevrez éventuellement un message d'erreur de type "`[...] archive contains more than 65535 entities`". Afin d'éviter cela, il est recommandé d'utiliser Ant 1.9 ou plus, et dans le fichier **build.xml**, d'ajouter `zip64mode="as-needed"` dans l'élément `<jar>`.

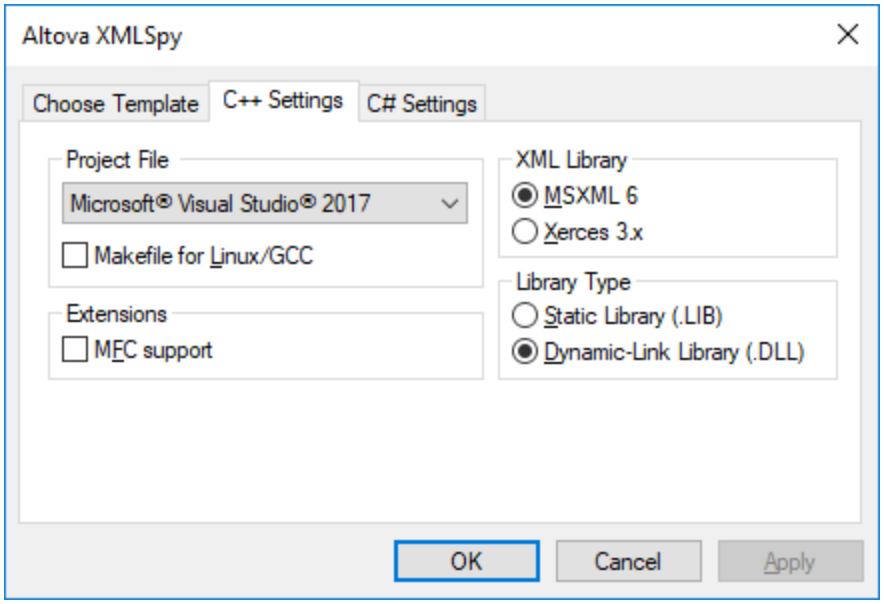
Réserver des noms de méthode

Lors de la personnalisation de la génération de code en utilisant les fichiers SPL fournis, il peut être nécessaire de réserver des noms pour éviter toute collision avec d'autres symboles. Pour ce faire :

1. Naviguer dans le répertoire d'installation du programme, par exemple, `C:\Program Files\Altova\XMLSpy2023`.
2. Dans le sous-répertoire **spl**, trouver le répertoire correspondant au langage de programme, par exemple, **..\spl\java**.
3. Ouvrir le fichier **settings.spl** et insérer une nouvelle ligne dans la section reserve par ex. **`reserve "myReservedWord"`**.
4. Régénérer le code de programme.

27.7 Options du Générateur de Code

L'option de menu **DTD/Schéma| Générer Code de Programme** vous permet de spécifier le langage de programmation cible ainsi que les paramètres de génération de code spécifiques.



Les paramètres disponibles sont les suivants

Paramètres C++	Définit les paramètre de compileur spécifique pour l'environnement C++, concrètement : • La version Visual Studio (2013, 2015, 2017, 2019, 2022) • Si un makefile pour Linux avec un compileur GCC doit être généré. • La bibliothèque XML (MSXML, Xerces 3.x) • Si les bibliothèques statiques ou dynamiques doivent être générées • Si le code doit être généré avec ou sans la prise en charge MFC L'option **Makefile pour Linux/GCC** ajoute des makefiles au code généré. Des fichiers de source C++ sont générés de manière à être portables en utilisant des constructions `#ifdef` pour prendre en charge des compileurs différents et des systèmes d'exploitation. Veuillez noter l'élément suivant si vous avez l'intention de compiler de code généré avec GCC (GNU Compiler Collection) sur Linux : • Pour la compilation Linux/GCC, la seule bibliothèque XML prise en charge est **Xerces 3.x**. • Cocher la case **Support MFC** n'a aucun effet sur la compilation avec Linux/GCC.

© 2016-2022 Altova GmbH

Paramètres C#	Sélectionnez l'option **Microsoft .NET Core 3.1** ou **Microsoft .NET 5.0** ou **Microsoft .NET 6.0** pour générer une solution Visual Studio ciblant les plate-formes respectives. Si vous devez cibler la plate-forme **.NET Framework** pour une version spécifique de Visual Studio, sélectionnez toute option de **Microsoft Visual Studio 2010-2019** - dans ce cas, la solution générée ciblera la version .NET Framework correspondant à la version respective de Visual Studio.

27.8 Référence SPL

Cette section donne un aperçu de SPL (Spy Programming Language), le langage de modèle du générateur de code.

Il set recommandé d'avoir une certaine expérience de la programmation, et de maîtriser les opérateurs, les fonctions, les variables et les classes, ainsi que les bases de la programmation orientée vers l'objet - très largement utilisé dans SPL. Vous devriez aussi avoir des connaissances détaillées du Schéma XML.

Les modèles utilisés par XMLSpy sont fournis dans le dossier ...\XMLSpy\spl. Vous pouvez utiliser ces fichiers en tant qu'aide pour vous aider à développer vos propres modèles.

Comment fonctionne le générateur de code

Les entrées dans le générateur de code sont les fichiers de modèle (.spl) et le modèle d'objet fourni par XMLSpy. Les fichiers de modèle contiennent des instructions SPL pour créer des fichiers, lire des informations depuis le modèle d'objet et effectuer des calculs, ponctué de fragments de code littéral dans le langage de programmation cible.

Le fichier de modèle est interprété par le générateur de code et les produit des fichiers de code source **.cpp, .java, .cs**, des fichiers de projet, ou tout autre type de fichier dépendant du modèle. Le code de source peut ensuite être compilé dans un fichier exécutable qui accède aux données XML décrites par le fichier de schéma.

Les fichiers SPL ont accès à une grande variété d'informations recueillie depuis les schémas de source. Veuillez noter qu'un fichier SPL n'est pas lié à un schéma spécifique, mais permet l'accès à tous les schémas. Veuillez vous assurer d'écrire vos fichiers SPL génériquement et d'éviter les structures qui s'appliquent aux schémas spécifiques.

Exemple : Créer un nouveau fichier dans SPL

Voici un fichier SPL de base. Il crée un fichier nommé **test.cpp,** et place l'instruction include à l'intérieur. La commande close achève le modèle.

```
[create "test.cpp"]
#include "stdafx.h"
[close]
```

27.8.1 Structure SPL de base

Un fichier SPL contient du texte littéral de sortie, mêlé d'instructions de générateur de code.

Les instructions du générateur de code sont contenues dans des crochets '**[**' et '**]**'. Plusieurs instructions peuvent être contenues dans une paire de crochets. Des instructions supplémentaires doivent être séparées par une nouvelle ligne ou un double point '**:**'.

Des exemples valides sont :

© 2016-2022 Altova GmbH

```
[$x = 42
$x = $x + 1]
```

ou

```
[$x = 42: $x = $x + 1]
```

Ajouter du texte à des fichiers

Le texte non contenu par **[** et **]**, est écrit directement dans le fichier de sortie actuel. S'il n'y a pas de fichier de sortie actuel, le texte est ignoré (voir Utiliser les fichiers comment créer un fichier de sortie).

Pour produire des crochets littéraux, les échapper avec une barre oblique inversée : **\[** et **\]**; pour produire une barre oblique inversée utiliser \\.

Commentaires

Les commentaires se trouvant dans un bloc d'instructions commencent toujours avec un caractère **'** , et se terminent sur la ligne suivante, ou à un caractère de bloc de fermeture **]**.

27.8.2 Déclarations

Les instructions suivantes sont évaluées tout en parsant le fichier de modèle SPL. Elles **ne sont pas** touchées par les instructions de contrôle du flux comme des conditions, des boucles ou des sous-routines, et sont toujours évaluées exactement une seule fois.

Ces mots-clé, comme tous les mots-clé dans SPL, ne sont pas sensibles à la casse.

Veuillez noter que toutes ces déclarations doivent se trouver dans un bloc délimité par des crochets.

map ... to ...

```
map mapname key to value [, key to value ]...
```

Cette instruction ajoute des informations dans un mappage. Voir ci-dessous pour des utilisations spécifiques.

```
map schemanativetype schematype to typespec
```

Le type de Schéma XML built-in spécifié sera mappé dans le type natif spécifié ou la classe, en utilisant le formateur spécifié. Ce paramètre s'applique uniquement à la génération de code pour la version 2007r3 et plus. Typespec est un type natif ou un nom de classe, suivi par une virgule, suivi par l'instance de classe de formateur.

Exemple :

```
map schemanativetype "double" en "double,Altova::DoubleFormatter"
```

map type ... to ...

```
map type schematype to classname
```

Le type de Schéma XML built-in spécifié sera mappé dans la classe spécifiée. Ce paramètre s'applique uniquement à la génération de code pour la version 2007 ou en-dessous.

Exemple :

```
map type "float" to "CSchemaFloat"
```

default ... is ...

```
default setting is value
```

Cette instruction vous permet de toucher la manière dont les noms de classe et de membre sont dérivés depuis le Schéma XML. Veuillez noter que les noms de paramètre sont sensibles à la casse.

Exemple :

```
default "InvalidCharReplacement" is "_"
```

Nom du paramètre	Explication
ValidFirstCharSet	Les caractères autorisés pour lancer un identifiant
ValidCharSet	Les caractères autorisés pour d'autres caractères dans un identifiant
InvalidCharReplacement	Le caractère qui remplacera tous les caractères dans les noms qui ne se trouvent pas dans ValidCharSet
AnonTypePrefix	Préfixe pour les noms des types anonymes*
AnonTypeSuffix	Suffixe pour les noms des types anonymes*
ClassNamePrefix	Préfixe pour des noms de classe générés
ClassNameSuffix	Suffixe pour les noms de classe générés
EnumerationPrefix	Préfixe pour les constantes symboliques déclarées pour les valeurs d'énumération
EnumerationUpperCase	"on" pour convertir les noms de constante d'énumération pour la casse majuscule

© 2016-2022 Altova GmbH

Nom du paramètre	Explication
FallbackName	Si un nom consiste uniquement en caractères qui ne sont pas dans ValidCharSet, utiliser celui-là

* Les noms des types anonymes sont construits à partir de AnonTypePrefix + nom d'élément + AnonTypeSuffix

reserve

```
reserve word
```

Ajoute le mot spécifié pour la liste des mots réservés. Cela garantit qu'il ne sera jamais généré en tant que classe ou de nom de membre.

Exemple :

```
reserve "while"
```

include

Inclut le fichier spécifié en tant que source SPL. Cela vous permet de partager votre modèle dans plusieurs fichiers pour une édition et une gestion plus simple.

```
include filename
```

Exemple :

```
include "Module.cpp"
```

27.8.3 Variables

Tout fichier SPL non-trivial nécessitera des variables. Certaines variables sont prédéfinies par le générateur de code, et des nouvelles variables peuvent être créées simplement en y attribuant des valeurs.

Le caractère **$** est utilisé lors d'une **déclaration** ou **utilisation** d'une variable, un nom de variable est toujours préfixé par **$**. Les noms de variable sont **sensibles à la casse**.

Types de variables :

- entier - aussi utilisé en tant que booléenne, où 0 est faux et tout le reste est vrai
- string
- objet - fourni par XMLSpy
- iterator - voir instruction foreach

Les types de variables sont déclarés par la première affectation :

```
[$x = 0]
```

x est maintenant un entier.

```
[$x = "teststring"]
```

x est maintenant traité en tant qu'un string.

Strings

Les constantes de string sont toujours entourées dans des guillemets doubles, comme dans l'exemple ci-dessus. **\n** et **\t** dans des guillemets doubles sont interprétés en tant que newline et tab, **\"** est un guillemet double littéral, et \\ est une barre oblique inversée. Les constantes de string peuvent aussi s'étendre sur plusieurs lignes.

La concaténation de string utilise le caractère **&**

```
[$BasePath = $outputpath & "/" & $JavaPackageDir]
```

Objets

Les objets représentent l'information contenue dans le schéma XML. Les objets ont des **propriétés**, qui peuvent être accédées en utilisant l'opérateur **.**. Il n'est pas possible de créer de nouveaux objets dans SPL (ils sont prédéfinis par le générateur de code, dérivés depuis le schema) d'entrée, mais il est possible d'attribuer des objets aux variables.

Exemple:

```
class [=$class.Name]
```

Cet exemple produit le mot "class", suivi par une espace et la valeur de la propriété **Name** property de l'objet **$class**.

27.8.4 Variables prédéfinies

Une fois que le fichier de schéma a été analysé par le générateur de code, les objets dans la table ci-dessous existent dans le Template Engine.

Nom	Type	Description
$schematype	entier	1 pour DTD, 2 pour Schema XML
$TheLibrary	Bibliothèque	La bibliothèque dérivée depuis le Schéma XML ou le DTD
$module	string	Nom du Schéma de source sans extension

© 2016-2022 Altova GmbH

Nom	Type	Description
$outputpath	string	Le chemin de sortie spécifié par l'utilisateur ou le chemin de sortie par défaut

Uniquement pour la génération C++ :

Nom	Type	Description
$domtype	entier	1 pour MSXML, 2 pour Xerces
$libtype	entier	1 pour static LIB, 2 pour DLL
$mfc	booléenne	True si la prise en charge MFC est activée
$VSVersion	entier	Spécifie la version Visual Studio. Valeurs valides : **0** Aucun projet Visual Studio **2010** Visual Studio 2010 **2013** Visual Studio 2013 **2015** Visual Studio 2015 **2017** Visual Studio 2017 **2019** Visual Studio 2019

Uniquement pour la génération C# :

Nom	Type	Description
$VSVersion	entier	Spécifie la version Visual Studio. Valeurs valides : **0** Aucun projet Visual Studio **2010** Visual Studio 2010 **2013** Visual Studio 2013 **2015** Visual Studio 2015 **2017** Visual Studio 2017 **2019** Visual Studio 2019

27.8.5 Créer des fichiers de sortie

Ces instructions sont utilisées pour créer des fichiers de sortie depuis une génération de code. Veuillez noter que toutes ces instructions doivent se trouver dans un bloc délimité par des crochets.

create

create *filename*

crée un nouveau fichier. Le fichier doit être fermé avec l'instruction **close**. Toutes les sorties suivantes sont écrites dans le fichier spécifié.

Exemple :

```
[create $outputpath & "/" & $JavaPackageDir & "/" & $application.Name & ".java"]
package [=$JavaPackageName];

public class [=$application.Name]Application {
...
}
[close]
```

close

ferme le fichier de sortie actuel.

=$*variable*

Writes écrit la valeur de la variable spécifiée dans le fichier de sortie actuel.

Exemple :

```
[$x = 20+3]
The result of your calculation is [=$x] - so have a nice day!
```

La fichier de sortie sera :

```
The result of your calculation is 23 - so have a nice day!
```

write

write *string*

écrit le string dans le fichier de sortie actuel.

Exemple :

© 2016-2022 Altova GmbH

```
[write "C" & $name]
```

Il peut aussi s'écrire :

```
C[=$name]
```

filecopy ... to ...

filecopy *source* **to** *target*

cope le fichier de source dans le fichier cible, sans interprétation.

Exemple :

```
filecopy "java/mapforce/mapforce.png" to $outputpath & "/" & $JavaPackageDir &
"/mapforce.png"
```

27.8.6 Opérateurs

Les opérateurs dans SPL fonctionnent comme dans la plupart des langages de programmation.

Liste des opérateurs SPL dans un ordre de précédence décroissant :

.	Accéder à la propriété d'objet
( )	Regroupement d'expression
true	constante booléenne "true"
false	constante booléenne "false"
&	Concaténation de string
-	Signe pour le nombre négatif
not	Négation logique
*	Multiplier
/	Diviser
%	Modulo
+	Ajouter
-	Soustraire
<=	Inférieur à ou égal
<	Inférieur à
>=	Supérieur à ou égal
>	Supérieur à
=	Égal

<> Non égal

and Conjonction logique (avec une évaluation de court circuit)
or Disjonction logique (avec une évaluation de court circuit)

= Attribution

27.8.7 Conditions

SPL vous permet d'utiliser les instructions "if" standard. La syntaxe est comme suit :

```
if condition
	statements
else
	statements
endif
```

ou, sans else :

```
if condition
	statements
endif
```

Note: La condition n'est pas contenue dans des parenthèses.

Comme dans tout autre langage de programmation, les conditions sont construites avec des opérateurs de logique et comparaison.

Exemple :

```
[if $namespace.ContainsPublicClasses and $namespace.Prefix <> ""]
	whatever you want ['inserts whatever you want, in the resulting file]
[endif]
```

Switch

SPL contient aussi une instruction à choix multiple.

Syntaxe :

```
switch $variable
	case X:
		statements
	case Y:
	case Z:
		statements
	default:
		statements
```


© 2016-2022 Altova GmbH

```
endswitch
```

Les libellés de cas doivent être des constantes ou des variables.

L'instruction switch dans SPL ne tombe pas à travers les cas (comme dans C), il n'y a donc pas de nécessité pour une instruction de "break".

27.8.8 Collections et foreach

Collections et itérateurs

Une collection contient plusieurs objets - comme un array ordinaire. Les itérateurs résolvent le problème du stockage et l'incrémentation des indexes array lorsque vous accédez aux objets.

Syntaxe :

```
foreach iterator in collection
       statements
next
```

Exemple :

```
[foreach $class in $classes
       if not $class.IsInternal
              ]      class [=$class.Name];
[      endif
next]
```

Exemple 2 :

```
[foreach $i in 1 To 3
 Write "// Step " & $i & "\n"
  ' Do some work
next]
```

Dans la première ligne :

$classes est l' **objet global** de tous les types générés. Il s'agit d'une collection d'objets de classe uniques.

Foreach passe à travers tous les items dans $classes, et exécute le code suivant l'instruction, jusqu'à l'instruction **next**, pour chacun d'entre eux.

Dans chaque itération, **$class** est attribué à l'objet de classe suivant. Il vous suffit de travailler avec l'objet de classe au lieu d'utiliser classes[i]->Name() comme pour un travail en C++.

Tous les itérateurs de collection possèdent les propriétés supplémentaires suivantes:

Index	L'index suivant, en commençant avec 0
IsFirst	true si l'objet actuel est le premier de la collection (index est 0)
IsLast	true si l'objet actuel est le derner de la collection
Current	L'objet actuel (cela est implicite s'il n'est pas spécifié et peut être exclu)

Exemple :

```
[foreach $enum in $facet.Enumeration
       if not $enum.IsFirst
              ], [
       endif
       ]"[=$enum.Value]"[
next]
```

27.8.9 Sous-routines

Le générateur de code prend en charge des sous-routines sous la forme de procédures ou de fonctions.

Fonctions :

- Passage de valeurs par valeur et par référence
- Paramètres locaux/globaux (local dans le cadre des sous-routines)
- Variables locales
- Invocation récursive (les sous-routines peuvent s'appeler elles-même)

27.8.9.1 Déclaration de sous-routine

Sous-routine

Exemple de syntaxe :

```
Sub SimpleSub()
       ... lines of code
 EndSub
```

- **Sub** est le mot-clé qui dénote la procédure.
- **SimpleSub** est le nom attribué à la sous-routine.
- Λα **parenthèse**Ronde peut contenir une liste de paramètre.
- Le bloc de code d'une sous-routine est lancé immédiatement après la parenthèse de fermeture de paramètre.
- **EndSub** dénote la fin du code de bloc.

Note: La **déclaration** récursive ou en cascade de la sous-routine n'est pas permise, donc une sous-routine peut ne pas contenir une autre sous-routine.

© 2016-2022 Altova GmbH

Paramètres

Les paramètres peuvent aussi être passés par des procédures utilisant la syntaxe suivante :

- Tous les paramètres doivent être des variables
- Les variables doivent être préfixées par le caractère **$**
- Les variables locales sont définies dans une sous-routine
- Les variables globales sont déclarées explicitement, en-dehors des sous-routines
- Plusieurs paramètres sont séparés par le caractère virgule "," dans des parenthèses arrondies
- Des paramètres peuvent passer des valeurs

Paramètres - des valeurs passées

Des paramètres peuvent être passées de deux manières, par des valeurs et par des références, en utilisant les mots-clés **ByVal** et **ByRef** respectivement.

Syntaxe :

```
' define sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
] ...
```

- **ByVal** spécifie que le paramètre est passé par valeur. Veuillez noter que la plupart des objets peut uniquement être passé par la référence.
- **ByRef** spécifie que le paramètre est passé par la référence. Il s'agit du défaut si ni ByVal ou ByRef est spécifié.

La fonction retourne des valeurs

Pour retourner une valeur depuis une sous-routine, utiliser l'instruction **return** . Une telle fonction peut être appelée depuis l'intérieur d'une expression.

Exemple :

```
' define a function
[Sub MakeQualifiedName( ByVal $namespacePrefix, ByVal $localName )
if $namespacePrefix = ""
 return $localName
else
 return $namespacePrefix & ":" & $localName
endif
EndSub
]
```

27.8.9.2 Invocation de sous-routine

Utiliser**call** pour invoquer une sous-routine, suivi par le nom de procédure et les paramètres, le cas échéant.

```
Call SimpleSub()
```

ou

```
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
```

Invocation de fonction

Pour invoquer une fonction (toute sous-routine qui contient une instruction **return**), il suffit d'utiliser son nom dans une expression. Ne pas utiliser l'instruction **call** pour appeler les fonctions. Exemple :

```
$QName = MakeQualifiedName($namespace, "entry")
```

27.8.9.3 Exemple de sous-routine

L'exemple suivant montre la déclaration de sous-routine et l'invocation.

```
[create $outputpath & $module & "output.txt"

' define sub SimpleSub()
Sub SimpleSub()
]SimpleSub() called
[endsub

' execute sub SimpleSub()
Call SimpleSub()

$ParamByValue        = "Original Value"
]ParamByValue        = [=$ParamByValue]
[$ParamByRef  = "Original Value"
]ParamByRef   = [=$ParamByRef]

' define sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
]CompleteSub called.
       param = [=$param]
       paramByValue = [=$paramByValue]
       paramByRef = [=$paramByRef]
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]      Set values inside Sub
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]CompleteSub finished.
[endsub

' run sub CompleteSub()
```


© 2016-2022 Altova GmbH

```
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
]
ParamByValue=[=$ParamByValue]
ParamByRef=[=$ParamByRef]
[
Fermer
]
```

27.8.10 Types intégrés

La section décrit les propriétés des types intégrés utilisés dans les variables prédéfinies qui décrivent le schéma parsé.

27.8.10.1 Bibliothèque

Cet objet représente la bibliothèque toute entière générée depuis le Schéma XML ou le DTD.

Propriétés	Type	Description
SchemaNamespaces	Espace de noms collection	Espaces de noms dans cette bibliothèque
SchemaFilename	string	Nom du fichier XSD ou DTD à partir duquel cette bibliothèque est dérivée
SchemaType	entier	1 pour DTD, 2 pour Schéma XML
Guid	string	Une ID unique globale
CodeName	string	Nom de bibliothèque généré (dérivé du nom de fichier de schéma)

27.8.10.2 Espace de noms

Un objet d'espace de noms par espace de noms de Schéma XML est généré. Les composants de schéma qui ne se trouvent pas dans un espace de noms sont contenus dans un objet d'espace de noms spécial avec un NamespaceURI vide. Veuillez noter que pour DTD, les espaces de noms sont aussi dérivés des attributs dont les noms commencent avec "xmlns".

Propriétés	Type	Description
CodeName	string	Nom pour le code généré (dérivé du préfixe)
LocalName	string	Préfixe d'espace de noms
NamespaceURI	string	URI d'espace de noms

Propriétés	Type	Description
Types	Type collection	Tous les types contenus dans cet espace de noms
Library	Bibliothèque	Bibliothèque contenant cet espace de noms

27.8.10.3 Type

Cet objet représente un type simple ou complexe. Il est utilisé pour générer une classe dans le langage cible. Il existe un type supplémentaire par bibliothèque qui représente le document, qui a tous les éléments racine possibles en tant que membres.

Des types anonymes ont un LocalName vide.

Propriétés	Type	Description
CodeName	string	Nom pour le code généré (dérivé depuis le nom local ou la déclaration de parent)
LocalName	string	Nom original dans le schéma
Namespace	Namespace	Espace de noms contenant ce type
Attributes	Membre collection	Attributs contenus dans ce type*
Elements	Membre collection	Élément enfant contenu dans ce type
IsSimpleType	booléen	True pour les types ssimples, false pour les types complexes
IsDerived	booléen	True si ce type est dérivé d'un autre type, qui est aussi représenté par un objet Type
IsDerivedByExtension	booléen	True si ce type est dérivé par extension
IsDerivedByRestriction	booléen	True si ce type est dérivé par restriction
IsDerivedByUnion	booléen	True si ce type est dérivé par union
IsDerivedByList	booléen	True si ce type est dérivé par list
BaseType	Type	Le type de base de ce type (si IsDerived est true)
IsDocumentRootType	booléen	True si ce type représente le document lui-même
Library	Library	Bibliothèque contenant ce type
IsFinal	booléen	True si déclaré en tant que final dans le schéma
IsMixed	booléen	True si ce type peut avoir du contenu mixte
IsAbstract	booléen	True si ce type est déclaré en tant qu'abstrait

© 2016-2022 Altova GmbH

Propriétés	Type	Description
IsGlobal	booléen	True si ce type est déclaré globalement dans le schéma
IsAnonymous	booléen	True si ce type est déclaré localement dans un élément

Pour les types simples uniquement :

Propriétés	Type	Description
IsNativeBound	booléen	True si la liaison de type native existe
NativeBinding	NativeBinding	Liaison native pour ce type
Facettes	Facettes	Facettes de ce type
Whitespace	string	Raccourci vers la facette Whitespace

* Les types complexes avec du contenu de texte (il s'agit des types avec du contenu mixte et des complexType avec simpleContent) ont un membre d'attribut non nommé supplémentaire qui représente le contenu de texte.

27.8.10.4 Membre

Cet objet représente un attribut ou un élément dans le Schéma XML. Il est utilisé pour créer des memnres de classe de types.

Propriétés	Type	Description
CodeName	string	Nom pour le code généré (dérivé depuis le nom local ou la déclaration de parent)
LocalName	string	Nom original dans le schéma. Vide pour le membre spécial représentant le contenu de texte des types complexes.
NamespaceURI	string	L'URI d'espace de noms de cet Élément/Attribut dans le cadre du documents/streams de l'instance XML.
DeclaringType	Type	Type déclarant le membre à l'origine (égal au ContainingType pour les membres non hérités)
ContainingType	Type	Type où cela est un membre de
DataType	Type	Type de données du contenu de ce membre
Bibliothèque	Bibliothèque	Bibliothèque contenant le DataType de ce membre

Propriétés	Type	Description
IsAttribute	booléenne	Vrai pour les attributs, faux pour les éléments
IsOptional	booléenne	Vrai si minOccurs = 0 ou attribut optionnel
IsRequired	booléenne	Vrai si minOccurs > 0 ou attribut requis
IsFixed	booléenne	Vrai pour des attributs fixes, la valeur se trouve dans la propriété Default
IsDefault	booléenne	Vrai pour des attributs par défaut, la valeur se trouve dans la propriété Default
IsNillable	booléenne	Vrai pour des éléments nillables
IsUseQualified	booléenne	Vrai si NamespaceURI n'est pas vide
MinOccurs	entier	minOccurs, comme dans le schéma. 1 pour des attributs requis
MaxOccurs	entier	maxOccurs, comme dans le schéma. 0 pour les attributs interdits, -1 pour non lié
Default	string	Valeur par défaut

27.8.10.5 NativeBinding

Cet objet représente la liaison d'un type simple vers un type natif dans le langage de programmation cible, comme spécifié par le mappage "schemanativetype".

Propriétés	Type	Description
ValueType	string	Type natif
ValueHandler	string	Instance de classe de formateur

27.8.10.6 Facettes

Cet objet représente toutes les facettes d'un type simple. Les facettes héritées sont fusionnées avec les facettes déclarées explicitement. Si une facette Length est en effet, MinLength et MaxLength sont définies pour la même valeur.

Propriétés	Type	Description
DeclaringType	Type	Les facettes de type sont déclarées sur
Whitespace	string	"preserve", "collapse" ou "replace"
MinLength	entier	Valeur de Facette

© 2016-2022 Altova GmbH

Propriétés	Type	Description
MaxLength	entier	Valeur de Facette
MinInclusive	entier	Valeur de Facette
MinExclusive	entier	Valeur de Facette
MaxInclusive	entier	Valeur de Facette
MaxExclusive	entier	Valeur de Facette
TotalDigits	entier	Valeur de Facette
FractionDigits	entier	Valeur de Facette
List	Collection de facette	Toutes les facettes en tant que liste

Facette

Cet objet représente une seule facette avec sa valeur calculée effective pour un type spécifique.

Propriétés	Type	Description
LocalName	string	Nom de facette
NamespaceURI	string	Espace de noms de facette
FacetType	string	Une de "normalization", "lexicalspace", "valuespace-length", "valuespace-enum" ou "valuespace-range"
DeclaringType	Type	Type sur lequel cette facette est déclarée
FacetCheckerName	string	Nom du vérificateur de facette (depuis le map schemafacet)
FacetValue	string ou entier	Valeur réelle de cette facette

28 Commandes de menu

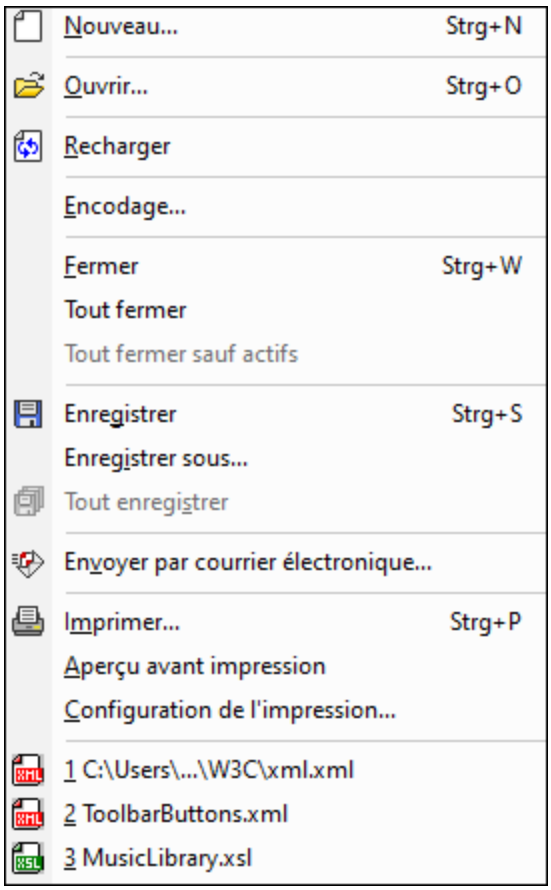
Cette section contient une description complète de toutes les commandes de menu de XMLSpy. Outre les descriptions générales des commandes, nous avons tenté de fournir des informations de fond lorsque cela nous a semblé être nécessaire. Néanmoins, si vous avez des questions qui ne sont pas traitées dans cette documentation, veuillez consulter les FAQs et les Forums de discussion sur le site Internet Altova. Si vous n'y trouvez pas de réponse satisfaisante, n'hésitez pas à contacter le Centre d'assistance Altova.

Vous trouverez les commandes standard de Windows comme (**Ouvrir, Enregistrer, Couper, Copier** et **Coller**) dans les menus Fichier et Éditer. Ces menus contiennent en outre des commandes liées à XML et Internet.

© 2016-2022 Altova GmbH

28.1 Menu Fichier

Le menu **Fichier** contient les commandes pour les opérations de fichier, classées comme dans la plupart des applications Windows. Outre les commandes standard Nouveau, Ouvrir, Enregistrer, Imprimer, Paramètres d'impression et Quitter, XMLSpy propose aussi des commandes spécifiques à XML et à l'application.



28.1.1 Nouveau

Cette section :

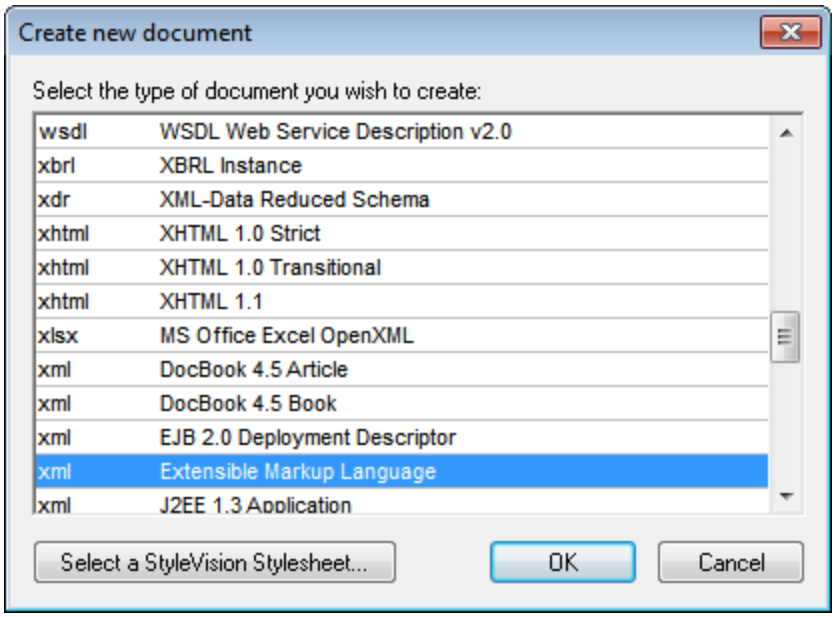
- Icône et raccourci
- Description
- Modèles pour nouveaux documents
- Attribuer un DTD ou un schéma XML à un nouveau document XML
- Spécifier l'élément racine d'un nouveau document XML
- Attribuer une SPS à un nouveau document XML
- Créer de nouvelles taxonomies XBRL avec l'Assistant de Taxonomie XBRL

Icône et raccourci

Icône :	
Raccourci :	**Ctrl+N**

Description

La commande **Nouveau** est utilisée pour créer un nouveau document. Cliquer sur **Nouveau** ouvre le dialogue Créer nouveau document (*capture d'écran ci-dessous*) dans lequel vous pouvez choisir le type de document que vous souhaitez créer. Si le type de document que vous souhaitez créer n'est pas recensé dans le dialogue, sélectionnez XML et changez l'extension de fichier lorsque vous enregistrez le fichier. Veuillez noter que vous pouvez ajouter de nouveaux types de document à cette liste de dialogue en utilisant la section Outils | Options | Types de fichier.



Modèles pour nouveaux documents

La liste du type de document contenue dans le dialogue Créer nouveau document peut aussi comporter des entrées pour des modèles de document définis par l'utilisateur pour n'importe quel type de document. Ces modèles peuvent être ouverts directement depuis le dialogue Créer nouveau document et édités. Pour créer votre propre modèle de document de manière à ce qu'il apparaisse dans la liste de types de document du dialogue Créer nouveau document, vous devez tout d'abord créer le document modèle, puis l'enregistrer dans le dossier conçu pour contenir les modèles de document.

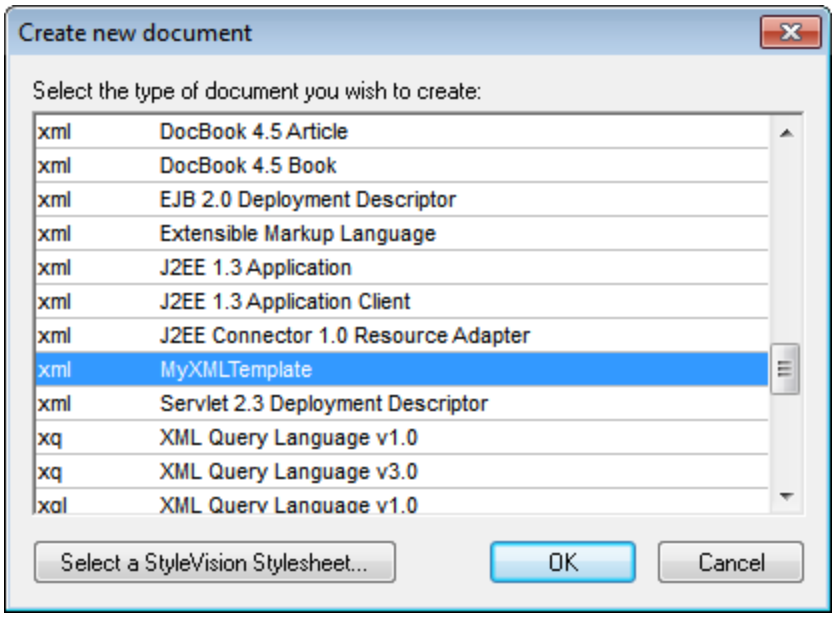
Créer un modèle de document comme suit :

1. Ouvrir le dossier `XMLSpy\Template` du dossier d'application **avec Windows Explorer ou avec votre outil de navigation de prédilection et choisir un fichier modèle**

© 2016-2022 Altova GmbH

rudimentaire parmi les fichiers nommés new.xxx (`.xxx` représentant une extension de fichier, comme `.xml` ou `.xslt`).

2. Ouvrir le fichier dans XMLSpy, et modifier le fichier selon vos besoins. Ce fichier sera le fichier modèle.
3. Une fois avoir terminé, choisir **Fichier | Enregistrer sous** pour réenregistrer le fichier dans le dossier `\Template` avec un nom approprié, par ex. `MyXMLTemplate.xml`. Vous disposez maintenant d'un modèle de document appelé `MyXMLTemplate`, qui apparaîtra dans la liste des types de document dans le dialogue Créer nouveau document.



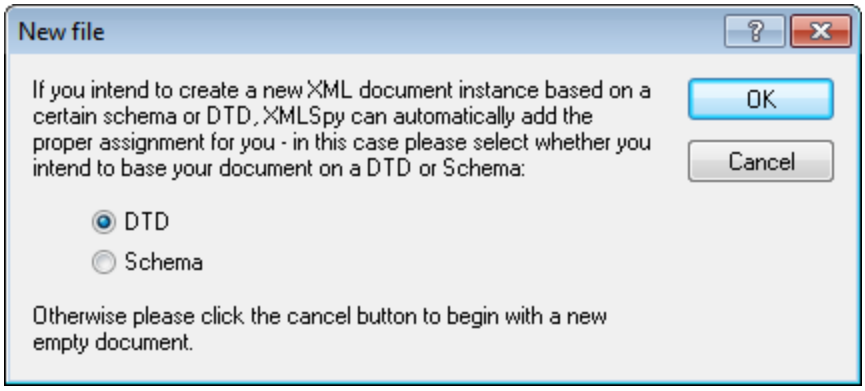
4. Pour ouvrir le modèle, choisir **Fichier | Nouveau**, et puis le modèle (`my-xml`, dans ce cas).

Pour supprimer un modèle de document de la liste des types de document, supprimer (ou déplacer) le fichier modèle du dossier modèle.

Attribuer un DTD ou un schéma XML à un nouveau document XML

Lorsque vous créez un nouveau document d'un certain type par le biais du dialogue Créer un nouveau document, le document est automatiquement ouvert avec l'association de Schéma XML ou DTD correcte. Par exemple, un fichier XHTML sera ouvert avec le DTD `http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd` y associé. Et un fichier de Schéma XML (`.xsd`) est associé avec le document de schéma `http://www.w3.org/2001/XMLSchema`.

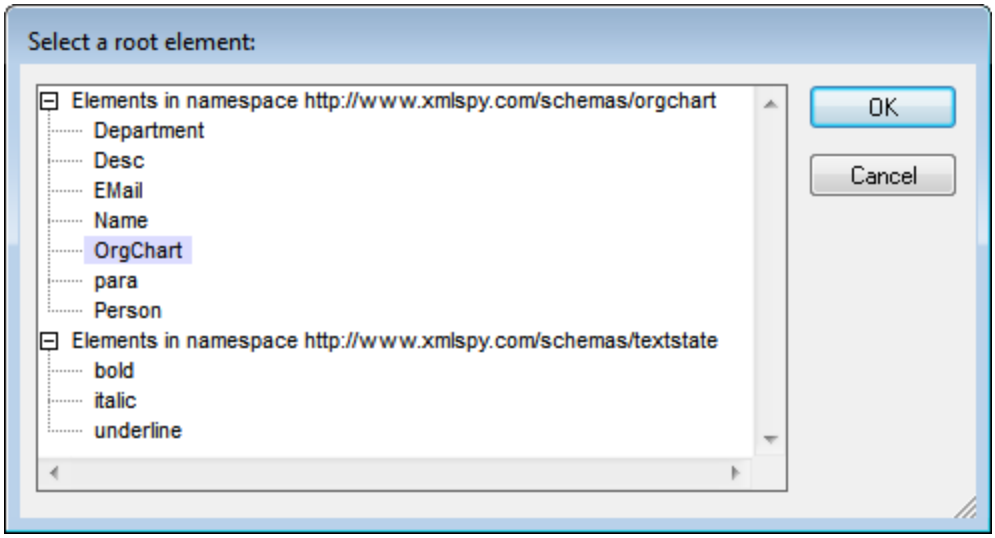
Si vous créez un nouveau document pour lequel le schéma n'est pas connu (par exemple, un fichier XML), alors vous serez invité à associer un schéma (DTD ou Schéma XML) au nouveau document (*capture d'écran ci-dessous*).



Si, dans le dialogue, vous choisissez DTD ou Schéma XML et que vous cliquez sur **OK**, vous pouvez chercher le schéma que vous souhaitez. Cliquer sur **Annuler** pour créer un nouveau fichier qui n'est pas associé avec un schéma.

Spécifier l'élément racine d'un nouveau document XML

Si un Schéma XML est choisi en tant que le schéma associé d'un document XML et que ce schéma comprend plus d'un élément global, chacun d'entre eux est un élément racine potentiel. Dans ce cas, le dialogue Choisir un élément racine (*capture d'écran ci-dessous*) s'ouvre, dans lequel vous pouvez choisir quel élément global sera l'élément racine du document XML. Dans la capture d'écran ci-dessous, l'élément global `OrgChart` a été choisi.



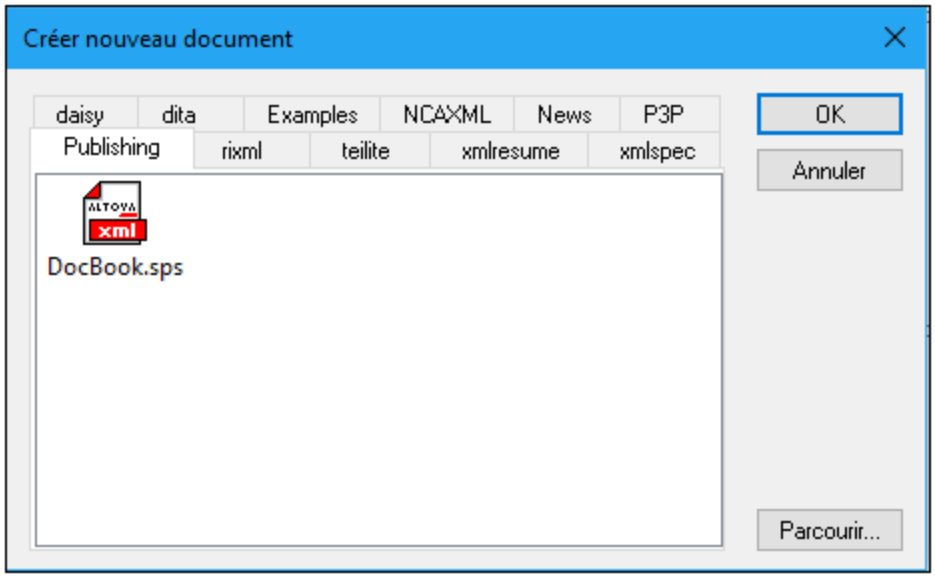
Cliquer sur **OK** maintenant pour créer un nouveau document XML avec cet élément (`OrgChart`) en tant que son élément racine.

Attribuer une SPS à un nouveau document XML

Lorsqu'un nouveau document XML est créé, vous pouvez y associer un StyleVision Power Stylesheet (fichier `.sps`) pour consulter le document dans Authentic View. Dans le dialogue Créer nouveau document (*voir*

© 2016-2022 Altova GmbH

première capture d'écran dans cette section), lorsque vous cliquez sur **Choisir une feuille de style StyleVision**, le dialogue Créer nouveau document (*affiché ci-dessous*) apparaît.



Vous pouvez chercher le SPS exigé dans les onglets du dossier ou vous pouvez cliquer sur la touche **Parcourir** pour chercher et choisir le SPS.

Créer de nouvelles taxonomies XBRL avec l'Assistant de Taxonomie XBRL

Dans le dialogue Créer nouveau document, si le schéma de taxonomie XBRL (`.xsd`) est sélectionné, un assistant vous guide à travers les étapes de création d'une nouvelle taxonomie XBRL. Cet assistant est décrit dans la section XBRL de la documentation.

28.1.2 Ouvrir

Icône et raccourci

Icône :	
Raccourci :	**Ctrl+O**

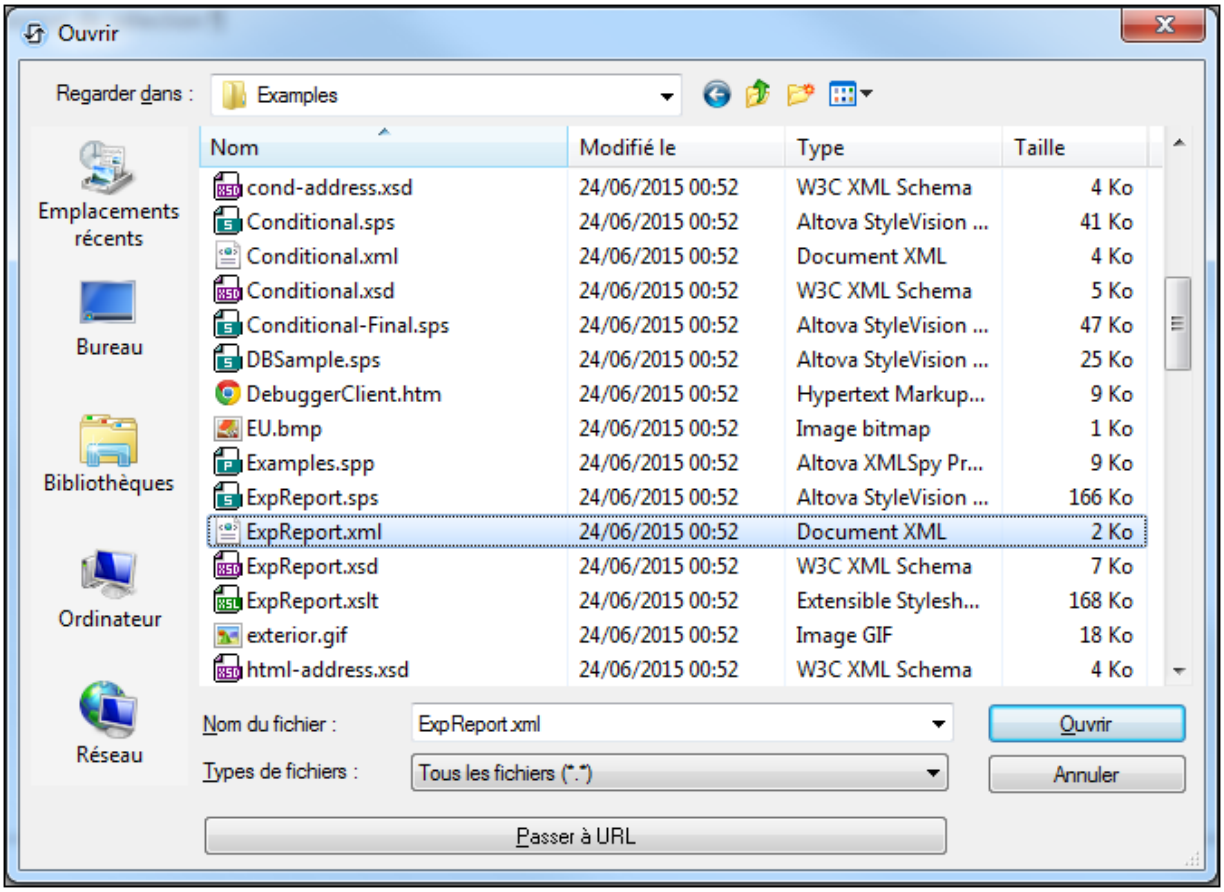
Description

La commande **Ouvrir** déclenche l'ouverture du dialogue Windows Ouvrir que vous connaissez bien et vous permet d'ouvrir tout document lié à XML ou tout document texte. Dans le dialogue Ouvrir, vous pouvez sélectionner plus d'un fichier à ouvrir. Utiliser la liste de choix Types de fichier pour limiter le type de fichiers affichés dans le dialogue. (La liste des types de fichiers disponibles peut être configurée dans la section Types de fichier du dialogue Options (**Outils | Options**).) Lorsqu'un fichier XML est ouvert, sa bonne formation est vérifiée. Si le fichier n'est pas bien formé, vous obtiendrez une erreur "fichier n'est pas bien formé". Réparez l'erreur et choisir la commande de menu **XML | Vérifier la bonne formation (F7)** pour revérifier. Si vous avez

choisi une validation sur ouverture automatique et que le fichier est invalide, vous obtiendrez un message d'erreur. Réparer l'erreur et choisir la commande de menu **XML | Valider XML (F8)** pour revalider.

- Sélectionner et enregistrer les fichiers via des URL et des Ressources globales

  Dans plusieurs dialogues Fichiers Ouvrir et Fichier Enregistrer, vous pouvez choisir de sélectionner le fichier nécessaire ou d'enregistrer un fichier via une URL ou une Ressource globale (*voir capture d'écran ci-dessous*). Cliquer sur **Passer à URL** ou sur **Passer à Ressources globales** pour vous rendre sur un de ces processus de sélection.

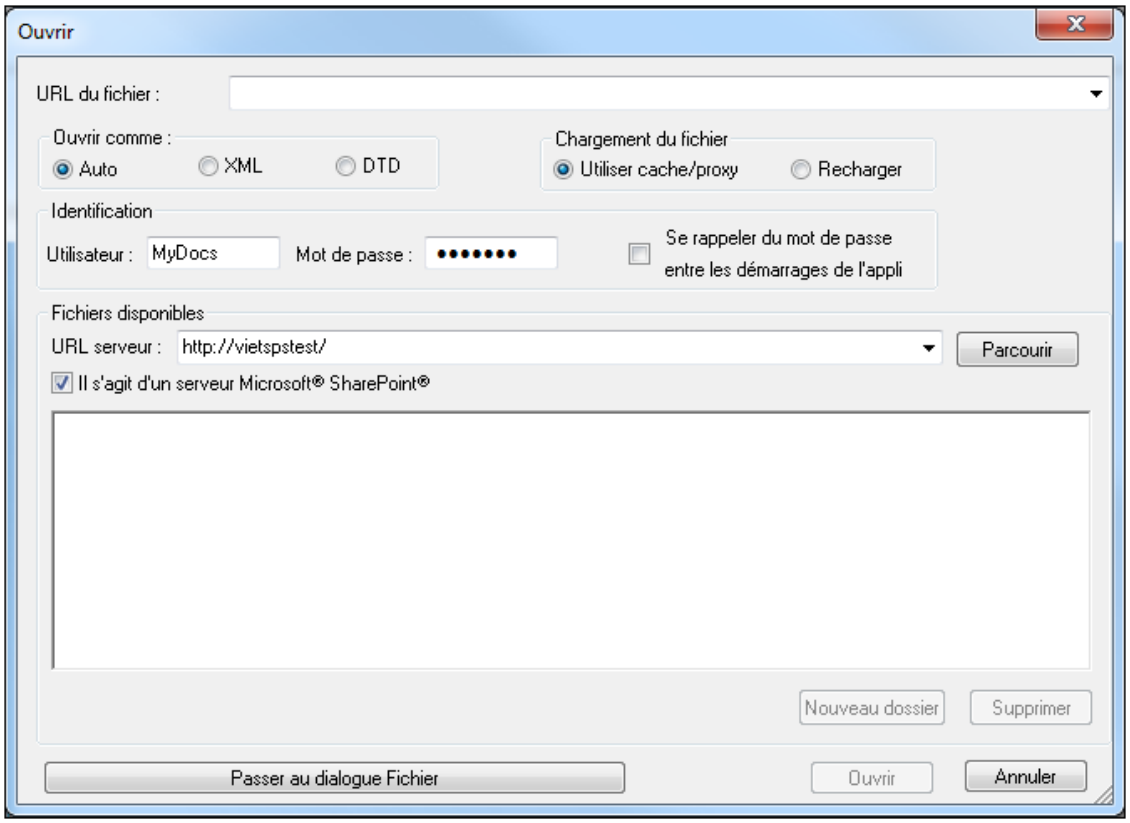


Sélectionner les fichiers via des URL

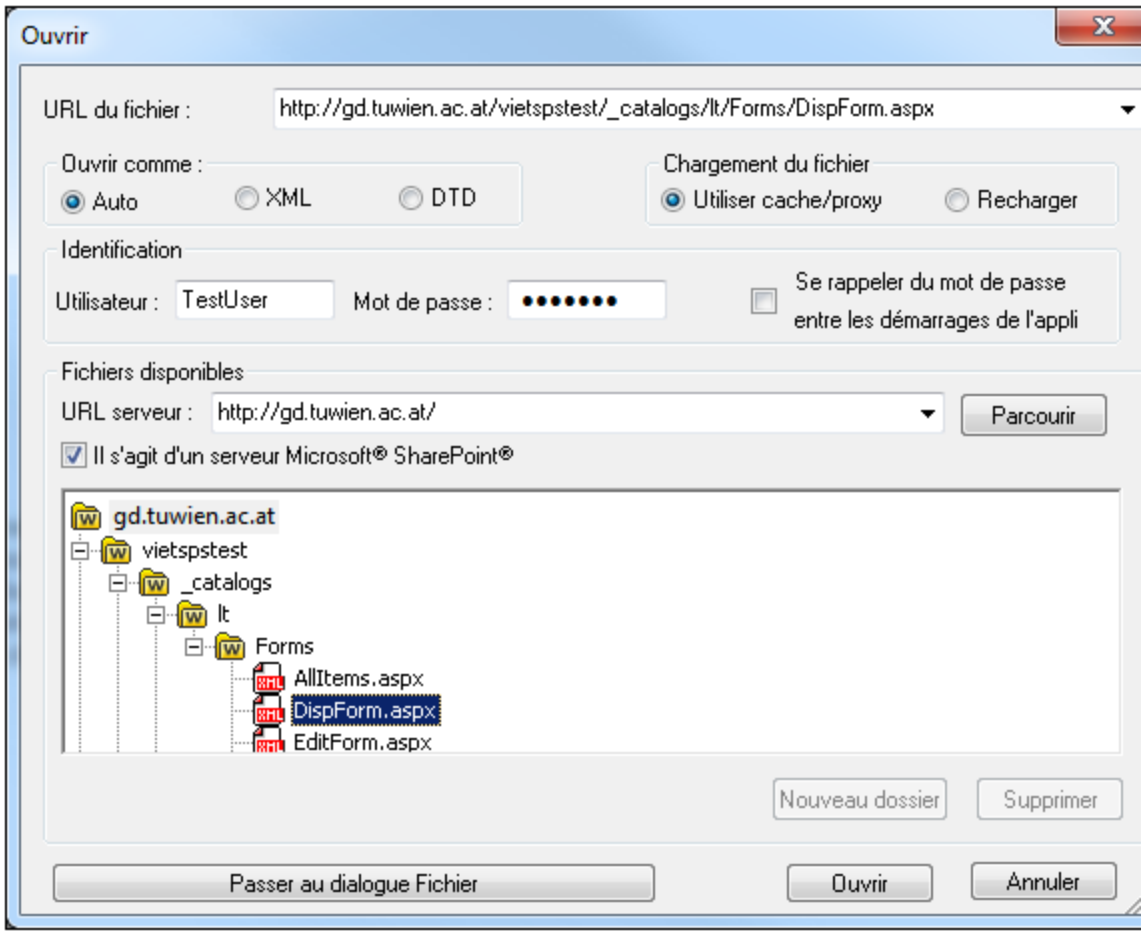
Pour sélectionner un fichier via une URL (pour l'ouvrir ou l'enregistrer), procédez comme suit :

1. Cliquer sur la commande **Passer à URL**. Cela vous permet de passer au mode URL du dialogue Ouvrir ou Enregistrer (*la capture d'écran ci-dessous montre le dialogue Ouvrir*).

© 2016-2022 Altova GmbH



2. Saisir l'URL à laquelle vous souhaitez accéder dans le champ *URL de serveur* (*capture d'écran ci-dessus*). Si le serveur est un Microsoft® SharePoint® Server, cochez la case correspondante *Microsoft® SharePoint® Server*. Voir les Notes Microsoft® SharePoint® Server ci-dessous pour plus d'informations concernant le travail avec des fichiers sur ce type de serveur.
3. Si le serveur est protégé par un mot de passe, veuillez saisir votre ID d'utilisateur et votre mot de passe dans les champs *Utilisateur* et *Mot de passe*.
4. Cliquer sur **Parcourir** pour consulter et naviguer dans la structure du répertoire du serveur.
5. Dans l'arborescence du dossier, recherchez le fichier que vous souhaitez charger puis cliquer dessus.



L'URL du fichier apparaît dans le champ URL de fichier (*voir capture d'écran ci-dessus*). Ce n'est qu'à ce moment que le bouton **Ouvrir** ou **Enregistrer** devient actif.

6. Cliquer sur **Ouvrir** pour charger le fichier ou **Enregistrer** pour l'enregistrer.

Notez les points suivants :

- La fonction Parcourir est uniquement disponible sur les serveurs qui prennent en charge WebDAV et sur les Microsoft SharePoint Servers. Les protocoles pris en charge sont FTP, HTTP et HTTPS.
- Pour vous donner un plus grand contrôle du processus de chargement lors de l'ouverture d'un fichier, vous pouvez choisir de charger le fichier par le biais du cache local ou d'un serveur proxy (ce qui accélère considérablement le processus si le fichier a été chargé avant). En alternative, vous pouvez choisir de recharger le fichier si vous travaillez par exemple avec un système de publication électronique ou de base de données ; dans ce cas, choisir l'option **Recharger**.

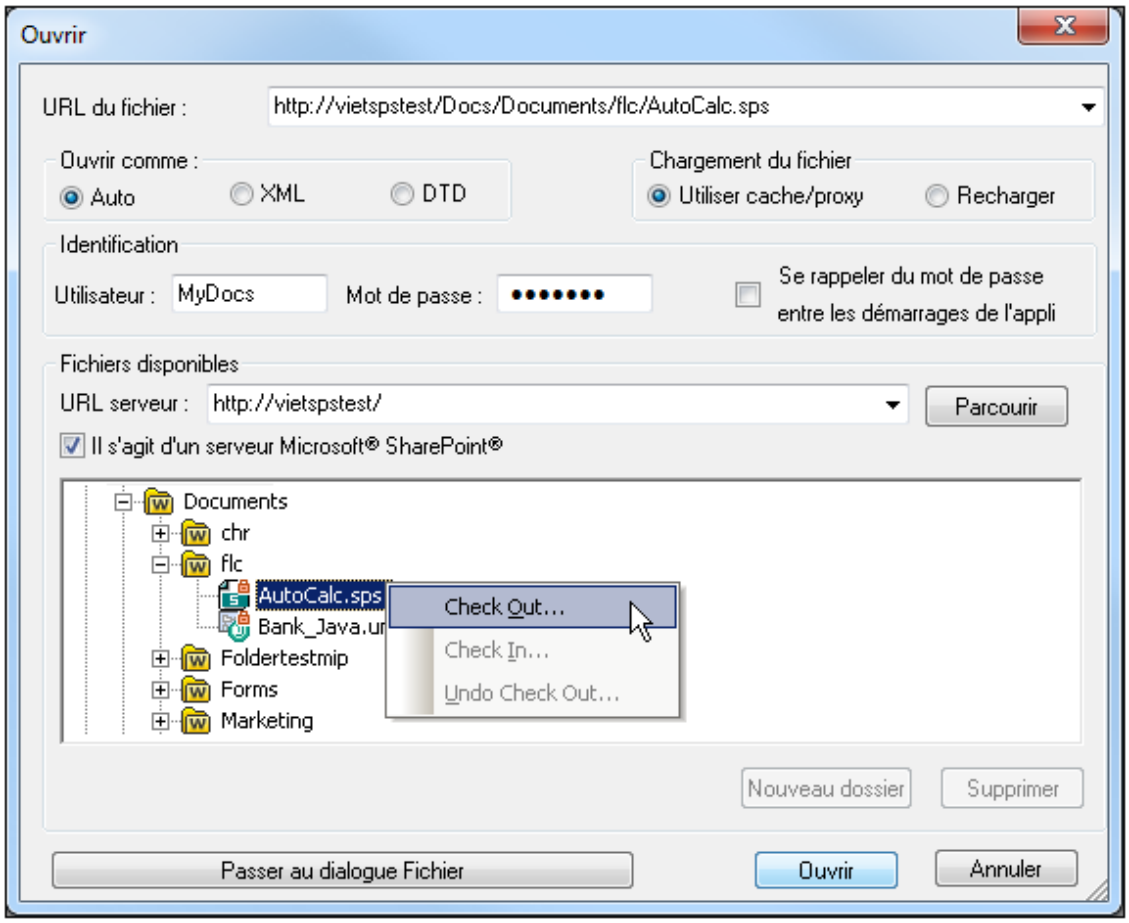
.

▼ Notes Microsoft® SharePoint® Server

 © 2016-2022 Altova GmbH

Noter les points suivants concernant les fichiers sur les Microsoft® SharePoint® Servers :

- Dans la structure du répertoire qui apparaît dans le panneau des Fichiers disponibles (*capture d'écran ci-dessous*), les icônes de fichier ont des symboles qui indiquent le statut de d'archivage/récupération des fichiers.



Cliquer avec la touche de droite sur un fichier, pour ouvrir un menu contextuel contenant les commandes disponibles pour ce fichier (*capture d'écran ci-dessus*).

- Les icônes de fichier sont présentées ci-dessous :

	Archivé. Disponible pour une récupération.
	Récupéré par un autre utilisateur. N'est pas disponible pour la récupération.
	Récupéré localement. Peut être édité et archivé.

- Après avoir effectué une récupération de fichier, vous pouvez l'éditer dans votre application Altova et l'enregistrer en utilisant **Fichier | Enregistrer (Ctrl+S)**.
- Vous pouvez faire un archivage du fichier édité par le biais du menu contextuel dans le dialogue Ouvrir URL (*voir capture d'écran ci-dessus*) ou via le menu contextuel qui s'ouvre sur un clic avec

le bouton de droite de la souris sur l'onglet de fichier dans la fenêtre principale de votre application (*capture d'écran ci-dessous*).



- Lorsqu'un fichier est récupéré par un autre utilisateur, il n'est plus disponible pour une récupération.
- Lorsqu'un fichier est récupéré localement par vous, vous pouvez annuler la récupération avec la commande Undo Check-Out (Annuler récupération) dans le menu contextuel. En conséquence, le fichier sera retourné sans aucun changement sur le serveur.
- Si vous procédez à une récupération d'un fichier dans une application Altova, vous ne pourrez pas faire de récupération dans une autre application Altova. Le fichier sera considéré comme s'il avait déjà été récupéré par vous. Les commandes disponibles à ce moment dans toute application Altova prenant en charge Microsoft® SharePoint® Server sera : **Check In (archiver)** et **Undo Check Out (Annuler récupération)**.

▼ Ouvrir et enregistrer des fichiers via Ressources globales

Pour ouvrir ou enregistrer un fichier via des ressources globales, cliquer sur **Passer à la Ressource globale**. Un dialogue s'ouvrira dans lequel vous pouvez sélectionner la ressource globale. Ces dialogues sont décrits dans la section Utiliser les Ressources globales. Pour une description générale des Ressources globales, consulter la section Ressources globales dans cette documentation.

28.1.3 Recharger

Icône

Icône :	

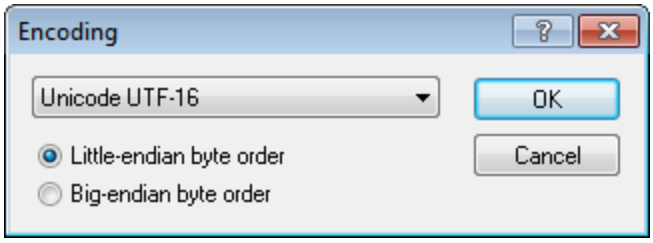
Description

Recharge tous les documents ouverts modifiés en dehors de XMLSpy. Si un ou plusieurs documents ont été modifiés en-dehors de XMLSpy, une invitation apparaît vous demandant si vous souhaitez recharger le/s document/s modifié/s. Si vous choisissez de recharger, toutes les modifications effectuées sur le fichier depuis le dernier enregistrement seront perdues.

28.1.4 Encodage

La commande **Encodage** vous permet de : (i) consulter l'encodage actuel du document actif (XML ou non XML), et (ii) sélectionner un encodage différent avec lequel le document actif sera enregistré la prochaine fois.

© 2016-2022 Altova GmbH



Dans les documents XML, si vous sélectionnez un encodage différent que celui utilisé actuellement, l'attribut d'encodage dans la déclaration XML sera modifié en conséquence. Pour les encodages de caractère two-byte et four-byte (UTF-16, UCS-2 et UCS-4) , vous pouvez également spécifier l'ordre des octets à utiliser pour le fichier. Un autre moyen de changer l'encodage d'un document XML est d'éditer l'attribut d'encodage de la déclaration XML du document directement. Les encodages par défaut pour des documents XML et non XML nouveaux et existants peuvent être définis dans la section d'encodage du dialogue des Options.

Note : Lors de l'enregistrement d'un document, XMLSpy vérifie automatiquement les spécifications d'encodage et vous permet de choisir l'encodage approprié par le biais du dialogue Encodage. Si votre document contient des caractères qui ne peuvent pas être représentés dans l'encodage sélectionné et que vous essayez d'enregistrer le fichier, vous recevrez un avertissement à cet effet.

28.1.5 Fermer, Fermer tout, Fermer tout sauf actives

Fermer

La commande **Fermer** ferme la fenêtre de document active. Si le fichier a été modifié (indiqué par un astérisque * placé après le nom du fichier dans la barre de titre), vous serez invité à enregistrer le fichier d'abord.

Tout fermer

La commande **Tout fermer** ferme toutes les fenêtres de document ouvertes. Si un document a été modifié (indiqué par un astérisque * placé après le nom du fichier dans la barre de titre), vous serez invité à enregistrer le fichier d'abord.

Tout fermer sauf actifs

La commande **Tout fermer sauf actifs** ferme toutes les fenêtres de document ouvertes sauf la fenêtre de document active. Si un document a été modifié (indiqué par un astérisque * placé après le nom du fichier dans la barre de titre), vous serez invité à enregistrer le fichier d'abord.

28.1.6 Enregistrer, Enregistrer sous, Enregistrer tout

Icônes et raccourcis

Commande	*Icône*	*Raccourci*

Enregistrer		**Ctrl+S**
Tout enregistrer		

Enregistrer

La **commande Enregistrer (Ctrl+S)** enregistre le contenu du document actif sous le fichier à partir duquel il a été ouvert. Lors de l'enregistrement d'un document, la bonne formation du fichier est vérifiée automatiquement. Le fichier sera également validé automatiquement si cette option a été configurée dans la section Fichier du dialogue Options (**Outils | Options**). La déclaration XML est également vérifiée en termes de spécifications d'encodage, et cet encodage est appliqué au document lorsque le fichier est enregistré.

Enregistrer sous

La commande **Enregistrer sous** déclenche l'ouverture du dialogue Windows "Enregistrer sous" que vous connaissez bien, dans lequel vous pouvez saisir le nom et l'emplacement du fichier actif que vous souhaitez enregistrer sous un nom. Les mêmes vérifications que pour la commande **Enregistrer** sont effectuées.

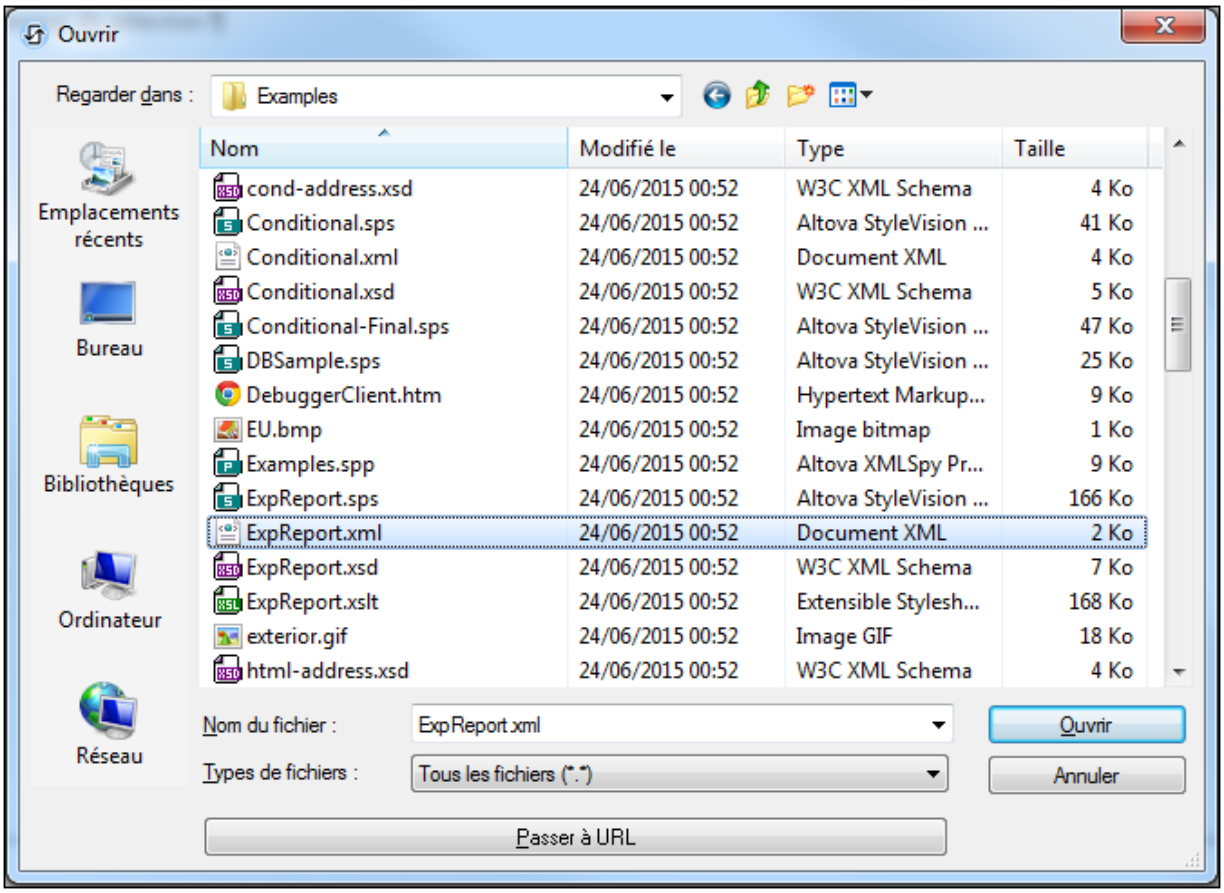
Tout enregistrer

La commande **Tout enregistrer** enregistre toutes les modifications qui ont été réalisées sur tout document ouvert. La commande est utile si vous éditez plusieurs documents simultanément. Si un document n'a pas été enregistré avant (par exemple après avoir été créé récemment), le dialogue Enregistrer sous est présenté pour ce document.

▼ Sélectionner et enregistrer les fichiers via des URL et des Ressources globales

Dans plusieurs dialogues Fichiers Ouvrir et Fichier Enregistrer, vous pouvez choisir de sélectionner le fichier nécessaire ou d'enregistrer un fichier via une URL ou une Ressource globale (*voir capture d'écran ci-dessous*). Cliquer sur **Passer à URL** ou sur **Passer à Ressources globales** pour vous rendre sur un de ces processus de sélection.

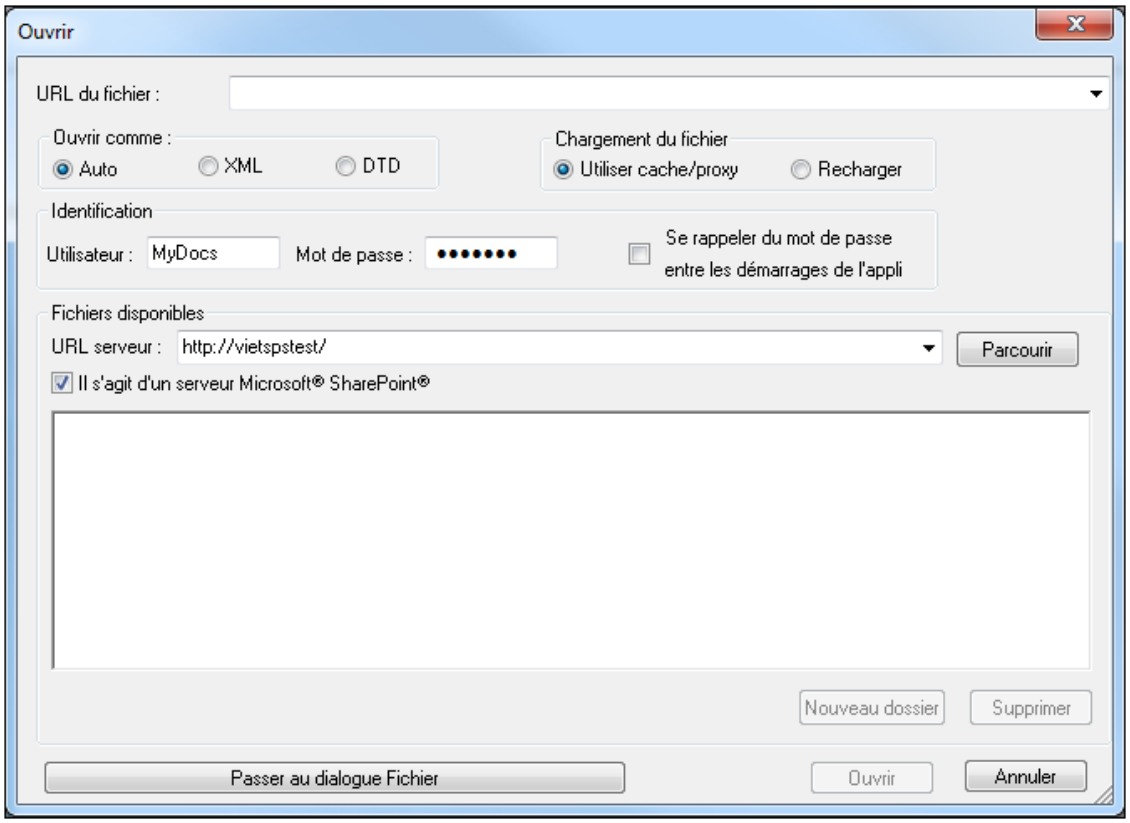
© 2016-2022 Altova GmbH



Sélectionner les fichiers via des URL

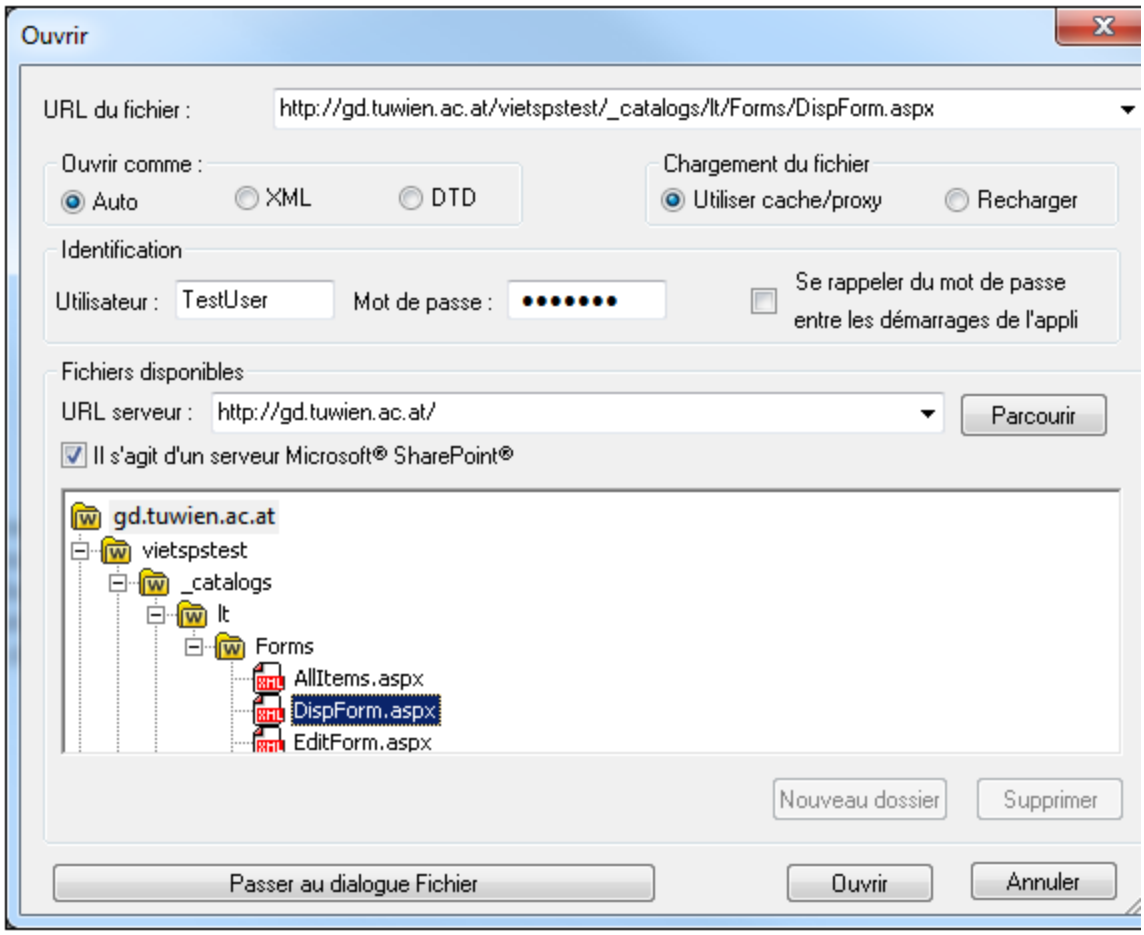
Pour sélectionner un fichier via une URL (pour l'ouvrir ou l'enregistrer), procédez comme suit :

1. Cliquer sur la commande **Passer à URL**. Cela vous permet de passer au mode URL du dialogue Ouvrir ou Enregistrer (*la capture d'écran ci-dessous montre le dialogue Ouvrir*).



2. Saisir l'URL à laquelle vous souhaitez accéder dans le champ *URL de serveur* (*capture d'écran ci-dessus*). Si le serveur est un Microsoft® SharePoint® Server, cochez la case correspondante *Microsoft® SharePoint® Server*. Voir les Notes Microsoft® SharePoint® Server ci-dessous pour plus d'informations concernant le travail avec des fichiers sur ce type de serveur.
3. Si le serveur est protégé par un mot de passe, veuillez saisir votre ID d'utilisateur et votre mot de passe dans les champs *Utilisateur* et *Mot de passe*.
4. Cliquer sur **Parcourir** pour consulter et naviguer dans la structure du répertoire du serveur.
5. Dans l'arborescence du dossier, recherchez le fichier que vous souhaitez charger puis cliquer dessus.

© 2016-2022 Altova GmbH



L'URL du fichier apparaît dans le champ URL de fichier (*voir capture d'écran ci-dessus*). Ce n'est qu'à ce moment que le bouton **Ouvrir** ou **Enregistrer** devient actif.

6. Cliquer sur **Ouvrir** pour charger le fichier ou **Enregistrer** pour l'enregistrer.

Notez les points suivants :

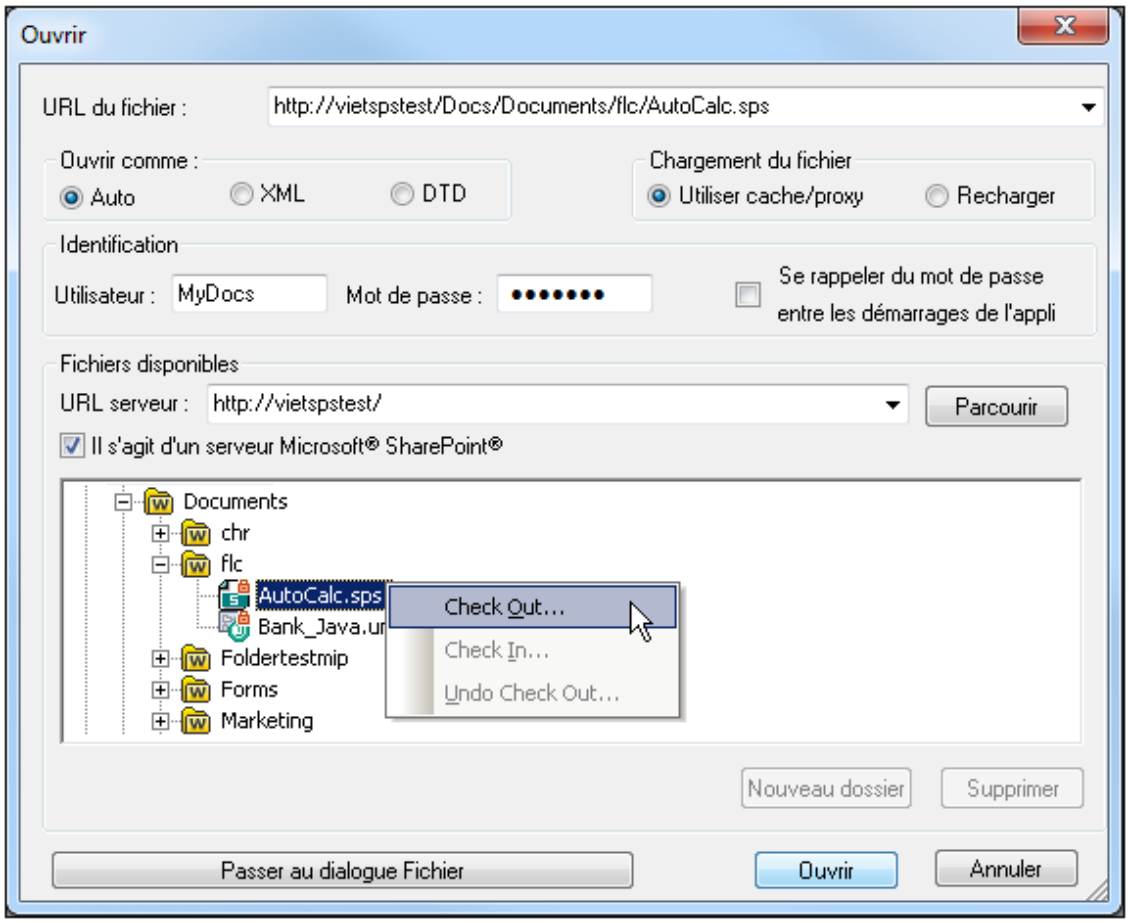
- La fonction Parcourir est uniquement disponible sur les serveurs qui prennent en charge WebDAV et sur les Microsoft SharePoint Servers. Les protocoles pris en charge sont FTP, HTTP et HTTPS.
- Pour vous donner un plus grand contrôle du processus de chargement lors de l'ouverture d'un fichier, vous pouvez choisir de charger le fichier par le biais du cache local ou d'un serveur proxy (ce qui accélère considérablement le processus si le fichier a été chargé avant). En alternative, vous pouvez choisir de recharger le fichier si vous travaillez par exemple avec un système de publication électronique ou de base de données ; dans ce cas, choisir l'option **Recharger**.

.

▼ Notes Microsoft® SharePoint® Server

Noter les points suivants concernant les fichiers sur les Microsoft® SharePoint® Servers :

- Dans la structure du répertoire qui apparaît dans le panneau des Fichiers disponibles (*capture d'écran ci-dessous*), les icônes de fichier ont des symboles qui indiquent le statut de d'archivage/récupération des fichiers.



Cliquer avec la touche de droite sur un fichier, pour ouvrir un menu contextuel contenant les commandes disponibles pour ce fichier (*capture d'écran ci-dessus*).

- Les icônes de fichier sont présentées ci-dessous :

	Archivé. Disponible pour une récupération.
	Récupéré par un autre utilisateur. N'est pas disponible pour la récupération.
	Récupéré localement. Peut être édité et archivé.

- Après avoir effectué une récupération de fichier, vous pouvez l'éditer dans votre application Altova et l'enregistrer en utilisant **Fichier | Enregistrer (Ctrl+S)**.
- Vous pouvez faire un archivage du fichier édité par le biais du menu contextuel dans le dialogue Ouvrir URL (*voir capture d'écran ci-dessus*) ou via le menu contextuel qui s'ouvre sur un clic avec

© 2016-2022 Altova GmbH

le bouton de droite de la souris sur l'onglet de fichier dans la fenêtre principale de votre application (*capture d'écran ci-dessous*).



- Lorsqu'un fichier est récupéré par un autre utilisateur, il n'est plus disponible pour une récupération.
- Lorsqu'un fichier est récupéré localement par vous, vous pouvez annuler la récupération avec la commande Undo Check-Out (Annuler récupération) dans le menu contextuel. En conséquence, le fichier sera retourné sans aucun changement sur le serveur.
- Si vous procédez à une récupération d'un fichier dans une application Altova, vous ne pourrez pas faire de récupération dans une autre application Altova. Le fichier sera considéré comme s'il avait déjà été récupéré par vous. Les commandes disponibles à ce moment dans toute application Altova prenant en charge Microsoft® SharePoint® Server sera : **Check In (archiver)** et **Undo Check Out (Annuler récupération)**.

▼ Ouvrir et enregistrer des fichiers via Ressources globales

Pour ouvrir ou enregistrer un fichier via des ressources globales, cliquer sur **Passer à la Ressource globale**. Un dialogue s'ouvrira dans lequel vous pouvez sélectionner la ressource globale. Ces dialogues sont décrits dans la section Utiliser les Ressources globales. Pour une description générale des Ressources globales, consulter la section Ressources globales dans cette documentation.

28.1.7 Envoyer par mail

Icône

Icône :	

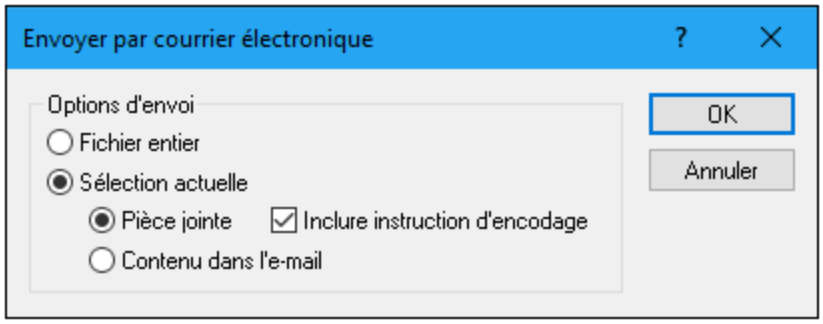
Description

La commande **Envoyer par mail** vous permet d'envoyer des documents XML ou des sélections depuis une document XML par e-mail. Selon son type, un document ou une sélection peut être attaché en tant que pièce jointe, contenu ou en tant que lien. Voir la table ci-dessous pour les détails.

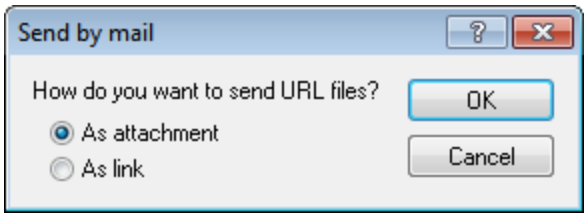
Ce que vous pouvez envoyer	Comment vous pouvez l'envoyer
Document XML actif	En tant que pièce jointe d'e-mail
Sélection de document XML actif	En tant que pièce jointe ou contenu d'e-mail

Un ou plusieurs fichiers dans la fenêtre Projet	En tant que pièce jointe d'e-mail
Une ou plusieurs URL dans la fenêtre Projet	En tant que pièce jointe ou lien

Lorsque la commande **Envoyer par mail** est invoquée dans le document XML actif, le dialogue Envoyer par e-mail (*capture d'écran ci-dessous*) s'ouvre et propose les options d'envoi montrées ci-dessous. Si la commande **Envoyer par mail** est invoquée sans texte sélectionné dans le fichier actif, le bouton radio *Fichier entier* (*voir capture d'écran ci-dessus*) est la seule option qui est activée ; les autres options sont désactivées.



Puisque les fichiers envoyés depuis la fenêtre Projet sont toujours envoyés en tant que pièce jointe uniquement, le dialogue Envoyer par mail est sauté et un email contenant les fichiers sélectionnés en tant que pièce jointe s'ouvre. Les URL contenues dans la fenêtre Projet peuvent être envoyées en tant que pièce jointe ou en tant que lien (*voir capture d'écran ci-dessous*). Choisir comment envoyer l'URL et cliquer sur **OK**.



28.1.8 Imprimer

Icône et raccourci

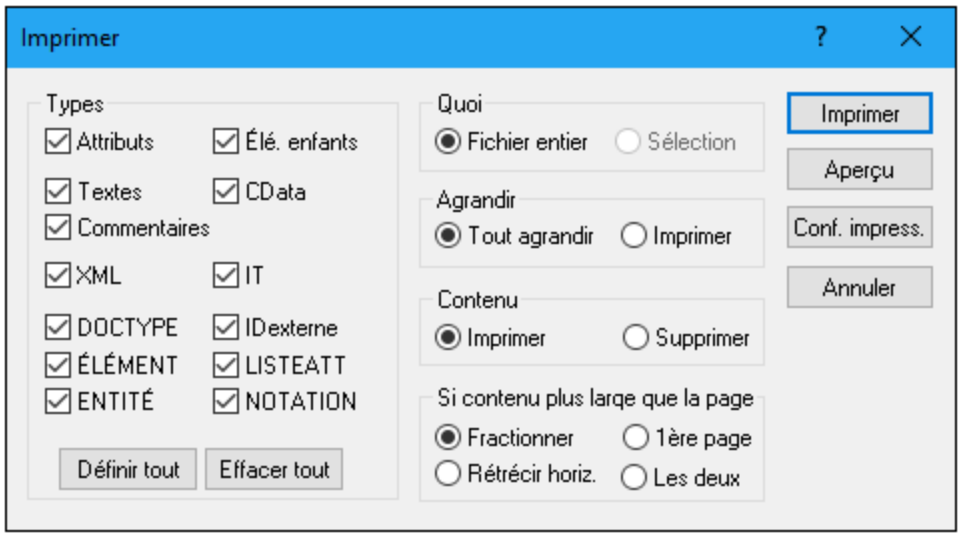
Icône :	
Raccourci :	**Ctrl+P**

Description

La commande **Imprimer** ouvre la boîte de dialogue d'impression, dans laquelle vous pouvez sélectionner les options d'impression pour imprimer le document actif actuellement.

© 2016-2022 Altova GmbH

Cliquer sur la commande **Imprimer** dans le Mode Grille pour ouvrir le dialogue Options d'impression (*capture d'écran ci-dessous*) qui vous permet de configurer les options d'impression pour le document XML. Cliquer sur **Imprimer** dans ce dialogue pour vous rendre au dialogue Impression pour les options de l'imprimante.



Les options disponibles pour l'impression en Mode Grille sont décrites ci-dessous :

- Dans le volet Types, vos sélectionnez les items que vous souhaitez inclure dans la documentation.
- Pour l'option Quoi, vous spécifiez la sélection actuelle ou bien vous imprimez le fichier entier.
- L'option Agrandir vous permet d'imprimer de document tel quel, ou bien avec tous les éléments descendant entièrement agrandis.
- L'option Contenus vous permet de choisir entre le contenu d'impression de tous les nœuds ou les noms de nœuds d'impression uniquement.
- Dans le panneau Si les contenus sont plus larges que la page, vous choisissez comment procéder si le contenu est plus large que la page. L'option "Fractionner les pages" imprime l'ensemble du document à sa taille normale, en répartissant le contenu sur plusieurs pages, aussi bien horizontalement que verticalement. Les pages devraient ensuite être collées ensemble pour former une affiche. L'option Première page n'imprime que la première page, de gauche de la zone d'impression. La zone qui déborde horizontalement n'est pas imprimée. Cette option est utile lorsque la plupart des informations importantes dans le Mode Grille de votre document est contenue dans le côté gauche. L'option Rétrécir horizontalement réduit la taille de la sortie (proportionnellement) jusqu'à ce qu'elle soit à la taille de la page, horizontalement ; le document peut s'étendre sur plusieurs pages. L'option Rétrécir les deux permet de rétrécir le document dans les deux sens jusqu'à ce qu'il entre parfaitement sur une feuille.
- La touche Imprimer imprime le document avec les options sélectionnées.
- La touche Aperçu ouvre un aperçu qui vous laisse voir la sortie finale avant de la consigner sur papier.
- La touche Configuration impression ouvre la boîte de dialogue Configuration impression et vous permet d'ajuster le format papier, l'orientation et d'autres options d'impression pour cette tâche d'impression uniquement. Voir aussi la commande Configuration d'impression.

Note : Vous pouvez changer la largeur des colonnes dans le Mode Grille pour optimiser la sortie d'impression.

Logo du programme

Si vous avez acheté une licence, vous aurez l'option de désactiver le logo de programme, la notice de copyright et les détails d'inscription lorsque vous imprimez un depuis XMLSpy. Cette option est disponible dans la section Affichage du dialogue Options.

Designs SDL et XBRL

Les modes graphiques des documents WSDL et XBRL, tels que vus sur l'écran, peuvent aussi être imprimés avec la commande **Imprimer**.

28.1.9 Aperçu d'impression, paramètres d'impression

Aperçu d'impression

La commande **Aperçu d'impression** cliquée dans le Mode Texte, le Mode Authentic et le Mode Navigation ouvre un aperçu à l'impression du document actuellement actif. De Grid View, Schema View, WSDL View et XBRL View, il ouvre la boîte de dialogue Impression, dans laquelle vous pouvez sélectionner les options d'impression, puis cliquer sur la touche **Aperçu** pour consulter l'aperçu d'impression.

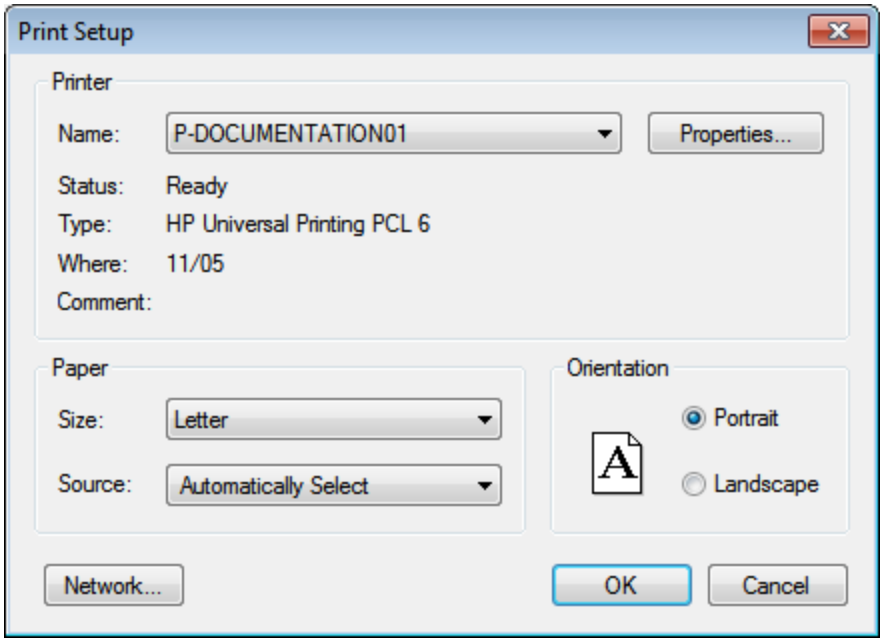
Dans le mode Aperçu d'impression, la barre d'outils Aperçu d'impression en haut à gauche de la fenêtre d'aperçu propose des options d'impression et d'aperçu. L'aperçu peut être agrandi ou rétréci en utilisant les boutons **Zoom avant** et **Zoom arrière**. Lorsque l'agrandissement de la page est tel que la longueur entière de la page s'affiche dans la fenêtre d'aperçu, le bouton **Une page / Deux pages** permet de voir une ou deux pages en même temps. Les boutons **Page suivante** et **Page précédente** peuvent être utilisés pour parcourir les pages. La barre d'outils contient également un bouton pour imprimer toutes les pages et un bouton pour fermer la fenêtre d'aperçu.

Note : afin de permettre les couleurs et les images d'arrière-plan dans l'Aperçu d'impression, suivez les étapes suivantes : (i) dans le menu **Outils** d'Internet Explorer, cliquez sur **Options Internet**, puis sur l'onglet Avancé ; (ii) dans Paramètres, sous Impression, cochez l'option *Imprimer les couleurs et images d'arrière-plan*, puis (iii) cliquez sur **OK**.

Paramètres d'impression

La commande **Configuration de l'impression** affiche le dialogue Paramètres d'impression spécifique à l'imprimante dans laquelle vous pouvez spécifier les paramètres d'impression concernant le format papier et l'orientation de page. Ces paramètres s'appliqueront à toutes les tâches d'impression ultérieures.

© 2016-2022 Altova GmbH



28.1.10 Fichiers récents, Quitter

Fichiers récents

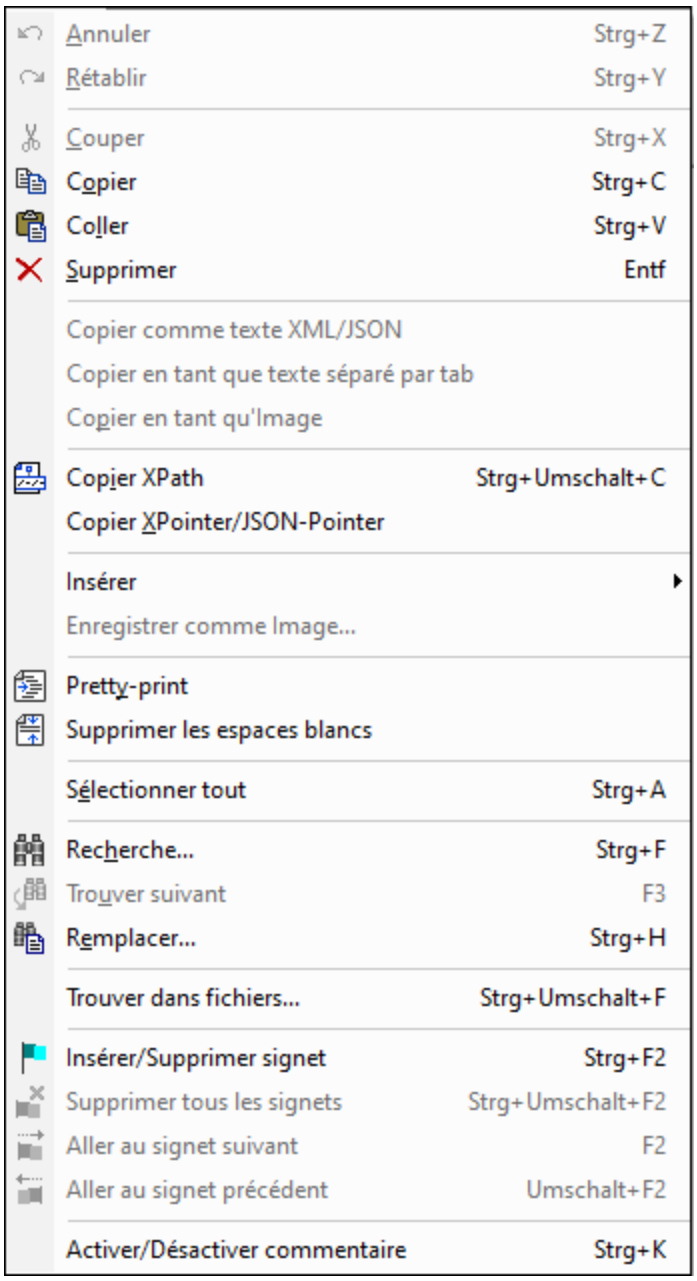
En bas du menu **Fichier**, vous trouverez une liste des neufs fichiers utilisés récemment, le fichier ouvert le plus récemment se trouvant tout en haut de la liste. Vous pouvez ouvrir ces fichiers en cliquant sur leur nom. Pour ouvrir un fichier de la liste à l'aide du clavier, appuyer **Alt+F** pour ouvrir le menu **Fichier**, puis appuyer sur le numéro du fichier que vous souhaitez ouvrir.

Quitter

Quitte XMLSpy. Si d'autres fichiers contenant des changements non enregistrés sont ouverts, vous serez invité à enregistrer ces changements. XMLSpy enregistre aussi les modifications des paramètres de programme et les informations concernant les fichiers utilisés récemment.

28.2 Menu Édition

Le menu **Éditer** contient des commandes pour éditer des documents dans XMLSpy. Ceux-ci contiennent les commandes connues Undo, Redo, Cut, Copy, Paste, Delete, Select All, Find, Find Next et Replace.



XMLSpy propose aussi des commandes spéciales pour :

- Copier une sélection vers le presse-papiers en tant que texte XML,
- Copier en tant que tab/ texte séparé
- Copier un sélecteur Xpath vers l'item sélectionné dans le presse-papier.

© 2016-2022 Altova GmbH

- Insérer et supprimer des signets, et naviguer vers les signets.

28.2.1 Annuler, Rétablir

Icônes et raccourcis

Commande	*Icône*	*Raccourci*
Annuler		**Ctrl+Z**
Rétablir		**Ctrl+Y**

Annuler

La commande **Annuler** prend en charge un nombre illimité de niveaux d'annulation. Chaque action peut être annulée et il est possible d'annuler une commande après l'autre. L'historique d'annulation est conservé après l'utilisation de la commande **Enregistrer**, vous permettant de retourner à l'état original du document avant l'enregistrement de vos changements. Les commandes **Annuler** et **Rétablir** vous permettent de vous déplacer à votre guise dans tout l'historique (*voir la commande **Rétablir** ci-dessous*).

Rétablir

La commande **Rétablir** vous permet de rétablir des commandes annulées précédemment, vous disposez donc d'un historique complet du travail effectué. Les commandes **Annuler** et **Rétablir** vous permettent de vous déplacer à votre guise sur tout l'historique.

28.2.2 Couper, Copier, Coller, Supprimer

Icônes et raccourcis

Commande	*Icône*	*Raccourci*
Couper		**Ctrl+X** ou **Shift+Del**
Copier		**Ctrl+C**
Coller		**Ctrl+V**
Supprimer		**Suppr**

Couper

La commande **Couper** copie le texte ou les éléments sélectionnés sur le presse-papier, supprimant la sélection de son emplacement actuel.

Copier

La commande **Copier** copie le texte ou les éléments sélectionnés sur le presse-papier. Cela peut être utilisé pour dupliquer des données dans XMLSpy ou pour déplacer des données dans une autre application.

Note : Lors de la copie dans le Mode grille, la sélection est copiée en utilisant une des deux méthodes : **Copier comme texte XML** et **Copier comme texte séparé par une tabulation**. La première méthode copie la sélection en tant que texte XML ; la deuxième copie la sélection en tant que table. Laquelle de ces deux méthodes est utilisée lorsque la commande **Copier** est invoquée sera spécifié dans la section Éditer du dialogue Outils | Options.

Coller

La commande **Copier** insère le contenu du presse-papier dans le positionnement actuel du curseur..

Supprimer

La commande **Supprimer** supprime le texte ou les éléments sélectionnés actuellement sans les placer dans le presse-papier.

28.2.3 Copier comme texte XML

La commande **Copier comme texte XML/JSON** copie les données XML ou JSON du Mode Grille comme texte XML (*affiché dans la liste ci-dessous*) ou texte JSON. Le texte sera copié dans le presse-papiers avec sa balise, et peut être collé dans d'autres emplacements du document. Veuillez noter que cette commande est uniquement disponible dans le Mode Grille.

```
<row>
   <para align="left">
      <bold>Check the FAQ</bold>
   </para>
   <para>
      <link mode="internal">
         <link_section>support</link_section>
         <link_subsection>faq2020</link_subsection>
         <link_text>XMLSPY 2020 FAQ</link_text>
      </link>
      <link mode="internal">
         <link_section>support</link_section>
         <link_subsection>faq2021</link_subsection>
         <link_text>XMLSPY 2021 FAQ</link_text>
      </link>
   <para>
<// row>
```

Le formatage du texte suit les paramètres actuels de pretty-printing actifs, qui sont spécifiés dans le dialogue des Options (**Outils | Options**). Le même effet peut être obtenu en passant au Mode Texte et copiant un fragment de texte XML avec **Ctrl+C** (**Éditer | Copier**).

© 2016-2022 Altova GmbH

28.2.4 Copier comme Texte Structuré

La commande **Copier comme texte séparé par tabulation** est activée uniquement si la sélection est une plage de cellules dans l'Affichage Table du Mode Grille XML et Mode Grille JSON. Cette commande peut être utilisée pour copier des données semblables à des tables dans un formulaire tabulaire ou des applications de feuilles de calcul.

Les captures d'écran ci-dessous montrent comment deux lignes dans le Mode Table sont copiées comme texte séparé par tabulation.

expense-item (4)		type	expto	Date	expense
	1	Lodging	Sales	2003-01-01	122.11
	2	Lodging	Development	2003-01-02	122.12

Les deux captures d'écran ci-dessous suivantes affichent les données qui ont été copiées en tant que texte structuré dans respectivement un document Microsoft Excel et dans un document Notepad.

	A	B	C	D	E
1	Lodging	Sales	1/1/2003	122.11	
2	Lodging	Development	1/2/2003	122.12	
3					

Veuillez noter qu'alors qu'Excel (*voir la capture d'écran ci-dessus*) formate automatiquement chaque cellule sur la base de sa forme lexicale de texte, Notepad (*voir la capture d'écran ci-dessous*) colle toutes les cellules en tant que strings.

*Untitled - Notepad

File Edit Format View Help

```
Lodging Sales          2003-01-01      122.11
Lodging Development    2003-01-02      122.12
```

Pour plus d'information, voir les sections Affichage Table (XML) etAffichage Table (JSON).

28.2.5 Copy as Image

La commande **Copier en tant qu'image** est désactivée uniquement si la cellule sélectionnée dans l'Affichage de la table du Mode Grille XML et Mode Grille JSON contient une image. La commande copie le string encodé Base64 de l'image sélectionnée. Si le string est collé dans un document où un string Base64-encoded peut être rendu en tant qu'image (par exemple vers une autre table dans l'Affichage Table), il sera rendu. Autrement, il sera collé en tant que string.

Pour plus d'information, voir les sections Menu contextuel dans le Mode Grille et Menu contextuel dans le Mode Grille JSON.

28.2.6 Copier XPath

La commande **Copier XPath** est disponible dans le Mode Texte et le Mode Grille, et crée une expression XPath qui situe le nœud actuellement sélectionné dans le document et copie l'expression XPath dans le presse-papiers. Cela vous permet de coller l'expression XPath dans un document (par exemple dans un document XSLT). Toutes les expressions commencent à partir de la racine du document. Par exemple, si un élément appelé `LastName` du troisième élément `Person` du second élément `Company` est sélectionné, l'expression XPath qui sera copiée sera : `/Companies/Company[2]/Person[3]/LastName`

Note : dans le Mode Grille, la commande **Copier XPath** peut aussi être accédée depuis le menu contextuel.

28.2.7 Copier XPointer

La commande **Copy XPointer/JSON-Pointer** est disponible dans le Mode Grille de documents XML et JSON et dans le Mode Texte de documents XML. Cette commande copie l'expression XPointer/JSON-Pointer vers le presse-papiers qui situe le nœud sélectionné.

- *Documents XML* La commande crée un schéma **`element()`** XPointer dans le/s nœuds sélectionné/s du document XML et copie XPointer dans le presse-papier. Par exemple, l'élément (1/3) XPointer sélectionne le troisième enfant de l'élément du document (ou élément racine).
- *Documents JSON :* La commande crée un JSON-Pointer dans le/s nœud/s sélectionné/s du document JSON et copie JSON-Pointer dans le presse-papier. Par exemple, JSON-Pointer **`/Artists/1/Albums/2/Tracks/3/Title`** sélectionne un nœud JSON comme suit : Recherchez le premier objet du niveau supérieur de la table **`Artists`** ; dans lequel vous rechercherez le deuxième objet de la table **`Albums`** ; dans lequel vous recherchez le troisième objet de la table **`Tracks`** ; dans lequel vous sélectionnerez l'objet **`Title`**.

Veuillez prendre note des points suivants :

- Les attributs XML ne peuvent pas être représentés en utilisant le schéma **`element()`**. Si un attribut est sélectionné, XPointer de l'élément parent de l'attribut est généré.
- Si de multiples éléments XML sont sélectionnés, le XPointer du premier élément est généré.
- Si une valeur JSON est sélectionnée, le JSON-Pointer de la clé de la valeur est généré.
- Si de multiples nœuds JSON sont sélectionnés, le JSON-Pointer du premier élément est généré.

Note : la commande **Copier XPointer/JSON-Pointer** est aussi accessible depuis le menu contextuel.

28.2.8 Insérer

Passer avec la souris sur ou sélectionner la commande **Insérer** pour faire défiler un sous-menu contenant trois commandes (décrit ci-dessous) :

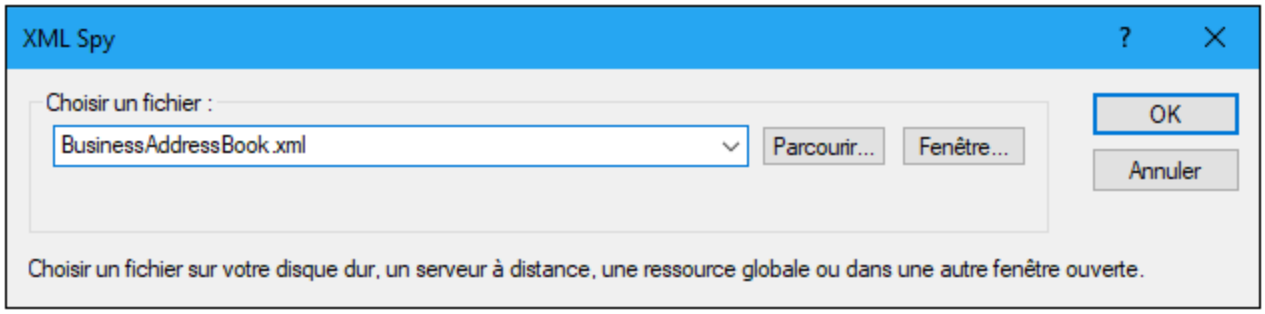
- Insérer le chemin de fichier
- Insérer XInclude

© 2016-2022 Altova GmbH

- Insérer le fichier externe encodé

Insérer le chemin de fichier

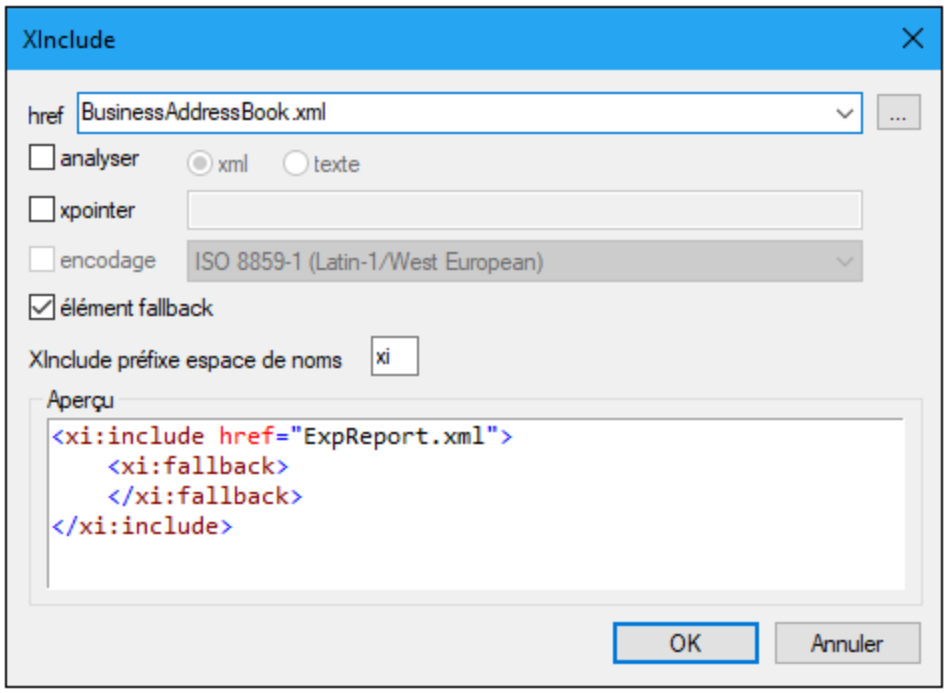
La commande **Chemin de fichier** est activée dans le Mode Texte et le Mode Grille des documents de tout type de fichier. Grâce à elle, vous pouvez insérer le chemin vers un fichier au niveau de la sélection du curseur. Cliquez sur la commande pour ouvrir un dialogue (*capture d'écran ci-dessous*) dans lequel vous pouvez sélectionner le fichier requis.



Le fichier requis peut être sélectionné en choisissant une des méthodes suivantes : (i) en parcourant le fichier, l'URL ou la ressource globale (utiliser la touche **Parcourir**) ; (ii) en choisissant la fenêtre dans laquelle le fichier est ouvert (la touche **Fenêtre**). Une fois terminé, cliquez sur **OK**. Le chemin vers le fichier sélectionné sera inséré dans le document actif au niveau de la sélection du curseur.

Insérer XInclude

La commande **XInclude** est disponible dans le Mode Texte et le Mode Grille et vous permet d'insérer un nouvel élément XInclude au point de sélection du curseur dans le Mode Texte ou avant l'item sélectionné dans les deux Modes Texte et Grille. Si dans le Mode Grille la sélection actuelle est un attribut, l'élément XInclude est inséré après l'attribut et avant le premier élément enfant de l'élément parent de l'attribut. Choisir cette commande pour ouvrir le dialogue XInclude (*capture d'écran ci-dessous*).



Le fichier XML à inclure est saisi dans le champ de texte `href` (en alternative, vous pouvez chercher le fichier en cliquant sur la touche **Chercher (...)** située à droite du champ de saisie). Le nom de fichier sera saisi dans le document XML en tant que la valeur de l'attribut `href`. Les attributs `parse`, `xpointer` et `encoding` de l'élément XInclude (`xi:include`) et l'élément enfant `fallback` de `xi:include` peuvent aussi être insérés par le biais du dialogue. Veuillez tout d'abord cocher la case appropriée puis sélectionner/saisir les valeurs requises. Dans le cas de l'élément `fallback`, le fait de cocher sa case insère uniquement l'élément vide. Le contenu de l'élément `fallback` doit être ajouté ultérieurement dans un des modes d'édition.

L'attribut `parse` détermine si le document inclus doit être parsé en tant que XML ou texte. (XML est la valeur par défaut et ne doit donc pas être spécifiée.) L'attribut `xpointer` identifie un fragment spécifique du document situé avec l'attribut `href` ; c'est ce fragment qui sera inclus. L'attribut `encoding` spécifie l'encodage du document inclus de manière à ce que XMLSpy puisse transcoder ce document (ou la partie devant être incluse) dans l'encodage du document incluant. Les contenus de l'élément enfant `fallback` remplacent l'élément `xi:include` si le document à inclure ne peut pas être situé.

Voici un exemple d'un document XML qui utilise XInclude pour inclure deux documents XML :

```
<?xml version="1.0" encoding="UTF-16"?>
<AddressBook xsi:schemaLocation="http://www.altova.com/sv/myaddresses AddressBook.xsd"
       xmlns="http://www.altova.com/stylevision/tutorials/myaddresses"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xi="http://www.w3.org/2001/XInclude">
       <xi:include href="BusinessAddressBook.xml"/>
       <xi:include href="PersonalAddressBook.xml"/>
</AddressBook>
```

Lorsque ce document XML est parsé, il remplacera les deux éléments XInclude avec les fichiers spécifiés dans les attributs `href` respectifs.

© 2016-2022 Altova GmbH

xml:base
Lorsque le validateur XML de XMLSpy lit un document XML et rencontre l'élément `include` dans l'espace de noms XInclude (dénommé ci-après `xi:include`), il remplace cet élément (`xi:include`) avec le document XML nommé dans l'attribut `href` de l'élément `xi:include`. L'élément de document (élément racine) du document XML inclus (ou de l'élément identifié par un XPointer) sera inclus avec un attribut de `xml:base` pour pouvoir préserver les URI de base de l'élément inclus. Si le document XML résultant (contenant le document XML inclus ou le fragment d'arborescence) doit être valide conformément à un schéma, alors l'élément de document du document inclus (ou l'élément de niveau supérieur du fragment d'arborescence) doit être créé avec un modèle de contenu qui permettra un attribut de `xml:base`. Si, conformément au schéma, l'attribut `xml:base` n'est pas autorisé sur cet élément, le document résultant sera invalide. Vous trouverez dans la section Attributs xml:Prefixed de la section Mode Schéma de la documentation une description pour définir un attribut `xml:base` dans un modèle de contenu d'un élément à l'aide du Mode de Schéma de XMLSpy .

XPointers

XMLSpy prend en charge les XPointers dans XInclude. Les recommandations W3C pertinentes sont les recommandations XPointer Framework et système XPointer element(). L'utilisation d'un XPointer dans un élément XInclude permet d'inclure une partie spécifique du document XML, au lieu de devoir inclure le document XML entier. Les XPointers sont utilisés dans un élément XInclude comme suit :

```
<xi:include href="PersonalAddressBook.xml" xpointer="element(usa)"/>
<xi:include href="BusinessAddressBook.xml" xpointer="element(/1/1)"/>
<xi:include href="BobsAddressBook.xml" xpointer="element(usa/3/1)"/>
<xi:include href="PatsAddressBook.xml" xpointer="element(usa)element(/1/1)"/>
```

Dans le système element() de XPointer, il est possible d'utiliser un NCName ou une séquence enfant dirigée par des entiers.

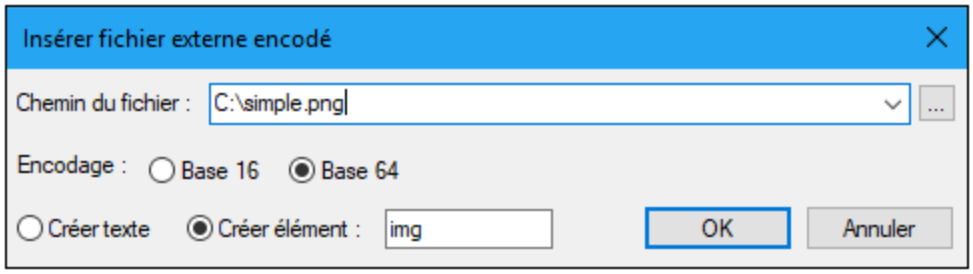
- Dans le premier élément `xi:include` nommé ci-dessus, l'attribut `xpointer` utilise le système d'élément avec un NCName de `usa`. Conformément au XPointer Framework, cet NCName identifie l'élément qui a une ID de `usa`.
- Dans le second `xi:include` listé ci-dessus, l'attribut `xpointer` avec une valeur de `element(/1/1)` identifie, dans une première étape, le premier élément enfant de la racine de document (qui, si le document est bien formé, sera son élément de document (ou de racine)). Dans une seconde étape, le premier élément enfant de l'élément situé dans l'étape précédente est situé ; dans notre exemple, il s'agit du premier élément enfant de l'élément de document.
- L'attribut `xpointer` du troisième `xi:include` listé ci-dessus utilise une combinaison de NCName et de la séquence enfant. Ce XPointer localise le premier élément enfant du troisième élément enfant ayant une ID de `usa`.
- Si vous n'êtes pas sûr si votre premier XPointer fonctionnera, vous pouvez le compléter avec un second, comme montré dans le quatrième `xi:include` listé ci-dessus : `xpointer="element(usa) element(/1/1)""`. Ici, s'il n'y a aucun élément avec une ID de `usa`, le XPointer de réserve spécifie que le premier élément enfant de l'élément de document doit être sélectionné. Des réserves supplémentaires sont autorisées. Des XPointers individuels peuvent ne pas être séparés ou ils peuvent être séparés par un espace blanc : par exemple, `xpointer="element(usa)element(addresses/1) element(/1/1)"`.

Note : Le contexte liant d'espace de noms n'est pas utilisé dans le système element() parce que celui-ci ne prend pas en charge les noms qualifiés.

Insérer le fichier externe encodé

La commande **Fichier externe encodé** est disponible dans le Mode Texte et le Mode Grille. Elle vous permet d'inclure un fichier externe en tant que texte encodé en Base-16 ou Base-64 à tout endroit dans le document XML. Cette fonction permet d'intégrer les fichiers externes dans le document XML. Cette fonction permet aux fichiers externes d'être incorporés dans le document XML.

Cliquer sur la commande **Insérer | Fichier externe encodé** ouvre le dialogue Insérer le fichier externe encodé (*capture d'écran ci-dessous*).



Vous pouvez chercher ou saisir le nom du fichier externe à encoder et à intégrer. Spécifiez un encodage Base-16 ou Base-64. Si vous souhaitez ajouter le texte encodé dans un élément, cochez la case *Créer élément* et spécifiez le nom de l'élément désiré (*voir la capture d'écran ci-dessus*). En alternative, sélectionnez *Créer texte* pour insérer le texte encodé directement à l'emplacement du curseur.

Cliquez sur **OK** pour insérer le texte encodé du fichier sélectionné au niveau du curseur, avec un élément contenant si cela a été précisé.

```
<img ext="png" encoding="xs:base64Binary">
  iVBORw0KGgoAAAANSUhEUgAAABAAAAAQAQMAAAAlPW0iAAAABlBMVEUAAAD/
  //+l2Z/dAAAAM0lEQVR4nGP4/5/h/1+G/58ZDrAz3D/McH8yw83NDDeNGe4U
  g9C9zwz3gVLMDA/A6P9/AFGGFyjOXZtQAAAAAElFTkSuQmCC
</img>
```

La liste ci-dessus montre le texte encodé d'un fichier d'image PNG. Un élément `img` a été créé autour du texte encodé.

28.2.9 Enregistrer comme image

Cette commande est activée lorsque une image encodée Base64 est sélectionnée en mode Texte ou en mode Grille. Elle convertit le string encodée Base64-encoded sélectionné en son format d'image. (Notez qu'il s'agit d'un string Base64 de l'image qui est affichée en mode Texte, mais c'est l'image générée depuis le string affiché en mode Grille.) Lorsque vous sélectionnez la commande, un dialogue Enregistrer sous apparaît. Dans celui-ci, sélectionnez l'emplacement sous lequel vous souhaitez enregistrer l'image et saisissez un nom pour le fichier d'image. L'extension du fichier d'image (`.png`, `.gif`, `.svg`, etc.) sera détectée automatiquement depuis l'encodage Base64 et apparaîtra dans le dialogue Enregistrer.

Pour plus d'information, voir les descriptions du mode Texte et du mode Grille dans les sections sur les documents XML et JSON.

© 2016-2022 Altova GmbH

28.2.10 Pretty-Print

Icône

Icône :	

Description

La commande **Pretty-Print** reformate le formatage texte de votre document XML ou JSON. Les propriétés de formatage sont spécifiées dans les paramètres Pretty-printing du dialogue des Options (**Outils | Options**). L'indentation de texte est spécifiée dans le dialogue Paramètres du Mode Texte (**Mode | Paramètres du Mode Texte**).

Veuillez noter les points suivants

- Le document XML doit être bien formé pour que la commande fonctionne.
- La fonction Pretty-printing ajoute des espaces ou des tabs au document lorsque celui-ci est enregistré.
- Pour supprimer tous les espaces (nouvelles lignes et indentation) créés avec la commande, utilisez la commande Supprimer les espaces blancs.

28.2.11 Supprimer les espaces blancs

Icône

Icône :	

Description

La commande **Supprimer les espaces blancs** supprime tous les espaces blancs du document. Cela peut contribuer à réduire la taille du fichier. Cette commande est utile si vous souhaitez supprimer un espace blanc généré par la commande **Pretty-Print** .

28.2.12 Sélectionner tout

La commande **Sélectionner tout** (**Ctrl+A**) sélectionne le contenu du document entier.

28.2.13 Recherche, Trouver suivant

Icônes et raccourcis

Commande	*Icône*	*Raccourci*
Recherche		**Ctrl+F**
Trouver suivant		**F3**

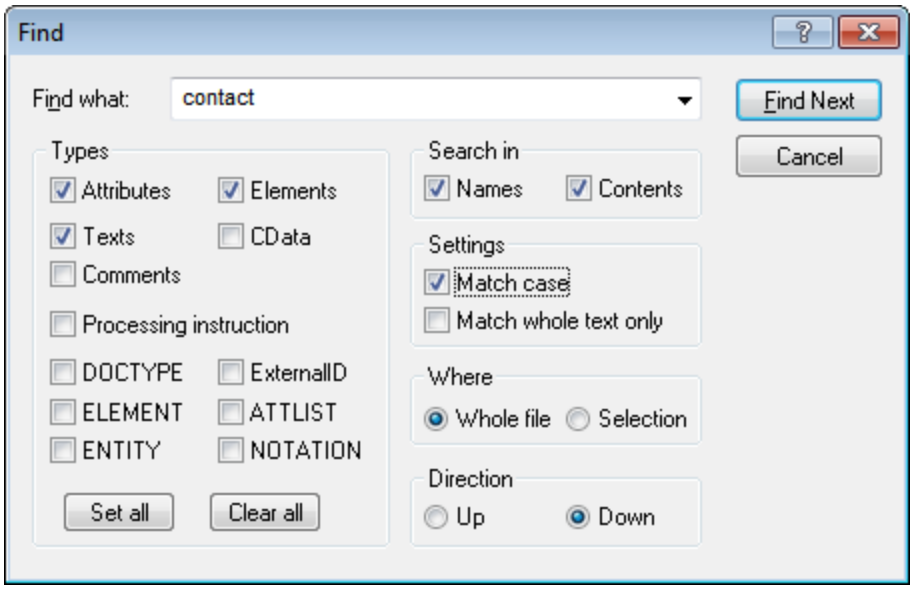
Recherche

La commande **Recherche** affiche le dialogue Recherche dans lequel vous pouvez spécifier le string que vous souhaitez trouver et les autres options pour la recherche. Selon le mode actuel, le dialogue Recherche affiche des options différentes. Pour trouver le texte, le saisir dans le champ de saisie ou utiliser la liste de choix pour sélectionner un terme parmi les 10 derniers termes recherchés, puis spécifier les options de recherche.

Les commandes **Recherche** et **Trouver suivant** peuvent aussi être utilisées pour trouver des noms de fichier et de dossier lorsqu'un projet est sélectionné pour la fenêtre Projet.

Mode Grille

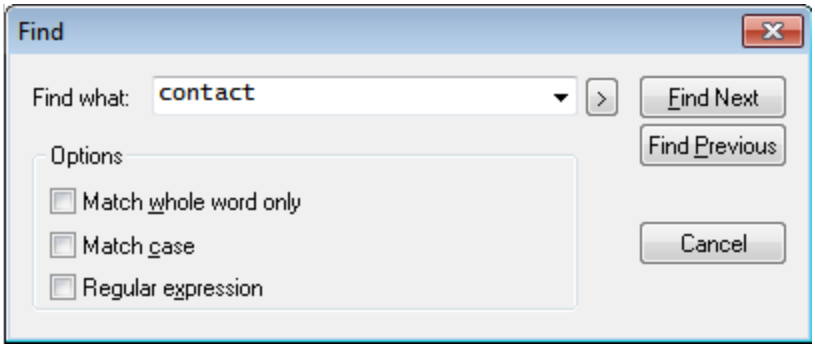
Dans le Mode Grille, le dialogue suivant apparaît. Sélectionnez les options dont vous avez besoin ou sélectionnez un bouton radio. Les options disponibles sont décrites ci-dessous.



- Le volet *Types* vous permet de choisir les nœuds et les composants du document XML que vous souhaitez inclure dans la recherche. Cela permet à la recherche de sauter des types de nœuds particuliers et d'aller plus loin. La touche **Définir tout**sélectionne tous les types de nœud ; **Effacer tout** désélectionne tous les types de nœud.
- Le volet *Rechercher dans* vous permet de définir si les noms d'un nœud, les contenus d'un nœud ou les deux doivent être recherchés pour le string de texte d'entrée.
- Le volet *Paramètres* vous permet de définir si la recherche doit être sensible à la casse et/ou correspondre au string d'entrée complet.
- Le volet *Où* vous permet de définir l'étendue de la recherche (le fichier entier ou le texte sélectionné).
- L'option *Direction* spécifie la recherche de la direction.

© 2016-2022 Altova GmbH

Mode Texte
Pour une description de la fonction Trouver des Modes Texte éditables, voir la section Mode Texte. Le dialogue Recherche des Modes de texte non éditables (tel que dans la fenêtre XPath/XQuery) est affiché dans la capture d'écran ci-dessous.



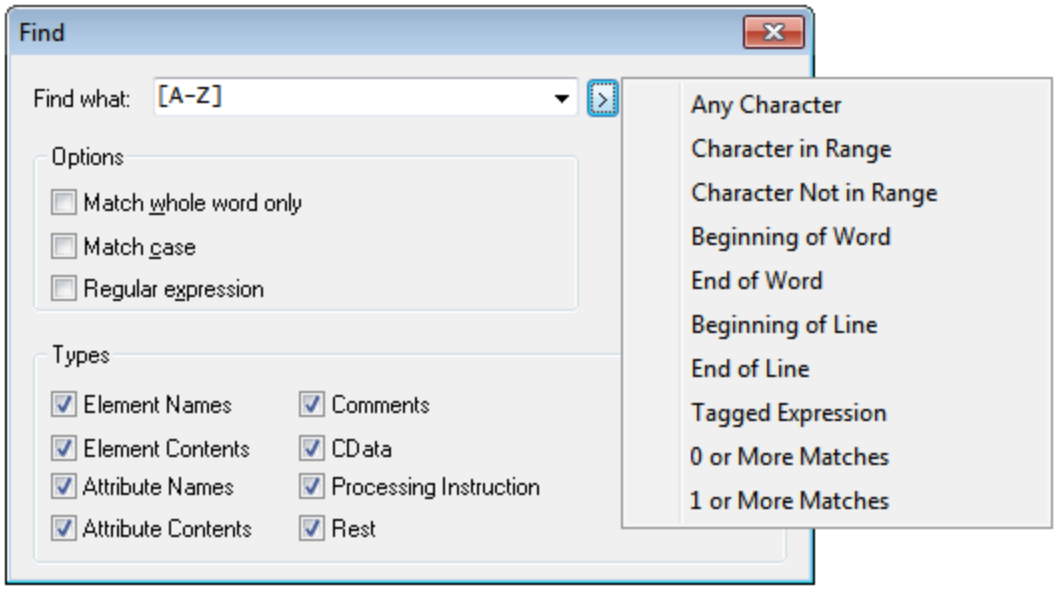
Les options disponibles sont comme suit:

- *Correspondance du mot entier*: Seuls les mots exacts contenus dans le texte seront comparés. Par exemple, pour le string entrée `fit`, avec *Match whole word only* vérifié, seul le mot `fit` correspondra au string de recherche ; `fit` dans `fitness`, par exemple, ne correspondra pas.
- *Respecter la casse*: Recherche sensible à la casse (`Address` n'est pas la même chose que `address`).
- *Expression régulière*: Recherche du texte spécifié par l'expression régulière que vous saisissez dans la zone de texte. Voir Expressions régulières ci-dessous pour une description des expressions régulières.

Veuillez prendre note des points suivants :

- Le dialogue Recherche est "dépourvu de mode", ce qui signifie qu'il peut rester ouvert alors que vous continuez à utiliser le Mode Texte. Appuyez sur **Enter** pendant que le dialogue est ouvert, pour le fermer. Si le texte est marqué avant d'ouvrir le dialogue, le texte marqué sera automatiquement inséré dans le champ de saisie Rechercher.
- Une fois que le dialogue Recherche est fermé, vous pouvez renouveler la recherche actuelle en appuyant sur **F3** pour une recherche vers l'avant ou **Shift+F3** pour une recherche vers l'arrière.
- La touche **Dérouler** située à droite de la touche *Rechercher* (la touche marquée **>**), ouvre une deuxième fenêtre que vous pourrez également utiliser pour saisir des expressions régulières dans le champ *Rechercher*.

Expressions régulières
Vous pouvez utiliser des expressions régulières pour affiner encore vos critères de recherche. Une liste pop-up est disponible pour vous aider à construire des expressions régulières. Pour accéder à cette liste, cliquez sur la touche **Dérouler** à droite du champ de saisie *Rechercher* (la touche marquée du symbole **>**)).



Cliquer sur la description d'expression requise permet d'insérer la syntaxe d'expression correspondante dans le champ d'entrée. Ci-dessous, vous trouverez une liste des caractères de syntaxe d'expression régulière.

`.`	Correspond à n'importe quel caractère. Il s'agit d'un espace réservé pour un seul caractère.
`(`	Marque le début d'une région pour marquer une correspondance.
`)`	Marque la fin d'une région marquée.
`\n`	Où `n` est `1` à `9` se réfère à la région marquée de 1 à 9 lors du remplacement. Par exemple, si le string de recherche était `Fred([1-9])XXX` et que le string de remplacement était `Sam\1YYY`, une fois appliqué à `Fred2XXX` cela générera `Sam2YYY`.
`\<`	Correspond au début d'un mot.
`\>`	Correspond à la fin d'un mot.
`\x`	Vous permet d'utiliser un caractère `x`, qui normalement n'aurait pas de signification particulière. Par exemple, `\[` sera interprété comme `[` et non pas comme le début d'un jeu de caractères.
`[...]`	Indique un *ensemble de caractères*. Par exemple, `[abc]` signifie tous les caractères `a`, `b` ou `c`. Vous pouvez aussi utiliser des plages : par exemple `[a-z]` pour tout caractère en minuscule.
`[^...]`	Le complément des caractères dans l'ensemble. Par exemple, `[^A-Za-z]` signifie n'importe quel caractère sauf un caractère alphabétique.
`^`	Correspond au démarrage d'une ligne (sauf si utilisé à l'intérieur d'un ensemble, *voir ci-dessus*).
`$`	Correspond à la fin d'une ligne. Exemple : `A+$` pour trouver un ou plusieurs `A` à la fin de la ligne.
`*`	Correspond à 0 ou plus. Par exemple, `Sa*m` correspond à `Sm`, `Sam`, `Saam`, `Saaam` etc.

 © 2016-2022 Altova GmbH

+	Correspond à 1 ou plus. Par exemple, `Sa+m` correspond à `Sam`, `Saam`, `Saaam` etc.

Note : Les expressions régulières ne sont pas prises en charge dans le champ Remplacer.

Trouver suivant

La commande **Trouver suivant** répète la dernière commande Chercher. Elle cherche l'occurrence suivante du texte saisi.

28.2.14 Remplacer

Icônes et raccourcis

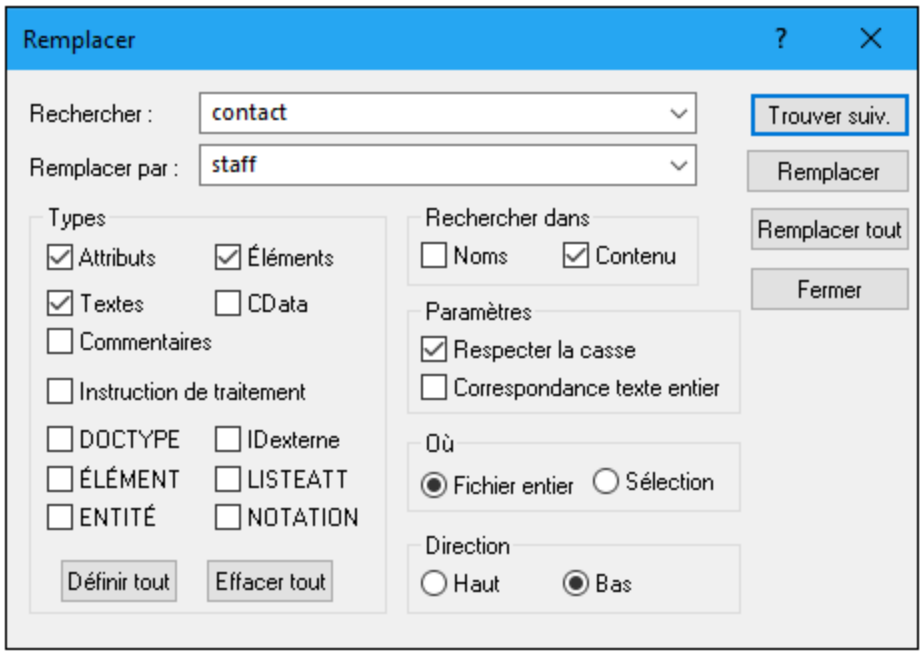
Commande	*Icône*	*Raccourci*
Remplacer		**Ctrl+H**

Description

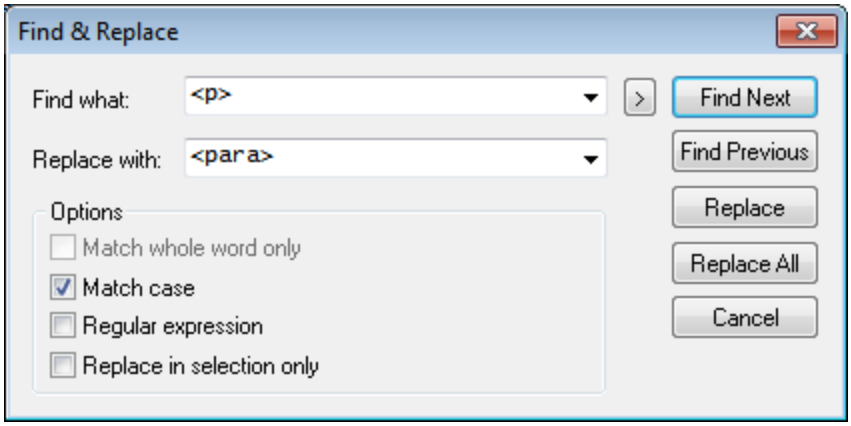
La commande **Remplacer** vous permet de trouver et de remplacer un string de texte après l'autre. Elle présente les mêmes options que la commande **Recherche**. Selon le mode que vous utilisez, le dialogue Remplacer affiche des options de recherche différentes. Vous pouvez remplacer les items individuellement ou vous pouvez utiliser la touche **Remplacer tout** pour effectuer une opération globale chercher-et-remplacer.

Mode Grille

Dans le Mode Grille, choisir la commande **Remplacer** pour ouvrir le dialogue Remplacer (*capture d'écran ci-dessous*). La capture d'écran montre les options variées Rechercher, qui sont présentées dans la description de la commande **Recherche** .



Mode Texte
Pour une description de la fonction Remplacer des Modes Texte éditables, voir la section Mode Texte. Le fonction Remplacer des Modes de texte non éditables (telle que dans la fenêtre XPath/XQuery) est décrite ci-dessous. Dans le Mode Texte, sélectionnez la commande **Remplacer** pour ouvrir le dialogue Chercher & Remplacer (*capture d'écran ci-dessous*). L'option Recherche est la même que pour la commande **Recherche** . L'option *Remplacer dans sélection uniquement* exécute l'opération rechercher et remplacer uniquement à l'intérieur de la sélection de texte.

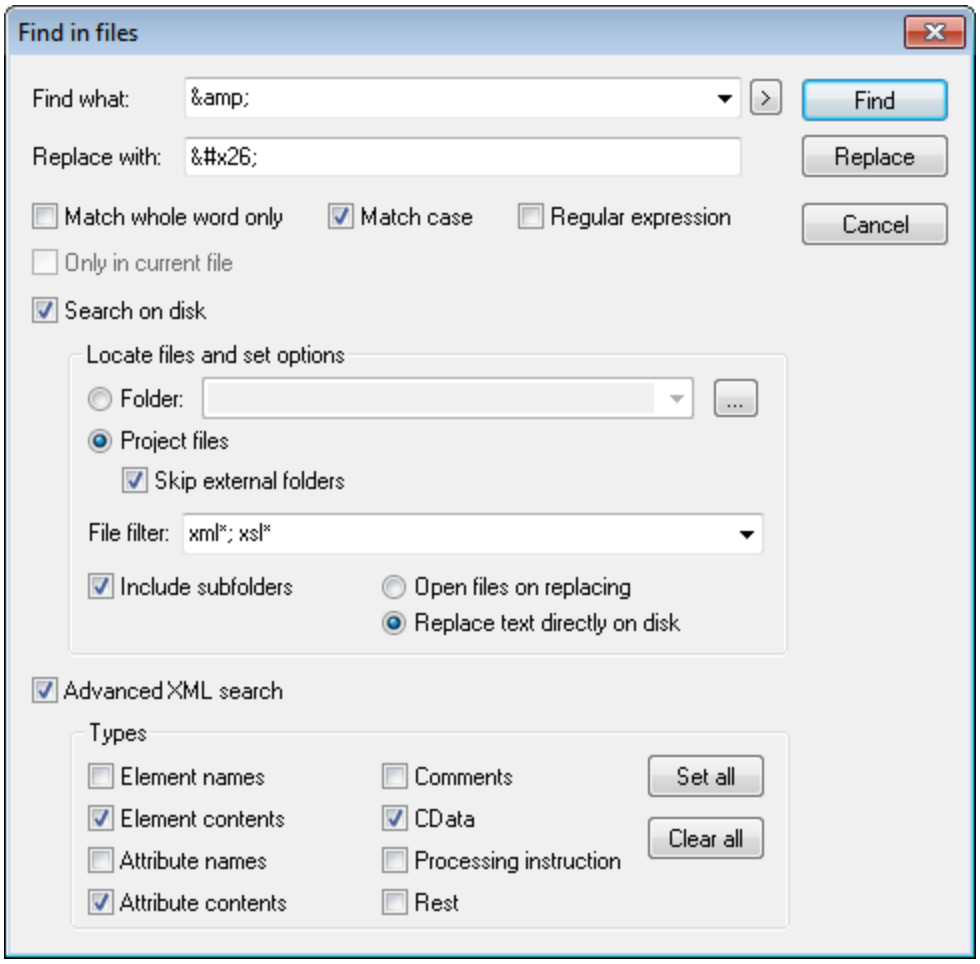


Note : Lorsque vous utilisez la commande **Remplacer tout**, chaque remplacement est enregistré en tant qu'opération individuelle, donc **Remplacer tout** peut être annulé étape par étape.

© 2016-2022 Altova GmbH

28.2.15 Chercher dans fichiers

La commande **Chercher dans fichiers** est une technique très efficace pour trouver et remplacer du texte rapidement dans de nombreux fichiers. Cliquer sur la commande pour faire apparaître le dialogue Chercher dans fichiers (*capture d'écran ci-dessous*). La commande **Chercher dans fichiers** est différente de la commande **Trouver** en ce sens qu'elle recherche tous les emplacements spécifiés pour le string Trouver en une fois et exécute les actions de remplacement directement. Un rapport est ensuite affiché dans la Fenêtre Rechercher dans fichiers. Néanmoins, dans le cas de la commande **Recherche**, l'utilisateur saisit le texte de recherche et parcourt le document actif (unique) un item à la fois.



Critères de recherche

Il existe deux grands critères de recherche : (i) ce que vous souhaitez chercher, et (ii) l'endroit où chercher. Pour la description pour savoir comment définir le texte qui doit être recherché (Quoi trouver), consultez la commande **Trouver**. Si le texte saisi dans la case « Find What » est une expression régulière, alors la case à cocher Expression régulière doit être cochée. Un assistant à la saisie pour les expressions régulières peut être accédé en cliquant sur la touche **Dérouler** située à droite du champ de saisie *Rechercher* (le bouton marqué du signe **>**) . L'utilisation des expressions régulières pour la recherche est expliquée dans la section concernant la commande **Recherche**.

Afin de spécifier quels types et parties de nœud d'un document XML doit être cherché, cocher la case Recherche avancée XML, puis sélectionner les types de nœud requis.

Vous pouvez spécifier le type de fichiers que vous souhaitez chercher en cochant soit *Dans le présent fichier seulement* ou *Chercher sur le disque*. Si vous choisissez de chercher sur le disque, vous pouvez choisir un dossier ou un projet à rechercher (après avoir coché la case *Chercher sur le disque*). Lorsqu'un dossier de projet a été sélectionné, les dossiers externes rajoutés au projet peuvent être sautés. Les fichiers à chercher peuvent être filtrés par les extensions de fichier et un astérisque (`xml*` ou `xsl*`, par exemple). Le séparateur entre les deux extensions de fichier peut être une virgule ou un point-virgule (`xml*;xsl*`, par exemple). Le caractère de l'astérisque peut aussi être utilisé en tant que caractère générique.

Les instances du string Rechercher à tous les endroits de la recherche sont listés dans la fenêtre Rechercher dans fichiers. Cliquez sur un des items listés pour ouvrir ce fichier dans le Mode Texte et le marquer.

Remplacer

La chose la plus importante à noter est que le fait de cliquer sur la touche **Remplacer** remplace toutes les instances du string Rechercher avec le string Remplacer. Si l'option *Ouvrir les fichiers au remplacement* a été cochée dans le dialogue Rechercher dans les fichiers, le fichier sera ouvert dans le Mode Texte ; sinon le remplacement s'effectuera silencieusement. Tous les strings remplacés sont listés dans la fenêtre Rechercher dans fichiers. Cliquez sur un des items listés pour ouvrir ce fichier dans le Mode Texte et le marquer.

Note : Les expressions régulières ne sont pas prises en charge dans le champ Remplacer.

28.2.16 Commandes signet

Icônes et raccourcis

Commande	*Icône*	*Raccourci*
Insérer/Supprimer signet		**Ctrl+F2**
Supprimer tous les signets		**Ctrl+Shift+F2**
Aller au signet suivant		**F2**
Aller au signet précédent		**Shift+F2**

Insérer/Supprimer signet

La commande **Insérer/Supprimer signet** insère un signet dans la position actuelle du curseur, ou supprime le signet si le curseur se trouve dans une ligne qui a été mise sous signet précédemment. Cette commande est uniquement disponible dans le Mode Texte.

Les lignes contenant un signet sont affichées de l'une des manières suivantes :

- Si la marge à signet a été activée, une ellipse bleu apparaît à la gauche du texte dans la marge à signet.

© 2016-2022 Altova GmbH

- Si la marge à signets n'a pas été activée, toute la ligne contenant le curseur sera marquée.

La touche **F2** passe à travers tous les signets contenus dans le document.

Supprimer tous les signets

La commande **Supprimer tous les signets** supprime tous les signets actuellement définis. Cette commande est uniquement disponible dans le Mode Texte. Veuillez noter que la commande **Annuler** n'annule pas les effets de **Supprimer tous les signets**.

Aller au signet suivant

La commande **Aller au signet suivant** place le curseur du texte au début de la ligne contenant le signet suivant. Cette commande est uniquement disponible dans le Mode Texte.

Aller au signet précédent

La commande **Aller au signet précédent** place le curseur du texte au début de la ligne contenant le signet précédent. Cette commande est uniquement disponible dans le Mode Texte.

28.2.17 Commenter/Décommenter

La commande **Commenter/Décommenter** est disponible dans le Mode Texte et est utilisée pour commenter et décommenter des fragments de texte XML. Le texte dans un document XML peut être désactivé en utilisant les délimiteurs de début de commentaire et de fin de commentaire XML, respectivement `<!--` et `-->`. Dans XMLSpy, ces délimiteurs de commentaire peuvent être insérés autour d'une sélection de texte en utilisant le menu de commande **Commenter/Décommenter**.

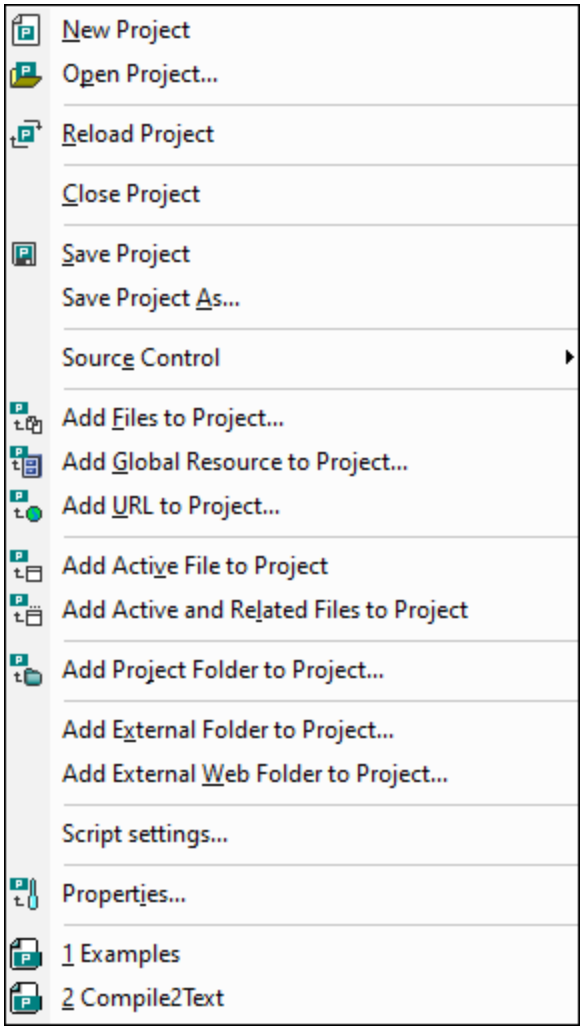
Pour décommenter un bloc de texte, sélectionner le texte à décommenter, puis sélectionner la commande **Commenter/Décommenter**, soit depuis le menu **Édition**, soit depuis le menu contextuel (touche de droite de la souris sur le texte sélectionné). Le texte commenté sera grisé (*voir capture d'écran ci-dessous*).

```
<Department>
    <Name>Administration</Name>
    <Person>
    <Person>
    <Person>
    <!--<Person>
        <First
        <Last></Last>
        <PhoneExt></PhoneExt>
        <EMail></EMail>
        <LeaveTotal></LeaveTotal>
        <LeaveUsed></LeaveUsed>
        <LeaveLeft></LeaveLeft>
    </Person>-->
</Department>
```

Pour décommenter un texte commenté, placer le curseur dans le bloc commenté et sélectionner la commande **Commenter/Décommenter**, soit depuis le menu **Édition**, soit depuis le menu contextuel (touche de droite de la souris sur le texte sélectionné). Les délimiteurs de commentaire seront supprimés et le texte ne sera plus grisé.

28.3 Menu Projet

XMLSpy utilise le mode bien connu d'arborescence pour gérer des fichiers ou les URL multiples dans les projets XML. Les fichiers et URL peuvent être réunis dans des dossiers regroupés par une extension en commun ou tout autre critère arbitraire, permettant une structuration et une manipulation batch en toute simplicité.



Veuillez noter : La plupart des commandes liées au projet sont aussi disponibles dans le menu contextuel, qui apparaît lorsque vous cliquez avec la touche de droite sur un item quelqconque dans la fenêtre de projet.

Chemins absolus et relatifs

Chaque projet est enregistré en tant que fichier de projet et a l'extension `.spp`. Ces fichiers sont en fait des documents XML que vous pouvez éditer comme tout fichier XML habituel. Dans le fichier de projet, par exemple, les chemins absolus sont utilisés pour des fichiers/dossiers de même niveau ou plus haut, et les chemins relatifs pour les fichiers/dossiers dans le dossier actuel ou dans les sous-fichiers. Par exemple, si votre structure de répertoire ressemble à cela :

© 2016-2022 Altova GmbH

```
|-Folder1
|       |
|       |-Folder2
|               |
|               |-Folder3
|                       |
|                       |-Folder4
```

Si votre fichier `.spp` est situé dans le `Folder3`, alors les références vers les fichiers dans `Folder1` et `Folder2` auront environ l'aspect suivant :

```
c:\Folder1\NameOfFile.ext
c:\Folder1\Folder2\NameOfFile.ext
```

Les références aux fichiers dans le `Folder3` et `Folder4` auront environ l'aspect suivant :

```
.\NameOfFile.ext
.\Folder4\NameOfFile.ext
```

Si vous souhaitez vous assurer que tous les chemins seront relatifs, enregistrer les fichiers `.spp` dans le répertoire racine de votre disque de travail.

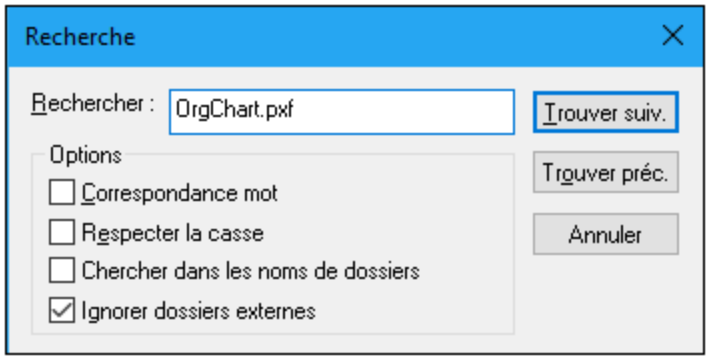
Glisser/déposer

Dans la fenêtre Projet, un dossier peut être glissé dans un autre dossier ou dans un autre emplacement dans le même dossier. Un fichier peut être glissé dans un autre dossier, mais ne peut pas être déplacé dans le sein du même dossier (dans lequel les fichiers sont classés par ordre alphabétique). De plus, les fichiers et dossiers peuvent être glissés depuis Windows File Explorer vers la fenêtre de projet.

Rechercher dans le projet

Vous pouvez chercher les fichiers de projet et les dossiers en utilisant leurs noms ou une partie de leur nom. Si la recherche aboutit, les fichiers ou les dossiers trouvés seront marqués un par un.

Pour lancer une recherche, Choisir la fenêtre du Projet en cliquant dessus (ou dedans), puis sélectionner la commande **Édition | Recherche** (ou le raccourcis **Ctrl+F**). Dans le dialogue Recherche qui apparaît (*capture d'écran ci-dessous*), saisir le texte que vous souhaitez chercher et sélectionner ou désélectionner les options de recherche (*explications ci-dessous*) conformément à vos exigences.



Les options de recherche suivantes sont disponibles :

- La correspondance mot est plus limitée puisque le texte entier doit correspondre à un mot entier dans le nom du fichier ou du dossier. Dans les noms du fichier, les parties avant et après le point (sans le point) sont toutes traitées en tant qu'un mot.
- Vous pouvez préciser que la casse du mot recherché doit correspondre exactement au string contenu dans le nom du fichier ou du dossier.
- Les noms de dossier peuvent être inclus à la recherche. Dans le cas contraire, seuls les noms de fichiers seront recherchés.
- Les dossiers externes peuvent être inclus ou exclus de la recherche. Les dossiers externes sont des dossiers réels dans le système ou le réseau, contrairement aux dossiers de projet qui sont créés dans le cadre du projet et non dans le système.

Si la recherche aboutit, le premier item correspondant est marqué dans la barre latérale du Projet. Vous pouvez ensuite parcourir tous les items correspondants retournés en cliquant sur les boutons **Trouver suivant** et **Trouver précédant** dans le dialogue Recherche.

Réinitialiser les projets

Si vous effectuez une modification dans un dossier externe, cette modification ne sera pas réfléchie dans la Fenêtre du Projet tant que le projet est réinitialisé.

Ressources globales dans le menu contextuel

Lorsque vous cliquez avec la touche de droite dans un dossier contenu dans la fenêtre Projet, dans le menu contextuel qui apparaît, vous pouvez sélectionner l'item de menu **Ajouter Ressource globale** pour ajouter une ressource globale. La commande de menu elle-même apparaît dans le dialogue « Choose Global Resource », qui liste les ressources globales file-type et folder-type dans le fichier XML des Ressources globales actives actuelles. Choisir la ressource globale exigée et elle sera ajoutée au dossier de projet sélectionné.

Fournisseurs de projets et de contrôle de source

Si vous prévoyez d'ajouter un projet XMLSpy à un archivage de contrôle de source, veuillez vous assurer que le positionnement des fichiers de projet dans la structure de système de fichier hiérarchique est une structure qui vous permet d'ajouter des fichiers uniquement depuis les niveaux inférieurs (le répertoire de racine constituant le sommet de l'arborescence du répertoire).

En d'autres termes, le répertoire où le **fichier de projet** est situé, représente essentiellement le **répertoire racine** du projet dans l'archivage de contrôle de source. Les fichiers ajoutés depuis les niveaux supérieurs (le répertoire de racine du projet) seront ajoutés au projet XMLSpy, mais leur emplacement dans l'archivage peut être inattendu, s'ils peuvent y être placés.

Par exemple, selon la structure de répertoire affichée ci-dessus, si un fichier de projet est enregistré dans le `Folder3` puis placé sous contrôle de source :

- Fichiers ajoutés au Folder1 ne peut pas être placé sous le contrôle source,
- Les fichiers ajoutés au Folder2 sont ajoutés au répertoire racine de l'archivage au lieu du dossier de projet mais se trouvent encore sous contrôle de source,
- Les fichiers ajoutés au Folder3 et au Folder4 fonctionnent comme prévu et sont placés sous contrôle de source.

© 2016-2022 Altova GmbH

28.3.1 Nouveau projet



La commande **Nouveau projet** crée un **nouveau** projet dans XMLSpy. Si vous travaillez actuellement avec un autre projet, une invite s'affichera vous demandant si vous souhaitez fermer tous les documents faisant partie du projet actuel. Le nom du projet est assigné quand vous enregistrez le projet en tant que `.spp` file.

28.3.2 Ouvrir le projet



La commande **Ouvrir le projet...** ouvre un projet existant dans XMLSpy. Si vous travaillez actuellement avec un autre projet, le projet précédent est tout d'abord fermé.

28.3.3 Recharger le projet



La commande **Recharger le projet** recharge le projet actuel depuis le disque. Si vous travaillez dans un environnement à utilisateurs multiples, il peut parfois devenir nécessaire de recharger le projet depuis le disque parce que d'autres utilisateurs peuvent avoir effectué des changements au projet.

Veuillez noter : Les fichiers de projet (fichiers `.spp`) sont en fait des documents XML que vous pouvez éditer comme tout fichier XML habituel.

28.3.4 Fermer projet

La commande **Fermer projet** ferme le projet actuel. Si le projet a été modifié, vous serez invité à enregistrer le projet d'abord. Lorsqu'un projet est modifié d'une manière ou d'une autre, un astérisque sera ajouté au nom du projet dans la fenêtre Projet.

28.3.5 Enregistrer projet, Enregistrer projet sous



La commande **Enregistrer projet** enregistre le projet actuel. Vous pouvez aussi enregistrer un projet en activant la fenêtre de projet et en cliquant sur l'icône.

La commande **Enregistrer projet sous** enregistre le projet actuel avec un nouveau nom que vous pouvez saisir lorsque vous y êtes invité.

28.3.6 Contrôle de source

Votre application Altova prend en charge Microsoft SourceSafe et d'autres référentiels compatibles. Une liste des systèmes pris en charge est indiquée dans la section Systèmes de contrôle de source pris en charge. Cette section décrit les commandes dans le sous-menu **Projet | Contrôle de source** qui sont utilisés pour fonctionner avec le système de contrôle de source depuis votre application Altova.

Aperçu de la fonction Contrôle de source

Le mécanisme pour placer les fichiers dans un projet d'application sous contrôle de source est le suivant :

1. Dans XMLSpy, un dossier de projet d'application contenant les fichiers à placer sous contrôle de source est créé. Généralement, le dossier de projet d'application correspondra à un dossier local dans lequel se trouvent les fichiers de projet. Le chemin menant vers le dossier local est référé en tant que le chemin local.
2. Dans la base de données du système de contrôle de source (également appelé contrôle de source ou référentiel), un dossier est créé qui contiendra les fichiers à placer sous contrôle de source.
3. Les fichiers de projet d'application sont ajoutés au contrôle de source avec la commande **Projet | Contrôle de source | Ajouter au Contrôle de source**.
4. Les actions de contrôle de source, comme l'archivage, l'extraction et la suppression de fichiers du contrôle de source, peuvent être effectuées en utilisant les commandes dans le sous-menu **Projet | Contrôle de source**. Les commandes dans ce sous-menu sont recensées dans les sous-sections de cette section.

Note : si vous souhaitez modifier le fournisseur actuel du contrôle de source, vous avez deux possibilités : (i) par le biais des options du contrôle de source (**Outils | Options | Contrôle de source**), ou (ii) dans le dialogue Changer Contrôle de source (**Projet | Contrôle de source | Changer Contrôle de source**).

Note : veuillez noter qu'un projet de contrôle de source n'est pas identique à un projet d'application. Les projets de contrôle de source sont dépendants des répertoires alors que les projets XMLSpy sont des constructions logiques sans dépendance directe à un répertoire.

Pour plus d'informations, voir la section Contrôle de source.

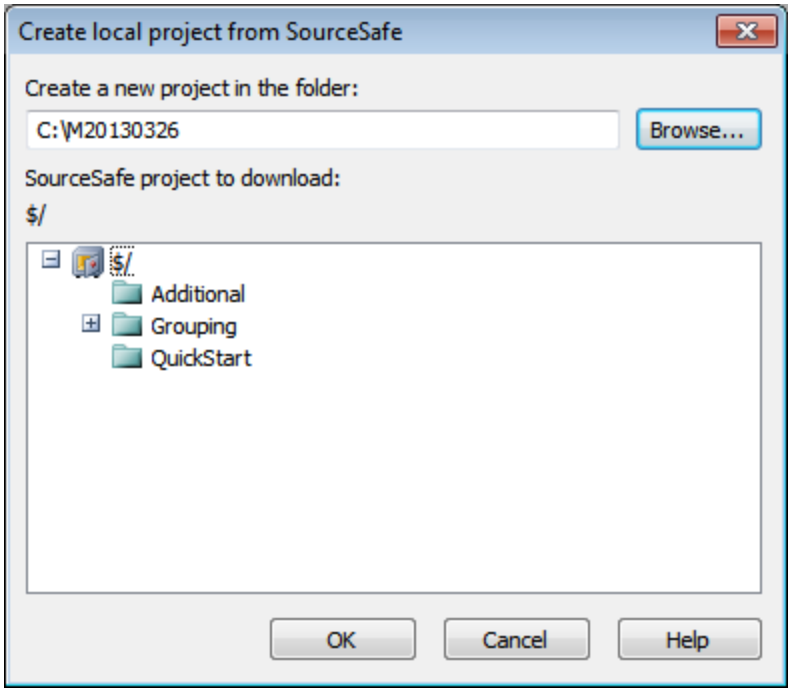
© 2016-2022 Altova GmbH

28.3.6.1 Ouvrir depuis le Contrôle de source

La commande **Ouvrir depuis le Contrôle de source** crée un nouveau projet d'application depuis un projet se trouvant sous contrôle de source.

Créer le nouveau projet comme suit :

1. Selon le système de contrôle de source utilisé, il peut être nécessaire, avant de créer un nouveau projet depuis le contrôle de source, de vous assurer qu'aucun fichier n'est extrait provenant du projet.
2. Il n'est pas nécessaire d'ouvrir un projet dans l'application, mais il est possible de le faire.
3. Choisir la commande Projet | Contrôle de source | Ouvrir depuis le Contrôle de source.
4. Le système de contrôle de source actuellement configuré ouvrira sa vérification et les dialogues de connexion. Établissez la connexion au référentiel que vous souhaitez, c'est-à-dire au dossier lié dans l'archivage qui correspond au dossier local.
5. Dans le dialogue qui s'ouvre (*capture d'écran ci-dessous*), recherchez le dossier local dans lequel les contenus du dossier lié dans le référentiel (auquel vous venez de vous connecter) doivent être copiés. Dans la capture d'écran ci-dessous, le dossier lié est appelé `MyProject` et est représenté par le signe `$` ; le dossier local est `C:\M20130326`.



6. Cliquez sur **OK**. Les contenus du dossier lié (`MyProject`) seront copiés dans le dossier local `C:\M20130326`., et un dialogue s'ouvre pour vous inviter à choisir le fichier de projet (fichier `.spp`) qui doit être créé en tant que le nouveau projet.
7. Choisir le fichier `.spp` qui aura été copié dans le dossier local. Dans notre exemple, tceci sera situé dans `MyProject.spp` dans le dossier `C:\M20130326`. Un nouveau projet appelé `MyProject` sera créé dans l'application et sera affiché dans la fenêtre Projet. Les fichiers de projet se trouveront dans le dossier `C:\M20130326`.

Symboles de contrôle de source

Les fichiers et dossier de projet affichent certains symboles dont la signification est expliquée ci-dessous.

	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

28.3.6.2 Activer le Contrôle de source

La commande **Activer le contrôle de source** vous permet d'activer ou de désactiver le contrôle de source pour un projet d'application. Choisir cette option sur n'importe quel fichier ou dossier pour activer/désactiver le contrôle de source pour l'ensemble du projet. Une fois que le contrôle de source est activé, les statuts d'archivage/d'extraction des différents fichiers sont extraits et affichés dans la fenêtre Projet.

	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non-disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

28.3.6.3 Obtenir la dernière version

La commande **Obtenir la dernière version** (dans le menu **Projet | Contrôle de source**) extrait et place la dernière version du contrôle de source du/des fichier/s sélectionné/s dans le répertoire de travail. Les fichiers sont extraits en lecture seule et ne sont pas extraits. Cette commande fonctionne comme la commande Obtenir, mais n'affiche pas le dialogue Obtenir. Si les fichiers sélectionnés sont actuellement extraits, alors l'action prise dépendra de comment votre système de contrôle de source gère une telle situation. Généralement, le système de contrôle de source vous demandera si vous souhaitez remplacer, fusionner ou conserver le fichier extrait tel quel.

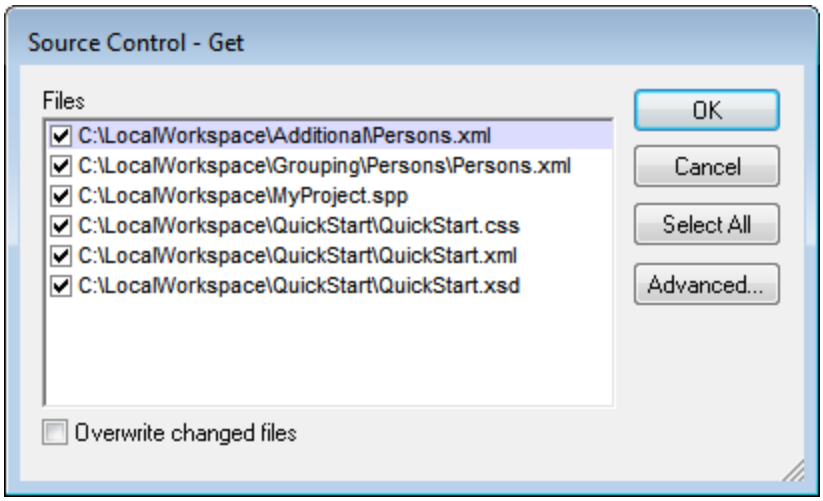
Note : Cette commande est récursive lorsqu'elle est effectuée sur un dossier, c'est-à-dire qu'elle touche tous les fichiers se trouvant en-dessous du fichier actuel dans la hiérarchie du dossier.

28.3.6.4 Obtenir, Obtenir les dossiers

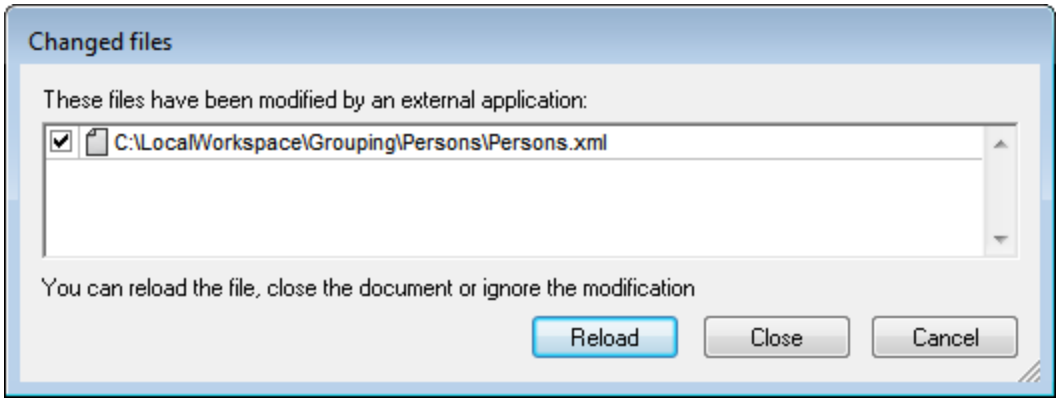
La commande **Get** (dans le menu **Projet | Contrôle de source**) extrait les fichiers du référentiel comme fichiers en lecture seule. (Pour éditer un fichier, il faut l'extraire.) Le dialogue Obtenir recense les fichiers dans l'objet (projet ou dossier) dans lequel la commande **Obtenir** a été exécutée (*voir capture d'écran ci-dessous*). Vous pouvez choisir les fichiers à extraire en les archivant.

© 2016-2022 Altova GmbH

Note : La commande **Obtenir dossier** vous permet de choisir des sous-dossiers individuels dans l'archivage si cela est permis par votre système de contrôle de source.

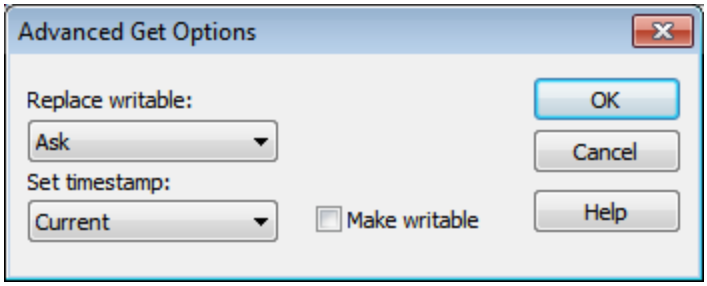


Vous pouvez décider d'écraser des fichiers extraits ayant été modifiés en cochant cette option située en bas du dialogue Obtenir. Cliquez sur **OK** pour écraser les fichiers. Si l'un des fichiers écrasés est actuellement ouvert, un dialogue apparaît (*capture d'écran ci-dessous*) demandant si vous souhaitez recharger le/s fichier/s (touche **Recharger**), fermer le/s fichier/s (**Fermer**) ou conserver le mode actuel du fichier (**Annuler**).



Options Obtenir avancées

Le dialogue Options Obtenir avancées (*capture d'écran ci-dessous*) est accessible depuis la touche **Avancé** dans le dialogue Obtenir (*voir première capture d'écran dans cette section*).



Ici, vous pouvez définir les options pour (i) remplacer les fichiers accessibles en écriture qui sont extraits, (ii) l'horodatage, et (iii) si vous souhaitez changer la propriété lecture seule du fichier extrait pour le rendre accessible en écriture.

28.3.6.5 Extraire, Archiver

Une fois qu'un fichier de projet a été placé sous la source de contrôle, il peut être extrait ou archivé en sélectionnant le fichier (dans la fenêtre de projet) et en cliquant la commande respective dans le menu **Projet | Contrôle de source** : **Extraire** et **Archiver**.

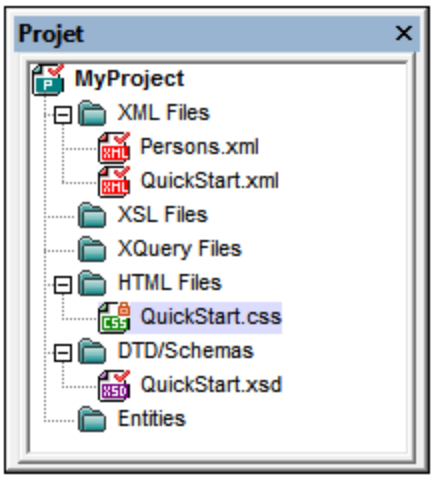
Lorsqu'un fichier est extrait, une copie provenant de l'archivage est placée dans le dossier local. Un fichier qui est extrait peut être édité. Si un fichier se trouvant sous contrôle de source n'est pas extrait, il ne pourra pas être édité. Une fois qu'un fichier a été édité, les changements peuvent être enregistrés dans l'archivage en validant le fichier. Même si vous n'enregistrez pas le fichier, en le validant, vous pourrez enregistrer les changements dans l'archivage. Un coche ou un cadenas dans l'icône vous indique que le fichier est extrait.

Les fichiers et dossier de projet affichent certains symboles dont la signification est expliquée ci-dessous.

	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

En choisissant le projet ou un dossier dans le projet, vous sélectionnez tous les fichiers se trouvant dans l'objet sélectionné. Pour sélectionner de multiples objets (fichiers et dossiers), appuyez sur la touche Ctrl tout en cliquant sur les objets. La capture d'écran ci-dessous montre un projet qui a été extrait. Le fichier `QuickStart.css` a été archivé par la suite.

© 2016-2022 Altova GmbH



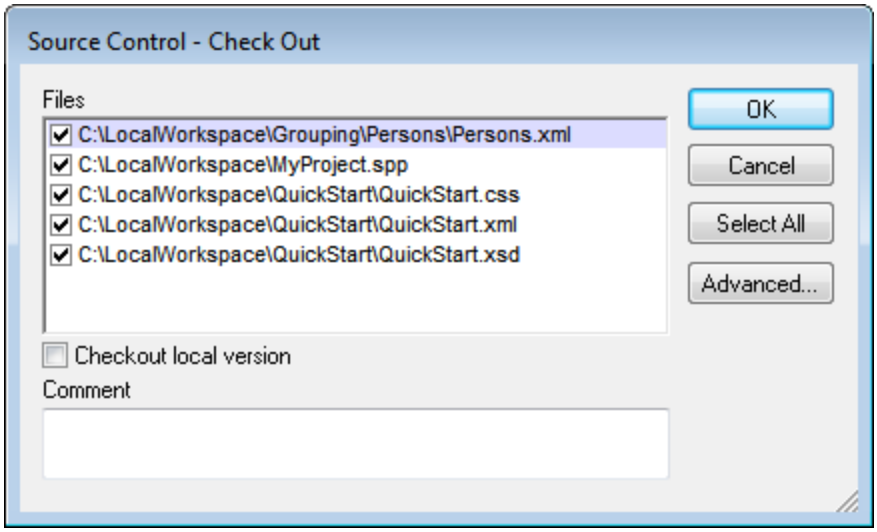
Enregistrer et rejeter des modifications d'édition

Veuillez noter que, lorsque vous archivez un fichier, vous pouvez choisir de laisser ce fichier extrait. De fait, les modifications d'édition sont enregistrées dans l'archivage alors que le fichier reste extrait, ce qui peut être utile lorsque vous souhaitez enregistrer des modifications d'édition de temps en temps puis poursuivre l'édition.

Si vous avez extrait un fichier puis que vous l'avez édité, mais que vous souhaitez annuler ces changements, vous pouvez retourner à la version de document enregistrée dans l'archivage en choisissant la commande **Projet | Contrôle de source | Annuler extraction**.

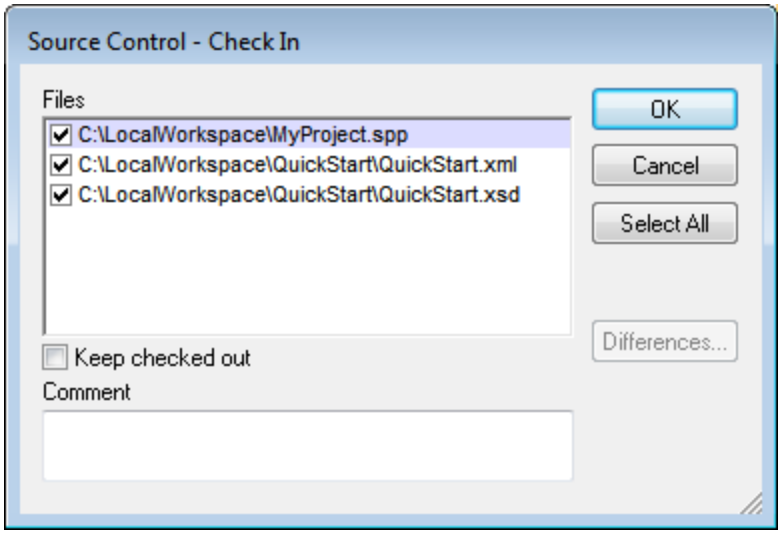
Extraction

Le dialogue Extraction (*capture d'écran ci-dessous*) vous permet de : (i) sélectionner les fichiers à extraire, et (ii) de choisir si vous souhaitez extraire la version de l'archivage ou la version locale.



Archivage

Le dialogue Archiver (*capture d'écran ci-dessous*) vous permet de : (i) sélectionner les fichiers à archiver, et (ii) si vous le souhaitez, de garder le fichier extrait.



Note : dans les deux dialogues (Extraire et Archiver), plusieurs fichiers apparaissent si l'objet sélectionné (projet ou dossiers de projet) contient plusieurs fichiers.

28.3.6.6 Annuler Extraction

Si vous avez extrait un fichier puis que vous l'avez édité, mais que vous souhaitez annuler ces changements, vous pouvez retourner à la version de document enregistrée dans l'archivage en choisissant la commande **Projet | Contrôle de source | Annuler extraction**.

Les fichiers et dossier de projet affichent certains symboles dont la signification est expliquée ci-dessous.

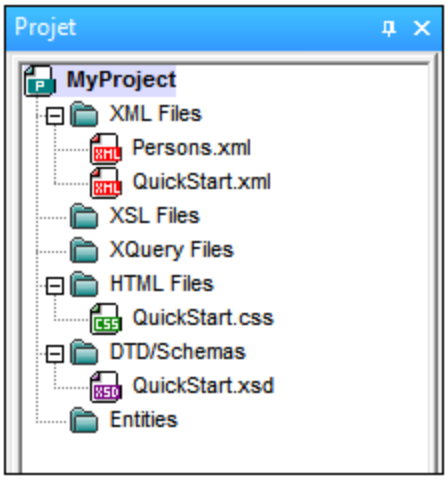
	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

28.3.6.7 Ajouter au Contrôle de source

Une fois qu'un projet a été ajouté au contrôle de source, vous pouvez ajouter des fichiers individuellement ou en groupe dans le contrôle de source. Choisir les fichiers dans la fenêtre Projet et cliquer sur la commande **Projet | Contrôle de source | Ajouter au Contrôle de source**. Pour sélectionner des fichiers multiples, maintenez appuyée la touche **Ctrl** en cliquant sur les fichiers que vous souhaitez ajouter. Exécuter la commande dans un

© 2016-2022 Altova GmbH

dossier de projet (vert) (*voir la capture d'écran ci-dessous*) ajoute tous les fichiers au dossier et ses sous-dossiers au contrôle de source.



Lorsque des fichiers sont ajoutés au contrôle de source, la hiérarchie de dossier local est répliquée dans le référentiel (pas la hiérarchie du dossier du projet). Donc, si un fichier est dans un sous-fichier X dans les abysses du dossier local, alors le dossier parent du fichier et tous les autres dossiers ancêtre sont automatiquement créés dans le référentiel.

Lorsque le premier fichier d'un projet est ajouté au contrôle de source, les liaisons correctes sont créées dans le référentiel et le fichier de projet (fichier `.spp`) est ajouté automatiquement. Pour plus de détails, voir la section Ajouter au Contrôle de source.

Symboles de contrôle de source

Les fichiers et dossier de projet affichent certains symboles dont la signification est expliquée ci-dessous.

	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non-disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

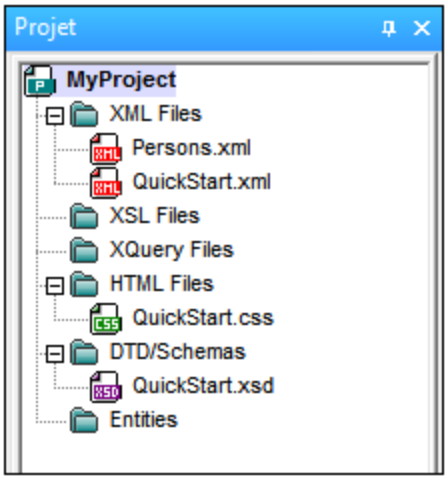
28.3.6.8 Supprimer du Contrôle de source

Pour supprimer un fichier du contrôle de source, choisir le fichier et cliquer sur la commande **Projet | Contrôle de source | Supprimer du Contrôle de source**. Vous pouvez aussi supprimer : (i) les fichiers dans un dossier de projet en exécutant la commande sur le dossier, (ii) de multiples fichiers que vous sélectionnez en maintenant la touche **Ctrl** appuyée et (iii) le projet entier en exécutant la commande sur le projet.

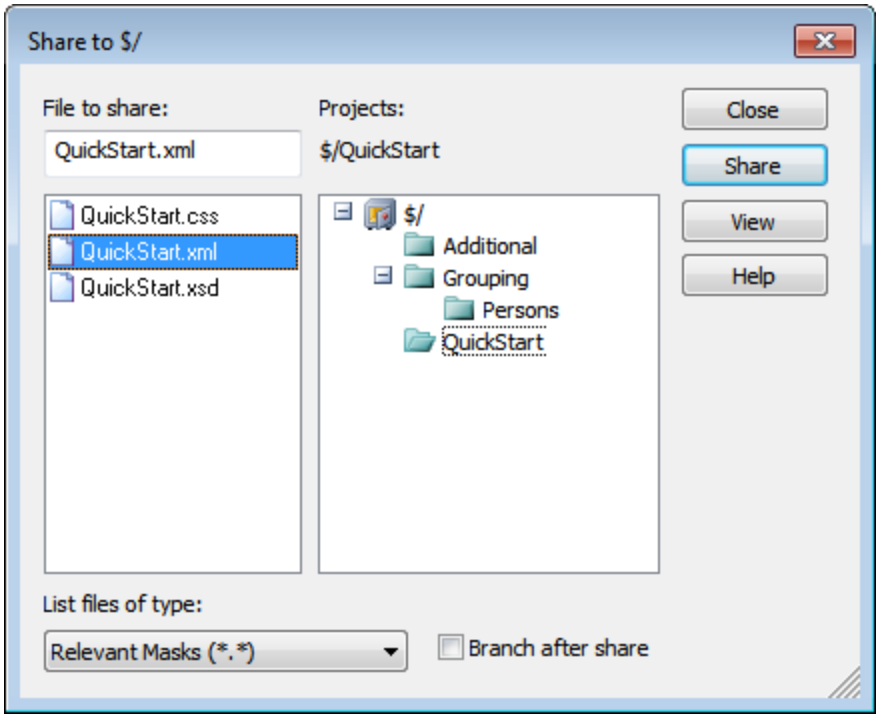
28.3.6.9 Partager depuis le Contrôle de source

La commande **Partager depuis le Contrôle de source** est prise en charge lorsque le système de contrôle de source utilisé prend en charge les partages. Vous pouvez partager un fichier de manière à ce qu'il soit disponible dans plusieurs endroits locaux. Une modification dans un de ces fichiers locaux sera reflétée dans toutes les autres versions « partagées ».

Dans la fenêtre d'application du projet, sélectionnez d'abord le projet (*en surbrillance dans la capture d'écran ci-dessous*). Ensuite, cliquez sur **Partager depuis le Contrôle de source**.



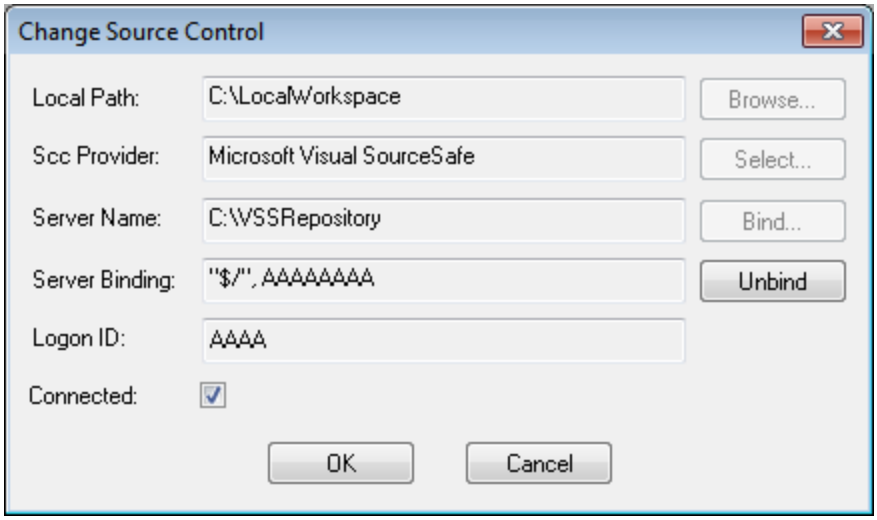
Le dialogue Partager avec [Dossier] (*capture d'écran ci-dessous*) s'ouvre.



© 2016-2022 Altova GmbH

Pour sélectionner les fichiers à partager, choisissez tout d'abord dans l'arborescence de projet située dans le panneau de droite le dossier dans lequel se trouvent les fichiers. Ces fichiers sont affichés dans le volet de gauche. Choisir le fichier que vous souhaitez partager (plusieurs fichiers en appuyant sur la touche **Ctrl** et en cliquant sur les fichiers que vous souhaitez partager). Les fichiers sélectionnés seront affichés dans le champ de saisie *Fichiers à partager* (*en haut à gauche*). Cliquez sur **Partager**, puis sur **Fermer** pour copier les fichiers sélectionnés dans le dossier de partage local.

Le dossier de partage est noté dans le nom du dialogue Partager avec [Dossier]. Dans la capture d'écran ci-dessus, il s'agit du dossier local (puisque le signe `$` est le dossier dans le référentiel dans lequel le dossier local est lié). Vous pouvez voir et configurer le dossier de partage dans le dialogue Changer Contrôle de source (*capture d'écran ci-dessous*, **Changer Contrôle de source**) en modifiant le chemin local et la liaison au serveur.

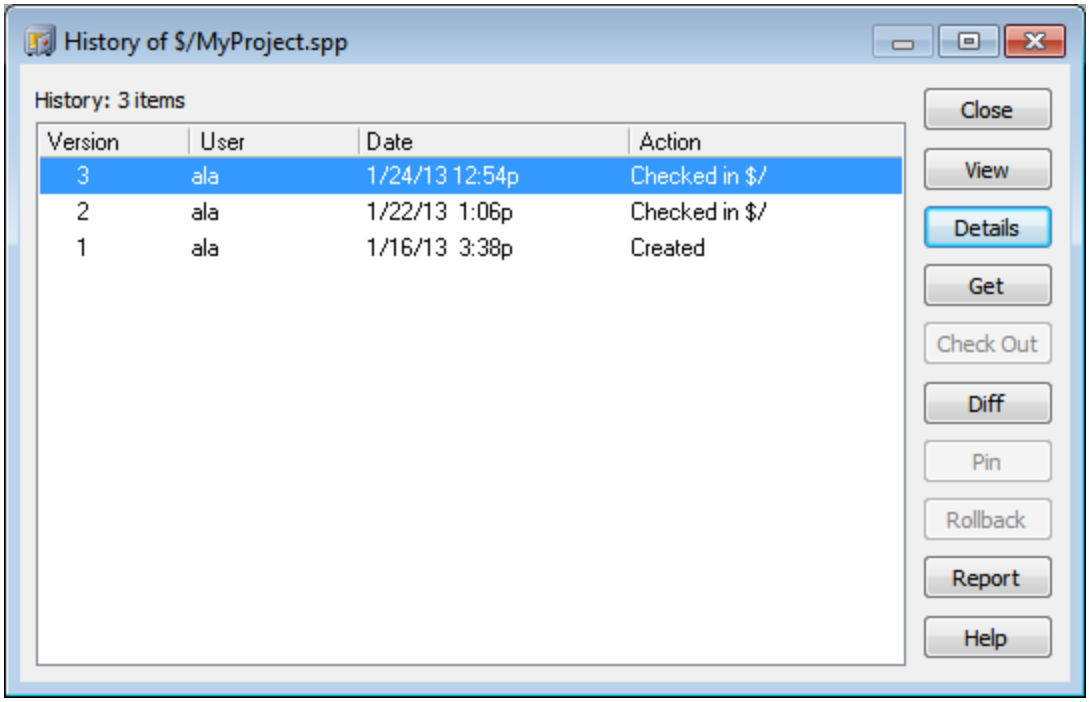


Pour plus de détails concernant le partage de votre système de contrôle de source, voir la documentation de l'utilisateur du système de contrôle de source.

28.3.6.10 Afficher Historique

La commande **Afficher historique** active la fonction Afficher historique du système de contrôle de source. Elle affiche l'historique du fichier sélectionné dans la fenêtre de projet. Sélectionnez le titre de projet dont vous voulez afficher l'historique (fichier `.spp`). Vous pouvez consulter des informations concernant les versions précédentes d'un fichier et les différences ou d'extraire les versions précédentes d'un fichier.

La capture d'écran ci-dessous montre le dialogue de l'historique du système de contrôle de source Visual SourceSafe. Elle recense les différentes versions du fichier `MyProject.spp`.



Ce dialogue Historique fournit différentes manières de comparer et obtenir des versions spécifiques du fichier en question. Double-cliquez sur une entrée dans la liste ouvre la boîte de dialogue Détails de l'Historique pour ce fichier. Les touches du dialogue ont les fonctions suivantes :

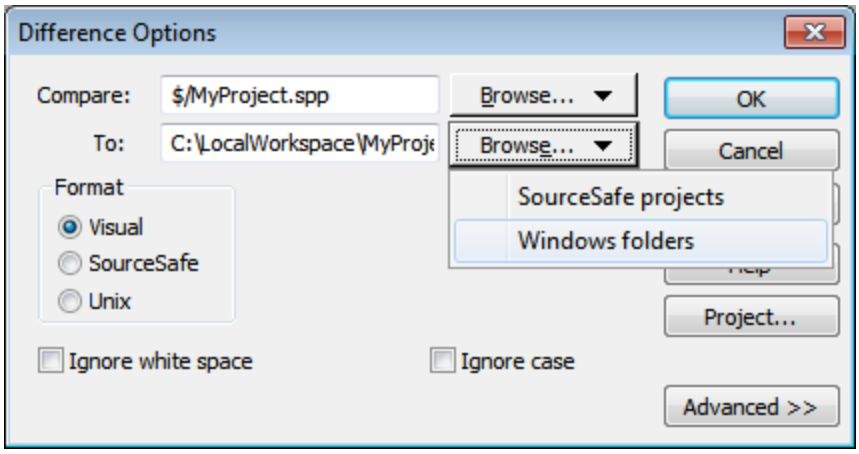
- *Fermer :* Ferme ce dialogue.
- *Mode :* Ouvre un autre dialogue dans lequel vous pouvez sélectionner le type d'affichage avec lequel vous souhaitez voir le fichier.
- *Détails :* Ouvre un dialogue dans lequel vous pouvez voir les propriétés du fichier actif actuellement.
- *Get :* Extrait une version de fichier précédente et la place dans le répertoire de travail.
- *Extraire :* Vous permet d'extraire une version précédente du fichier.
- *Diff :* Ouvre le dialogue options de différenciation pour une différenciation entre deux versions de fichier. Utiliser **CTRL+Clic** pour marquer les deux versions de fichier dans cette fenêtre, puis cliquez sur Diff pour consulter les différences entre elles.
- *Pin :* Épingle et détache une version du fichier, vous permettant de définir la version du fichier spécifique à utiliser lors de la différenciation de deux fichiers.
- *Annuler :* Annule et retourne à la version sélectionnée du fichier.
- *Rapport :* Génère un rapport d'historique que vous pouvez envoyer à l'imprimante, au fichier ou au presse-papiers.
- *Aide :* Ouvre l'aide en ligne du plugin du fournisseur du contrôle de source.

28.3.6.11 Afficher les différences

La commande **Afficher les différences** est activée lorsqu'un fichier est sélectionné dans la fenêtre Projet. Pour sélectionner le fichier de projet (fichier `.spp`), choisissez le titre de projet dans la fenêtre Projet. La commande **Afficher les différences** lance l'outil de différenciation du système de contrôle de source de manière à ce que les différences entre les fichiers peuvent être vérifiées directement depuis l'application Altova.

© 2016-2022 Altova GmbH

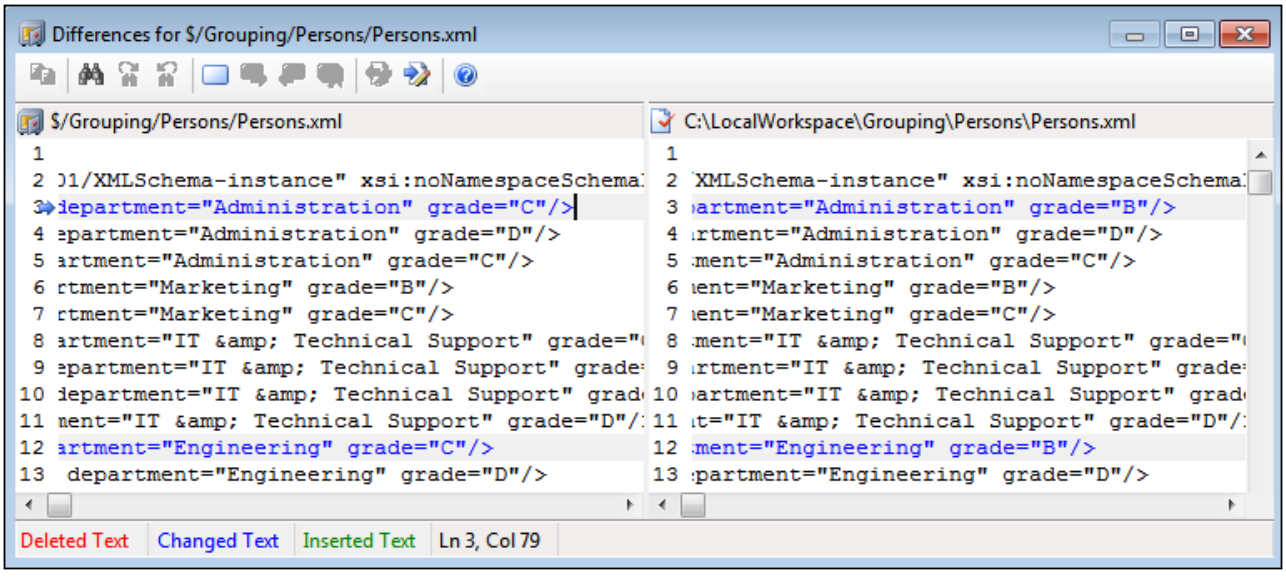
La capture d'écran ci-dessous montre les outils de différenciation du système de contrôle de source Visual SourceSafe.



Le référentiel et les versions locales sont affichés par défaut dans les champs de saisie *Comparer* et *vers* respectivement. Vous pouvez chercher d'autres fichiers de la manière suivante :

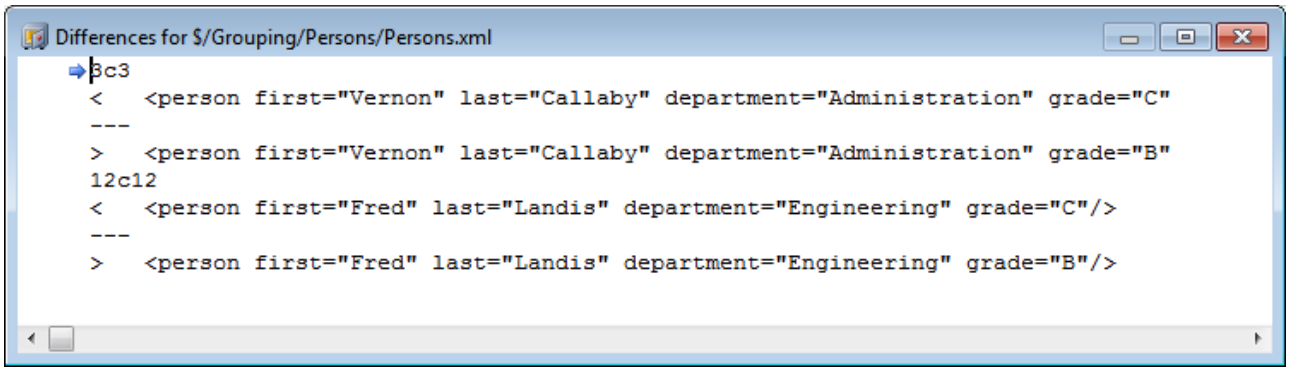
1. À partir de la liste déroulante dans le bouton **Chercher**, choisir des projets SourceSafe (pour chercher des fichiers d'archivage) ou des dossiers Windows (pour chercher des dossiers locaux).
2. Cherchez les fichiers que vous souhaitez, puis sélectionnez-les.

Sélectionnez les options que vous souhaitez et cliquez sur **OK** pour exécuter la vérification. Les résultats de différenciation sont affichés dans une fenêtre séparée. Les captures d'écran ci-dessous montrent les résultats d'une vérification en deux formats.



La capture d'écran ci-dessus montre les résultats de différenciation Visual SourceSafe dans un format Visual (*voir dialogue d'Options ci-dessus*), alors que la capture d'écran ci-dessous montre les résultats dans un format

Unix. Les deux présentent deux différences, chacune d'entre elle est une modification du degré (grade) de C à B.



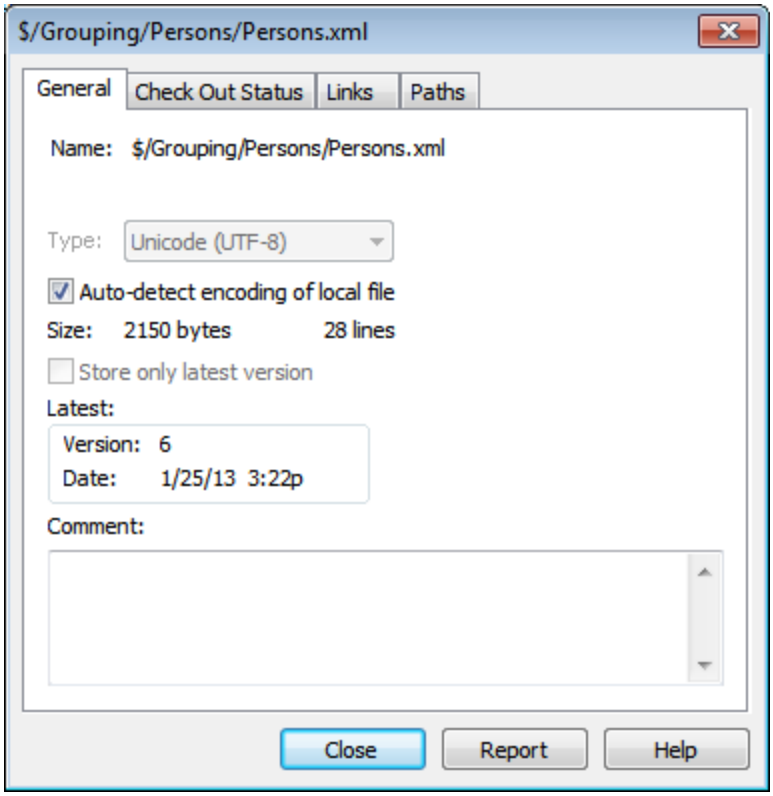
Pour une description détaillée de la manière dont votre système de contrôle de source gère la différenciation, voir la documentation de l'utilisateur du système de contrôle de source.

28.3.6.12 Afficher Propriétés

La commande **Afficher Propriétés** affiche les propriétés du fichier sélectionné actuellement (*voir la capture d'écran ci-dessous*). Les propriétés affichées dépendent du système de contrôle de source que vous utilisez. La capture d'écran ci-dessous montre les propriétés de Visual SourceSafe lorsqu'il est le système de contrôle de source actif.

Veuillez noter que cette commande peut uniquement être utilisée sur des fichiers individuels.

© 2016-2022 Altova GmbH



Pour plus de détails, voir la documentation de l'utilisateur du système de contrôle de source.

28.3.6.13 Réinitialiser statut

La commande **Réinitialiser statut** réinitialise le statut de tous les fichiers de projet, indépendamment de leur statut actuel.

28.3.6.14 Gestionnaire de Contrôle de source

La commande **Gestionnaire de Contrôle de source**lance votre logiciel de contrôle de source avec son interface d'utilisateur native.

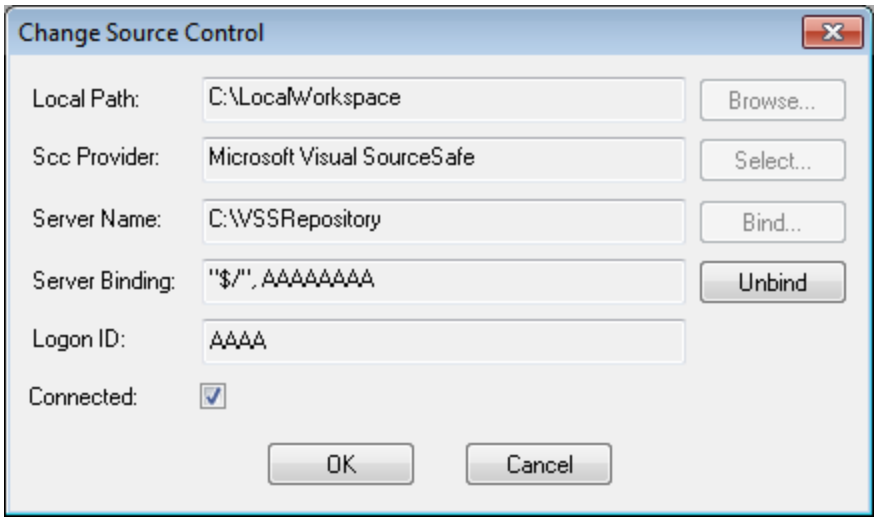
28.3.6.15 Modifier Contrôle de source

La liaison actuelle est celle que le projet d'application actif utilisera pour se connecter à la base de donnée de contrôle de source, c'est pourquoi la liaison actuelle doit être correcte. Cela signifie que le fichier de projet d'application (fichier.spp) doit se trouver dans le chemin de dossier local et le dossier lié dans l'archivage doit

être la base de données où ces fichiers du projet sont stockés. Généralement, le dossier lié et sa sous-structure correspondront au dossier de l'espace de travail local et à sa sous-structure.

Dans le dialogue Changer le Contrôle de source (*voir la capture d'écran ci-dessous*), vous pouvez changer le système de contrôle de source (*Fournisseur SCC*), le dossier local (*Chemin local*) et la liaison du référentiel (*Nom de serveur* et *Liaison de serveur*).

Ce n'est qu'après avoir délié la liaison actuelle que les configurations peuvent être modifiées. Délier la liaison actuelle avec la touche **Délier**. Toutes les configurations sont maintenant éditables.



Modifier les paramètres du contrôle de source comme suit :

1. Utiliser la touche **Chercher** pour chercher dans le dossier local et sur la touche **Choisir** pour choisir parmi les systèmes de contrôle de source installés.
2. Ensuite, vous pouvez lier le dossier local à une base de données d'archivage. Cliquer sur la touche **Lier** pour ce faire. Le dialogue de connexion de votre système de contrôle de source s'ouvre.
3. Si vous avez saisi une *ID de login*, elle sera transmise au système de contrôle de source ; sinon, vous devrez saisir vos détails de connexion dans le dialogue de connexion.
4. Choisir la base de données dans le référentiel que vous souhaitez lier à ce dossier local. Cette configuration peut être étalée sur plusieurs dialogues.
5. Une fois que la configuration a été créée, cliquez sur **OK** dans le dialogue Changer Contrôle de source.

28.3.7 Ajouter des fichiers au projet



La commande **Projet** | **Ajouter fichier au projet** ajoute le fichier actif au projet actuel. Utiliser cette commande pour ajouter des fichiers à n'importe quel dossier dans votre projet. Vous pouvez soit sélectionner un seul fichier soit un groupe de fichiers (avec **Ctrl+ clic**) dans le dialogue Ouvrir. Si vous ajoutez des fichiers au projet, ils seront distribués parmi les dossiers respectifs basés sur les Extensions de type de fichier définies dans le dialogue Propriétés de projet.

© 2016-2022 Altova GmbH

28.3.8 Ajouter une Ressource globale au Projet

La commande **Projet** | **Ajouter Ressource globale au projet** permet d'ouvrir le dialogue Choisir Ressource globale, dans lequel vous pouvez choisir une ressource globale d'un fichier ou d'un type de dossier à ajouter au projet. Une fois qu'une ressource globale de type fichier a été sélectionnée, le fichier est ajouté au dossier approprié basé sur l'extension de type de fichier définie dans le dialogue Propriétés de projet. Si une ressource globale de type fichier a été sélectionnée, ce dossier sera ouvert dans un dialogue à fichiers ouverts et vous serez invité à choisir un fichier ; le fichier sélectionné sera ajouté au dossier approprié basé sur l'extension de type de fichier définie dans le dialogue Propriétés de projet. Pour une description de ressources globales, voir la section Ressources globales de cette documentation.

28.3.9 Ajouter une URL au projet



La commande **Projet** | **Ajouter URL au projet** ajoute une URL au projet actuel. Les URL contenues dans un projet entraînent l'inclusion de l'objet cible de l'URL dans le projet. Quelle que soit l'opération batch effectuée sur une URL ou dans un dossier qui contient un objet à URL, XMLSpy extrait le document depuis l'URL et exécute l'opération nécessaire.

28.3.10 Ajouter fichier actif au projet



La commande **Projet** | **Ajouter fichier actif au projet** ajoute le fichier actif au projet actuel. Si vous venez d'ouvrir un fichier depuis votre disque dur ou par le biais d'une URL, vous pouvez ajouter le fichier au projet actuel avec cette commande.

28.3.11 Ajouter des fichiers actifs et liés au projet



La commande **Projet** | **Ajouter fichiers actifs et liés à un projet** ajoute le document XML actuellement actif et tous les fichiers liés au projet. Lorsque vous travaillez sur un document XML basé sur un DTD ou un Schéma, cette commande ajoute non seulement le document XML mais aussi tous les fichiers liés (par exemple, le DTD et toutes les entités parsées externes auxquelles le DTD se réfère) au projet actuel.

Veuillez noter : Les fichiers référencés par les instructions de traitement (tels que les fichiers XSLT), ne sont pas considérés être des fichiers liés.

28.3.12 Ajouter un dossier de projet au projet



La commande **Projet** | **Ajouter Dossier de projet au projet** ajoute un nouveau dossier au projet actuel. Utiliser cette commande pour ajouter un nouveau dossier au projet actuel ou un sous-dossier à un dossier de projet. Vous pouvez aussi accéder à cette commande depuis le menu contextuel en cliquant avec la touche de droite sur un dossier dans la fenêtre de projet.

Note : Un dossier de projet peut être glissé et déposé dans un autre dossier de projet ou dans n'importe quel endroit dans le projet. Un dossier peut aussi être glissé depuis Windows (File) Explorer et déposé dans tout autre dossier de projet.

Note : Les dossiers de projet sont verts, alors que les dossiers externes sont jaunes.

28.3.13 Ajouter un dossier externe au projet

La commande **Projet** | **Ajouter dossier externe au projet** ajoute un nouveau dossier externe au projet actuel. Utiliser cette commande pour ajouter un dossier local ou de réseau au projet actuel. Vous pouvez aussi accéder à cette commande depuis le menu contextuel si vous cliquez avec la touche de droite un dossier dans la fenêtre de projet.

Note : les dossiers de externes sont jaunes, alors que les dossiers de projet sont verts.

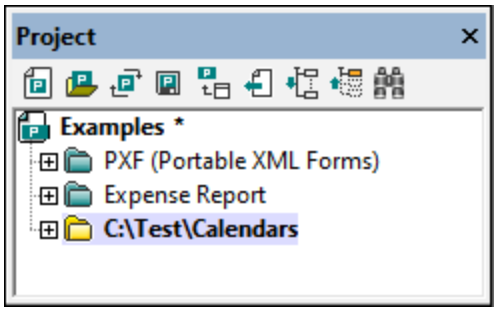
Note : les fichiers contenus dans des dossiers externes ne peuvent pas être placés sous le contrôle de source.

Ajouter des dossiers externes aux projets

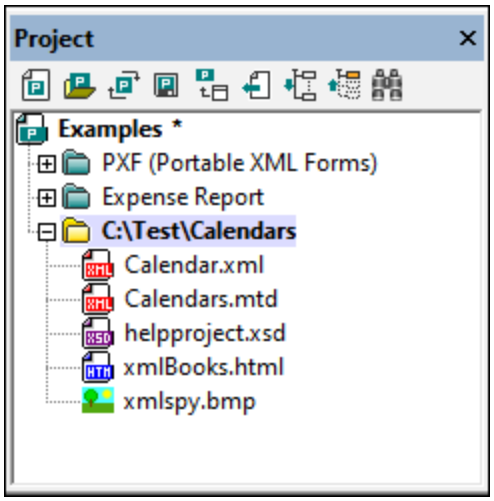
Pour ajouter un dossier externe au projet :

1. Choisir l'option de menu **Projet | Ajouter dossier externe au projet**.
2. Choisir le dossier que vous souhaitez inclure et cliquez sur **OK** pour confirmer. Le dossier sélectionné apparaîtra alors dans la fenêtre Projet.

© 2016-2022 Altova GmbH



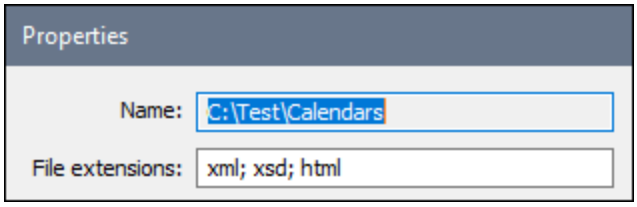
3. Cliquer sur l'icône plus pour consulter le contenu du dossier.



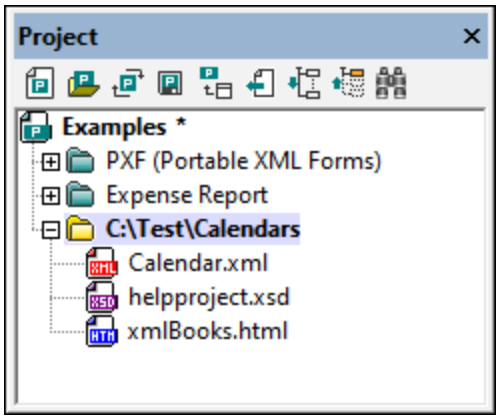
Filtrer les contenus des dossiers

Pour filtrer les contenus du dossier :

1. Cliquez avec la touche de droite sur le dossier externe que vous avez ajouté et sélectionnez les **Propriétés**. Cela permet d'ouvrir le dialogue Propriétés.



2. Cliquez dans le champ *extensions de fichier* et saisissez les extensions de fichier des types de fichier que vous voulez voir, séparant les types de fichiers par un point virgule (*voir la capture d'écran ci-dessus*).
3. Cliquer sur **OK** pour confirmer.



La fenêtre de projet ne montre plus que les types de fichiers sélectionnés.

Valider des dossiers externes

Pour valider et vérifier la bonne-formation d'un dossier externe :

1. Choisir les types de fichier que vous souhaitez voir ou vérifier depuis le dossier externe.
2. Cliquer sur le dossier et cliquer sur la commande de menu **XML | Vérifier la bonne formation** ou **Valider XML**(raccourcis **F7** or **F8**, respectivement). Tous les fichiers visibles sous le dossier sont vérifiés. Si un fichier est mal formé ou invalide, ce fichier sera ouvert dans la fenêtre principale et vous pourrez l'éditer.
3. Corriger l'erreur et exécuter le processus de validation une fois de plus et revérifier.

Mettre à jour un dossier de projet

Vous pouvez ajouter ou supprimer des fichiers dans le répertoire local ou de réseau à tout moment. Pour mettre à jour l'aperçu du dossier, cliquez avec la touche de droite sur le dossier externe et choisissez l'option de menu popup **Réinitialiser**.

Supprimer les dossiers externes et les fichiers contenus

Choisir un dossier externe et appuyer sur la touche **Supprimer** pour supprimer le dossier depuis la fenêtre Projet. Il est également possible de cliquer avec la touche de droite sur le dossier externe et choisir la commande **Supprimer**. Chacune de ces actions ne supprime que le dossier externe de la fenêtre de projet. Le dossier externe n'est pas supprimé du disque dur ou du réseau.

Pour supprimer un fichier dans un dossier externe, vous devez le supprimer physiquement du disque dur ou du réseau. Pour voir le changement dans le projet, réinitialiser les contenus du dossier externe (cliquer avec la touche de droite sur le dossier externe et choisir **Réinitialiser**).

Note : un dossier externe peut être glissé et déposé dans un dossier de projet ou dans tout autre emplacement dans le projet (mais pas dans un autre dossier externe). Aussi, un dossier externe peut être glissé de Windows (File) Explorer et déposé dans tout emplacement de la fenêtre de projet excepté dans un autre dossier externe.

© 2016-2022 Altova GmbH

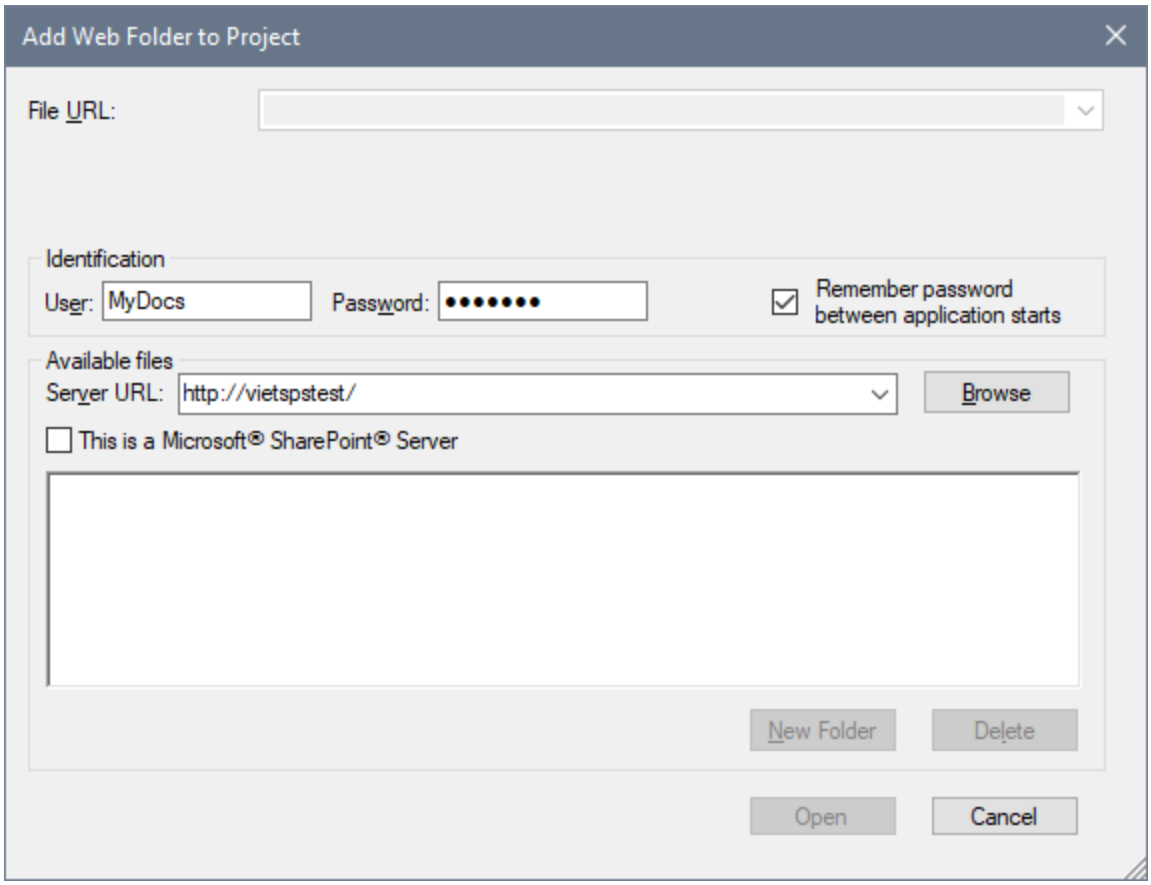
28.3.14 Ajouter un dossier web externe au projet

Cette commande ajoute un nouveau dossier Web externe au projet actuel. Vous pouvez aussi accéder à cette commande depuis le menu contextuel si vous cliquez avec la touche de droite un dossier dans la fenêtre de projet. Notez que les fichiers contenus dans des dossiers externes ne peuvent pas être placés sous le contrôle de source.

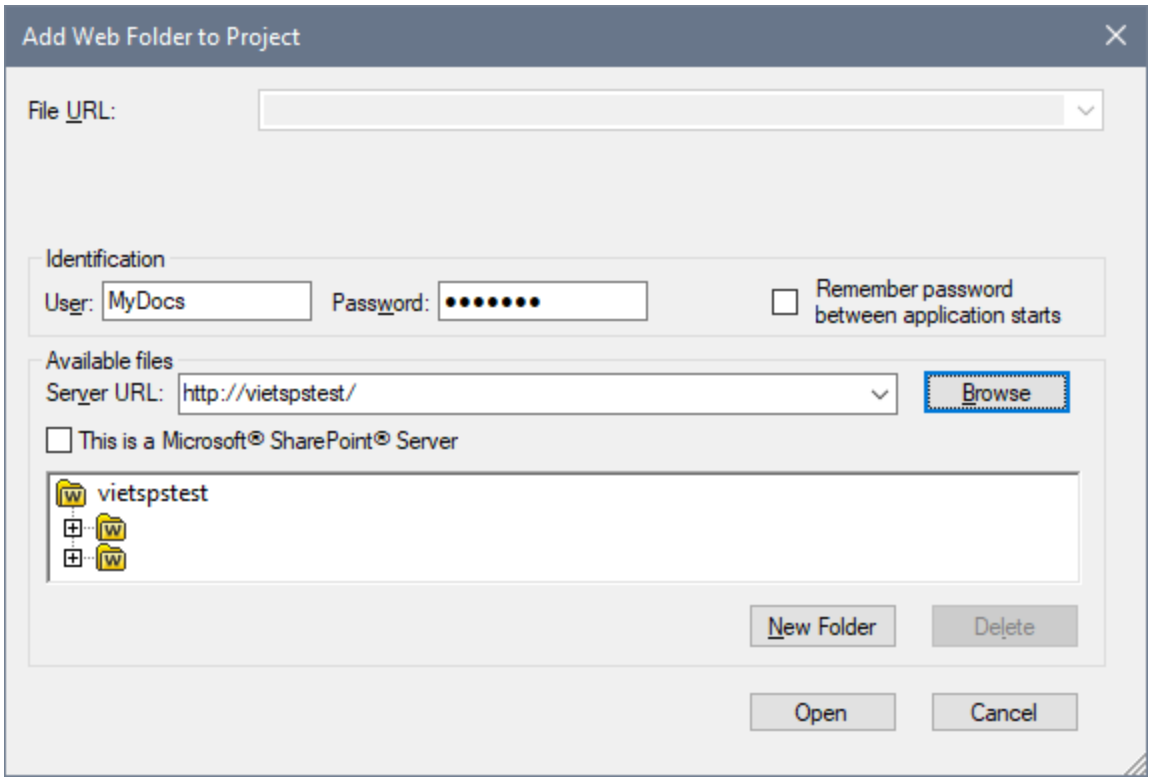
Ajouter un dossier Web externe au projet

Pour ajouter un dossier Web externe au projet, procédez comme suit :

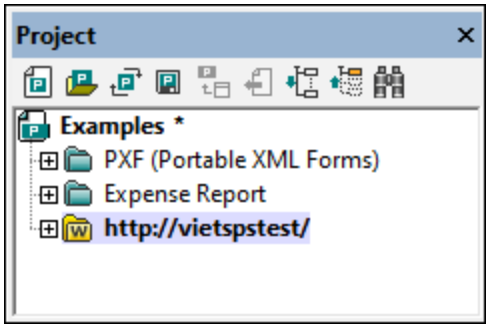
1. Choisir l'option de menu option **Projet | Ajouter dossier Web externe au projet**. Cela ouvre le dialogue Paramètres de requête SOAP (*capture d'écran ci-dessous*).



2. Cliquer dans le champ URL de Serveur et saisir l'URL de l'URL de serveur. Si le serveur est un Microsoft® SharePoint® Server, cochez cette option. Voir la section *Dossier sur un Microsoft® SharePoint® Server* ci-dessous pour plus d'informations concernant le travail avec des fichiers sur ce type de serveur.
3. Si le serveur est protégé par un mot de passe, veuillez saisir votre ID d'utilisateur et votre mot de passe dans les champs *Utilisateur* et *Mot de passe*.
4. Cliquez sur **Parcourir** pour vous connecter au serveur et consultez les dossiers disponibles.

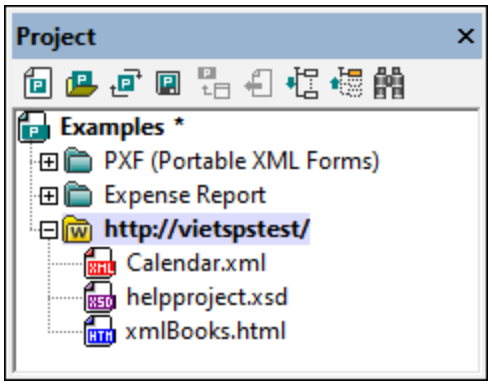


5. Cliquez sur le dossier que vous souhaitez ajouter au mode de projet. Le bouton **Ouvrir** ne s'activera qu'après avoir effectué cette étape. L'URL du dossier apparaît maintenant dans le champ URL de fichier.
6. Cliquez sur **Ouvrir** pour ajouter le dossier au projet.



7. Cliquez sur l'icône plus pour consulter le contenu du dossier.

© 2016-2022 Altova GmbH



Filtrer les contenus de dossier

Pour filtrer les contenus d'un dossier, cliquez avec la touche de droite sur le dossier et sélectionnez **Propriétés** depuis le menu contextuel. Dans le dialogue Propriétés qui s'ouvre, cliquez dans le champ *Extensions de fichier* et saisir les extensions de fichier des types de fichier que vous souhaitez voir (par exemple, des fichiers XML et XSD). Séparez chaque type de fichier avec un point-virgule (par exemple : `xml; xsd; sps`). La fenêtre de projet n'affichera que ce dossier avec des fichiers ayant une extension spécifiée.

Valider et vérifier la bonne formation d'un dossier

Pour vérifier les fichiers dans un dossier pour la bonne formation ou pour les valider, sélectionnez le dossier, puis cliquez sur l'icône de la commande de menu **XML | Vérifier la bonne formation** ou **XML | Valider XML** (raccourci **F7** ou **F8**, respectivement). Tous les fichiers visibles dans le dossier sont vérifiés. Si un fichier est mal formé ou invalide, ce fichier sera ouvert dans la fenêtre Principale et vous pourrez l'éditer. Corrigez l'erreur et redémarrez le processus pour revérifier le reste du dossier. Veuillez noter que vous pouvez sélectionner des fichiers discontinus dans le dossier en maintenant appuyée la touche **Ctrl** et en cliquant individuellement sur les fichiers. Seuls ces fichiers seront contrôlés lorsque vous appuierez sur **F7** ou **F8**.

Mettre à jour les contenus du dossier de projet

Des fichiers peuvent être ajoutés ou supprimés du dossier Web à tout moment. Pour mettre à jour l'aperçu du dossier, cliquez avec la touche de droite sur le dossier externe et choisissez l'option du menu contextuel **Réinitialiser**.

Supprimer les dossiers et les fichiers

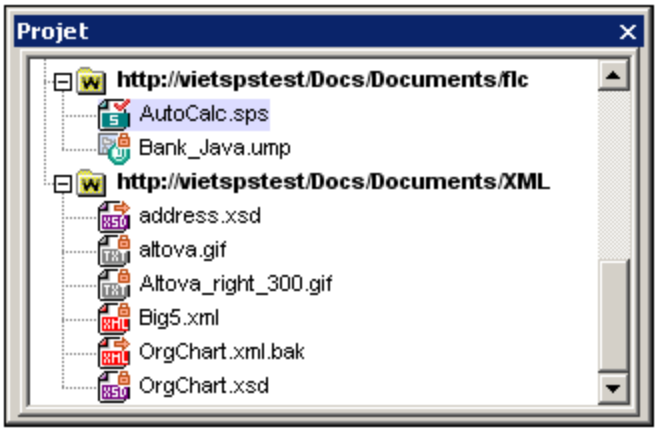
Puisque c'est le dossier Web qui a été ajouté au projet, seul le dossier Web peut être supprimé du projet (et non pas les fichiers s'y trouvant). Vous pouvez supprimer un dossier Web d'un projet, soit (i) en cliquant avec la touche de droite sur le dossier et en choisissant **Supprimer**, ou (ii) en choisissant le dossier et en appuyant sur la touche **Supprimer**. Cela supprime le dossier uniquement de l'aperçu Projet ; mais ne supprime rien du tout dans le serveur web.

Note : Double-cliquer sur un seul fichier et appuyer sur la touche **Supprimer** ne supprime pas un fichier depuis la fenêtre de projet. Vous devez le supprimer physiquement sur le serveur puis rafraîchir les contenus du dossier externe.

Les dossiers sur un Microsoft® SharePoint® Server

Lorsqu'un dossier dans un Microsoft® SharePoint® Server a été ajouté à un projet, les fichiers contenus dans le dossier peuvent être extraits et archivés par le biais de commandes dans le menu contextuel de la liste des fichiers dans la fenêtre Projet (*voir la capture d'écran ci-dessous*). Pour accéder ces commandes, cliquez avec la touche de droite sur le fichier sur lequel vous voulez travailler et sélectionnez la commande souhaitée (**Check Out**, **Check In**, **Undo Check Out**).

L'ID de l'utilisateur et le mot de passe peuvent être enregistrés dans les propriétés des dossiers individuels dans le projet, vous permettant ainsi de sauter le processus de vérification à chaque fois que le serveur est accédé.



Dans la fenêtre de projet (*capture d'écran ci-dessous*), les icônes de fichier ont des symboles qui indiquent le statut d'archivage/de récupération des fichiers. Les différents icônes de fichier sont affichés ci-dessous :

	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non disponible pour une extraction.
	Extrait localement. Peut être édité et archivé.

Veuillez noter les points suivants :

- Une fois avoir extrait un fichier, vous pouvez l'éditer dans votre application Altova et l'enregistrer avec **Fichier | Enregistrer (Ctrl+S)**.
- Vous pouvez archiver le fichier édité par le biais du menu contextuel dans la fenêtre Projet (*voir la capture d'écran ci-dessus*), ou par le biais du menu contextuel qui s'ouvre lorsque vous cliquez avec la touche de droite dans l'onglet Fichier de la fenêtre principale de votre application (voir la *capture d'écran ci-dessous*).



- Lorsqu'un fichier est extrait par un autre utilisateur, il ne peut pas être extrait.
- Lorsqu'un fichier est extrait localement par vous, vous pouvez annuler l'extraction avec la commande Annuler l'extraction dans le menu contextuel. Cela a pour effet de retourner le fichier dans le serveur sans les modifications.

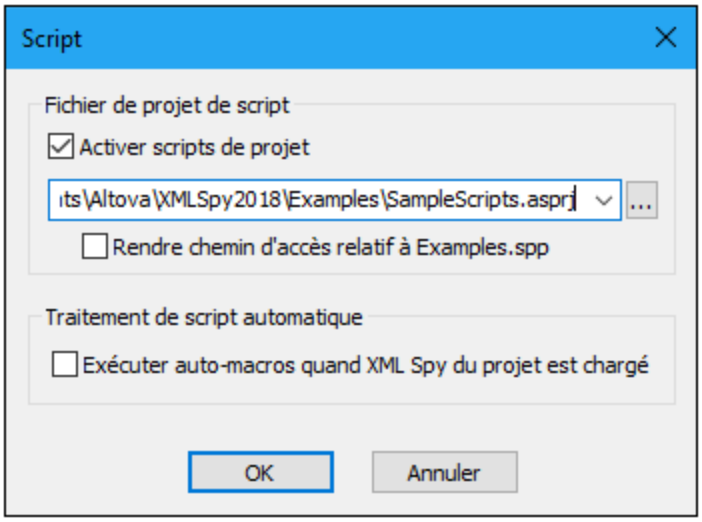
© 2016-2022 Altova GmbH

- Si vous extrayez un fichier dans une des applications Altova, vous ne pouvez pas l'extraire dans une autre application Altova. Le fichier est considéré être déjà extrait par vous. Les commandes disponibles dans ce cas pour les applications Altova prenant en charge Microsoft® SharePoint® Server seront : **Archiver** et **Annuler l'extraction**.

28.3.15 Paramètres du script

Un projet de script est attribué à un projet de XMLSpy comme suit :

1. Dans la GUI de XMLSpy, ouvrir le projet d'application exigé.
2. Choisir la commande de menu **Projet | Paramètres de script**. Le dialogue Script (*capture d'écran ci-dessous*) s'ouvre.



3. Cochez la case *Activer les scripts de projet* et sélectionnez le projet de script requis (fichier `.asprj`). Si vous souhaitez exécuter des auto-macros lorsque le projet de XMLSpy est chargé, cocher la case *Exécuter auto-macros*.
4. Cliquez sur **OK** pour finir.

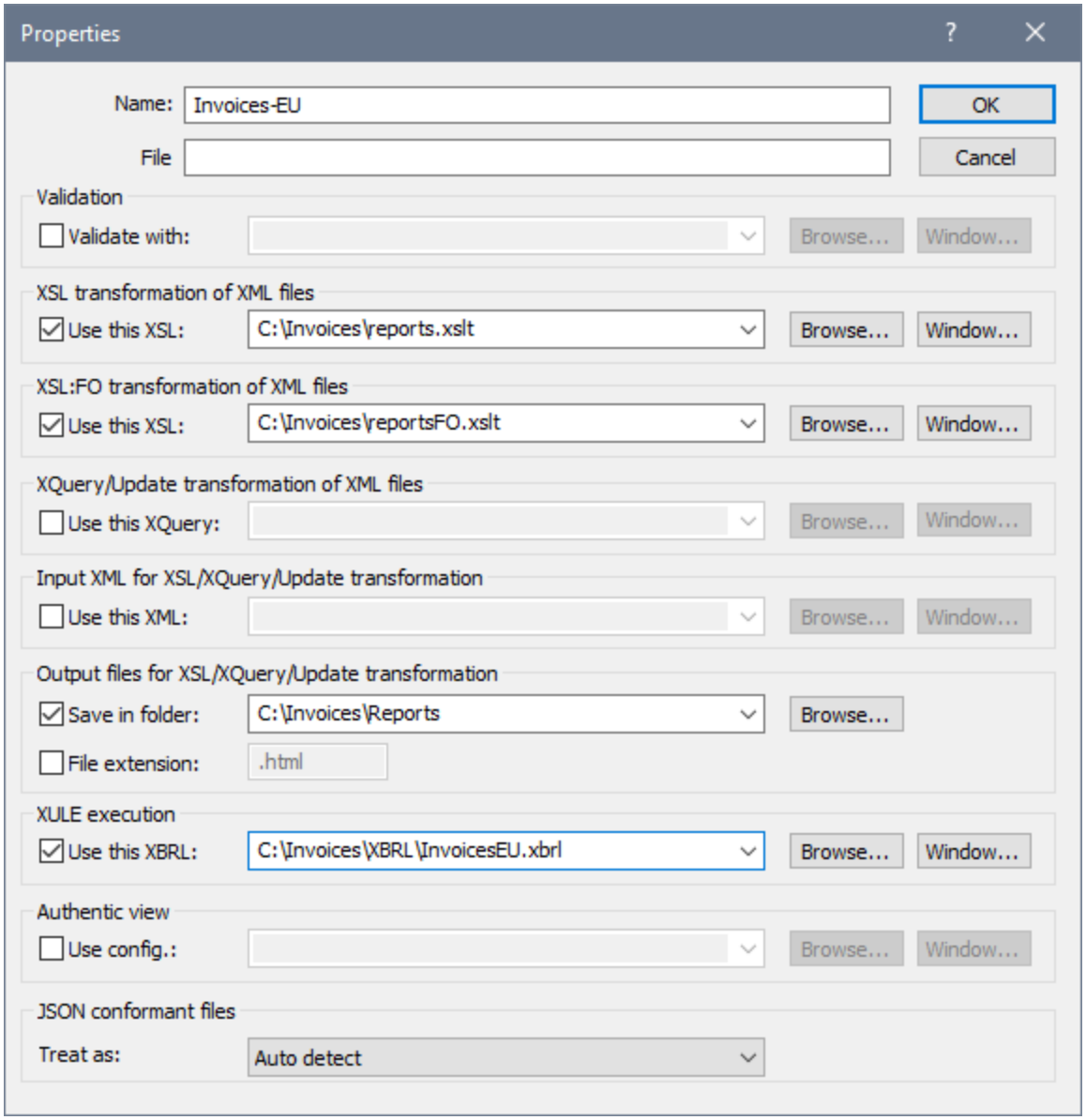
Note : Pour désactiver (donc, désattribuer) le projet de script d'un projet XMLSpy, décochez la case *Activer scripts de projet*.

28.3.16 Propriétés



La commande **Projet** | **Propriétés de projet** ouvre le dialogue Propriétés (*capture d'écran ci-dessous*) du projet actif. Si vous cliquez avec la touche de droite sur un dossier dans la fenêtre Projet (contrairement au dossier de projet) et que vous sélectionnez **Propriétés**, le dialogue Propriétés de ce dossier est ouvert. Les paramètres de dialogue sont décrits ci-dessous.

Note : Si votre fichier de projet se trouve sous contrôle de source, une invite apparaîtra vous demandant si vous souhaitez extraire le fichier de projet (`.spp`). Cliquez sur **OK** si vous voulez éditer les paramètres et pouvoir les enregistrer.



Paramètres

Extensions de fichier
Le paramètre *Extensions de fichier* est activé pour les dossiers individuels, et non pas pour le dossier de projet. Lorsqu'un fichier est ajouté à un projet, il sera ajouté au dossier pour lequel son extension de fichier a été définie. Par exemple, si un fichier nommé `MyReport.xml` est ajouté au projet. Si les extensions de fichier `.xml` ont été définies dans le dossier `Invoices-EU` (tel qu'affiché dans la capture d'écran ci-dessus), alors

© 2016-2022 Altova GmbH

`MyReport.xml` sera ajouté au dossier `Invoices-EU`. S'il y a plus d'un dossier auquel vous souhaitez ajouter des fichiers XML, vous devriez ajouter des fichiers XML individuels directement au dossier (au lieu du projet).

ID d'utilisateur et mot de passe pour les dossiers externes
Dans les dossiers externes (y compris les dossiers Web externes), vous pouvez enregistrer l'ID d'utilisateur et le mot de passe qui peuvent être nécessaires pour accéder au serveur.

Validation
Le DTD, Schéma XML, ou le schéma JSON schéma JSON qui doit être utilisé pour valider les fichiers dans le dossier actuel (ou tout le projet si les propriétés sont celles du projet).

Transformation XSL des fichiers XML
La feuille de style XSLT à utiliser pour la transformation XSLT des fichiers XML dans le dossier.

Transformation XSL-FO des fichiers XML
La feuille de style XSLT pour transformer des fichiers XML dans le dossier pour XSL-FO.

Transformation XQuery/Update des fichiers XML
Le fichier XQuery ou XQuery Update à utiliser pour des exécutions XQuery ou des exécutions XQuery Update de fichiers XML dans le dossier.

XML d'entrée pour la transformation XSL/XQuery/Update de fichiers XML
Le fichier XML à utiliser en tant qu'entrée pour les transformations XSLT ou les exécutions XQuery/XQuery Update avec les fichiers XSLT, XQuery ou XQuery Update respectifs dans le dossier.

Fichiers de sortie pour la transformation XSL/XQuery/Update
Le répertoire de destination des transformations, et, en option, l'extension de fichier du document de résultat.

Exécution XULE
Le fichier d'instance XBRL à traiter avec le document XULE qui est actif dans la fenêtre d'application XMLSpy.

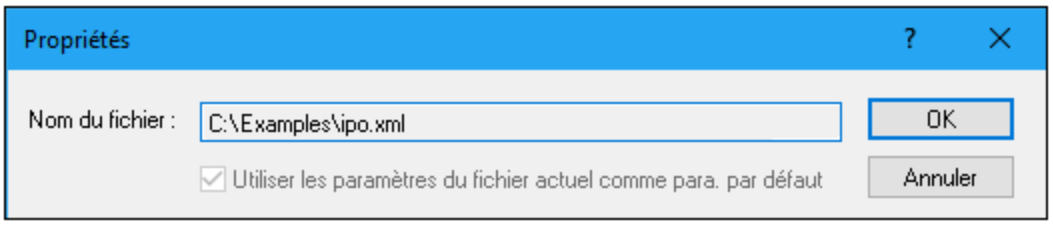
Mode Authentic
Utiliser config spécifie le StyleVision Power Stylesheet (fichier SPS) à utiliser pour l'affichage Authentic View des fichiers XML dans le dossier. Veuillez noter que le fichier XML doit être valide par rapport au même schéma utilisé pour le SPS.

Fichiers conformes à JSON
Cette propriété spécifie si un dossier de projet contient des fichiers de schéma JSON ou des fichiers d'instance JSON. Il peut être très utile d'aider à identifier les fichiers de schéma JSON si les fichiers ne sont pas clairement identifiés en tant que de fichier de schéma JSON par le mot-clé **`$schema`** et les fichiers se référencent l'un l'autre. Vous pouvez les définir comme *Instance JSON, Schéma JSON,* ou *Auto detect*. Le paramètre par défaut de *Auto-detect* amènerait XMLSpy à vérifier la structure et le contenu des fichiers JSON pour déterminer son type.

Notes à propos des propriétés du projet

Notes concernant la précédence :

- Lorsque des validations ou des transformations XSLT/XQuery sont effectuées par le biais des menus contextuels du dossier de projet, la validation ou les fichiers de transformation spécifiés dans ce dialogue prévalent sur toute attribution dans le fichier XML. De même, les paramètres spécifiés pour les dossiers de projet individuels prévalent sur les paramètres définis pour les dossiers d'ancêtre.
- Si un fichier est présent dans plusieurs dossiers du projet et que plusieurs fichiers de validation ou de transformation lui ont été attribués dans des dossiers différents, vous pouvez définir quelle attribution utiliser lorsque le fichier est traité en dehors du projet. Spécifiez cela comme suit : Trouver dans le dossier du projet le fichier dont vous souhaitez utiliser l'attribution. Cliquer avec la touche de droite dans ce dossier de projet et choisir **Propriétés**. Dans le dialogue qui apparaît (*capture d'écran ci-dessous*), choisir *Utiliser les paramètres dans le dossier actuel en défaut*. (Le dossier actuel est le dossier de projet dans lequel se trouve le fichier.) Si l'option est désactivée, cela signifie que les paramètres du dossier actuel sont déjà sélectionnés en tant que les paramètres par défaut à utiliser. Si vous choisissez une instance de fichier qui se trouve dans un dossier de projet qui n'est pas l'instance par défaut, l'option sera activée et vous pouvez faire passer les paramètres par défaut pour en faire les paramètres de ce dossier. Veuillez noter que si le fichier a une attribution locale (c'est-à-dire une attribution dans le fichier lui-même), alors l'attribution locale sera utilisée et les paramètres de dossier par défaut seront ignorés.



28.3.17 Projets les plus récents

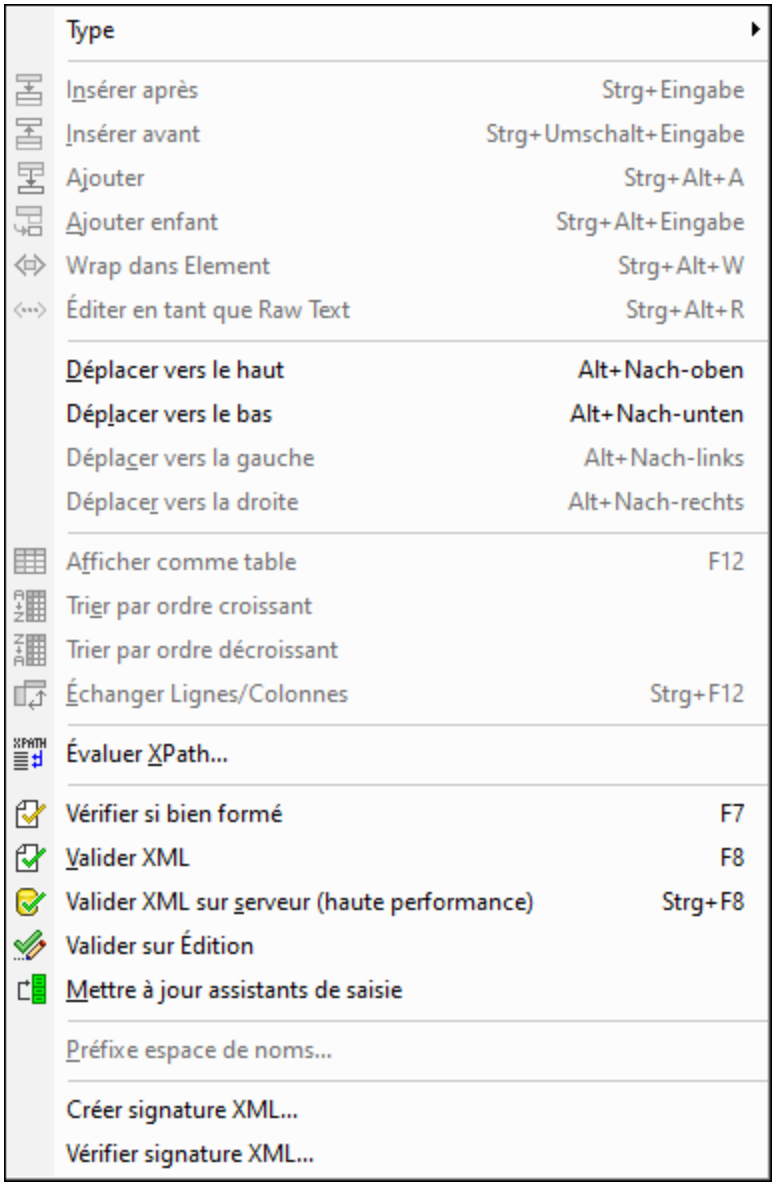
Cette commande affiche le nom de fichier et le chemin pour les neuf derniers projets utilisés, permettant un accès rapide à ces fichiers.

Notez aussi que XMLSpy peut automatiquement ouvrir le dernier projet que vous avez utilisé dès que vous lancez XMLSpy. (Section **Outils | Options | Fichier**, Projet | Ouvre le dernier projet au démarrage du programme).

© 2016-2022 Altova GmbH

28.4 Menu XML

Le menu **XML** contient des commandes qui sont communément requises lorsqu'on travaille avec des documents XML. Vous trouverez des commandes pour insérer ou apposer des éléments, modifier la hiérarchie de l'élément, définir un préfixe d'espace de nom, ainsi qu'évaluer XPaths dans le contexte de documents XML individuels.



Parmi les tâches XML les plus fréquemment utilisées font partie les vérifications pour la bonne formation des documents et la validité des documents XML. Les commandes pour ces tâches sont dans le menu.

28.4.1 Type

La commande **Type** a un sous-menu qui contient une liste de types de nœud XML. Vous pouvez changer le type du nœud que vous venez de choisir dans le Mode Grille en un nouveau type depuis cette liste. Les types de nœud dans le sous-menu sont activés uniquement dans le Mode Grille, et seuls les types de nœud sont activés sur lesquels le nœud actuellement choisi peut être changé.

Voir le sujet XML | Contenu du document pour plus d'information.

28.4.2 Insérer avant/après

Les commandes **Insérer avant** et **Insérer après** sont activées lorsqu'un nœud est sélectionné dans le Mode Grille. Elles ajoutent un nœud d'élément au même niveau, respectivement, après et avant l'item sélectionné. Changez le nom du nouvel élément ajouté en double-cliquant sur le nom de la cellule et pour éditer le nom. Changez le type du nœud en cliquant sur l'icône de l'élément (à gauche de son nom), puis en sélectionnant le type de nœud souhaité, ou en utilisant la commande **XML | Type**.

Voir le sujet XML | Structure du document pour plus d'informations.

28.4.3 Apposer, Ajouter enfant

Les commandes **Apposer** et **Ajouter enfant** sont activées lorsqu'un nœud est sélectionné dans le Mode Grille. La commande **Apposer** ajoute un nouveau nœud d'élément en tant que dernier sibling de l'item sélectionné. La commande **Ajouter enfant** appose un nouvel nœud d'élément en tant qu'enfant. Changez le nom du nouvel élément ajouté en double-cliquant sur le nom de la cellule et pour éditer le nom. Changez le type du nœud en cliquant sur l'icône de l'élément (à gauche de son nom), puis en sélectionnant le type de nœud souhaité, ou en utilisant la commande **XML | Type**.

Voir le sujet XML | Structure du document pour plus d'informations.

28.4.4 Wrapper dans array

La commande **Wrapper dans array** est activée lorsqu'un nœud est sélectionné dans le Mode Grille. Le nœud sélectionné reçoit un élément parent avec un nom par défaut. Changez le nom du nouvel élément parent ajouté en double-cliquant sur le nom de la cellule et pour éditer le nom.

Voir le sujet XML | Structure du document pour plus d'informations.

© 2016-2022 Altova GmbH

28.4.5 Éditer en tant que Texte brut

La commande **Éditer en tant que Texte brut** est activée lorsqu'un nœud est sélectionné dans le Mode Grille. Permet d'éditer le contenu du texte de l'item sélectionné en tant que texte brut. Ceci est utile si, par exemple, vous êtes en train d'éditer du contenu complexe tel que du code HTML.

Pour plus d'informations, voir la section XML dans le Mode Grid.

28.4.6 Déplacer vers le Haut/Bas/Gauche/Droite

Ces commandes **Déplacer** sont activés lorsqu'un nœud est sélectionné dans le Mode Grille. S'il est possible de déplacer le nœud vers le haut, le bas, la gauche ou la droite de son emplacement actuel dans la grille, la/les commande(s) correspondante(s) sera(-ont) activée(s). Choisir la commande respective pour effectuer le déplacement.

28.4.7 Afficher en tant que Table

La commande **Afficher en tant que Table** est activée lorsqu'un élément répétitif est sélectionné dans le Mode Grille. C'est une commande de bascule qui active l'afficahge du set d'éléments répétitifs entre le Mode Grille et l'Affichage Table. L'Affichage Table vous permet de voir des éléments répétitifs en tant que table dans lesquels les lignes représentent des occurrences tandis que les colonnes représentent les nœuds enfant.

Voir le sujet Affichage de la table pour plus d'informations.

28.4.8 Tri Montant/Descendant

Les commandes **Tri montant** et **Tri ascendant** sont activées dans l'Affichage Table lorsqu'une colonne ou une cellule est sélectionnée. Elles trient la colonne sélectionnée soit par ordre alphabétique ou numérique, dépendant du type de données de la colonne.

Voir le sujet Affichage Table pour plus d'informations.

28.4.9 Échanger Lignes/Colonnes

La commande **Échanger Lignes/Colonnes** est activée dans Afficahge Table lorsque la cellule supérieure gauche est sélectionnée (*marquée en rouge dans la capture d'écran ci-dessous*). La commande bascule des lignes vers les colonnes et vice versa.

expense-item (4)		type	expto	Date	expense
	1	Lodging	Sales	2003-01-01	122.11
	2	Lodging	Development	2003-01-02	122.12
	3	Lodging	Marketing	2003-01-02	299.45
	4	Entertainment	Development	2003-01-02	13.22

Voir le sujet Affichage Table pour plus d'informations.

28.4.10 Évaluer XPath



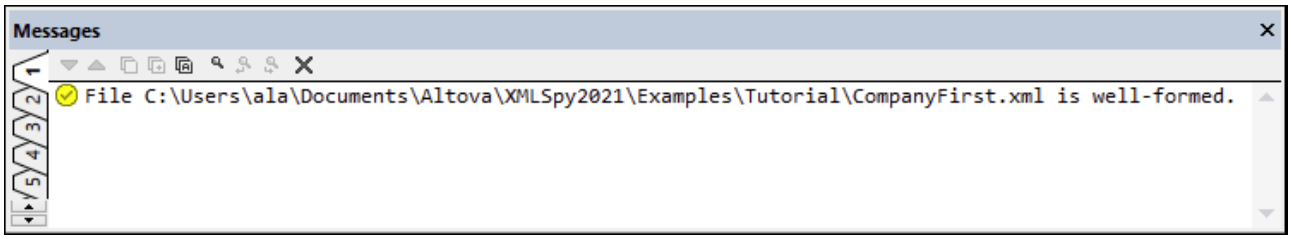
La commande **XML** | **Évaluer XPath** ouvre les Fenêtres de sortie si celles-ci ne sont pas ouvertes et active l'onglet XPath dans les Fenêtres de sortie. Dans l'onglet XPath, vous pouvez évaluer une expression XPath sur le document actif et voir si les résultats dans la Fenêtre de sortie.

28.4.11 Vérifier bonne formation

 **F7**

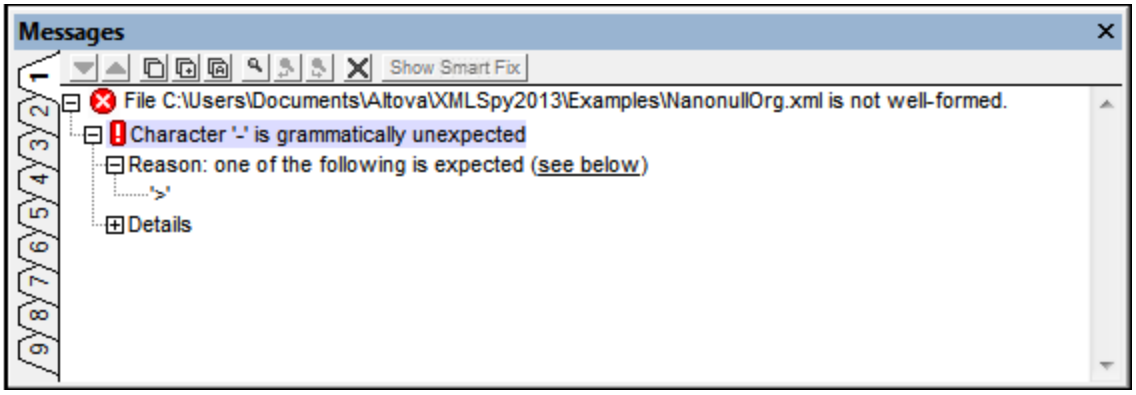
La commande **XML** | **Vérification de la bonne formation (F7)** vérifie le document actif pour la bonne formation avec les définitions de la spécification XML 1.0. Chaque document XML **doit** être bien formé. XMLSpy vérifie la bonne formation qu'un document soit ouvert ou enregistré, ou lorsque le mode est changé du Mode Texte en un autre mode. Vous pouvez aussi vérifier la bonne formation à tout moment pendant l'édition en utilisant cette commande.

Si la vérification de la bonne formation réussit, un message s'affichera dans la fenêtre Messages (*capture d'écran ci-dessous*).

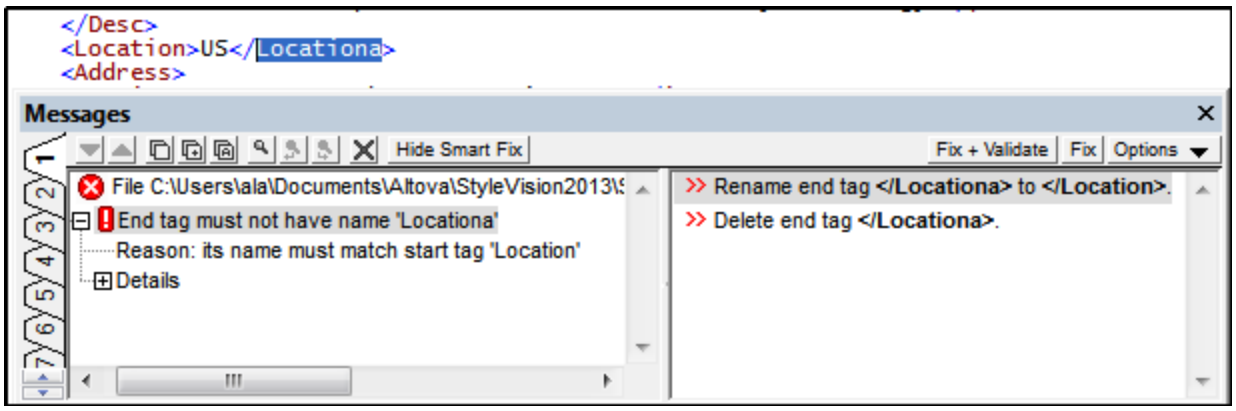


Si vous rencontrez une erreur pendant la vérification de la bonne formation, un message d'erreur correspondant s'affiche (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Sur la base de l'information dans le schéma, les options pour un smart fix sont également suggérées si la vérification de la bonne formation a été effectuée dans le **Mode Texte ou le Mode Grille**. Pour consulter une liste d'options smart fix, cliquer sur la touche **Afficher Smart Fix** (*voir capture d'écran ci-dessus*). Un panneau contenant des suggestions d'options smart fix apparaît dans la fenêtre Messages (*voir capture d'écran ci-dessous*). Veuillez noter que les erreurs dans la fenêtre Messages sont affichées une erreur à la fois.



Dans le volet Smart Fix, choisir un des smart fix suggérés et cliquez soit sur la touche **Réparer + Valider** ou sur la touche **Réparer** (*voir capture d'écran ci-dessus*). L'erreur sera corrigée avec le smart fix. En alternative, vous pouvez double-cliquer sur le smart fix que vous voulez. Cette action permet soit de réparer l'erreur, soit de réparer l'erreur et de valider le document, selon l'option sélectionnée dans la liste déroulante *Options*. La commande **Réparer + Valider** est utile parce que le validateur sera maintenant en mesure de valider au-delà de l'erreur réparée et de détecter l'erreur de bonne formation suivante ou l'erreur de validation, le cas échéant.

Pour dissimuler le volet Smart Fix, cliquer sur la touche **Cacher Smart Fix** (*voir capture d'écran ci-dessus*).

Note : La fenêtre Messages contient neuf onglets. Le résultat de validation est toujours affiché dans l'onglet actif. Vous pouvez donc valider un document XML dans l'onglet 1 et conserver le résultat dans cet onglet. Pour valider un second document, passez à l'onglet 2 (ou l'onglet 3 si vous préférez) avant d'effectuer la vérification. Si vous ne changez pas d'onglet, l'onglet 1 (ou l'onglet actif) sera écrasé avec les résultats de la dernière validation.

Validation depuis la fenêtre Projet

La commande **Valider** peut aussi être appliquée à un fichier, un dossier ou un groupe de fichiers dans le projet actif. Choisir le fichier ou le dossier nécessaire dans la fenêtre Projet (en cliquant dessus). Ensuite, cliquez sur **XML | Valider XML**. Vous pouvez aussi utiliser la commande **Valider XML sur le Serveur (haute performance)**. Les fichiers invalides se trouvant dans un projet seront ouverts et rendus actifs dans la Fenêtre principale et le message d'erreur *Fichier n'est pas valide* s'affichera.

Note : La fenêtre Messages contient neuf onglets. Le résultat de validation est toujours affiché dans l'onglet actif. Vous pouvez donc valider un document XML dans l'onglet 1 et conserver le résultat dans cet onglet. Pour vérifier la bonne formation d'un second document, passez à l'onglet 2 (ou l'onglet 3 si vous préférez) avant d'effectuer la vérification. Si vous ne changez pas d'onglet, l'onglet 1 (ou l'onglet actif) sera écrasé avec les résultats de la dernière vérification

Il n'est généralement pas permis d'enregistrer un document XML mal formé, mais XMLSpy vous propose une option Enregistrer tout de même. Cela peut être utile lorsque vous souhaitez interrompre votre travail temporairement (en état mal formé) pour le reprendre plus tard.

Note : Vous pouvez aussi utiliser la commande **Vérification de la bonne formation** sur tout autre fichier, dossier ou groupes de fichiers dans la fenêtre de projet active. Cliquer sur l'item respectif et puis sur l'icône Vérifier la bonne formation.

28.4.12 Valider XML



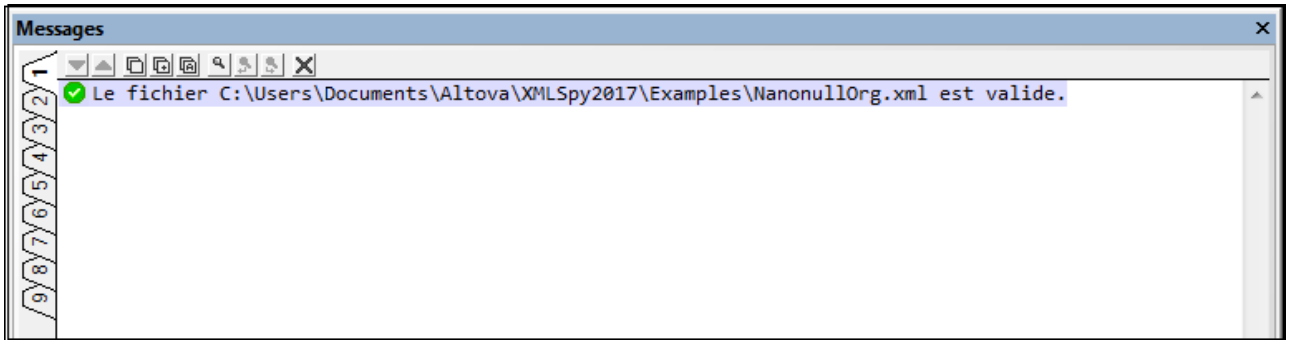
F8

La commande **XML | Valider (F8)** vous permet de valider des documents XML par rapport aux DTD, Schémas XML et autres schémas. La validation est effectuée automatiquement lorsque vous passez du Mode Texte à un autre mode. Vous pouvez spécifier qu'un document soit automatiquement validé lorsqu'un fichier est ouvert ou enregistré (**Outils | Options | Fichier**). La commande **Valider** effectue aussi une vérification de la bonne formation avant de vérifier la validité, il n'est donc pas nécessaire d'utiliser la commande **Vérifier la bonne formation** avant d'utiliser la commande **Valider**.

Note : vous pouvez activer la commande **Valider sur Édition** pour valider lorsque vous éditez des données.-

Si un document est valide, un message de validation réussi est affiché dans la fenêtre Messages.

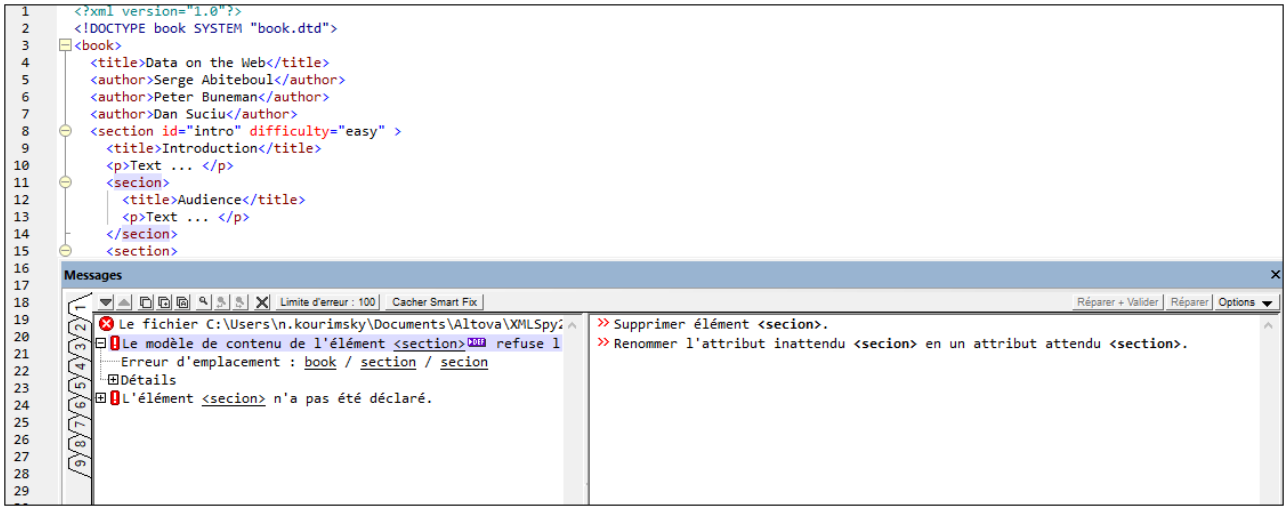
© 2016-2022 Altova GmbH



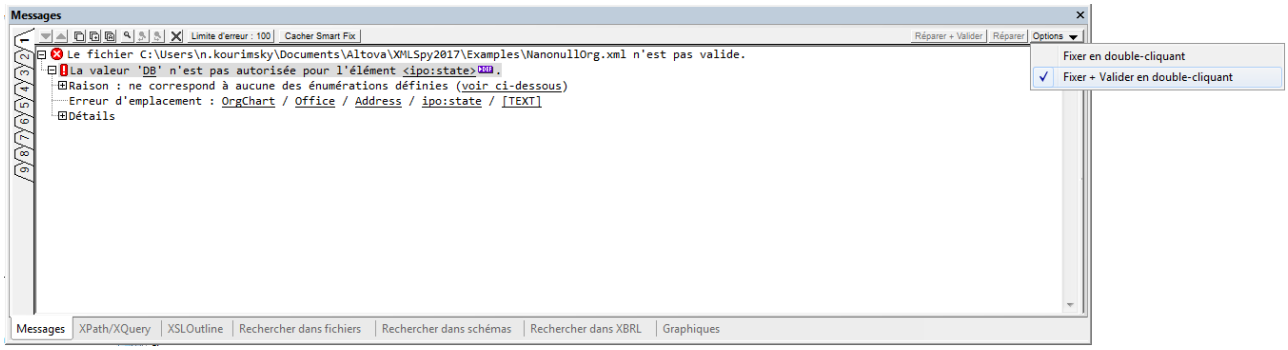
Sinon, un message qui décrit l'erreur est affiché. Vous pouvez cliquer sur les liens dans le message d'erreur pour passer au nœud dans le document XML où l'erreur a été trouvée. Voir la section suivante pour voir une description du message d'erreur et pour consulter comment réparer les erreurs de validation avec les smart fixes de XMLSpy.

Erreurs de validation et leurs réparations

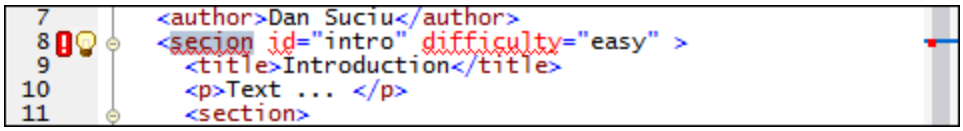
Lorsqu'une erreur de validation est affichée dans la fenêtre Messages, les causes de l'erreur sont affichées dans le volet de gauche (*voir la capture d'écran ci-dessous*). Si une cause est sélectionnée dans le panneau de gauche, des smart fixes, si disponibles, sont affichés dans le panneau de droite. Des suggestions de Smart fix sont disponibles dans le **Mode Texte** et le **Mode Grille**, et sont basées sur des informations se trouvant dans le schéma associé. Pour consulter les smart fixes, cliquer sur la touche **Afficher Smart Fix**. Cliquer sur **Dissimuler Smart Fix** si vous ne souhaitez pas que ces suggestions soient affichées. Veuillez noter que les erreurs de bonne formation (comme des balises de démarrage et de fin qui ne correspondent pas), le cas échéant, sont affichées avant que les erreurs de validation soient affichées. Donc la touche **Afficher/Dissimuler Smart Fix** sera activée uniquement lorsqu'une erreur de validation sera atteinte (c'est-à-dire, après que toutes les erreurs de bonne formation aient été corrigées).



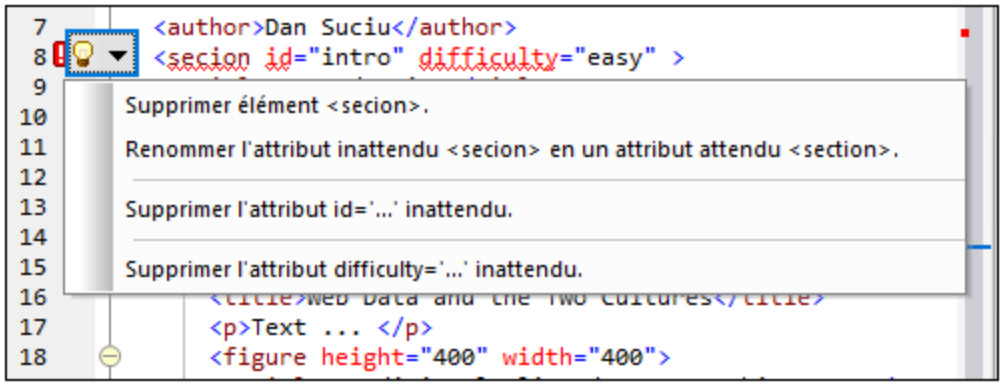
Pour appliquer un smart fix, soit le (i) double-cliquer, soit (ii) le sélectionner et cliquer soit sur l'option **Fix** ou **Fix + Valider** (*voir capture d'écran ci-dessous*). La commande **Réparer + Valider** validera outre l'erreur fixée et choisira l'erreur suivante, le cas échéant.



Dans le Mode Texte, vous trouverez deux indicateurs supplémentaires d'une erreur de validation (*voir capture d'écran ci-dessous*): (i) un point d'exclamation rouge dans la marge numérotée et (ii) un carré de marquage rouge dans la barre de défilement (à la droite de la fenêtre).



L'icône ampoule à côté de l'icône du point d'exclamation (*voir la capture d'écran ci-dessus*) est l'icône du smart fix. Si vous passez sur l'icône, tous les smart fix sur toutes les causes de l'erreur sont affichés (*voir la capture d'écran ci-dessous*). Sélectionnez un smart fix pour l'appliquer.



Note : les indicateurs d'erreur de validation et les smart fixes décrits ci-dessus sont réinitialisés uniquement lorsque la commande **XML | Valider (F8)** est exécutée ; ils ne sont pas mis à jour en arrière-plan. Donc, si vous corrigez une erreur, vous devrez exécuter la commande**Valider (F8)** à nouveau pour vous assurer que l'erreur a bien été réparée.

Note : la fenêtre Messages a neuf onglets. Le résultat de validation est toujours affiché dans l'onglet actif. Donc vous pouvez valider un document XML dans l'onglet 1 et conserver le résultat dans l'onglet. Pour valider un second document, passer à l'onglet 2 (ou l'onglet 3 si vous préférez) avant d'effectuer le contrôle. Si vous ne souhaitez pas changer d'onglets, l'onglet 1 (ou l'onglet actif) sera écrasé avec les résultats de la dernière validation.

© 2016-2022 Altova GmbH

Valider depuis la fenêtre Projet

La commande **Valider** peut aussi s'appliquer à un fichier, dossier ou un groupe de fichiers dans le projet actif. Choisir le fichier ou le dossier requis dans la fenêtre Projet (en cliquant dessus). Puis, cliquez sur **XML | Valider** ou **F8**. Les fichiers invalides se trouvant dans un projet seront ouverts et rendus actifs dans la Fenêtre principale, et le message *Fichier n'est pas valide* s'affichera.

Valider des documents XML

Pour valider un fichier XML, rendre le document XML actif dans la Fenêtre principale et cliquez sur **XML | Valider** ou **F8**. Le document XML est validé par rapport au schéma référencé dans le fichier XML. Si un document est valide, un message de validation réussi est affiché dans la fenêtre Messages. Tant que le document XML est ouvert, le schéma est conservé dans la mémoire (voir Flush Memory Cache dans le menu DTD/Schéma).

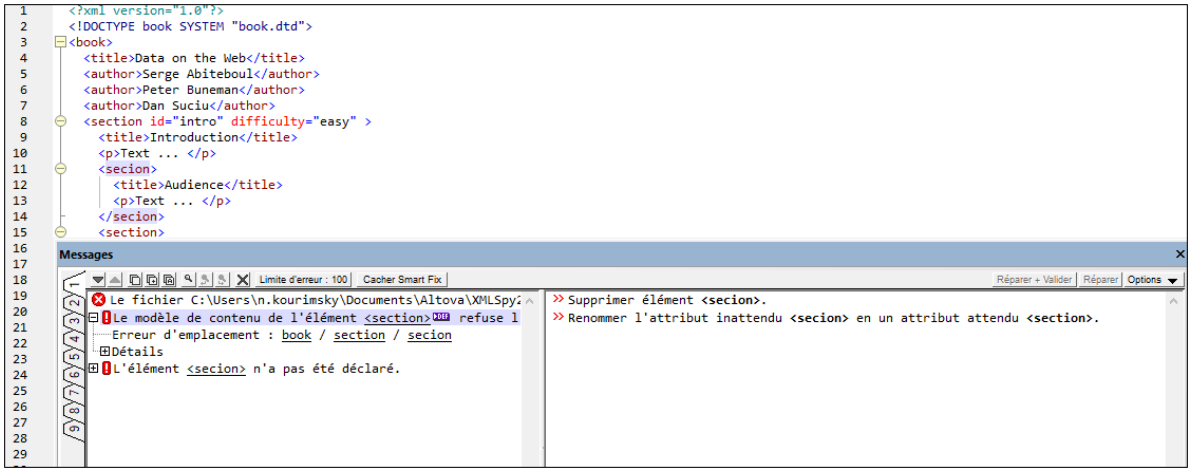
Valider des documents de schéma (DTD et Schéma XML)

XMLSpy prend en charge les dialectes de schéma principaux, y compris les DTD et les Schémas XML. Pour valider un document de schéma, rendre le document actif dans la Fenêtre principale et cliquez sur **XML | Valider** ou **F8**.

Messages de validation

Il existe deux types de messages :

- Si le schéma (DTD ou Schéma XML) est valide, un message de réussite de la validation est affiché dans la fenêtre Messages.
- Si le schéma n'est pas valide, un message d'erreur est affiché dans la fenêtre Messages (*voir la capture d'écran ci-dessous*).



Un message d'erreur affiche chaque cause possible de cette erreur séparément Par exemple, dans la capture d'écran ci-dessus, quatre causes possibles d'erreur de validation sont rapportées ; la première est agrandie, les trois autres sont réduites. Chaque cause est divisée en trois parties :

1. Une description de la cause possible. La description contient des liens vers la définition pertinente dans le document de schéma associé. Vous pouvez vous rendre rapidement à la définition de schéma spécifique pour voir quelle était la raison exacte de l'invalidité du document.
2. Le chemin d'emplacement vers le nœud dans le document XML qui a entraîné l'erreur. Cliquez sur un nœud dans ce chemin d'emplacement pour marquer ce nœud dans le document.
3. Vous trouverez des informations détaillées concernant l'erreur et un lien menant au paragraphe pertinent dans la spécification de schéma. C'est là que les règles de schéma qui spécifient la légalité pertinente sont spécifiées.

Note : si la validation est effectuée dans le Mode Texte, cliquer sur un lien dans la Fenêtre Messages pour *marquer* la définition correspondante dans le Mode Texte. Si la validation est effectuée dans le Mode Schéma, cliquer sur un lien de définition pour *ouvrir* la définition dans le Mode Schéma et vous permettre d'*éditer le composant directement*.

Catalogues

Pour toute information sur la prise en charge dans XMLSpy, voir la section Catalogs in XMLSpy.

Validation automatique avec Raptor XML 2023

RaptorXML est l'application autonome d'Altova pour une validation XML, une transformation XSLT et une transformation XQuery. Elle peut être utilisée depuis la ligne de commande, par le biais d'une interface COM, dans des programmes Java et dans les applications .NET. Les tâches de validation peuvent donc être automatisées avec l'utilisation de RaptorXML. Par exemple, vous pouvez créer un fichier batch qui appelle RaptorXML pour effectuer une validation sur un ensemble de documents et envoie la sortie dans un fichier texte. Voir la documentation RaptorXML pour tout détail.

28.4.13 Valider XML sur serveur (performance élevée)

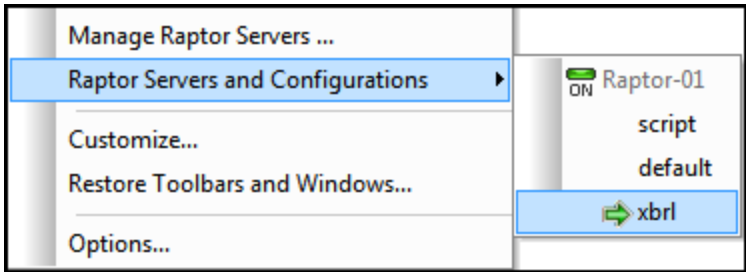
 **Ctrl+F8**

La commande **XML | Valider sur le serveur (performance élevée) (Ctrl+F8)** valide le document XML actif en utilisant le serveur RaptorXML actuellement actif et sa configuration active. La commande effectue immédiatement la validation et affiche les résultats dans la Fenêtre Messages.

Note : La performance dépend du nombre de cœurs de processeur PC utilisés par le Serveur RaptorXML pour la validation : plus de nombre est élevé, plus le traitement sera rapide.

Si vous avez défini plusieurs configurations sur plusieurs serveurs, vous pouvez sélectionner un serveur et une de ses configurations en tant que la configuration active. Celle-ci sera utilisée pour les validations suivantes. En plaçant le curseur sur la commande **Outils | Serveurs Raptor et Configurations** (*capture d'écran ci-dessous*), un sous-menu apparaît qui contient tous les serveurs ajoutés, avec la configuration de chacun d'entre eux. Sélectionner la configuration de serveur que vous souhaitez pour rendre la configuration active. Dans la capture d'écran ci-dessous, la configuration `xbrl` du serveur nommé `Raptor-01` a été sélectionnée en tant que la configuration active (indiquée par la flèche verte).

© 2016-2022 Altova GmbH



La commande **Valider XML sur le serveur (performance élevée) (Ctrl+F8)** est aussi disponible dans l'assistant à la saisie Projet. Cliquer avec la touche de droite sur le projet, sur un dossier ou un fichier et choisir **Valider XML sur le serveur** pour valider les données XML ou XBRL dans l'objet sélectionné.

Note: La validation Raptor est disponible dans le Mode Texte, le Mode Grille et le Mode XBRL.

28.4.14 Validation des fichiers WSDL

Un document WSDL n'est pas seulement un document WSDL mais aussi un document XML. En résultat, il peut être validé en tant que XML et aussi en tant que WSDL. La liste suivante contient des informations importantes concernant le comportement de la validation WSDL dans les *Éditions Enterprise* et *Professional* de XMLSpy :

- L'Édition *Professional* exécute une validation de schéma simple, elle donc traite le fichier WSDL en tant que fichier XML et le valide conformément au schéma défini sous http://schemas.xmlsoap.org/wsdl/.
- L'Édition *Enterprise* propose une validation WSDL qui dépasse la validation XML fournie par l'Édition *Professional*. Elle ne valide pas par rapport à http://schemas.xmlsoap.org/wsdl/. Au lieu, elle utilise le document http://www.altova.com/specs_wsdl.html#_document-s ainsi que sa propre logique. Cela apporte une information de validation supplémentaire dans le contexte de WSDL. Ainsi, il peut se produire qu'un fichier WSDL est valide dans l'Édition *Professional* mais n'est pas valide dans l'Édition *Enterprise* (*voir exemple ci-dessous*).
- Il existe une différence entre http://schemas.xmlsoap.org/wsdl/ et http://www.altova.com/specs_wsdl.html#_document-s. L'ancien schéma ne contient pas de définitions d'**éléments d'extensibilité**, qui sont définies dans la spécification WSDL. Il semblerait que ce défaut soit une erreur dans le schéma W3C officiel ; le défaut est adressé dans le dernier schéma (utilisé par *Enterprise Edition*).
- Puisque l'Édition *Professional* utilise http://schemas.xmlsoap.org/wsdl/ pour la validation, les éléments d'extensibilité seront rapportés comme étant invalides dans l'Édition *Professional,* mais valides dans l'Édition *Enterprise* (qui utilise http://www.altova.com/specs_wsdl.html#_document-s).
- Puisque le schéma W3C est un schéma officiel fourni par le groupe de travail W3C, toute erreur pouvant se trouver dans ce schéma sont malheureusement indépendants du contrôle d'Altova.

Exemple

L'exemple suivant fait partie d'un fichier WSDL. Veuillez noter l'élément "getCityTime" qui a été déclaré dans le fichier. Cet élément est référencé à tort par "getCityTimes". L'Édition *Enterprise* contrôle si les éléments qui sont référencés ont précédemment été déclarés dans le fichier ; l'Édition *Professional* n'est pas déclarée. Ce fichier (en partant du principe que le reste du fichier est valide) sera considéré valide dans l'Édition

Professional, mais invalide dans l'Édition *Enterprise* (en partant du principe que "getCityTimes" n'est pas défini à un autre endroit du fichier).

```
<s:element name="getCityTime">
	<s:complexType>
		<s:sequence>
			<s:element minOccurs="0" maxOccurs="1" name="city" type="s:string"/>
		</s:sequence>
	</s:complexType>
</s:element>
<s:element name="abc">
	<s:complexType>
		<s:sequence>
			<s:element ref="getCityTimes"/>
		</s:sequence>
	</s:complexType>
</s:element>
```

28.4.15 Valider sur Édition

La commande **Valider sur Édition** active/désactive le mode *Valider sur Édition*, qui permet la validation quand vous tapez dans les Mode Texte, Mode Grille JSON, ou Mode Authentic. Le mode peut également être activé/désactivé par le biais de la touche de la barre d'outils de la commande ou de l'option *Validation* > *Sur Édition* de la Section fichier du dialogue des options.

28.4.16 Mise à jour Assistants à la saisie

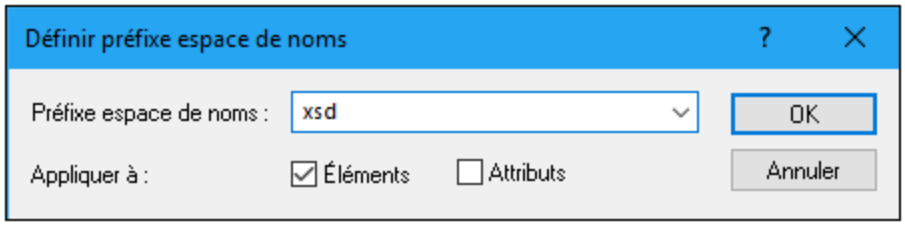


La commande **Mettre à jour les assistants de saisie** met à jour les fenêtres des assistants de saisie en rechargeant DTD ou Schéma. Si vous avez modifié le Schéma DTD ou XML sur lequel un document XML ouvert est basé, vous devriez mettre à jour les assistants de saisie de manière à ce que l'information d'édition intelligente reflète les changements dans le schéma.

28.4.17 Préfixe d'espace de noms

La commande **XML** | **Préfixe d'espace de noms** est disponible dans le Mode Grille et ouvre un dialogue dans lequel vous pouvez définir le préfixe d'espace de noms de l'élément ou de l'attribut sélectionné, et, dans le cas des éléments, de ses descendants également.

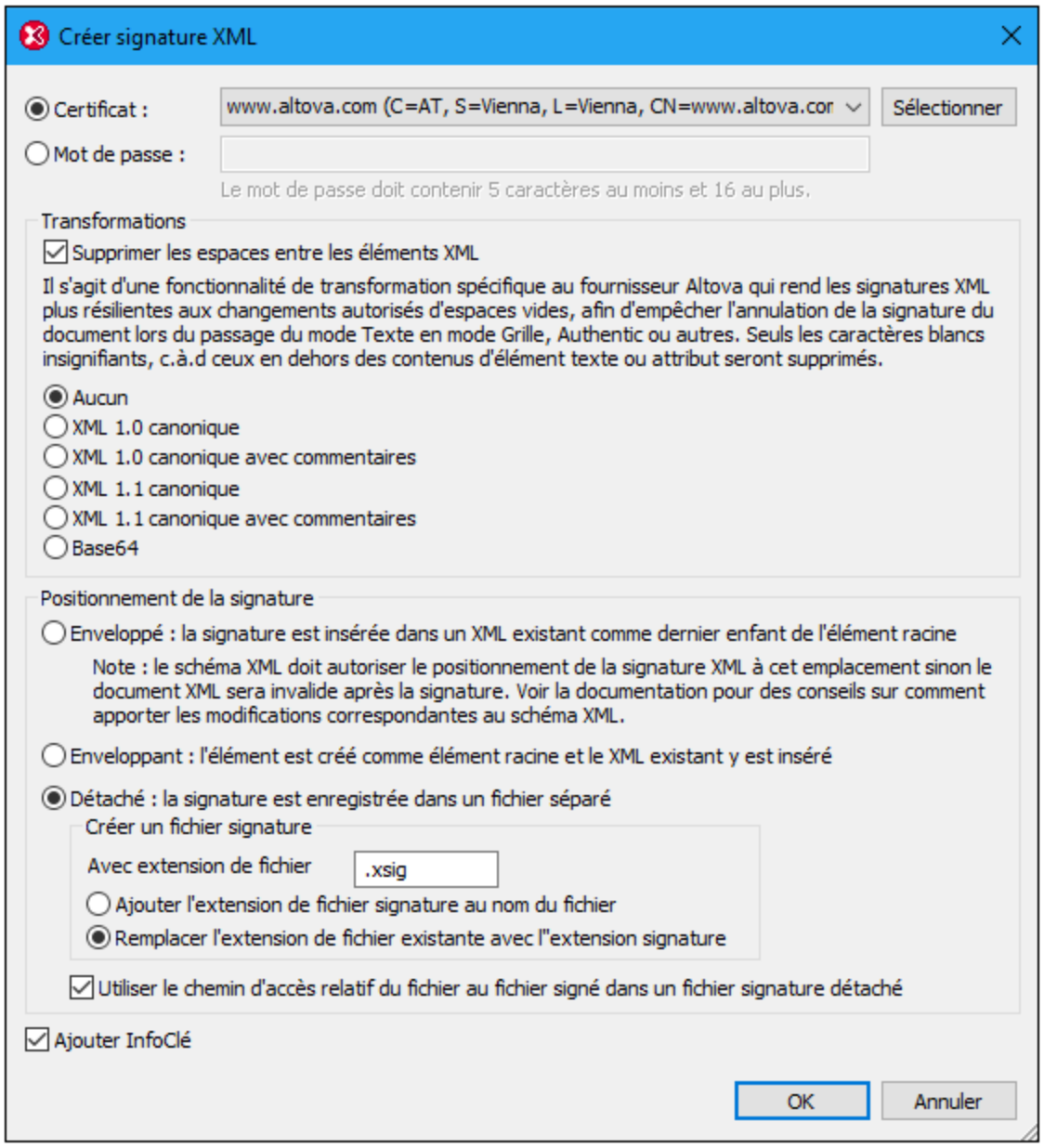
© 2016-2022 Altova GmbH



Vous pouvez choisir de définir le préfixe d'espace de noms soit sur les éléments, les attributs ou les deux. Le préfixe d'espace de noms s'applique à l'élément ou l'attribut sélectionné et, si un élément est sélectionné, aux nœuds descendants de l'élément sélectionné.

28.4.18 Créer Signature XML

La commande **Créer Signature XML** est activée dans le Mode Texte, le Mode Grille, le Mode Schéma, le Mode WSDL et le Mode XBRL, et vous permet de créer une signature XML pour le document XML actif. Cliquer sur la commande pour ouvrir le dialogue Créer Signature XML (*capture d'écran ci-dessous*), dont les paramètres sont expliqués ci-dessous.



Méthode d'authentification : certificat ou mot de passe

La signature peut être basée sur un certificat ou un mot de passe. Choisir le bouton radio de la méthode que vous souhaitez utiliser.

- *Certificat :* Si vous souhaitez utiliser un certificat, celui-ci doit avoir une clé privée et doit se situer dans un stockage de certificats accessible . La signature est générée en utilisant la clé privée du certificat. Afin de vérifier la signature, un accès au certificat (ou une version de clé publique) est nécessaire. La clé publique du certificat est utilisée pour vérifier la signature. Afin de sélectionner le certificat de clé privé-public que vous souhaitez utiliser, cliquer sur la touche **Sélectionner** et chercher le certificat. Pour plus de détails concernant les certificats, voir la section Travailler avec des Certificats.

© 2016-2022 Altova GmbH

- *Mot de passe :* Saisir un mot de passe d'une longueur de cinq à seize caractères. Ce mot de passe sera nécessaire pour vérifier la signature.

Transformations

Les données XML sont transformées et le résultat de la transformation est utilisé pour la création de la signature. Vous pouvez spécifier l'algorithme de canonicalisation pour qu'il soit appliqué dans les données XML du fichier (le contenu `SignedInfo`) avant d'effectuer les calculs de la signature. Vous trouverez ci-dessous les différences principales entre les différents algorithmes :

- *Canonique XML avec ou sans commentaires :* si des commentaires sont inclus pour le calcul de la signature, tous les changements apportés aux commentaires dans les données XML aboutiront à des échec de vérification. Sinon, les commentaires peuvent être modifiés ou ajoutés au document XML une fois que le document a été signé, et la signature sera néanmoins encore vérifiée comme étant authentique.
- *Base64:* L'élément root (ou document) du document XML est considéré comme étant encodé en Base64, et est lu dans sa forme binaire. Si l'élément root n'est pas encodé en Base64, une erreur est retournée ou l'élément est lu en tant que vide, selon le type d'élément rencontré.
- *Aucun :* Aucune transformation n'est effectuée et les données XML provenant du fichier binaire et enregistrées sur le disque sont transmises directement pour la création de la signature. Toute modification ultérieure des données résultera en un échec de la vérification de la signature. Néanmoins, si la case *Supprimer les espaces blancs* est cochée, tous les espaces blancs seront éliminés et les modifications dans les espaces blancs seront ignorés. La différence principale entre l'option *Aucun* et une option *Canonicalisation* est que la canonicalisation produit un flux de données XML, dans lequel certaines différences comme un ordre d'attribut, sont normalisées. Ainsi, une transformation de canonicalisation normalisera toutes les modifications comme par exemple l'ordre des attributs (donc la vérification réussira), alors qu'une non-transformation reflètera une telle modification (la vérification échouera).

Placement de la signature

La signature peut être placée dans le cadre du fichier XML ou être créée en tant que fichier séparé. Les options suivantes sont disponibles :

- *Enveloppé:* L'élément de signature est créé en tant que le dernier élément enfant de l'élément root (document).
- *Enveloppant:* L'élément de signature est créé en tant que l'élément root (document) et le document XML est inséré en tant qu'élément enfant.
- *Détaché:* La signature XML est créée en tant que fichier séparé. Dans ce cas, vous pouvez spécifier l'extension de fichier du fichier de signature et si le nom du fichier est créé avec : (i) l'extension apposée au nom du fichier XML (par exemple, `test.xml.xsig`), ou (ii) l'extension remplaçant l'extension XML du fichier XML (par exemple, `test.xsig`). Vous pouvez aussi spécifier si, dans le fichier de la signature, la référence au fichier XML est un chemin relatif ou absolu.

Note : Les signatures XML pour les fichiers XML Schema (`.xsd`) et pour les fichiers XBRL peuvent uniquement être créées en tant que fichiers de signature externes. En ce qui concerne les fichiers WSDL, les signatures peuvent être créées en tant que fichiers externes et peuvent être "enveloppées" dans le fichier WSDL.

Note : Si la signature XML est créée en tant que fichier séparé, le fichier XML et le fichier de signature sont associés l'un à l'autre par le biais d'une référence dans le fichier de signature. Par conséquent, la vérification de signature dans les cas où la signature est un fichier externe doit être effectuée lorsque le fichier de signature est actif, et non pas avec le fichier XML activé.

Apposer une information de clé

L'option *Apposer des infos de clé* sont disponibles lorsque la signature est basée sur certificat. Elle n'est pas disponible si la signature est basée sur mot de passe.

Si l'option est sélectionnée, l'information de clé publique est placée dans la signature, sinon, l'information de clé ne sera pas incluse dans la signature. L'avantage d'inclure des informations de clé est que le certificat lui-même (en particulier l'information de clé publique qui s'y trouve) ne sera pas nécessaire pour le processus de vérification (puisque l'information de clé est présente dans la signature).

28.4.19 Vérifier Signature XML

Une signature XML sera vérifiée correctement si le fichier XML n'a pas été modifié depuis sa signature. Sinon, la vérification échouera. La commande **Vérifier Signature XML** exécute le processus de vérification et affiche les résultats de la vérification dans les fenêtres Messages. Les différents scénarios de vérification dans XMLSpy sont décrits ci-dessous :

- Le fichier XML contient une signature à base de certificat, les informations clé sont incluses dans la signature
- Le fichier XML contient une signature à base de certificat, les informations clé ne sont pas incluses dans la signature
- Signature à base de certificat dans le fichier externe, les informations clé sont incluses dans la signature
- Signature à base de certificat dans le fichier externe, les informations clé ne sont pas incluses dans la signature
- Le fichier XML contient une signature à base de mot de passe
- Signature à base de mot de passe dans le fichier externe

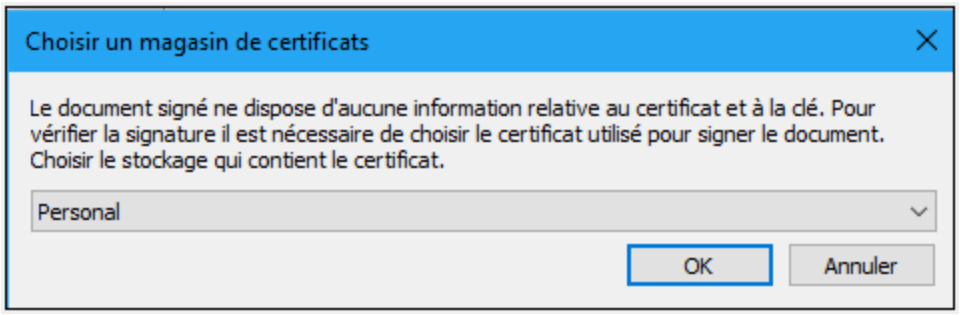
Le fichier XML contient une signature à base de certificat, les informations clé sont incluses dans la signature

Pour vérifier la signature XML dans ce scénario, rendre le fichier XML actif dans XMLSpy. Cliquer sur la commande **XML | Vérifier la signature XML** pour exécuter le processus de vérification et afficher le résultat dans la fenêtre Messages (la vérification a réussi ou a échoué).

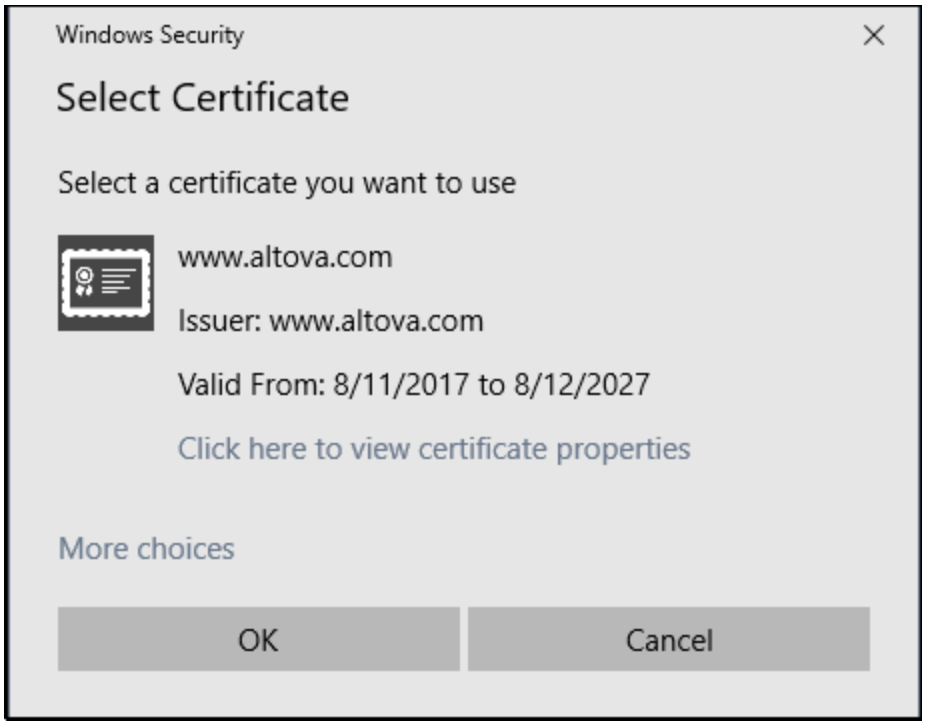
Le fichier XML contient une signature à base de certificat, les informations clé ne sont pas incluses dans la signature

Si aucune information clé n'est contenue dans la signature à base de certificat, XMLSpy vous invitera à choisir le certificat à partir duquel l'information à clé publique pour la vérification sera lue. Pour effectuer la vérification, activer le fichier XML dans XMLSpy. Cliquer sur la commande **XML | Vérifier la signature XML**, vous serez invité à choisir le stockage de certificat dans lequel le certificat est stocké (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



Une fois avoir choisi un stockage de certificat et avoir cliqué sur **OK**, un dialogue affichant les certificats dans ce stockage s'affiche (*capture d'écran ci-dessous*). Choisir le certificat nécessaire pour la vérification et cliquer sur **OK**.



La signature est vérifiée et le résultat sera affiché dans la fenêtre Messages.

Signature à base de certificat dans le fichier externe, les informations clé sont incluses dans la signature

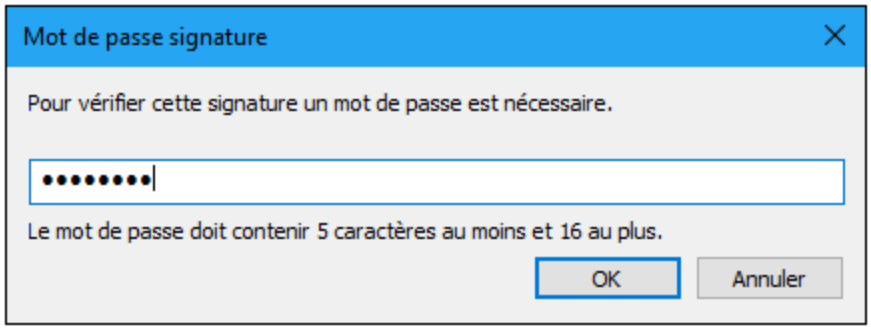
Si une signature XML basée sur un certificat est un fichier externe, la signature est vérifiée alors que le fichier de signature est activé dans XMLSpy. Cliquer sur la commande **XML | Vérifier la signature XML**, le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

Signature à base de certificat dans le fichier externe, les informations clé ne sont pas incluses dans la signature

Si une signature XML basée sur un certificat est un fichier externe, la signature est vérifiée alors que le fichier de signature est activé dans XMLSpy. Cliquer sur la commande **XML | Vérifier la signature XML**, XMLSpy vous invitera à choisir le certificat à partir duquel l'information à clé publique pour la vérification sera lue. Choisir le certificat en suivant les instructions dans la section : *Le fichier XML contient une signature à base de certificat, les informations clé ne sont pas incluses dans la signature*. Le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

Le fichier XML contient une signature à base de mot de passe

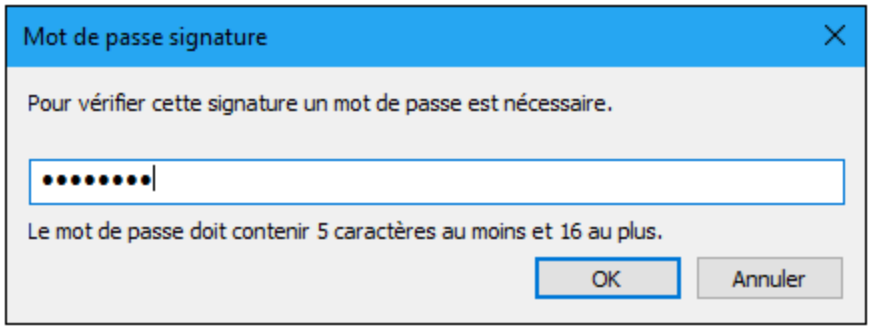
Si le fichier XML contient une signature XML basée sur un mot de passe, la signature est vérifiée alors que le fichier de signature est activé dans XMLSpy. Cliquer sur la commande **XML | Vérifier la signature XML**, un dialogue s'ouvre vous demandant de saisir un mot de passe (*capture d'écran ci-dessous*).



Saisir le mot de passe, qui doit avoir une longueur de cinq à seize caractères, puis cliquer sur **OK**. Le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

Signature à base de mot de passe dans le fichier externe

Si une signature XML basée sur un mot de passe est un fichier externe, la signature est vérifiée alors que le fichier de signature est activé dans XMLSpy. Cliquer sur la commande **XML | Vérifier la signature XML**, un dialogue s'ouvre vous demandant de saisir un mot de passe (*capture d'écran ci-dessous*).

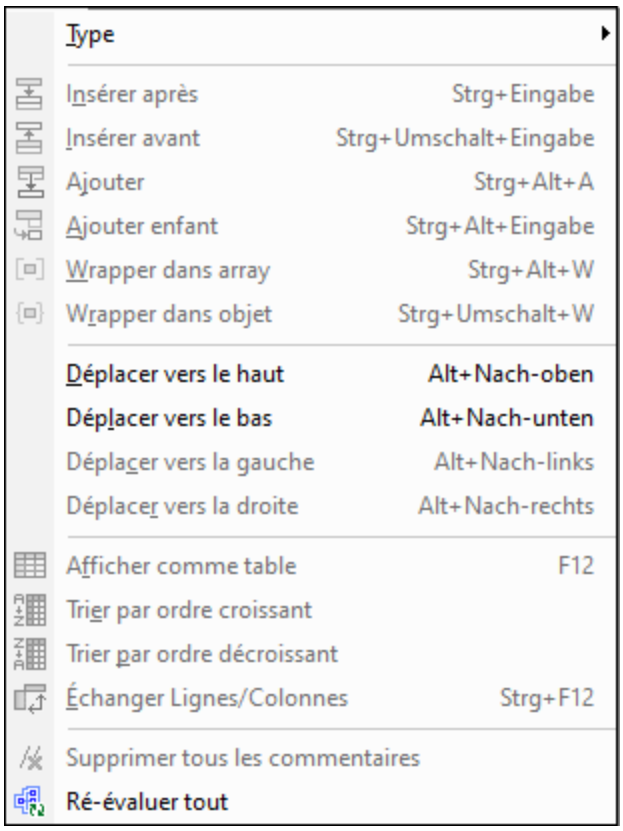


© 2016-2022 Altova GmbH

Saisir le mot de passe, qui doit avoir une longueur de cinq à seize caractères, puis cliquer sur **OK**. Le processus de vérification sera exécuté et le résultat sera affiché dans la fenêtre Messages (la vérification a réussi ou a échoué).

28.5 Menu JSON

Le menu **JSON** contient des commandes qui sont communément requises lorsqu'on travaille avec des documents JSON. Une majorité des commandes sont utilisées lorsque vous travaillez dans Mode JSON Grid. Si une commande n'est pas appliquée à l'emplacement du curseur actuel, elle est désactivée.



Le commandes ce menu sont décrites dans les sous-sections de cette section :

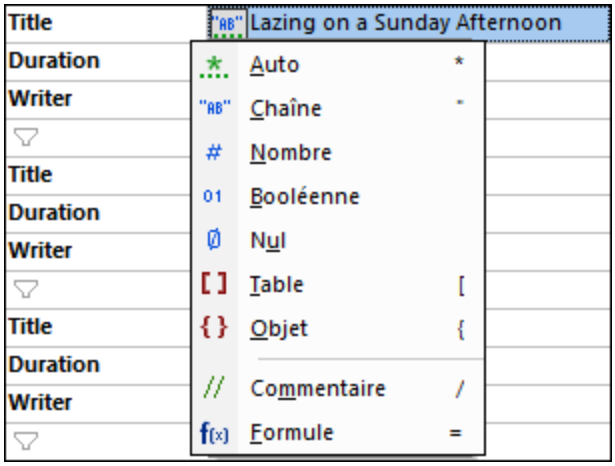
- Type
- Insérer avant&après, Apposer, Ajouter enfant
- Wrapper dans array/objet
- Déplacer
- Afficher en tant que Table
- Tri Montant/Descendant
- Échanger Lignes/Colonnes
- Supprimer commentaires, Tout réévaluer,

28.5.1 Type

La commande **Type** (**Ctrl + touche Menu**) affiche un menu contenant des types de données JSON *(capture d'écran ci-dessous).* Sélectionner un type depuis le menu pour l'attribuer aux cellules actuellement

© 2016-2022 Altova GmbH

sélectionnées dans le mode Grille. (La touche **Menu** est généralement située en bas à droite du clavier, à côté de la touche **Ctrl**. Son icône ressemble à :).



Pour plus d'informations concernant les types d'édition, voir Éditer le contenu de Document JSON.

28.5.2 Insérer, Apposer, Ajouter enfant

Les commandes **Insérer Après, Insérer Avant**, **Apposer**, et **Ajouter enfant** sont activées lorsque la sélection actuelle dans Mode Grid permet à un composant d'être respectivement, inséré, apposé ou ajouté en tant qu'enfant.

- **Insérer Après** insère un composant du même type que le composant sélectionné dans une ligne de grille au-dessous du composant sélectionné.
- **Insérer Avant** insère un composant du même type que le composant sélectionné dans une ligne de grille au-dessus du composant sélectionné.
- **Apposer** appose un composant du même type que le composant sélectionné dans une ligne de grille après tous les frères du composant sélectionné.
- **Ajouter enfant** ajoute un nouveau composant enfant en tant qu'un dernier enfant. Le type sera le même que celui de l'enfant qui était le dernier précédemment.

Pour plus d'informations concernant l'édition de la structure de document JSON, voir Éditer la structure de Document JSON.

28.5.3 Wrapper dans Array/Objet

Les commandes **Wrapper dans Array** et **Wrapper dans Objet** commands font chacun un wrap du composant sélectionné dans un array ou un objet, respectivement.
Pour plus d'informations concernant l'édition de la structure de document JSON , voir Éditer la structure de Document JSON.

28.5.4 Déplacer

S'il est possible de déplacer une composant vers le haut, le bas, la gauche ou la droite de son emplacement actuel dans la grille, la commande correspondante sera activée. Choisir la commande respective pour effectuer le déplacement.

28.5.5 Display as Table

The **Display as Table** command is enabled when a repeating component in Grid View is selected. It is a toggle command that switches the display of the repeating components between standard Grid View and Table Display. Table Display enables you to view repeated elements as a table in which the rows represent the occurrences while the columns represent child nodes.

See the topic Table Display for more information.

28.5.6 Tri Montant/Descendant

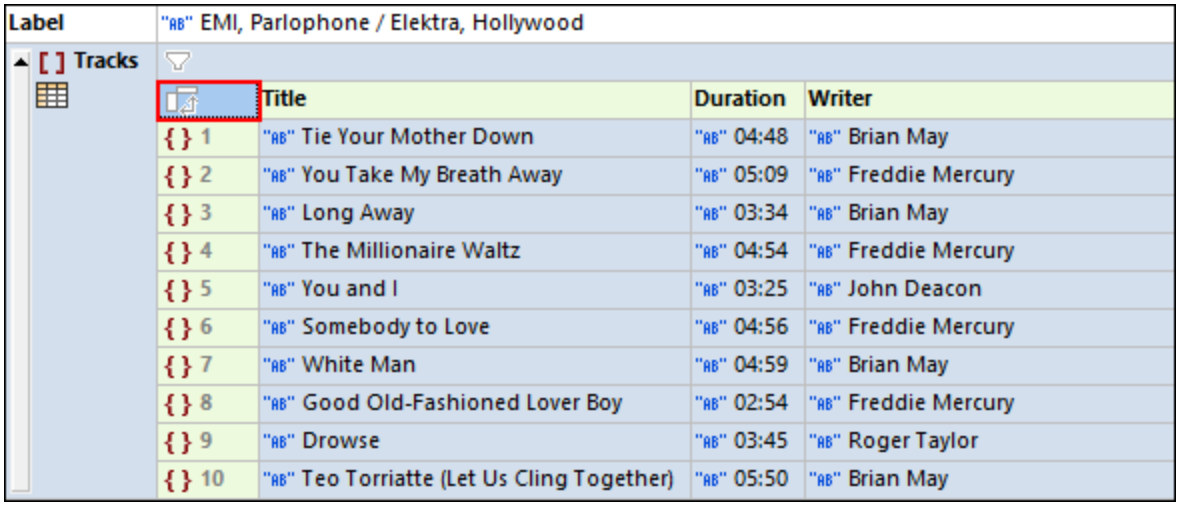
Les commandes **Tri montant** et **Tri ascendant** sont activées dans l'Affichage Table lorsqu'une colonne de l'affichage de table d'un composant est sélectionnée. Pour sélectionner une colonne, sélectionner son en-tête. Le tri est basé sur les valeurs dans la colonne.

Voir le sujet Affichage Table pour plus d'informations.

28.5.7 Échanger Lignes/Colonnes

The **Flip Rows/Columns** command is enabled in Table Display when the top left cell of a table is selected (*marked in red in screenshot below*). The command switches rows to columns and vice versa.

© 2016-2022 Altova GmbH



Label	"AB" EMI, Parlophone / Elektra, Hollywood		
[ ] Tracks			
	Title	Duration	Writer
{ } 1	"AB" Tie Your Mother Down	"AB" 04:48	"AB" Brian May
{ } 2	"AB" You Take My Breath Away	"AB" 05:09	"AB" Freddie Mercury
{ } 3	"AB" Long Away	"AB" 03:34	"AB" Brian May
{ } 4	"AB" The Millionaire Waltz	"AB" 04:54	"AB" Freddie Mercury
{ } 5	"AB" You and I	"AB" 03:25	"AB" John Deacon
{ } 6	"AB" Somebody to Love	"AB" 04:56	"AB" Freddie Mercury
{ } 7	"AB" White Man	"AB" 04:59	"AB" Brian May
{ } 8	"AB" Good Old-Fashioned Lover Boy	"AB" 02:54	"AB" Freddie Mercury
{ } 9	"AB" Drowse	"AB" 03:45	"AB" Roger Taylor
{ } 10	"AB" Teo Torriatte (Let Us Cling Together)	"AB" 05:50	"AB" Brian May

See the topic Table Display for more information.

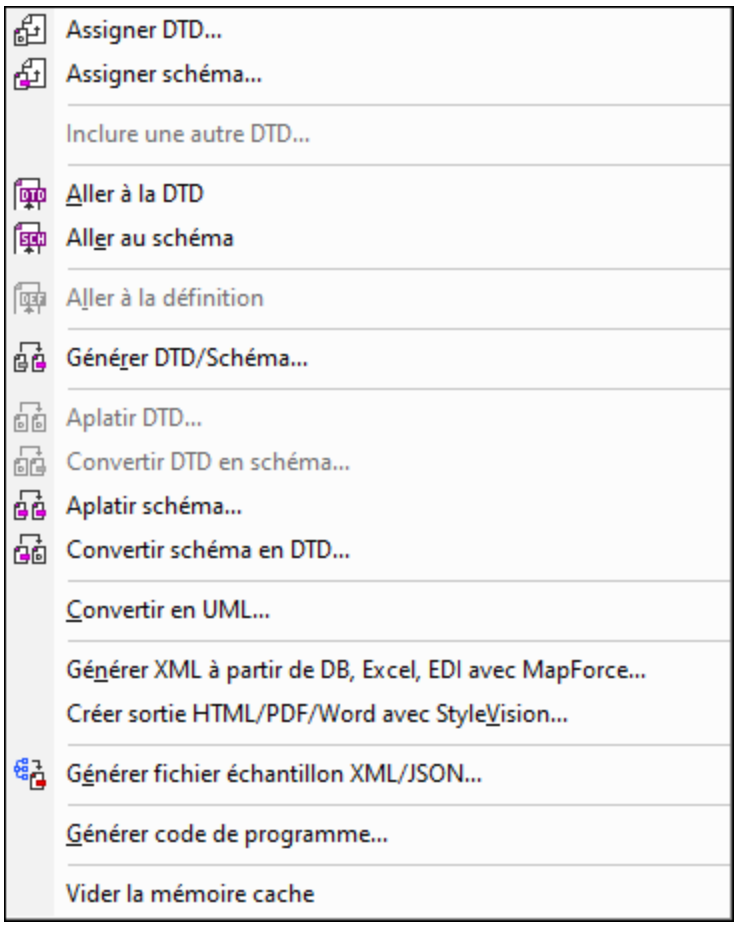
28.5.8 Supprimer commentaires, Tout réévaluer

La commande **Supprimer Commentaires** supprime tous les commentaires. La commande peut être utilisée dans le mode Grille JSON.

La commande **Tout réévaluer** réévalue tous les filtres et formules. Ceci est utile lorsque le document JSON accède des données dynamiques qui peuvent changer avec le temps (par exemple, des taux de change). La commande peut être utilisée en modes Texte et Grille JSON.

28.6 Menu DTD/Schéma

Le menu **DTD/Schéma** (*voir la capture d'écran ci-dessous*) contient des commandes qui fonctionnent avec des DTD et Schémas XML.



28.6.1 Assigner DTD



La commande **DTD/Schéma | Assigner DTD...** est activée lorsqu'un fichier XML est actif. Elle assigne une DTD à un document XML, permettant ainsi au document d'être validé et permettant également une édition intelligente du document. La commande ouvre le dialogue Assigner fichier pour vous permettre de spécifier le fichier DTD que vous souhaitez assigner. Vous pouvez aussi sélectionner un fichier par le biais d'une ressource globale ou une URL (cliquer sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**). Veuillez noter que vous pouvez faire du chemin du fichier DTD assigné un chemin relatif en cliquant la case Rendre le chemin relatif à.... Une fois que vous avez terminé, votre

© 2016-2022 Altova GmbH

document XML contiendra une déclaration DOCTYPE qui référence la DTD assignée. La déclaration DOCTYPE ressemblera à :

```
<!DOCTYPE main SYSTEM "http://link.xmlspy.com/spyweb.dtd">
```

Note : un DTD peut être assigné à un nouveau fichier XML au moment de la création du fichier.

28.6.2 Assigner Schéma



La commande **DTD/Schéma | Assigner Schéma...** est activée lorsqu'un document XML est activé. Elle attribue un Schéma XML à un document XML, permettant de valider le document et d'activer l'édition intelligente pour le document. La commande ouvre le dialogue Assigner fichier pour vous permettre de spécifier le fichier de Schéma XML que vous souhaitez assigner. Vous pouvez aussi sélectionner un fichier par le biais d'une ressource globale ou d'une URL (cliquer sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**). Veuillez noter que vous pouvez rendre le chemin du fichier assigné relatif en cliquant sur la case à cocher Rendre le chemin relatif à .... Lorsque vous avez terminé, votre document XML contiendra une assignation de Schéma XML avec les espaces de noms requis. L'assignation de schéma ressemblera à :

```
xmlns="http://www.xmlspy.com/schemas/icon/orgchart"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:schemaLocation="http://www.xmlspy.com/schemas/icon/orgchart
http://schema.xmlspy.com/schemas/icon/orgchart.xsd"
```

28.6.3 Inclure une autre DTD

La commande **DTD/Schéma | Inclure une autre DTD...** vous permet d'inclure une autre Document Type Definition (DTD) ou une entité parsée externe dans le sous-ensemble interne d'une définition de type de document, ou dans tout document DTD. Cela s'effectue en définissant une déclaration d'entité parsée externe correspondante et en utilisant cette entité dans la ligne suivante :

```
<!ENTITY % navigation.dtd SYSTEM "S:\xml\navigation.dtd">
%navigation.dtd;
```

La commande ouvre le dialogue Assigner fichier pour vous permettre de spécifier le fichier DTD que vous souhaitez inclure dans votre DTD.

Note : cette commande est uniquement activée dans le Mode Grille.

28.6.4 Aller à la DTD



La commande **DTD/Schéma | Aller à la DTD** ouvre la DTD sur laquelle le document XML actif est basé. Si aucune DTD n'est assignée, un message d'erreur s'affichera.

28.6.5 Aller au Schéma



La commande **DTD/Schéma | Aller au Schéma** ouvre le Schéma XML sur lequel le document XML actif est basé. Si aucun Schéma XML n'est assigné, un message d'erreur sera affiché.

28.6.6 Aller à la Définition



La commande **DTD/Schéma | Aller à la définition** affiche la définition exacte d'un élément ou un attribut dans la définition de type du document correspondant ou document de schéma.

Pour voir la définition d'item dans le Mode Grille

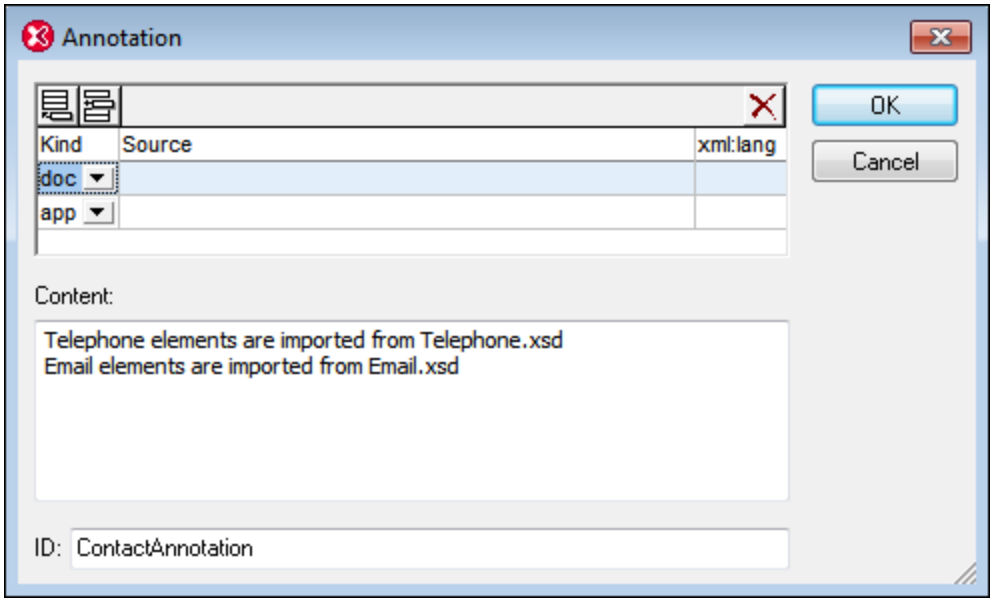
1. Cliquez à gauche sur l'item.
2. Sélectionnez l'item de menu **DTD/Schéma | Aller à la Définition** ou cliquez sur l'icône.

Pour voir la définition d'item dans le Mode Schéma

- Utiliser CTRL + Double clic sur l'item pour lequel vous souhaitez voir la définition de ou
- Cliquez sur l'item et sélectionnez l'option de menu **DTD/Schéma | Aller à la Définition** ou cliquez sur l'icône.

Dans les deux cas, la DTD correspondante ou le fichier de schéma est ouvert et la définition d'item est marquée.

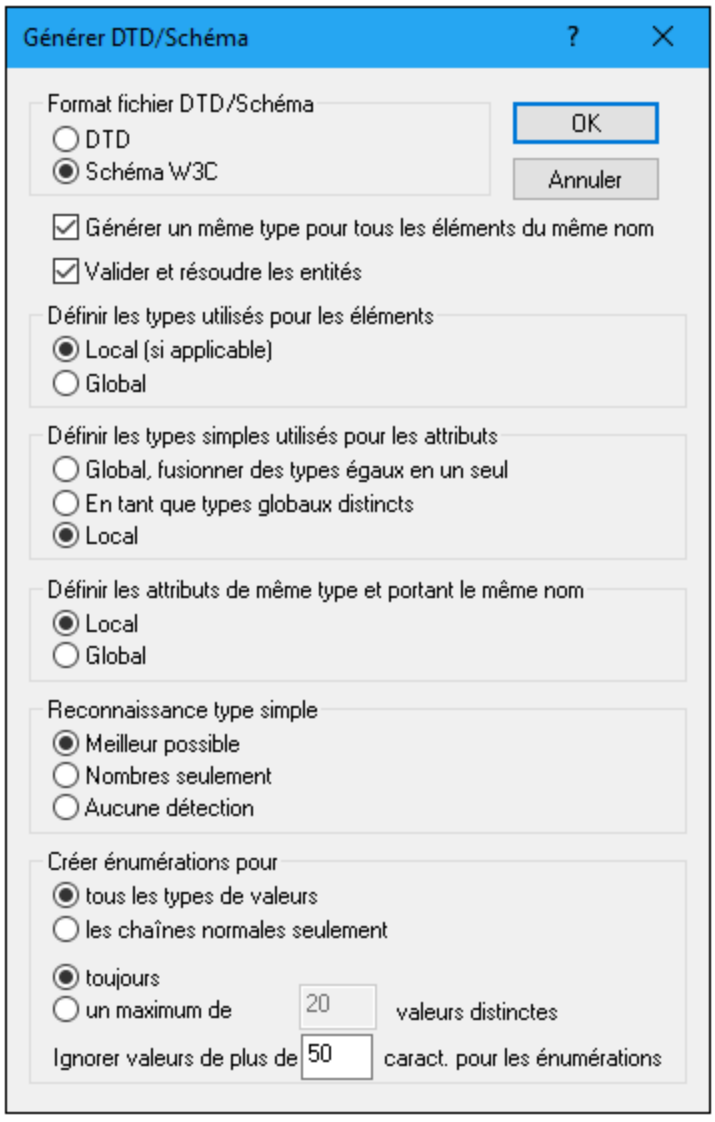
© 2016-2022 Altova GmbH



28.6.7 Générer DTD/Schéma



La commande **DTD/Schéma | Générer DTD/Schéma** génère une nouvelle DTD ou un Schéma XML W3C depuis un document XML (ou depuis un ensemble de documents XML contenu dans un dossier dans la fenêtre de projet). Cette commande est utile lorsque vous souhaitez générer une DTD ou un Schéma XML depuis les documents XML.



Si vous générez un Schéma XML, les options suivantes sont disponibles :

- ***Éléments*** : Le type d'éléments peut être défini soit de manière locale, soit globale (*Définir les types d'éléments*). Si les éléments ont le même nom, un type commun peut être déclaré pour une utilisation dans la définition de ces éléments (*Générer un type partagé*).
- ***Attributs*** : Les types simples d'attributs (*Définir les types simples pour attributs*) peuvent être définis comme (i) types globaux communs ; (ii) types globaux distincts ; (iii) types locaux. Les attributs avec le même nom et type peuvent être définis soit de manière locale, soit globale.
- ***Reconnaissance de type simple*** : La reconnaissance de types (*reconnaissance de type simple*) peut être définie comme : (i) la meilleure possible; (ii) la reconnaissance de types de données de nombre uniquement ; (iii) pas de reconnaissance de type de données, dans un tel cas, tous les types de données sont définis comme `xs:string`.
- ***Résolution d'entité*** : Dans le document XML, les entités peuvent apparaître en contenu d'élément et valeurs d'attribut. Le fait qu'elles soient résolues ou non, (*Valider et résoudre des entités*) est donc important pour les valeurs d'énumération. De plus, certaines entités (en particulier des entités parsées

 © 2016-2022 Altova GmbH

qui contiennent le markup) peuvent affecter le modèle de contenu différemment selon qu'elles sont résolus ou non. Veuillez noter que le document XML sera validé en tant que XML correct avant de générer le schéma. Si le document est invalide, le processus de génération de schéma sera interrompu.

- ***Énumérations***: Tous les types de valeurs, ou valeurs de string uniquement, peuvent être énumérés.

Si vous générez une DTD, la résolution d'entité et les options d'énumération sont disponibles.

La commande générer DTD/Schéma fonctionne normalement sur la fenêtre principale active, mais vous pouvez aussi utiliser la commande **Générer DTD/Schéma** sur tout fichier, dossier ou groupe de fichiers dans la fenêtre de projet active.

Si les éléments ou les attributs sont présents dans plus d'un espace de noms, XMLSpy génère un Schéma XML séparé pour chaque espace de nom distinct ; c'est pourquoi des fichiers multiples peuvent être créés sur le disque.

28.6.8 Aplatir DTD

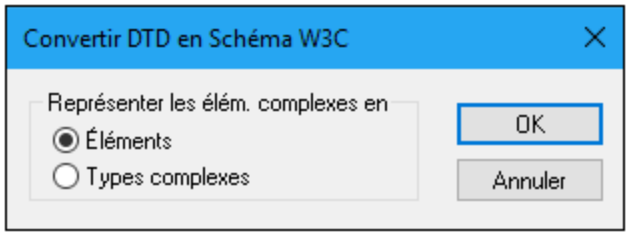
La commande **Aplatir DTD** est activée lorsqu'une DTD est le document actif. Elle crée une nouvelle DTD plate en supprimant des entités de paramètre et en produisant une seule DTD depuis une collection de modules. Elle supprime aussi des sections marquées `IGNORE` et supprime les entités de paramètre non utilisées.

La commande ouvre un dialogue Enregistrer, dans lequel vous pouvez sélectionner un emplacement sous lequel vous pourrez sauvegarder le fichier DTD généré. Cliquer sur **Enregistrer** pour effectuer la conversion. Le fichier DTD aplati est généré et ouvert dans XMLSpy.

28.6.9 Convertir DTD en Schéma

La commande **Convertir DTD en** Schéma est activée lorsqu'une DTD est le document actif. Elle convertit une DTD en un document de Schéma XML (XSD).

La commande ouvre le dialogue Convertir DTD en Schéma W3C (*capture d'écran ci-dessous*), dans laquelle vous pouvez sélectionner si vous souhaitez convertir des éléments complexes dans des éléments ou dans des types complexes. Cliquez sur **OK** pour recevoir une invitation à choisir un emplacement où enregistrer le fichier généré XSD. Cliquez sur **Enregistrer** pour exécuter la conversion. Le fichier XSD est généré et ouvert dans XMLSpy.



Lorsque vous convertissez une DTD en Schéma XML, XMLSpy fait quelques hypothèses seulement en raison du peu d'informations disponibles. En particulier, les valeurs de certains composants de DTD sont traitées

littéralement au lieu de parser leur sémantique. Cela est dû au fait que le programme ne peut pas savoir laquelle des utilisations possibles sera utilisée. Dans ces cas, vous devriez modifier la conversion générée.

Dans tous les cas, vous devriez examiner avec soin la conversion générée pour voir si vous pouvez l'améliorer. Quelques domaines dans lesquels des améliorations peuvent être nécessaires sont rassemblées ci-dessous.

Types de données Attribut
Les DTD permettent uniquement 10 types de données d'attribut, alors que les Schémas XML, par exemple, permettent plus de 40 types de données plus les types de données dérivés. Vous pouvez souhaiter améliorer un Schéma XML généré, par exemple, en utilisant un type de données plus restrictif. Veuillez noter que lorsqu'un Schéma XML est converti en DTD, l'information de type de données sera perdue.

Espaces de noms
Les DTD ne sont pas conscients de l'espace de noms. En conséquence, si les noms d'espaces doivent être spécifiés en DTD, ils doivent être codés en dur dans les noms d'élément et d'attribut. Cela peut poser des défis importants lors de la conversion d'un schéma à un autre.

Entités
Le Schéma XML n'a pas d'équivalent pour les déclarations d'entités générale de DTD. Lorsque XMLSpy convertit une DTD en un Schéma XML, il ignore les déclarations d'entité.

Déclarations de données non parsées
Les DTD et les Schémas XML utilisent des mécanismes différents pour gérer des données non parsées. Cela est expliqué plus en détail ci-dessous.

Les DTD utilisent le mécanisme suivant :

- Une notation est déclarée consistante d'un nom et d'un identifiant, par exemple :
  ```
  <!NOTATION gif SYSTEM "image/gif">
  ```
- Vous déclarez l'entité, par exemple :
  ```
  <!ENTITY cover_img SYSTEM "graphics/cover_img.gif" NDATA gif>
  ```
- Généralement, vous spécifiez un type d'attribut de `ENTITY` sur l'attribut pertinent, par exemple :
  ```
  :<!ELEMENT img EMPTY>
  <!ATTLIST img format ENTITY #REQUIRED>
  ```

Dans le Schéma XML, le mécanisme correspondant est comme suit :

- Déclarer une notation. Cela fonctionne de la même manière que pour la DTD.
  ```
  <xs:notation name="gif" public="image/gif"/>
  ```

  Veuillez noter que l'attribut `public` est obligatoire et maintient l'identifiant. Un attribut `system` optionnel conserve l'identifiant de système et est généralement un programme d'exécution qui peut se charger des ressources du type de notation.
- Vous associez la déclaration `notation` avec une valeur d'attribut donnée en utilisant le type de données `NOTATION`. Néanmoins, vous ne pouvez pas utiliser le type de données `NOTATION` directement, mais vous devez dériver un autre type de données depuis le type de données `NOTATION` datatype.
  ```
  <xs:simpleType name="formatType">
     <xs:restriction base="xs:NOTATION">
  ```

 © 2016-2022 Altova GmbH

```
      <xs:enumeration value="gif"/>
      <xs:enumeration value="jpeg"/>
   </xs:restriction>
</xs:simpleType>
```

- Vous associez l'attribut avec le type de données dérivé depuis le type de données `NOTATION`, par ex.

```
<xs:complexType name="imgType">
   <xs:attribute name="height"/>
   <xs:attribute name="width"/>
   <xs:attribute name="location"/>
   <xs:attribute name="format" type="formatType" use="required"/>
</xs:complexType>
<xs:element name="img" type="imgType"/>
```

Lorsque vous convertissez une DTD en un Schéma XML, XMLSpy, suivez les étapes suivantes :

- Un exemple comme

```
<!ATTLIST image format ENTITY #REQUIRED
...>
```

  est converti en

```
<xs:attribute name="format" type="xs:ENTITY" use="required"/>
```

- et un autre comme

```
<!NOTATION gif SYSTEM "image/gif">
```

  est converti en

```
<xs:notation name="gif" system="image/gif"/>
```

Vous devriez donc effectuer les modifications suivantes :

1. Dans des notations comme `<xs:notation name="gif" system="image/gif"/>`, remplacez system par public, et ajoutez un identifiant de système optionnel, le cas échéant.
2. Dérivez un type de données depuis le type de données `NOTATION` tel que décrit ci-dessus pour `formatType`.
3. Associez le type de données dérivé avec l'attribut pertinent.

Note: Conformément à la spécification de Schéma XML, vous ne nécessitez pas ou ne pouvez pas, selon le point de vue, déclarer une entité externe.

28.6.10 Aplatir Schéma

La commande **Aplatir Schéma** est activée lorsqu'un Schéma XML est le document actif. Elle génère un nouveau XSD plat en (i) ajoutant les composants de tous les schéma inclus en tant que composants globaux du schéma actif et en (ii) supprimant les schémas inclus.

La commande redirige vers la commande Aplatir Schéma du menu Conception de Schéma. Puisque la commande Aplatir Schéma est uniquement disponible dans le Mode Schéma, vous serez invité à choisir si vous souhaitez passer au Mode Schéma ou pas. Pour plus d'informations, voir la commande Aplatir Schéma.

28.6.11 Convertir Schéma en DTD

La commande **Convertir Schéma en DTD** est activée lorsqu'un Schéma XML est le document actif. Il convertit un document Schéma XML (XSD) dans une DTD.

La commande ouvre un dialogue Enregistrer, dans lequel vous choisissez un emplacement dans lequel enregistrer le fichier DTD généré. Cliquer sur **Enregistrer** pour effectuer la conversion. Le fichier DTD est généré et ouvert dans XMLSpy.

Veuillez noter les points suivants :

1. Lorsque vous convertissez un Schéma XML en une DTD, les préfixes d'espace de noms utilisés dans le Schéma XML, pas l'URI d'espace de noms ou les déclarations d'espace de noms, sont effectués jusqu'au noms des éléments correspondants et des attributs dans la DTD.
2. Puisque les parseurs XML ignorent les espaces de noms lors de la validation d'un document XML par rapport à une DTD, les déclaration d'espace de noms elles-mêmes ne sont pas converties.
3. Les attributs `elementFormDefault` et `attributeFormDefault` de l'élément `xs:schema` déterminent quels éléments et attributs ont leurs préfixes inclus dans le processus de conversion. Si définis sur non qualifié, seuls les éléments déclarés globalement et les attributs, respectivement, incluront des préfixes dans la conversion. Si définis sur qualifié, tous les éléments et les noms d'attribut ont leurs préfixes inclus dans la conversion.
4. Les préfixes sont convertis dans leurs valeur de string correspondantes, plus un double-point. Les éléments et les attributs dans les espaces de noms par défaut sont convertis en éléments et attributs avec des noms qui commencent avec le string : `default_NS_X`, où X est un entier (commençant avec `1` et ayant une valeur maximum égale au nombre d'espaces de noms par défaut utilisés dans le Schéma XML).
5. Dans la DTD, les noms d'élément sont composés d'entités de paramètre. Cela vous permet de changer facilement le préfixe dans la DTD si le préfixe dans le document XML devait changer à un moment ou un autre. Les définitions d'entité de paramètre peuvent être changées soit dans le document DTD lui-même soit en contournant les définitions d'entité de paramètre dans le sous-ensemble de DTD interne du document XML.

Note: Les espaces de noms n'ont pas de valeur sémantique dans les DTD, et les préfixes d'espaces de noms transférés depuis le Schéma XML deviennent une partie lexicale du nom de l'élément ou de l'attribut défini dans la DTD.

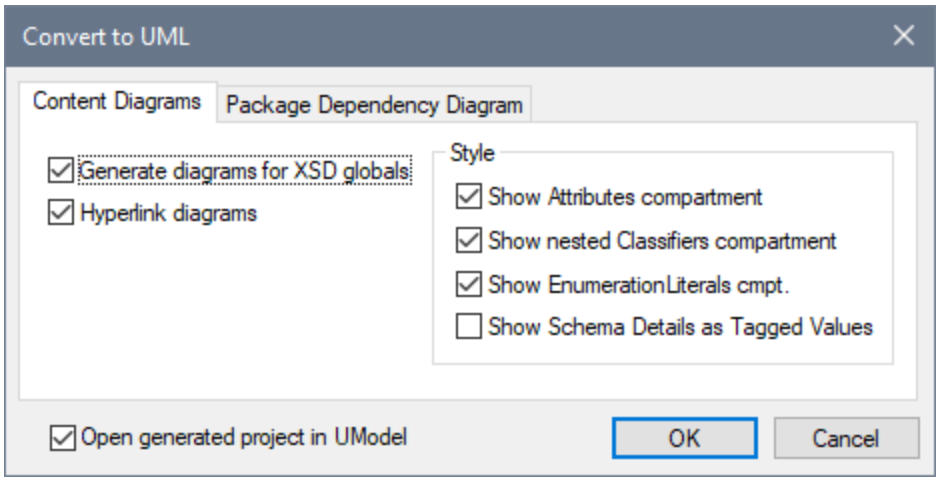
28.6.12 Convertir en UML

La commande **DTD/Schéma | Convertir en UML** convertit un Schéma XML W3C en un document Altova UModel Project (`.ump`) (ci-après nommé projet UModel). UMP est un format natif de Altova UModel, l'application de modélisation UML d'Altova. Les fichiers UMP peuvent ensuite être consultés et édités dans Altova UModel.

Pour convertir un schéma en UML, procédez comme suit:

1. Une fois le schéma ouvert, cliquez sur la commande **Convertir en UML**. Le dialogue Convertir en UML s'ouvre (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



2. Dans l'onglet Diagrammes de contenu, choisissez l'option Générer Diagrammes pour les Globaux XSD. Cela générera, dans le projet UModel, un diagramme de modèle de contenu pour chaque composant global.
3. Choisissez les options nécessaires provenant des options disponibles dans le dialogue. Ces options sont expliquées ci-dessous.
4. Si vous souhaitez immédiatement consulter le projet créé dans UModel, sélectionnez l'option pour ouvrir le projet dans UModel. Sinon, ne pas cocher cette option.
5. Cliquez sur **OK**.
6. Dans le dialogue Enregistrer sous qui apparaît, parcourez le dossier de destination et saisissez le nom du fichier UMP, puis cliquez sur **Enregistrer**.

Convertir en options UML

Les options suivantes sont disponibles dans le dialogue Convertir en UML.

Dans l'onglet ***Diagrammes de contenu*** :

- *Diagrammes hyperlien* crée un lien dans chaque diagramme à l'entrée de ce composant global dans le Mode Arborescence de modèle permettant ainsi de situer rapidement le composant dans la hiérarchie de schéma.
- Dans le volet *Style*, les options d'affichage des compartiments permettent d'afficher ou de cacher des compartiments variés.

Dans l'onglet ***Diagramme de paquetage de dépendance*** :

- L'option *Générer des diagrammes* détermine si un diagramme de dépendance de paquet est généré. Un diagramme de paquetage de dépendance propose un aperçu du paquet entier, montrant les relations des composants de paquet l'un par rapport à l'autre. Veuillez noter que les autres options dans cet onglet seront activées uniquement si l'option Générer diagramme est sélectionnée.
- Choisir l'option *Paquetage hyperlien vers diagramme* crée un lien depuis le diagramme de paquetage vers le Mode Arborescence de modèle.
- Quatre options sont disponibles pour la mise en page du diagramme de paquetage des dépendances : (i) mise en page non organisée (option Mise en page automatique non sélectionnée); (ii) mise en page hiérarchique (options Mise en page automatique et "hiérarchique" sélectionnées) ; iii) bloc (options Mise en page automatique et "bloc" sélectionnées) ; et (iv) réparti équitablement (options Mise en page

automatique et "force dirigée" sélectionnées). La mise en page peut être modifiée en éditant le diagramme dans UModel.

Note: la fonction Convertir en UML prend uniquement en charge le Schéma XML W3C.

28.6.13 Générer XML depuis BD, Excel, EDI avec MapForce

La commande **DTD/Schéma | Générer XML depuis BD, Excel, EDI avec MapForce** permet de lancer MapForce Altova si l'application est installée. MapForce vous permet de mapper un schéma sur une autre DTD, Schéma XML, ou une base de données et de générer un XML.

28.6.14 Créer sortie HTML/PDF/Word avec StyleVision...

La commande **DTD/Schéma | Créer sortie HTML/PDF dans StyleVision...** lance StyleVision Altova si l'application est installée. StyleVision vous permet de concevoir des feuilles de style pour une sortie HTML, PDF et RTF.

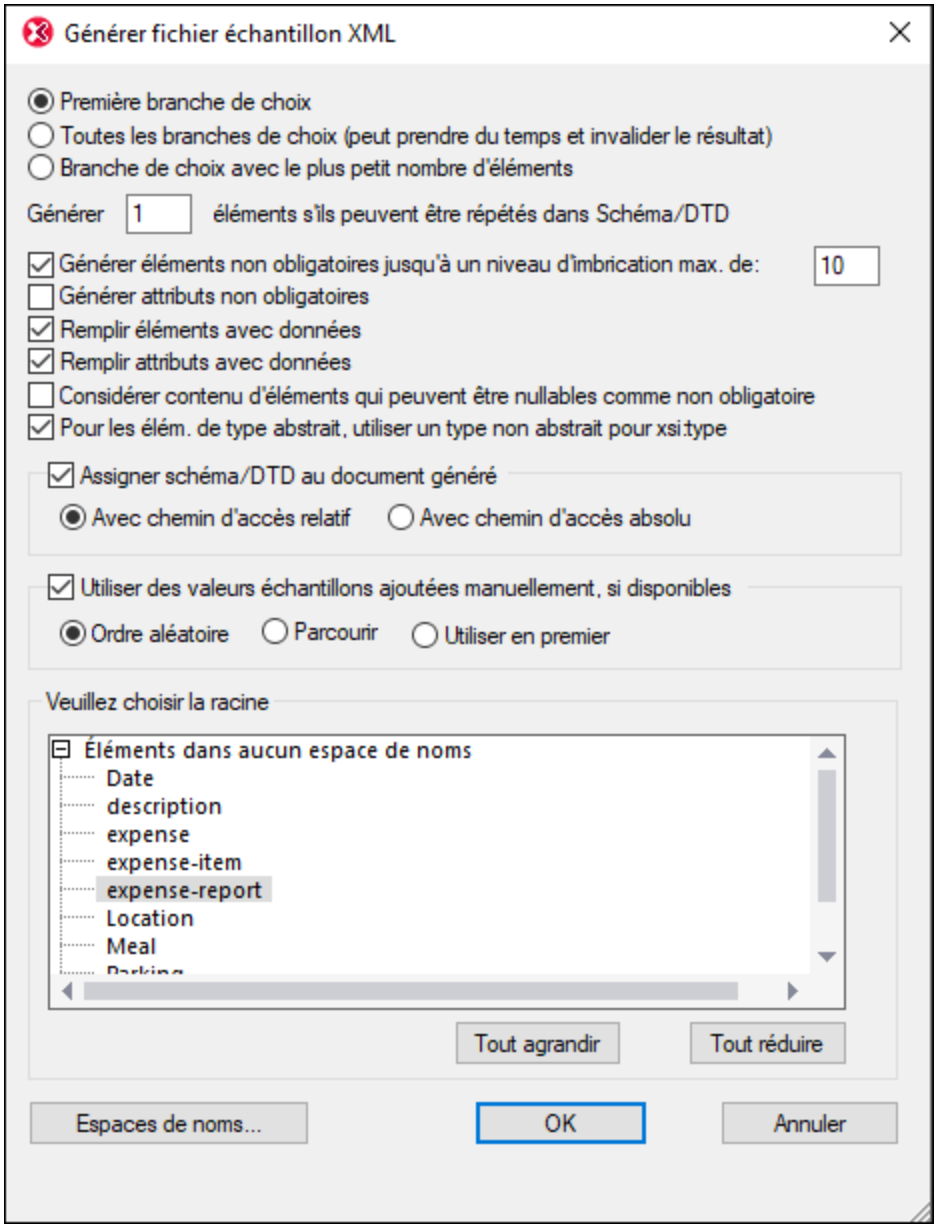
28.6.15 Générer fichier échantillon XML/JSON

La commande **Générer fichier échantillon XML/JSON** est activée dans le Mode Texte, Mode Grille et le Mode Schéma, et génère une instance XML ou JSON basée sur le fichier de schéma actuellement actif. Si le fichier actuellement actif est une DTD ou un Schéma XML, un fichier d'instance XML sera généré. Si le fichier actif actuellement est un schéma JSON, alors un fichier d'instance JSON est généré.

Générer des fichiers XML échantillon

Avec une DTD ou un Schéma XML actif, vous pouvez générer une instance XML échantillon basée sur le schéma. En cliquant sur la commande **Générer un fichier XML échantillon**, le dialogue Générer un fichier XML échantillon apparaît (*capture d'écran ci-dessous*), dans lequel vous pouvez spécifier les options pour la génération d'échantillon.

© 2016-2022 Altova GmbH



Groupes Éléments de choix

Un groupe de choix est un groupe d'éléments à partir duquel un élément peut être utilisé. Par exemple, si un élément appelé `items` est défini comme ayant un groupe `choice` consistant en les trois éléments : `cd`, `dvd`, `book`, alors `items` peut validement avoir un de ces trois éléments en tant qu'élément enfant (avec un nombre maximum d'occurrences tel que spécifié dans l'attribut `maxOccurs` de cet élément).

Dans le dialogue Générer un fichier XML échantillon, vous pouvez choisir si vous souhaitez générer (i) la première branche (élément) du groupe `choice`, (ii) toutes les branches, ou (iii) la branche avec le plus petit nombre d'éléments descendants. Veuillez noter que la sélection *Toutes les branches* pourrait générer un document invalide puisqu'une seule branche d'un groupe `choice` est autorisée.

Si une des branches du groupe `choice` est répétitive (donc qu'elle a une valeur `maxOccurs` plus grande que `1`), alors vous spécifiez, dans la première fenêtre à texte du dialogue, combien générer d'éléments répétitifs, jusqu'à un maximum de `99`. Si l'attribut `maxOccurs` du groupe `choice` est défini comme `unbounded` ou comme large nombre et *All branches* est sélectionné dans le dialogue Générer fichier XML échantillon, alors le gropue `maxOccurs` de `choice` est également limité au nombre d'éléments répétitifs que vous spécifiez dans la première zone de texte du dialogue.

Générer des éléments non obligatoires
Activer cette option pour générer aussi bien les éléments obligatoires et non obligatoires définis dans le schéma. Si activé, vous pouvez spécifier le niveau d'imbrication que vous souhaitez. Une profondeur d'imbrication plus importante permet aux éléments non obligatoires d'être générés jusqu'au niveau d'imbrication que vous spécifiez.

Générer des attributs non obligatoires
Activer cette option pour générer aussi bien les attributs obligatoires et non obligatoires définis dans le schéma.

Générer X éléments s'ils peuvent être répétés dans Schéma/DTD
Activer cette option génère le nombre d'éléments répétitifs que vous souhaitez saisir dans la zone de texte. Cela s'applique à tous les éléments, y compris à ceux se trouvant dans les groupes `choice`.

Remplir les éléments et attributs avec des données
Activer cette option pour insérer les valeurs de type de donnée des éléments et attributs respectifs. Par exemple, si un élément est défini comme étant un type de données pf `string`, alors l'élément est donné comme valeur factice de `string`.

Éléments annulables et types abstraits
Les contenus des éléments nillables peuvent être traités en tant que non obligatoires et les éléments avec un type abstrait peuvent utiliser un type non abstrait pour son attribut `xsi:type`.

Assigner schéma pour le fichier XML généré
Le schéma utilisé pour générer le fichier XML qui peut être assigné au fichier XML généré avec un chemin relatif ou absolu.

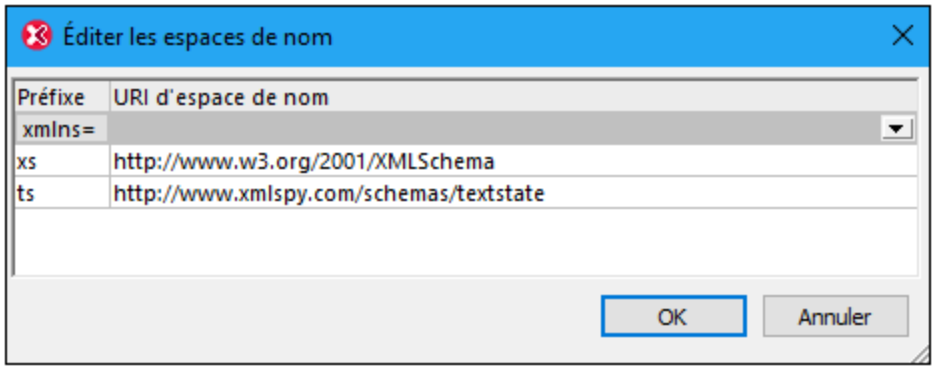
Utiliser des valeurs échantillon ajoutées manuellement, si disponibles
Si des valeurs d'échantillon sont assignées au composant de schéma, celles-ci seront utilisées en tant que la valeur ou le contenu de ce composant. Pour des composants individuels, des valeurs échantillon sont assignées dans l'Assistant à la saisie Facettes, dans l'onglet Échantillons. Vous pouvez spécifier quelle valeur provenant des valeurs d'échantillon disponibles est sélectionnée pour qu'une seule génération de fichier soit spécifiée :

- Une sélection aléatoire.
- Chaque valeur d'échantillon à son tour pour chaque instance du composant. Pour chaque génération de fichier, le cycle démarre une nouvelle fois.
- La première valeur toujours.

© 2016-2022 Altova GmbH

Élément racine
Si le schéma contient plus d'un élément global, ceux-ci sont énumérés et l'élément racine requis pour le fichier XML échantillon peut être sélectionné depuis la liste.

Espaces de noms
Cliquez sur la touche **Espaces de noms** pour ouvrir le dialogue Éditer espaces de noms (*voir la capture d'écran ci-dessous*), les espaces de noms qui sont définies dans le schéma, plus tout schéma XML standard, qui sont requis dans le fichier XML échantillon apparaîtront dans ce dialogue.

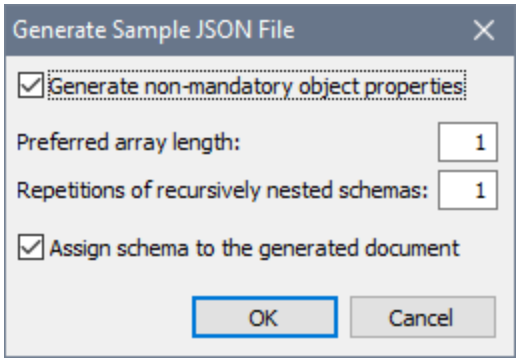


Vous pouvez éditer les éléments suivants :

- Le préfixe d'espace de noms qui est lié à un des espaces de noms du document. Les préfixes d'espace de noms qui sont définis dans ce dialogue seront utilisés (dans le fichier XML généré) aux nœuds de préfixe qui se trouvent dans l'espace de nom correspondant. Par exemple, la capture d'écran indique que les nœuds dans l'espace de noms `http://www.xmlspy.com/schemas/textstate` seront préfixés avec **`ts:`** dans le fichier d'échantillon.
- Vous pouvez déterminer un des espaces de noms du document en tant que l'espace de noms par défaut (`xmlns=`) en sélectionnant, dans la liste de choix `xmlns=` l'espace de noms que vous souhaitez. Les nœuds dans l'espace de noms qui sont sélectionnés comme espace de noms par défaut seront ensuite générés sans un préfixe d'espace de noms.

Générer des fichiers JSON échantillon

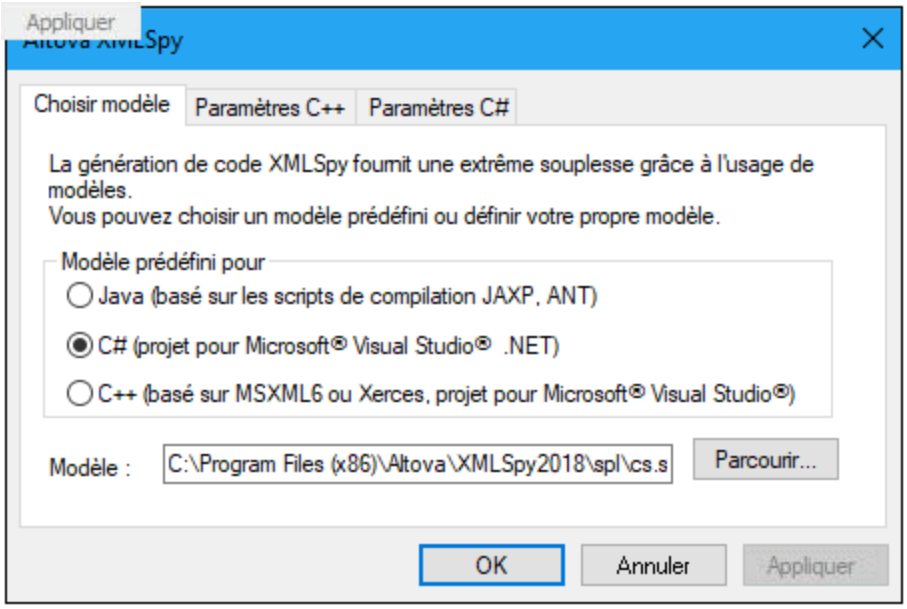
Avec un schéma JSON actif, vous pouvez générer une instance JSON échantillon sur la base du schéma JSON. Cliquez sur la commande pour faire apparaître le dialogue Générer fichier échantillon JSON (*capture d'écran ci-dessous*), dans lequel vous pouvez spécifier les options pour la génération d'échantillons.



Vous pouvez choisir soit de générer des propriétés d'objet non obligatoires, des propriétés de motif non obligatoires, la longueur des tableaux, et les répétitions de définitions récursives. Vous pouvez aussi spécifier si le schéma JSON actif doit être assigné automatiquement au fichier échantillon JSON généré ou pas.

28.6.16 Générer code de programme

La commande **DTD/Schéma | Générer code de programme...** génère automatiquement des fichiers de classe Java, C++ ou C# depuis des définitions dans le document de schéma actif (DTD ou Schéma XML).



Veuillez voir la section du Générateur de Code dans cette documentation pour plus de détails concernant la génération de code.

© 2016-2022 Altova GmbH

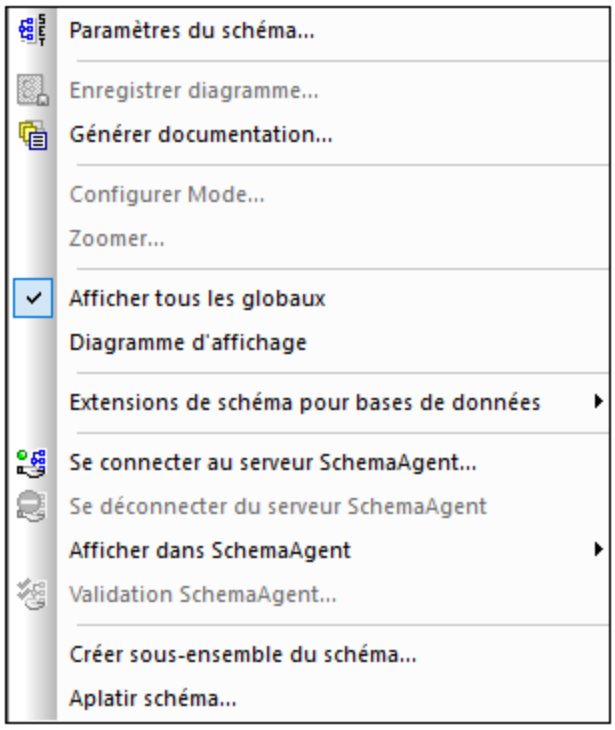
28.6.17 Vider cache mémoire

La commande **DTD/Schéma | Vider le cache mémoire** vide tous les documents de schéma mis sous cache (DTD et Schéma XML) provenant de la mémoire. Pour accélérer la validation et l'édition intelligente, XMLSpy met sous cache les documents de schéma utilisés récemment et les entités externes parsées dans la mémoire. Les informations provenant de ces documents cachés sont aussi affichées lorsque la commande Aller à la définition est invoquée.

Vider le cache de mémoire si vous n'avez pas suffisamment d'espace mémoire sur votre système, ou si vous avez récemment utilisé des documents basés sur des schémas différents.

28.7 Menu Conception de schéma

Le menu **Conception de schéma** vous permet de configurer le Mode Schéma de XMLSpy. Ce mode vous permet de concevoir des Schémas XML dans une GUI. Il est disponible lorsqu'un document de Schéma XML est actif dans le Mode Schéma.



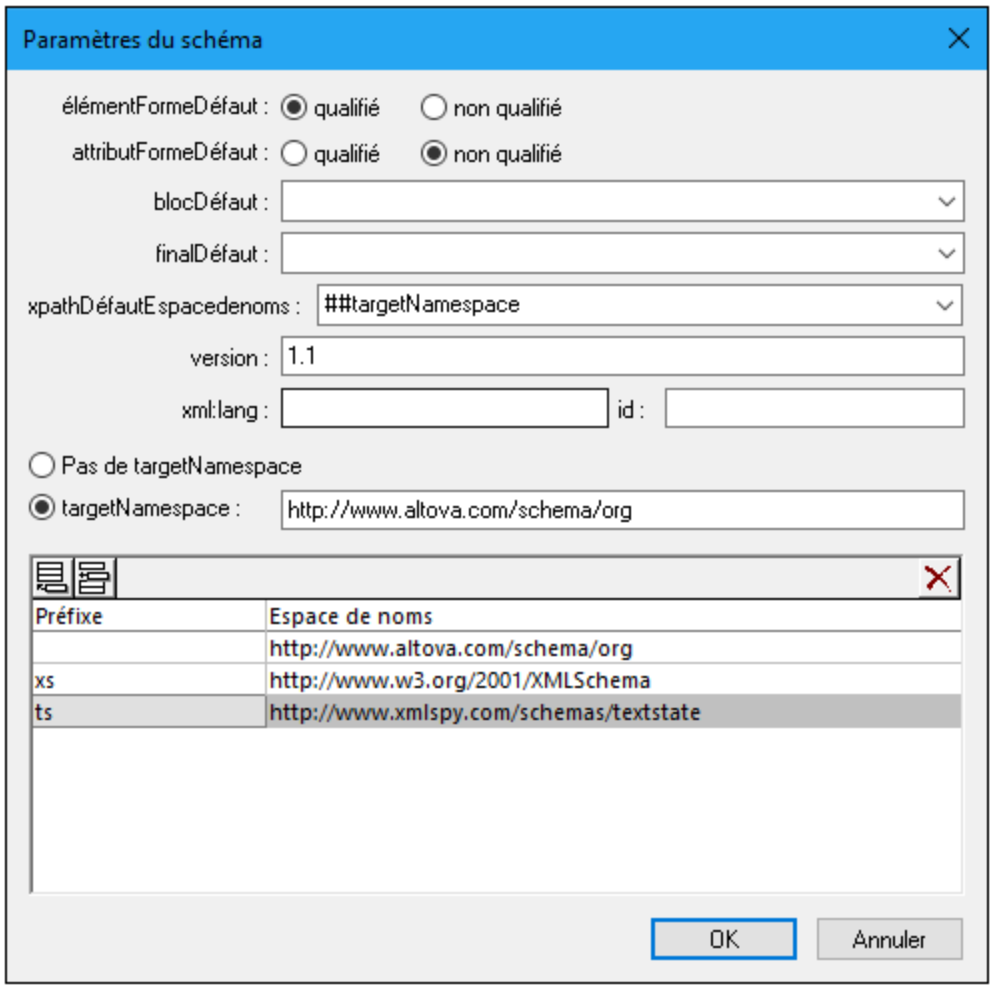
Les commandes disponibles dans ce menu sont décrites dans cette section.

28.7.1 Paramètres du schéma



La commande **Conception de schéma** | **Paramètres du schéma** est activée dans le Mode Schéma et vous permet de définir des paramètres globaux pour le schéma actif. Ces paramètres sont les attributs de l'élément `xs:schema`.

© 2016-2022 Altova GmbH



Les paramètres définis dans le dialogue Conception de schéma affichés ci-dessus (lorsque le mode XSD est défini à 1.1) créera l'élément `xs:schema` suivant.

```
<xs:schema xmlns="http://www.altova.com/schemas/org"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       targetNamespace="http://www.altova.com/schemas/org"
       elementFormDefault="qualified"
       attributeFormDefault="unqualified"
       xpathDefaultNamespace="##targetNamespace"
       version="1.1"
       defaultAttributes="Contact">
```

Note : Ce qui se trouve dans le dialogue de Paramètres du schéma changera conformément au mode XSD actif. Lorsque XSD 1.0 est le mode actif, les attributs XSD 1.1 ne sont pas présents dans le dialogue.

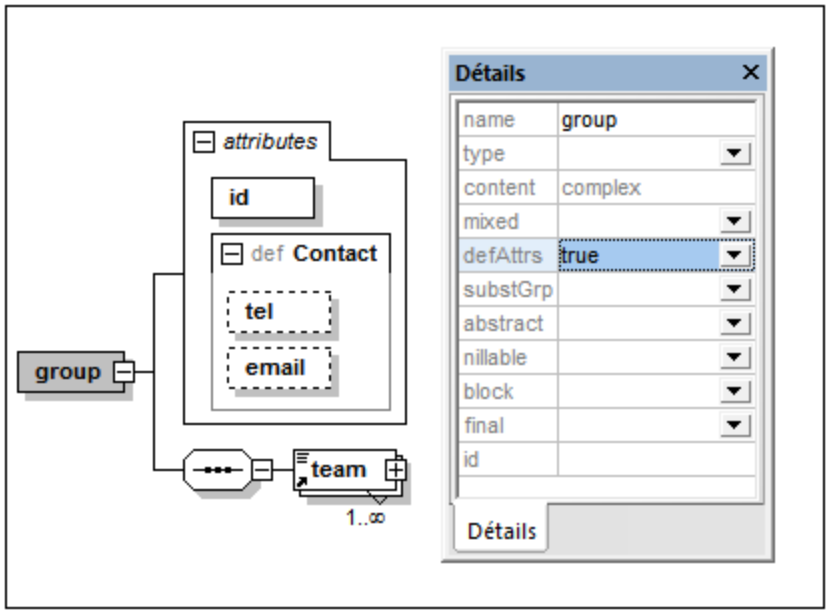
Les attributs `defaultAttributes` et `xpathDefaultNamespace` sont des fonctions Schéma XML 1.1 et seront uniquement disponibles dans le mode XSD 1.1. Les autres attributs sont disponibles aussi bien dans XSD 1.0 que XSD 1.1.

L'attribut **version**

L'attribut version est la version du document. Il ne s'agit pas de la version XSD du document.

L'attribut par défaut **defaultAttributes**

L'attribut defaultAttributes vous permet de choisir un groupe d'attributs en tant que le groupe d'attribut par défaut de tous les types complexes dans le schéma. Le groupe d'attribut par défaut est affiché dans le modèle de contenu de ces types complexes. Dans la capture d'écran ci-dessous, le groupe d'attribut `Contact` est le groupe d'attribut par défaut (*voir aussi capture d'écran ci-dessus, où il a été réglé*), et est disponible automatiquement sur l'élément `group`. Pour désactiver le groupe d'attribut, régler le `defaultAttributesApply` du type complexe sur `false`. Vous pouvez le faire par le biais de la propriété *defAttrs* dans l'Assistant à la saisie Détails du type complexe (*voir capture d'écran ci-dessous*).



L'attribut **xpathDefaultNamespace**

L'attribut `xpathDefaultNamespace` définit l'espace de noms par défaut pour les éléments dans les expressions XPath utilisées dans le schéma. Une fois défini dans le dialogue Paramètres du schéma, l'attribut s'applique à l'élément de niveau supérieur `xs:schema`. Donc l'étendue de la déclaration sera le document entier. Vous pouvez contourner la déclaration sur `xs:schema` avec des déclarations sur les éléments où l'attribut est autorisé :

- `xs:assert` and `xs:assertion`
- `xs:alternative`
- `xs:selector` and `xs:field` (in identity constraints)

Vous pouvez changer l'espace de noms par défaut XPath dans l'assistant à la saisie Détails des éléments cités ci-dessus.

L'attribut `xpathDefaultNamespace` peut avoir une des trois valeurs autorisées suivantes :

© 2016-2022 Altova GmbH

- `##targetNamespace`: L'espace de noms par défaut XPath sera le même que l'espace de noms cible du schéma
- `##defaultNamespace`: L'espace de noms par défaut XPath sera le même que l'espace de noms cible du schéma
- `##local`: Il n'y a pas d'espaces de noms XPath par défaut

Si aucun espace de noms par défaut XPath n'est déclaré dans le document, des éléments non préfixés dans les expressions XPath ne se trouveront dans aucun espace de noms. La déclaration d'espaces de noms XPath par défaut ne s'applique pas aux attributs.

28.7.2 Enregistrer diagramme



La commande **Conception de schéma | Emregistrer diagramme...** permet d'enregistrer le diagramme du Modèle de contenu affiché actuellement dans la fenêtre principale comme fichier d'image en format PNG à l'emplacement de votre choix.

28.7.3 Générer la documentation



La commande **Conception de schéma | Générer Documentation** génère une documentation détaillée de votre schéma XML ou JSON (*voir capture d'écran ci-dessous*) en format HTML, MS Word, RTF ou PDF. La documentation générée par cette commande peut être modifiée et utilisée à volonté ; une permission de la part d'Altova n'est pas nécessaire. La documentation est générée pour les composants que vous avez sélectionnés dans le dialogue Documentation de Schéma (JSON) (qui apparaît lorsque vous sélectionnez la commande **Générer Documentation**). Les éléments liés (éléments enfants, types complexes, etc.) sont généralement liés par hyperlien dans la sortie sur écran, vous permettant de naviguer de composant en composant. Les composants avec un modèle de contenu ont donc des liens vers les définitions de modèle de contenu. Veuillez noter que la documentation de schéma est aussi générée pour les **composants de schéma inclus et importés**. Les options de génération-documentation variées pour le schéma XML sont décrites dans la section Options de Documentation. Les options de documentation de schéma JSON sont décrites dans la section Générer une Documentation de schéma JSON.

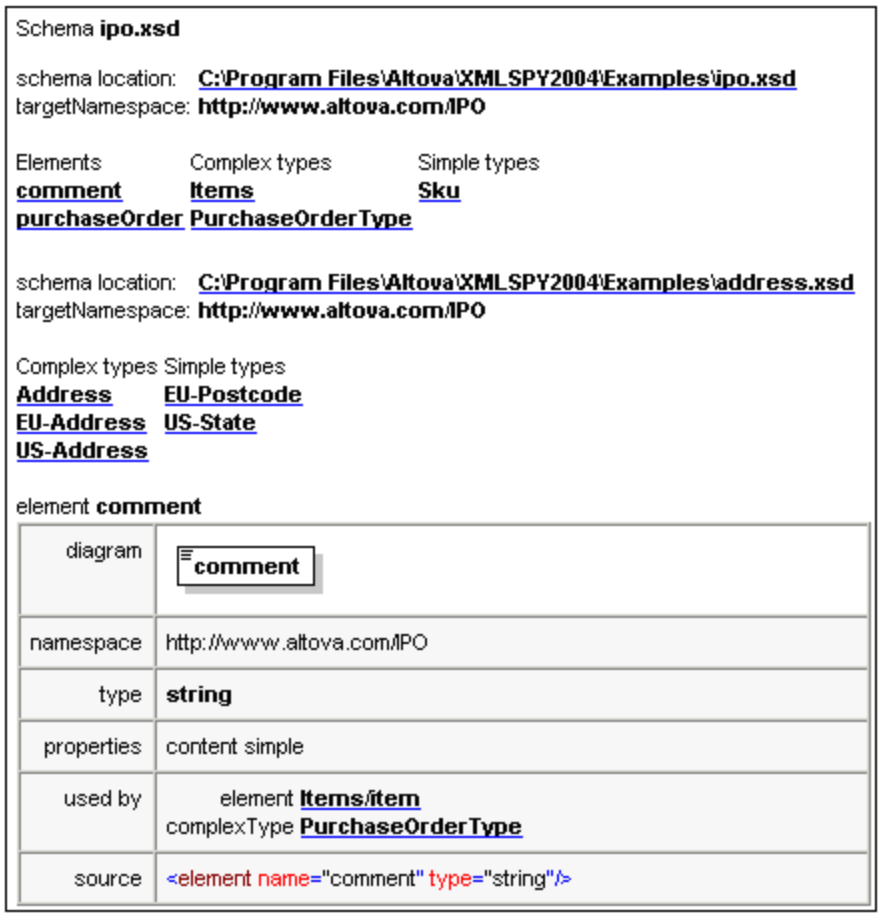
Veuillez noter que les Options de Documentation sont appliquées en plus des paramètres que vous spécifiez dans le dialogue de Configuration d'affichage de schéma.

Note : Afin de générer une documentation en format MS Word, vous devez installer MS Word (version 2000 ou plus).

Vous pouvez soit utiliser le design standard fixe de XMLSpy pour le document généré ou vous pouvez utiliser une SPS StyleVision pour le design. L'utilisation d'une SPS StyleVision vous permet de personnaliser le

design de la documentation générée ainsi que de générer un PDF en tant que format de sortie supplémentaire. Le travail avec une SPS est expliqué dans la section, Design défini par l'utilisateur.

Note : Pour pouvoir utiliser une SPS pour générer une documentation de schéma, vous devez installer StyleVision sur votre appareil.


Schema **ipo.xsd**

schema location: **C:\Program Files\Altova\XMLSPY2004\Examples\ipo.xsd**
targetNamespace: **http://www.altova.com/IPO**

Elements	Complex types	Simple types
comment	**Items**	**Sku**
purchaseOrder	**PurchaseOrderType**	

schema location: **C:\Program Files\Altova\XMLSPY2004\Examples\address.xsd**
targetNamespace: **http://www.altova.com/IPO**

Complex types	Simple types
Address	**EU-Postcode**
EU-Address	**US-State**
US-Address	

element **comment**

diagram	comment
namespace	http://www.altova.com/IPO
type	**string**
properties	content simple
used by	element **Items/item** complexType **PurchaseOrderType**
source	<element name="comment" type="string"/>

La capture d'écran ci-dessus montre la documentation de schéma générée avec un index (tous les schémas liés avec leurs composants globaux organisés par type de composant) en haut du document.

Note : Lors de la génération de documentation pour des documents de schéma W3C, XMLSpy utilise des versions internes d'application de ces documents. Par conséquent, d'autres emplacement de ces documents ne sont pas pris en compte et les redéfinitions et autres modifications de schéma ne seront pas reflétées dans la documentation.

28.7.3.1 Options de documentation

La commande **XBRL | Générer Documentation** génère une documentation détaillée de la taxonomie XBRL actuelle : XML Schéma ou JSON Schéma. Cette section décrit la génération de la documentation de Schéma XML. La procédure de génération de la documentation de schéma JSON est semblable. Pour plus de détails

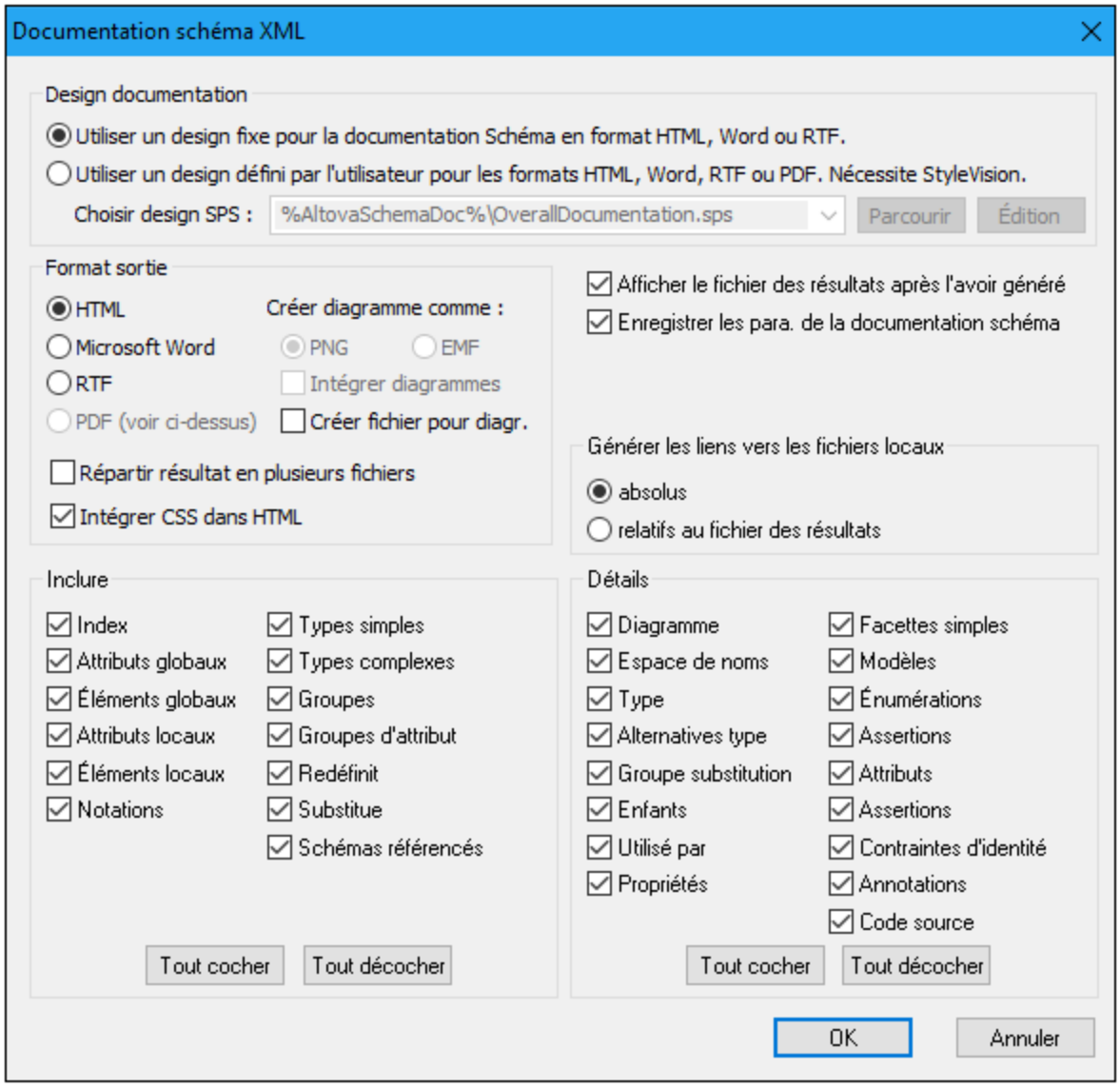
© 2016-2022 Altova GmbH

sur comment générer une documentation JSON Schéma et une description des paramètres de génération de la documentation, voir la section Générer une documentation JSON Schéma.

Générer une documentation de Schéma XML

Lorsqu'un document de Schéma XML est actif et que vous cliquez sur la commande **Générer documentation**, le dialogue Documentation de schéma (*capture d'écran ci-dessous*) s'affiche. Dans ce dialogue, vous pouvez sélectionner des options pour la documentation.

Dans le volet Design de documentation du dilaogue, vous pouvez sélectionner si vous utilisez le design fixe XMLSpy pour la documentation générée ou si vous voulez utiliser un design personnalisé créé dans une SPS StyleVision. Choisir l'option que vous souhaitez. Veuillez noter que la sortie PDF est disponible uniquement pour une documentation générée avec une SPS StyleVision, et non pas pour une documentation générée à l'aide d'un design fixe. Voir la section Design défini par l'utilisateur pour découvrir comment travailler avec une SPS.



Les autres options dans le dialogue de Documentation de schéma sont expliquées ci-dessous :

- Le format requis est spécifié dans le panneau de format de sortie : soit HTML, Microsoft Word, RTF ou PDF. (Le format de sortie PDF est uniquement disponible si vous utilisez une SPS StyleVision pour générer la documentation.) En cliquant sur **OK**, vous serez invité à saisir le nom du fichier de sortie et l'emplacement où il a dû être enregistré.
- Les documents Microsoft Word sont créés avec l'extension de fichier `.doc` lorsqu'ils sont générés à l'aide d'un design fixe, et avec une extension de fichier `.docx` lorsqu'ils sont générés avec une SPS de StyleVision.
- La documentation peut être générée soit en tant que fichier unique soit entre plusieurs fichiers. Si plusieurs fichiers sont générés, chaque fichier correspond à un composant. Les composants inclus dans la sortie sont spécifiés à l'aide de cases à cocher dans le volet Inclure. Dans les designs fixes, les liens entre les documents multiples sont créés automatiquement.
- En ce qui concerne la sortie HTML, les définitions de style CSS peuvent être sauvegardées soit dans un fichier CSS séparé soit intégrées dans le fichier HTML (dans l'élément `<head>`). Si un fichier CSS séparé est créé, il recevra le même nom que le fichier HTML, mais aura une extension `.css`. Cocher ou décocher la case *Intégrer CSS dans HTML*pour définir l'option requise.
- L'option *Intégrer Diagrammes* est activée pour les options de sortie MS Word et PDF. Lorsque cette option est cochée, les diagrammes sont intégrés dans le fichier de résultat, soit dans un format PNG ou EMF. Sinon, les diagrammes sont créés en tant que fichiers PNG ou EMF, qui sont affichés dans le fichier de résultat par des liens d'objet.
- Lorsque la sortie est HTML, tous les diagrammes sont créés en tant que fichier PNG externes. Si la case à cocher *Créer dossier pour les diagrammes* est cochée, un dossier sera créé dans le même dossier que le fichier HTML, et les fichiers PNG seront enregistrés à l'intérieur. Ce dossier aura un nom du format *`HTMLFilename`*`_diagrams`. Si la case *Créer dossier pour les diagrammes* n'est pas cochée, les fichiers PNG seront enregistrées dans le même dossier que le fichier HTML.
- Les liens vers les fichiers locaux (comme les fichiers d'image de diagramme et le fichier externe CSS) peuvent être relatifs ou absolus. Dans le volet *Générer liens vers des fichiers locaux*, choisir le bouton radio approprié conformément à l'option que vous préférez.
- Dans le volet Include, vous choisissez les types d'items que vous souhaitez inclure dans la documentation. Chaque item des types sélectionnés sera affiché dans la documentation générée. Par exemple, si *Attributs locaux* est coché, alors la description de chaque attribut local sera affichée en tant qu'entrée séparée. L'option Index liste tous les schémas liés en haut du fichier, et leurs composants globaux sont organisés par types de composant. Les touches **Tout cocher** et **Tout décocher** vous permettent de sélectionner et désélectionner rapidement toutes les options du volet. Veuillez noter que l'option *Include* n'affecte pas l'affichage d'un type d'item dans le cadre des définitions graphiques. Cet affichage est contrôlé par les paramètres que vous effectuez dans le dialogue Configuration d'affichage de schéma. Donc si vous souhaitez désactiver l'affichage des attributs dans le cadre de la représentation graphique d'un item de schéma, décocher l'option Attributs dans le dialogue Configuration d'affichage de schéma.
- Le volet Détails liste les détails qui peuvent être inclus pour chaque composant. Sélectionner les détails que vous souhaitez inclure dans la documentation. Les touches **Tout cocher** et **Tout décocher** vous permettent de sélectionner et désélectionner rapidement toutes les options du volet.
- L'option *Afficher le fichier de résultat* est activée pour toutes les options de sortie. Lorsque cette option est cochée, les fichiers de résultat sont affichés dans le Mode Navigateur (sortie HTML), MS Word (sortie MS Word), et les applications par défaut pour les fichiers `.rtf` (sortie RTF) et les fichiers `.pdf` (sortie PDF).

Valeurs de paramètre

Si la SPS StyleVision contient une ou plusieurs définitions de paramètres, en cliquant sur **OK**, un dialogue s'ouvre listant tous les paramètres définis dans la SPS. Vous pouvez saisir des valeurs de paramètre dans ce dialogue pour contourner les valeurs de paramètre par défaut qui ont été assignées dans la SPS.

© 2016-2022 Altova GmbH

28.7.3.2 Design défini par l'utilisateur

À la place du standard fixe du design XMLSpy, vous pouvez créer un design personnalisé pour la documentation de schéma. Le design personnalisé est créé dans une SPS StyleVision qui est un modèle de design pour le document de sortie.

Créer la SPS

Une StyleVision Power Stylesheet (ou SPS) est créée à l'aide d'un produit StyleVision d'Altova. Une SPS pour la génération de la documentation de schéma doit être basée sur un Schéma XML qui spécifie la structure de la documentation de schéma. Ce schéma est appelé `SchemaDocumentation.xsd`, et est fourni avec votre package XMLSpy. Il est stocké dans le dossier : `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Documentation\Schema`.

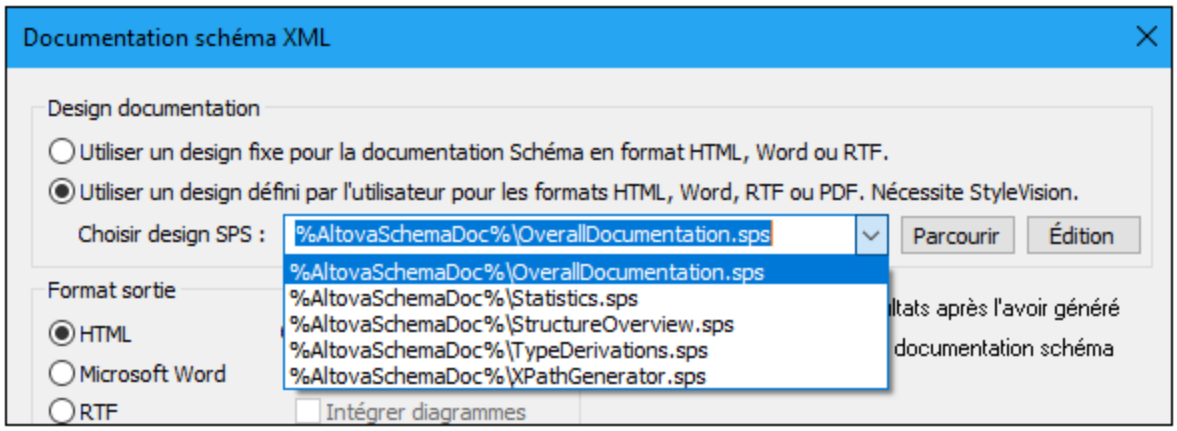
Lors de la création du design SPS dans StyleVision, les nœuds du schéma `SchemaDocumentation.xsd` sont placés dans le modèle de design et des styles et propriétés leur sont assignés. Des composants additionnels, tels que des liens, tables et images peuvent aussi être ajoutés au design SPS. De cette manière, le document de sortie complet peut être conçu dans la SPS. Voir le manuel d'utilisation StyleVision pour consulter des explications détaillées de la création d'un design SPS dans StyleVision.

L'avantage d'utiliser une SPS pour générer une documentation de schéma est que vous détenez le contrôle sur le design de documentation du schéma. Veuillez aussi noter qu'une sortie PDF de la documentation schéma est uniquement disponible si une SPS définie par l'utilisateur est utilisée ; la sortie PDF n'est pas disponible si le design fixe de XMLSpy est utilisé.

Spécifier la SPS pour utiliser la documentation de schéma

Une fois qu'une SPS a été créée, elle peut être utilisée pour générer une documentation de schéma. La SPS que vous souhaitez utiliser pour générer la documentation WSDL est sélectionnée dans le dialogue de Documentation Schéma (accédé par le biais de la commande **Design Schéma | Générer documentation**). Dans le volet de design de la documentation de ce dialogue (*voir la capture d'écran ci-dessous*), choisir le bouton radio *Utiliser un design défini par l'utilisateur*. Vous pouvez ensuite cliquer sur la touche **Parcourir** et chercher la SPS que vous souhaitez. Cliquez sur la touche **OK** du dialogue et, dans le dialogue Enregistré qui s'ouvre, sélectionnez le dossier pour et saisissez le nom du fichier de sortie.

Note : le fichier SPS doit situer correctement le schéma sur lequel il est basé : `SchemaDocumentation.xsd` (*voir ci-dessus*).



Les designs SPS éditables suivants pour la génération de documentation de schéma sont fournis avec XMLSpy. Ils se trouvent dans le sous-dossier `Altova\XMLSpy2023\Documentation\Schema\` du dossier (Mes Documents).

- `OverallDocumentation.sps`, qui génère une documentation complète sur le schéma
- `Statistics.sps`, qui liste le nombre d'éléments globaux et locaux, des attributs et des groupes d'attributs, et des types simples et complexes pour le schéma principal et pour chaque fichier de schéma indépendamment.
- `StructureOverview.sps`, qui crée une sortie de la structure d'éléments globaux et de types complexes jusqu'à une profondeur configurable
- `TypeDerivations.sps`, qui liste les types simples et complexes et tous leurs types dérivés directement et indirectement sous la forme d'une arborescence
- `XPathGenerator.sps`, qui génère toutes les déclarations XPath possibles jusqu'à une profondeur configurable

Ces fichiers, avec d'autres fichiers SPS que vous avez récemment parcourus, seront disponibles dans la liste de choix de l'option *Utiliser un design défini par l'utilisateur* (*voir la capture d'écran ci-dessus*).

Cliquer sur la touche **Éditer** dans le volet Design de Documentation pour lancer StyleVision et ouvrir la SPS sélectionnée dans une fenêtre StyleVision. Afin de préconsulter le document de résultat dans StyleVision, vous nécessiterez un fichier de travail XML. Un fichier XML échantillon a déjà été attribué aux designs SPS listés ci-dessus `Sample.xml`, qui est situé dans le dossier (Mes Documents), dans le sous-dossier suivant :

```
Altova\XMLSpy2023\Documentation\Schema\SampleData
```

Note : afin d'utiliser une SPS pour générer une documentation de schéma, vous devez avoir installé StyleVision sur votre appareil.

28.7.4 Configurer mode

La commande **Conception de schéma | Configurer mode** est active dans le Mode Modèle de contenu et vous permet de configurer le Mode Modèle de contenu. Cliquer sur la commande pour ouvrir le dialogue Configuration du mode schéma situé en bas à droite de la fenêtre XMLSpy, vous permettant de voir les effets de vos paramètres alors que vous les saisissez dans le dialogue. Les paramètres prennent effet lorsque vous cliquez sur la touche **OK** du dialogue, et que vous les appliquez au Mode de modèle de contenu de tous les

© 2016-2022 Altova GmbH

fichiers de Schéma XML qui sont ouverts par la suite. Ces paramètres s'appliquent aussi à la sortie de documentation de schéma et à la sortie impression. Par exemple, si vous souhaitez désactiver l'affichage des attributs dans le cadre de la représentation graphique d'un item de schéma dans la sortie de documentation de schéma, décocher l'option *Attributs* dans le dialogue de Configuration de l'affichage de schéma (*capture d'écran ci-dessous*).

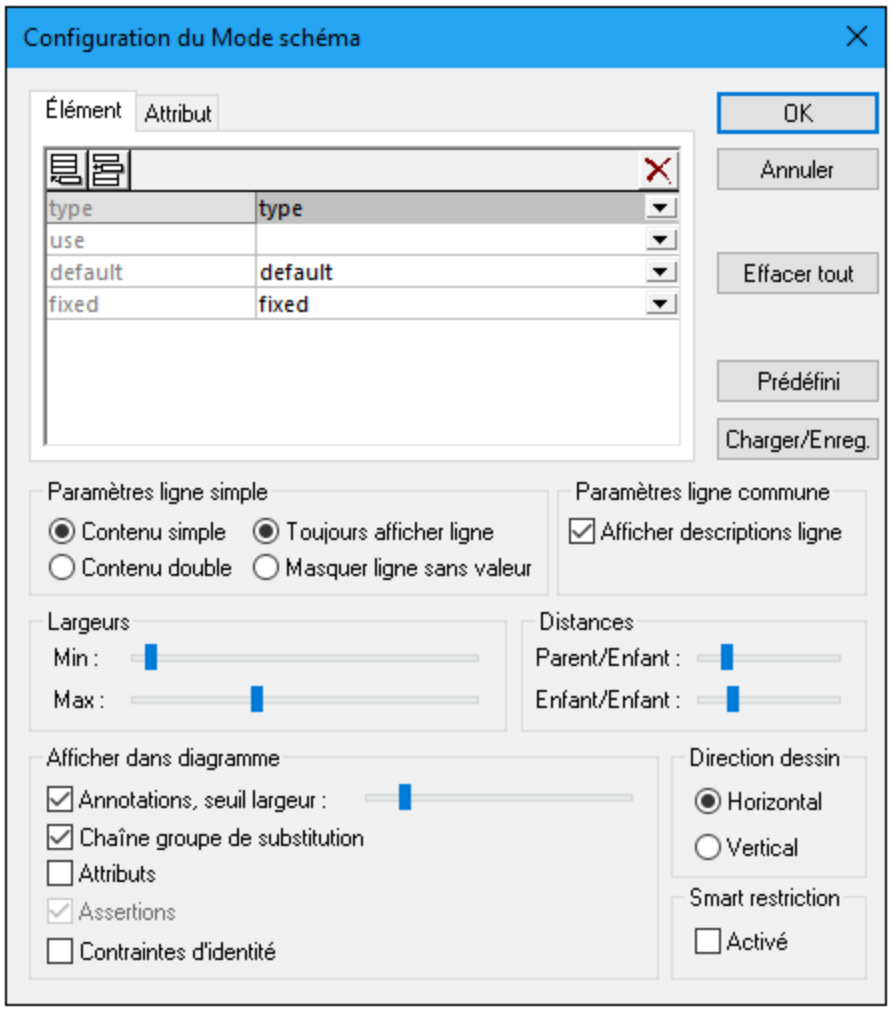
Note: Pour une description de la manière dont configurer le Mode de Design de schéma JSON, voir la section Configurer le Mode de conception de schéma JSON.

Définir les lignes de description de la propriété pour le modèle de contenu

Vous pouvez définir quelles propriétés des éléments et attributs sont affichées dans le Mode de modèle de contenu. Ces propriétés apparaissent en tant que lignes de grille dans les champs de composant.

Pour définir des lignes de description de la propriété :

1. Choisir **Conception de schéma | Configurer mode**. Le dialogue de configuration de l'affichage de schéma apparaît.
2. Dans l'onglet **Élément** ou **Attribut**, cliquer sur la l'icône Ajouter  ou Insérer  pour ajouter une ligne de description de la propriété. La ligne est ajoutée dans le dialogue et dans les champs d'élément dans le Mode de modèle de contenu.
3. À partir de la liste de choix, sélectionner la propriété que vous souhaitez afficher. *Voir capture d'écran.*
4. Renouveler les étapes 2 et 3 pour autant de propriétés que vous le souhaitez.



Le Mode de modèle de contenu est mis à jour et montre les lignes de description de la propriété définie pour tous les éléments pour lesquels ils existent.

Note : En ce qui concerne les attributs, la configuration que vous définissez apparaît uniquement lorsque des attributs sont affichés dans le diagramme, (par opposition à leur affichage dans un volet situé en-dessous du Mode de modèle de contenu). Le mode configuré s'applique à tous les Modes de modèle de contenu ouverts après la définition de la configuration.

Supprimer une ligne de description de propriété depuis le Mode de Modèle de contenu

Pour supprimer des lignes de description de propriété individuelles, dans le dialogue de Configuration d'affichage du schéma, sélectionner la ligne de description de propriété que vous souhaitez supprimer et cliquer sur l'icône Supprimer .

© 2016-2022 Altova GmbH

Paramètres de configuration du Mode de Modèle de contenu

Le Mode de modèle de contenu peut être configuré avec les paramètres contenus dans le dialogue de Configuration d'affichage du schéma. Vous trouverez une description de la manière dont définir quelles lignes de description de la propriété sont affichées dans le Mode de modèle de contenu ci-dessus. Les autres paramètres sont décrits ci-dessous.

Paramètres ligne simple
Vous pouvez définir si une ligne de description de la propriété doit contenir un contenu unique ou double et si des lignes individuelles doivent apparaître pour chaque élément ou uniquement pour les éléments qui contiennent cette propriété. Utiliser des boutons radio appropriés pour définir vos paramètres. Veuillez noter que ces deux paramètres peuvent être définis pour des lignes individuelles séparément (choisir la ligne requise et effectuer le paramètre).

Paramètres ligne commune
Cette option permet d'activer et de désactiver les descriptions de ligne (c.à.d. le nom de la propriété).

Largeurs
Ces coulisses permettent de définir la taille minimum et maximum des rectangles d'élément dans le Mode de modèle de contenu. Changer la taille si le texte de description n'est pas entièrement visible ou si vous souhaitez standardiser votre affichage.

Distances
Ces coulisses permettent de définir les distances horizontales et verticales entre les différents éléments sur l'écran.

Afficher dans diagramme
Les cases à cocher Annotations activent et désactivent l'affichage des annotations, ainsi que la largeur d'annotation du texte au moyen de la coulisse. Vous pouvez aussi activer et désactiver l'affichage des groupes de substitution. Les Attributs et les Contraintes d'identité apparaissent dans le diagramme de Modèle de contenu si leur case à cocher sont sélectionnées ; sinon, elles apparaissent en tant qu'onglets dans un volet de la fenêtre du Modèle de contenu.

Direction dessin
Ces options définissent l'orientation horizontale ou verticale de l'arborescence de l'élément sur l'écran.

Éditer le modèle de contenu dans le diagramme lui-même
Vous pouvez changer les propriétés d'élément directement dans le diagramme du modèle de contenu. Pour ce faire, double-cliquer la propriété que vous souhaitez éditer et commencez à saisir des données. Si une sélection est disponible, une liste déroulante apparaît à partir de laquelle vous pourrez choisir une option. Sinon, saisir une valeur et confirmer avec **Entrée**.

Boutons dans le dialogue de configuration de l'affichage de schéma
Ce dialogue présente les boutons suivants :

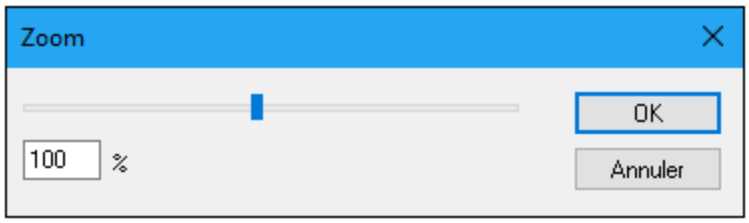
- La touche **Charger/Enregistrer** vous permet de charger et d'enregistrer les paramètres que vous effectuez ici.
- La touche **Prédéfini** remet la configuration d'affichage aux valeurs par défaut.
- La touche **Effacer tout** vide la fenêtre de toutes les entrées de liste.

Activer les smart restrictions

Pour activer les smart restrictions, cocher la case Activer Restrictions de schéma.

28.7.5 Zoomer

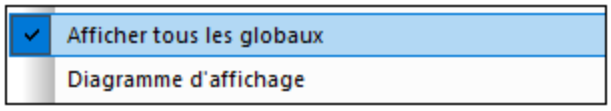
La commande **Conception de schéma | Zoom** contrôle le facteur de zoom du Mode Modèle de contenu. Cette fonction est utile si vous disposez d'un grand modèle de contenu et que vous souhaitez effectuer un zoom arrière afin que l'ensemble du modèle de contenu entre dans la Fenêtre principale. Vous pouvez zoomer à une taille entre 10% et 200% de la taille réelle.



Pour zoomer en avant et en arrière, faire glisser la coulisse ou cliquer dans le champ de saisie et indiquer un pourcentage.

28.7.6 Afficher tous les Globaux

La commande **Conception de schéma | Afficher tous les globaux** passe du Mode de Modèle de contenu à l'Aperçu de schéma pour afficher tous les composants globaux dans le schéma. Cette commande alterne avec la commande Afficher le diagramme. La fonction actuellement sélectionnée est indiquée avec une coche dans la partie gauche (*voir capture d'écran*).

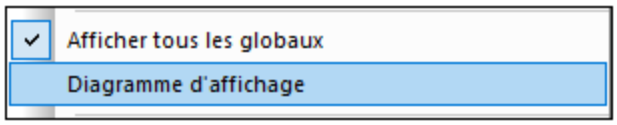


En alternative, vous pourriez utiliser l'icône **Afficher tous les globaux**  située en haut de Mode Modèle de contenu pour passer à l'Aperçu de schéma.

28.7.7 Afficher le diagramme

La commande **Conception de schéma | Afficher le diagramme** passe au Mode de Modèle de contenu du composant global sélectionné, si le composant sélectionné a un modèle de contenu. Les composants globaux qui ont un modèle de contenu (types complexes, éléments et groupes d'élément) sont indiqués par l'icône  à leur gauche. La commande Afficher diagramme alterne avec la commande Afficher tous les globaux. La commande activée actuellement est indiquée par une coche à sa gauche (*capture d'écran ci-dessous*).

 © 2016-2022 Altova GmbH



En alternative, vous pouvez utiliser les méthodes suivantes pour passer au Mode Modèle de contenu :

- Cliquer sur l'icône à côté du composant, le modèle de contenu que vous souhaitez afficher.
- Double-cliquer sur le nom d'un composant dans l'Assistant à la saisie Navigateur de composant (en haut à droite).

28.7.8 Extensions de schéma pour les bases de données

Ce point de menu ouvre un sous-menu contenant des commandes pour les extensions de schéma Oracle et MS SQL Server.

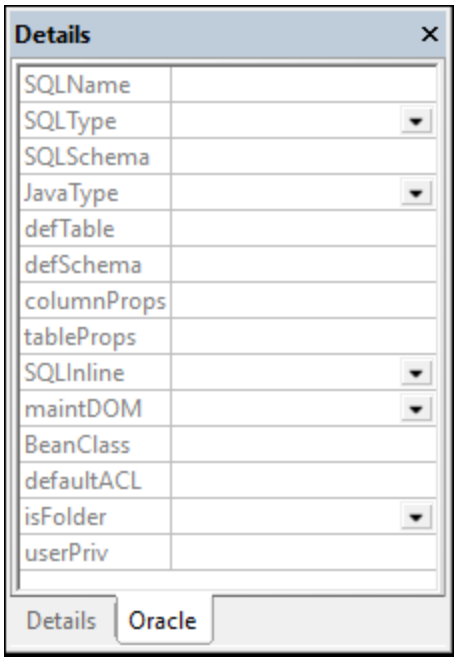
- Activer les extensions de schéma Oracle
- Paramètres de schéma Oracle
- Activer les extensions de schéma Microsoft SQL Server
- Relation de schéma nommées
- Relations d'éléments non-nommés

28.7.8.1 Activer les extensions de schéma Oracle

XMLSpy prend en charge les extensions de schéma Oracle à utiliser avec Oracle 9i Project XDB. Ces extensions de schéma vous permettent de configurer et de personnaliser la manière dont Oracle 9i Project XDB stocke les documents XML. Ceux-ci sont ensuite accessibles par le biais de requêtes SQL et d'outils de legacy. Veuillez consulter le site Web Oracle pour plus d'informations.

Lorsque vous sélectionnez la commande **Activer les extensions de Schéma Oracle**, la chose suivante se produit :

- L'espace de noms XDB est déclaré dans l'élément `schema` : `xmlns:xdb="http://xmlns.oracle.com/xdb"`.
- Un onglet Oracle est créé dans l'Assistant à la saisie Détails, vous permettant d'ajouter des attributs, y compris des attributs spécifiques à XDB aux éléments de schéma comme `xsd:complexType` et `xsd:element`.

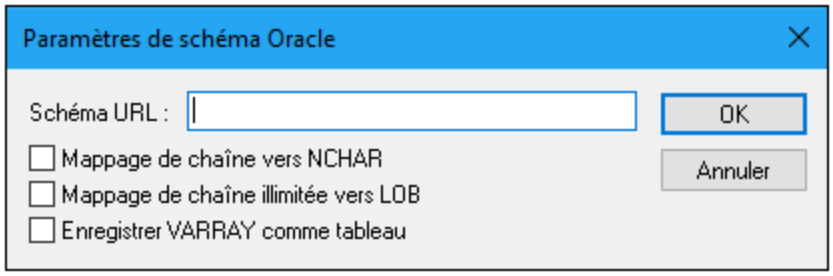


Les extensions Oracle peuvent être définies pour des types complexes, des éléments, des attributs. Utiliser l'Assistant à la saisie comme vous le feriez dans XMLSpy.

Note : Les extensions peuvent être activées ou désactivées. Lorsque des extensions sont activées, la commande est affichée avec une coche à sa gauche. La désactivation des extensions supprime la déclaration d'espace de noms XDB et toutes les extensions XDB dans le fichier. Un message d'avertissement apparaît car cette action ne peut pas être annulée.

28.7.8.2 Paramètres de schéma Oracle

La commande **Paramètres de schéma Oracle** vous permet de définir les paramètres globaux pour les extensions de schéma Oracle.



Afin d'accéder à ce dialogue, les extensions de schéma Oracle doivent être activées (avec la commande Activer les extensions de schéma Oracle).

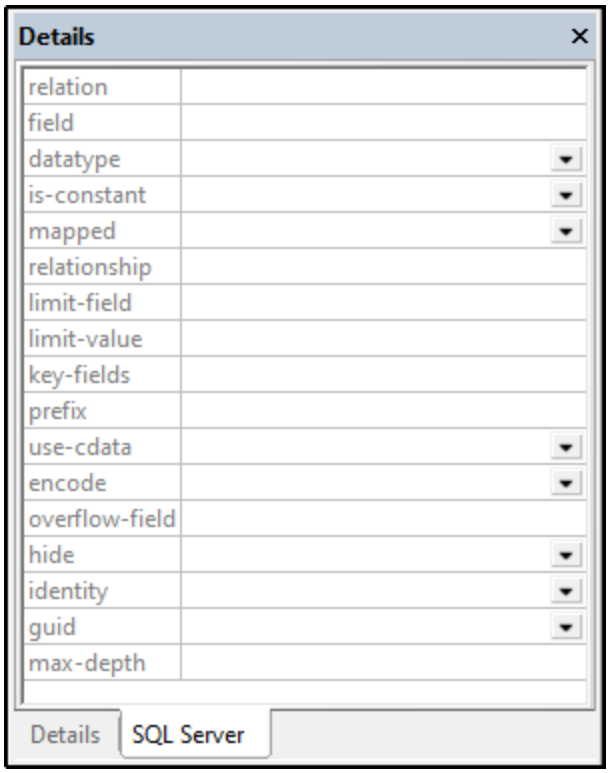
© 2016-2022 Altova GmbH

28.7.8.3 Activer les extensions de schéma Microsoft SQL Server

1XMLSpy prend en charge les extensions de schéma pour Microsoft SQL Server 2000 pour une utilisation avec Microsoft SQL Server. Ces extensions de schéma vous permettent de configurer et de personnaliser la manière dont Microsoft SQL Server stocke les documents XML. Ces documents XML sont ensuite accessibles par le biais de requêtes SQL et d'outils de legacy. Veuillez consulter le site Microsoft pour plus d'informations.

Lorsque vous choisissez la commande **Activer extension de schéma du serveur Microsoft SQL** :

- L'espace de noms du serveur SQL est déclaré sur l'élément `schema` : `xmlns:sql="urn:schemas-microsoft-com:mapping-schema"`.
- Un onglet de serveur SQL est créé dans l'Assistant à la saisie Détails, vous permettant d'ajouter des attributs aux éléments de schéma comme `xsd:element`.



Lorsque les extensions de serveur SQL peuvent être définies pour un composant de schéma, l'onglet de serveur SQL est disponible dans l'Assistant à la saisie Détails au moment de la sélection du composant. Utiliser l'Assistant à la saisie tel que vous l'utiliseriez normalement dans XMLSpy.

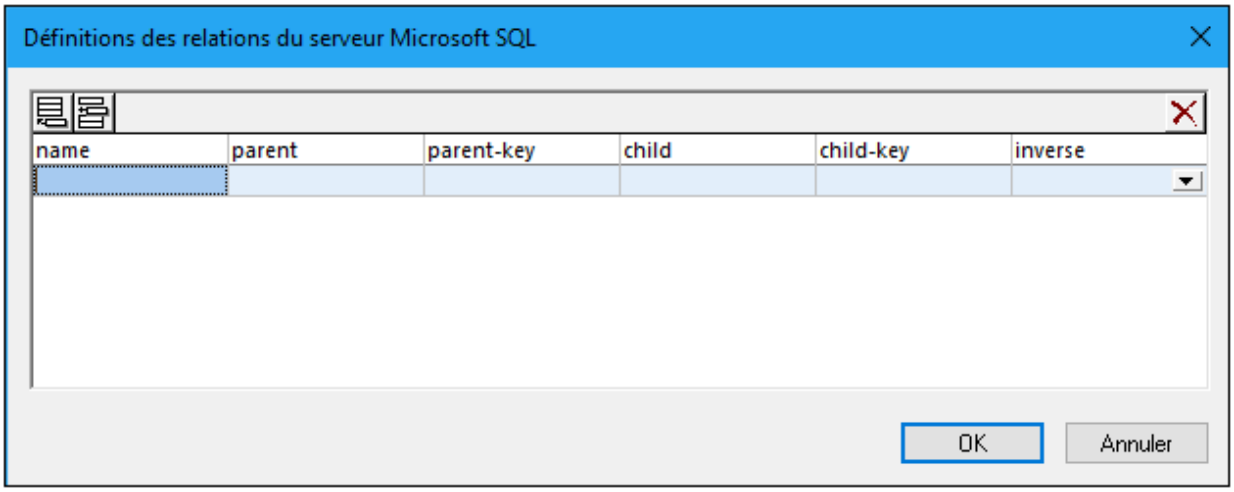
Note: Cette commande de menu peut être activée et désactivée. Les extensions peuvent donc être désactivées ou désactivées. Lorsque les extensions de serveur SQL sont activées, la commande est affichées avec une coche à sa gauche. En désactivant les extensions de serveur SQL (en cliquant sur la commande activée) la déclaration d'espace de nom du serveur SQL et toutes les extensions SQL sont supprimées dans le fichier. Un message d'avertissement apparaît, car cette action ne peut pas être annulée.

28.7.8.4 Relations de schéma nommées

La commande **Relations de schéma nommées** permet à la définition des relations nommées de fournir les informations nécessaires pour créer la hiérarchie de document. Vous devrez avoir préalablement activé les extensions de schéma SQL Server, à l'aide de l'option de menu "Activer les extensions de Schéma SQL Server", afin de pouvoir accéder à cette option de menu.

Pour créer une relation de schéma nommée :

1. Cliquer sur l'icône insérer ou apposer, pour ajouter une nouvelle ligne dans le dialogue.
2. Cliquer sur le champ et saisir le nom de relation correspondant
3. Cliquer sur **OK** pour confirmer.



Cela génère un élément de relation SQL, qui se place juste derrière la déclaration d'espace de noms.

Note : Cliquer sur l'icône supprimer, pour supprimer une ligne depuis la fenêtre du dialogue.

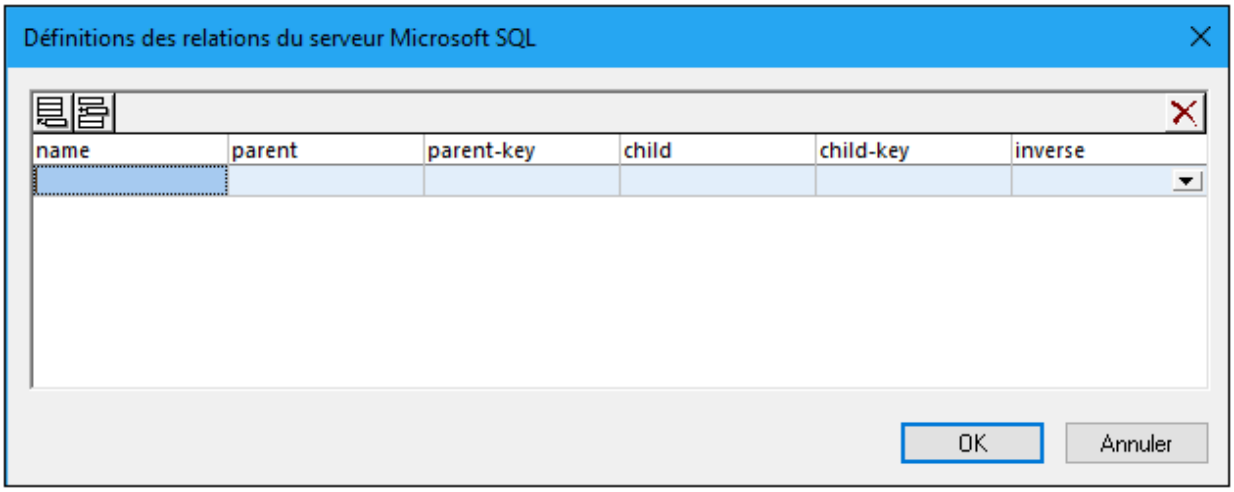
28.7.8.5 Relations d'éléments sans nom

La commande **Relations de schéma sans nom** permet à la définition des relations sans nom de fournir les informations nécessaires pour créer la hiérarchie de document. Vous devrez avoir préalablement activé les extensions de schéma SQL Server, à l'aide de l'option de menu **Activer les extensions de schéma Microsoft SQL Server** afin de pouvoir accéder à cette option de menu.

Pour créer une relation de schéma sans nom :

1. Cliquer sur l'icône insérer ou apposer, pour ajouter une nouvelle ligne dans le dialogue.
2. Cliquer sur le champ et saisir le nom correspondant
3. Cliquer sur **OK** pour confirmer.

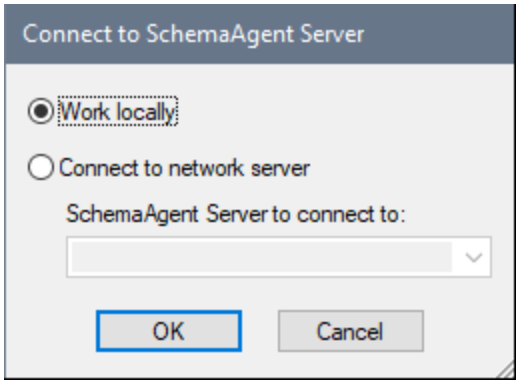
© 2016-2022 Altova GmbH



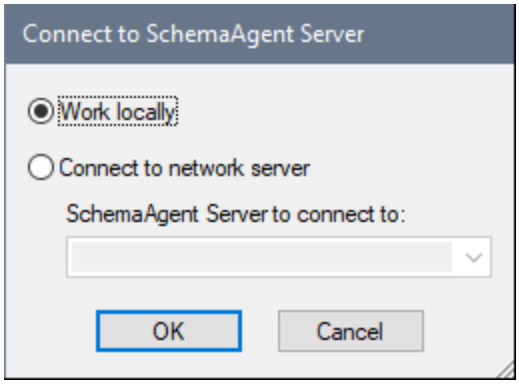
Cela génère un élément de relation SQL, pour l'élément de schéma sélectionné actuellement.

Note : Cliquer sur l'icône supprimer , pour supprimer une ligne depuis la fenêtre du dialogue.

28.7.9 Se connecter à SchemaAgent Server



La commande **Conception de schéma | Se connecter à SchemaAgent Server** est activée lorsqu'un document de Schéma XML est actif. Vous pouvez vous connecter à un SchemaAgent Server. Toutefois, vous pourrez uniquement vous connecter à un serveur SchemaAgent si un produit SchemaAgent Altova se trouvant sous licence est installé sur votre appareil. Lorsque vous cliquez sur cette commande, le dialogue Se connecter à SchemaAgent (*capture d'écran ci-dessous*) s'ouvrira :



Vous pouvez soit utiliser le serveur local (le serveur SchemaAgent qui est contenu dans Altova SchemaAgent) soit un serveur de réseau (le produit Altova SchemaAgent Server qui est disponible gratuitement). Si vous choisissez **Travailler localement**, le serveur local de SchemaAgent sera lancé lorsque vous cliquez sur **OK** et une connexion sera établie. Si vous choisissez **Se connecter au serveur de réseau**, le SchemaAgent Server sélectionné doit être en cours d'exécution afin de pouvoir établir une connexion.

Une fois la connexion à SchemaAgent Server établie, XMLSpy fonctionne comme un client SchemaAgent et propose des fonctions d'édition et de gestion de schéma puissantes et améliorées. Pour plus de détails concernant SchemaAgent, l'installation de SchemaAgent Server, et la connexion à SchemaAgent Server, voir SchemaAgent dans la section DTD et Schéma XML de ce manuel d'utilisateur. Pour plus d'informations concernant l'installation et le travail avec ces deux produits, voir le manuel d'utilisateur SchemaAgent fourni avec ces produits.

Une fois vous être connecté à SchemaAgent Server, un message apparaît dans la barre en haut de la Fenêtre principale comportant des informations concernant la connexion. Vous avez maintenant le plein accès à tous les schémas et les composants de schémas dans les chemins d'accès (aux dossiers) définis pour le serveur SchemaAgent auquel XMLSpy est connecté.

Note : Pour que la connexion réussisse, vous devez installer le logiciel Altova SchemaAgent Client avec une licence valide sur le même appareil que celui sur lequel XMLSpy est installé.

28.7.10 Déconnecter de SchemaAgent Server



La commande **Déconnecter de SchemaAgent Server** est activée lorsqu'une connexion à un SchemaAgent Server a été établie avec succès. Le fait de choisir cette commande déconnecte XMLSpy de SchemaAgent Server.

© 2016-2022 Altova GmbH

28.7.11 Afficher dans SchemaAgent

Le point de menu **Afficher dans SchemaAgent** permet d'afficher le schéma actif et, en option, les schémas liés dans le produit in the SchemaAgent Altova. (Ce produit doit être installé sur le même appareil que XMLSpy si vous souhaitez utiliser les fonctions de SchemaAgent). Le/s schéma/s est/sont ouvert/s dans un nouveau SchemaAgent Design dans SchemaAgent.

Passer avec la souris sur le point de menu **Afficher dans SchemaAgent** pour ouvrir un sous-menu avec des options concernant les schémas à afficher dans SchemaAgent. Ces options sont décrites dans SchemaAgent dans la section DTD et Schéma XML de ce manuel de l'utilisateur.

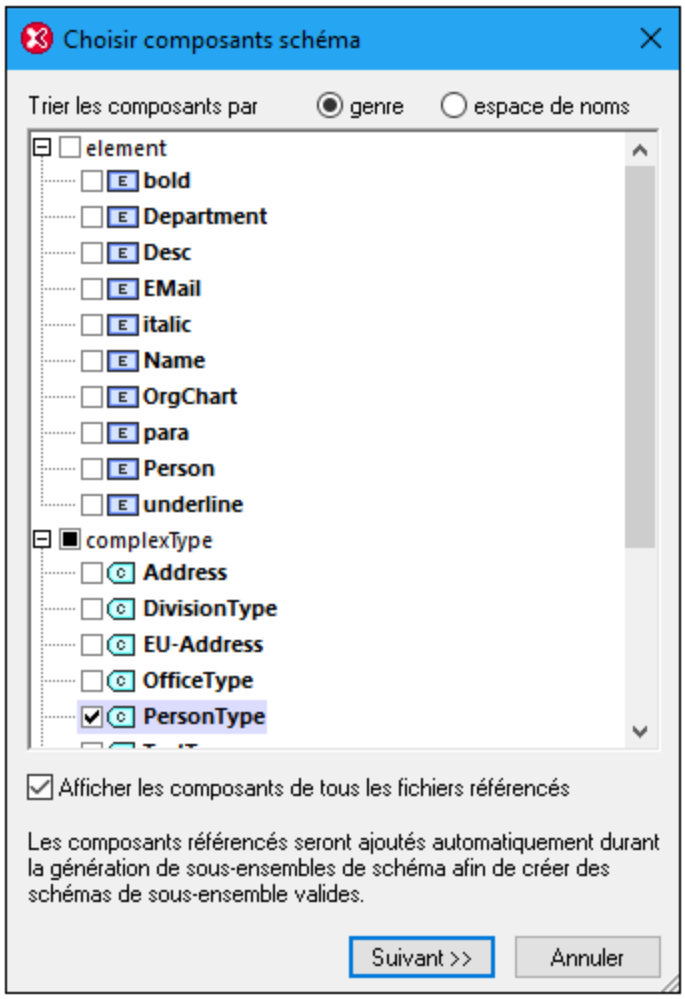
28.7.12 Validation SchemaAgent



La commande **SchemaAgent Validation** vous permet de valider le schéma actif actuellement ainsi que les schémas liés au schéma actif actuellement. Cette fonction est décrite en détail dans la section Validation SchemaAgent dans la section Mode Schéma de ce manuel de l'utilisateur.

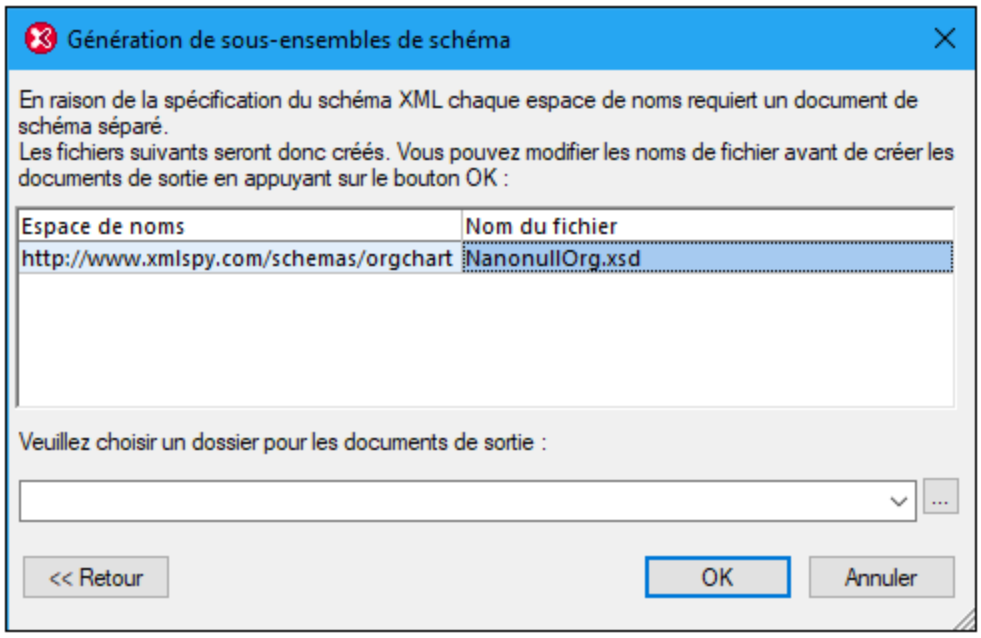
28.7.13 Créer un sous-ensemble de schéma

La commande **Créer un sous-ensemble de schéma** ouvre le dialogue Choisir composant de schéma (*capture d'écran ci-dessous*). Dans ce dialogue, vous cochez le ou les composants que vous souhaitez créer en tant qu'un sous-ensemble de schéma unique puis cliquez sur **Suivant**. (Veuillez noter qu'une case à cocher en bas du volet permet de lister les composants depuis tous les fichiers référencés pour la sélection.)



Dans le dialogue Génération de sous-ensemble de schéma qui apparaît maintenant (*capture d'écran ci-dessous*), saisir les noms que vous souhaitez donner aux fichiers du pack de sous-ensemble du schéma. Vous devez aussi spécifier le dossier dans lequel les nouveaux fichiers de sous-ensemble de schéma seront enregistrés. Un pack de sous-ensemble du schéma peut avoir plusieurs fichiers si un ou plusieurs composants créés sont des composants importés dans le schéma original. Un fichier de schéma séparé est créé pour chaque espace de noms dans le sous-ensemble du schéma. Les noms de fichier affichés dans le dialogue sont, par défaut, les noms des fichiers originaux. Mais puisque vous n'êtes pas autorisé à écraser les fichiers originaux, utilisez de nouveaux noms de fichier si vous souhaitez enregistrer les fichiers dans le même dossier que les fichiers originaux.

© 2016-2022 Altova GmbH

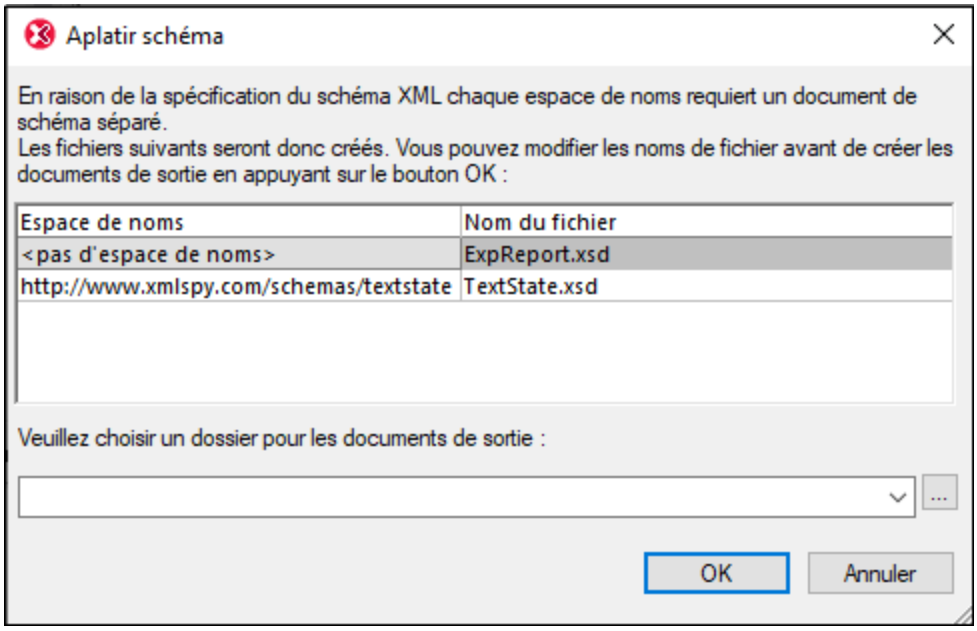


Cliquer sur **OK** pour ouvrir dans le Mode Schéma le fichier de sous-ensemble de fichier avec l'espace de noms correspondant à celui du fichier actif. Tout autre fichier dans le package sera créé mais pas ouvert dans le Mode Schéma.

28.7.14 Aplatir le schéma

Pour aplatir le schéma actif dans le Mode Schéma il faut : (i) ajouter les composants de tous les schémas inclus en tant que composants globaux du schéma actif et (ii) supprimer les schémas inclus.

Pour aplatir le schéma actif, sélectionner la commande **Conception de schéma | Aplatir schéma**. Le dialogue Aplatir schéma s'ouvre (*capture d'écran ci-dessous*), il contient les noms des fichiers séparés, un nom pour chaque espace de noms qui se trouvera dans le schéma aplati. Ces noms par défaut sont les mêmes que dans les noms de fichier originaux. Mais puisque vous n'êtes pas autorisé à écraser les fichiers originaux, vous devrez changer les noms de fichier si vous souhaitez les enregistrer dans le même dossier que le fichier actif. Vous pouvez parcourir un dossier dans lequel le schéma aplati et ses fichiers associés seront enregistrés.



Cliquer sur **OK** pour ouvrir le fichier de schéma dans le Mode Schéma.

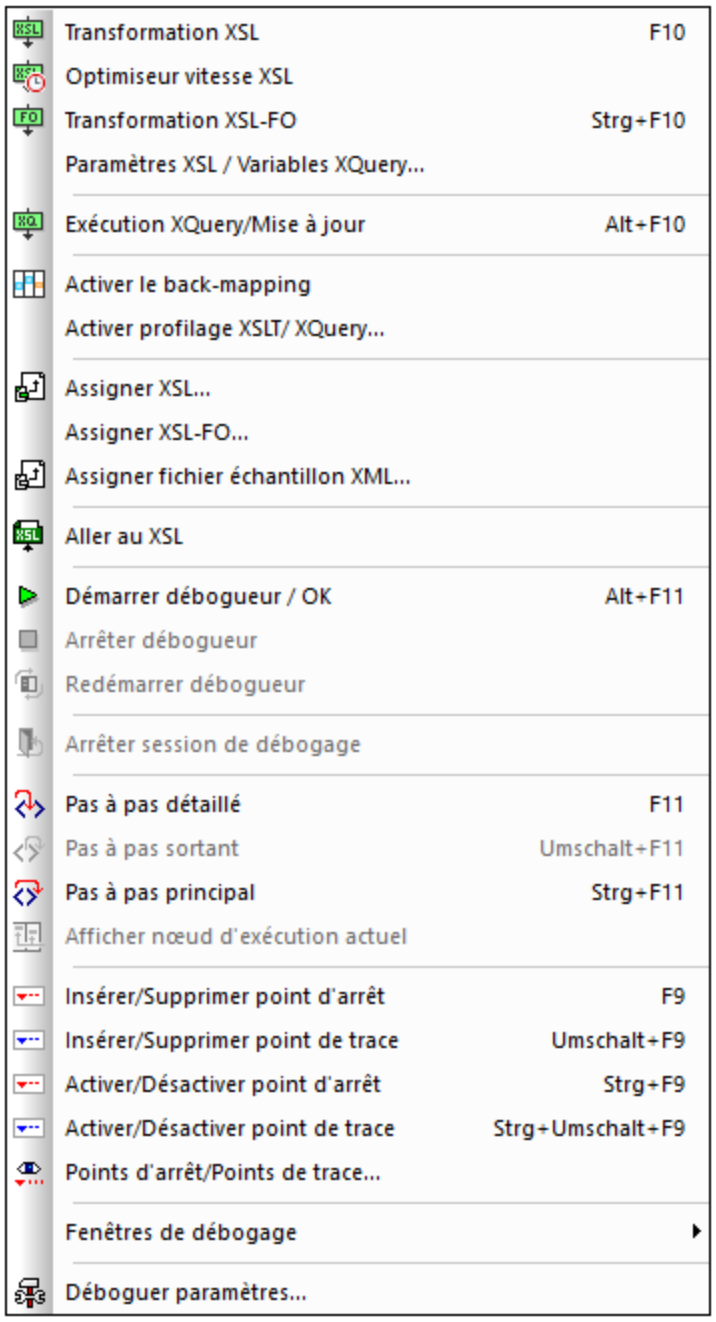
© 2016-2022 Altova GmbH

28.8 Menu XSL/XQuery

Le langage de transformation XSL vous permet de spécifier comment un document XML devrait être converti en d'autres documents XML ou fichiers de texte. Un type de document XML qui est généré avec un document XSLT est un document FO, qui peut ensuite être traité pour générer une sortie PDF. XMLSpy est doté de processeurs XSLT intégrés (pour XSLT 1.0, XSLT 2.0, et XSLT 3.0) et peut être lié à un processeur FO dans votre système pour transformer des fichiers XML et générer toute une variété de sorties. L'emplacement du processeur FO doit être spécifié dans la section XSL du dialogue Options (**Outils | Options**) afin de pouvoir l'utiliser directement depuis l'interface XMLSpy.

XMLSpy dispose également d'un moteur XQuery intégré, qui peut être utilisé pour exécuter des documents XQuery (avec ou sans référence à un document XML).

Les commandes gérant toutes les transformations mentionnées sont accessibles depuis le menu **XSL/XQuery**. En outre, ce menu contient également des commandes pour travailler avec le Débogueur XSLT/XQuery Altova.



28.8.1 Transformation XSL

F10

La commande **XSL/XQuery | Transformation XSL** transforme un document XML à l'aide d'une feuille de style XSLT assignée. La transformation peut s'effectuer en utilisant le moteur XSLT Altova approprié (moteur XSLT

© 2016-2022 Altova GmbH

1.0 Altova Engine pour les feuilles de style XSLT 1.0 ; moteur XSLT 2.0 moteur pour les feuilles de style XSLT 2.0 ; moteur XSLT 3.0 Altova pour les feuilles de style XSLT 3.0), le module MSXML fourni par Microsoft, ou un moteur XSLT externe. Le processeur qui est utilisé en conjonction avec cette commande est spécifié dans la section XSL du dialogue Options (**Outils | Options**).

Si votre document XML contient une référence à une feuille de style XSLT, cette feuille de style sera utilisée pour la transformation. (Une feuille de style XSLT peut être assignée à un document XML en utilisant la commande Assigner XSL. Si le document XML fait partie d'un projet, une feuille de style XSLT peut être spécifiée sur la base d'un dossier dans le dialogue Propriétés de projet. Cliquer avec la touche de droite dans le/s dossier/s ou le/s fichier/s que vous souhaitez transformer et choisir Transformation XSL.) Si une feuille de style XSLT n'a pas été assignée à un fichier XML, vous serez invité à choisir la feuille de style XSLT à utiliser. Vous pouvez aussi choisir un fichier par le biais d'une ressource globale ou une URL (cliquer sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**).

Automatiser la validation avec RaptorXML 2023

RaptorXML est l'application autonome d'Altova pour la validation XML, la transformation XSLT, et la transformation XQuery. Elle peut être utilisée depuis la ligne de commande, par le biais d'une interface COM, dans des programmes Java et dans des applications .NET. Les tâches de transformation XSLT peuvent donc être automatisées avec l'aide de RaptorXML. Par exemple, vous pouvez créer un fichier batch qui commande à RaptorXML d'exécuter des transformations XSLT sur un ensemble de documents et envoie la sortie vers un fichier de texte. Voir la documentation RaptorXML pour plus de détails.

Transformations en fichiers ZIP

Afin d'appliquer la sortie à un fichier ZIP, y compris des fichiers Open Office XML (OOXML) de type `.docx`, il faut spécifier le protocole ZIP dans le chemin de fichier du fichier de sortie. Par exemple :

```
filename.zip|zip/filename.xxx

filename.docx|zip/filename.xxx
```

Note : La structure du répertoire devra éventuellement être créée avant d'exécuter la transformation. Si vous générez des fichiers pour un archive Open Office XML, vous devrez comprimer les fichiers d'archive pour créer le fichier OOXML de niveau supérieur (par exemple, `.docx`).

28.8.2 Optimiseur de vitesse XSL



La commande **Optimiseur de vitesse XSL** est activée lorsqu'un document XSLT ou XML est actif. Elle lance l'Optimiseur de vitesse XSL, qui analyse la possibilité d'effectuer des transformations plus rapides en utilisant la feuille de style XSLT en cours d'analyse. L'Optimiseur fonctionne en exécutant la feuille de style XSLT à optimiser sur un ensemble de données XML (un ou plusieurs documents XML), et en analysant la performance de la feuille de style. Une stratégie d'optimisation est dérivée depuis cette analyse et peut être enregistrée avec la feuille de style XSLT (en tant qu'instruction de traitement à la fin de la feuille de style). La feuille de style optimisée peut être utilisée ultérieurement pour produire des transformations plus rapides.

En cliquant sur la commande, vous serez invité à choisir, selon le fait qu'un document XML ou XSLT soit actif, respectivement, un document XML ou une feuille de style XSLT. En cliquant sur **OK**, l'analyse démarre. Si le document XSLT ou XML a déjà, respectivement, une assignation XML ou une assignation XSLT dans le document, cette étape est sautée et l'analyse commence immédiatement. Pour plus de détails concernant l'utilisation de l'Optimiseur, voir la section Optimiseur de vitesse XSL. Les paramètres de l'Optimiseur peuvent être définis dans l'onglet Optimiseur de vitesse XSL du dialogue Options (**Outils | Options**).

28.8.3 Transformation XSL-FO



Ctrl+F10

FO est un format XML qui décrit des documents paginés. Un processeur FO, comme le FOP du projet XML Apache, prend un fichier FO en tant qu'entrée et génère un PDF en sortie. La production d'un document PDF depuis un document XML est donc un processus en deux étapes.

1. Le document XML est transformé en un document FO à l'aide d'une feuille de style XSLT.
2. Le document FO est traité par un processeur FO pour générer un PDF (ou un sortie alternative).

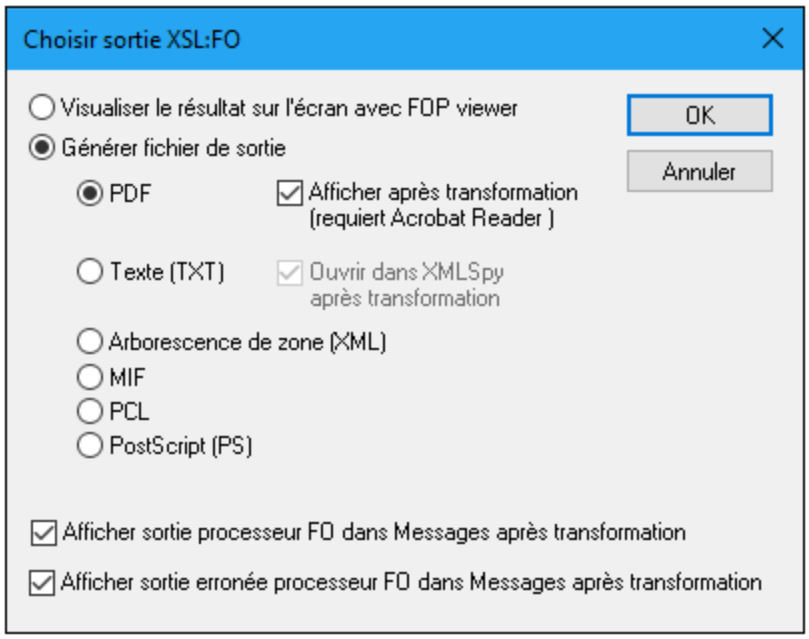
La commande **XSL/XQuery | Transformation XSL:FO** transforme un document XML ou un document FO en un PDF.

- Si la commande **Transformation XSL:FO** est exécutée sur un document XML de source, les deux étapes listées ci-dessus sont exécutées, en séquence, une après l'autre. Si la feuille de style XSLT nécessaire pour la transformation en FO n'est pas référencée dans le document XML, vous êtes invité à en assigner une pour la transformation. Vous pouvez aussi sélectionner un fichier par le biais de la ressource globale ou d'une URL (cliquer sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**). La transformation depuis XML en XSL-FO est effectuée par le processeur XSLT spécifié dans la section XSL du dialogue Options (**Outils | Options**). Par défaut, le processeur XSLT sélectionné est le processeur XSLT intégré de XMLSpy. Le document FO résultant est traité directement avec le processeur FO spécifié dans la section XSL du dialogue Options (**Outils | Options**).
- Si la commande **Transformation XSL:FO** est exécutée sur un document FO, alors le document est traité avec le processeur FO spécifié dans la section XSL du dialogue Options (**Outils | Options**).

Sortie de transformation XSL:FO

La commande **Transformation XSL:FO** ouvre le dialogue Choisir Sortie XSL:FO (*capture d'écran ci-dessous*). (Si le document actif est un document XML sans une assignation XSLT, vous serez tout d'abord invité à choisir un fichier XSLT.)

© 2016-2022 Altova GmbH



Vous pouvez consulter la sortie du processeur FO directement sur l'écran à l'aide de l'affichage FOP ou bien, vous pouvez générer un fichier de sortie dans un des formats suivants : PDF, texte, une arborescence de zone XML, MIF PCL, ou PostScript. Vous pouvez aussi activer les messages provenant du processeur FO pour afficher (i) le message de sortie standard du processeur dans la fenêtre Messages ; et (ii) les messages d'erreur du processeur dans la fenêtre Messages. Pour passer à une de ces deux options, cocher l'option appropriée située en bas du dialogue.

Note :

- À moins d'avoir désélectionné l'option pour installer le processeur FOP du Projet XML Apache, il aura été installé dans le dossier `C:\ProgramData\Altova\SharedBetweenVersions`. Une fois installé, le chemin y menant aura été saisi automatiquement dans la section XSL du dialogue Options (**Outils | Options**) en tant que le processeur FO à utiliser. Vous pouvez définir le chemin vers tout processeur FO que vous souhaitez utiliser.
- La commande Transformation XSL:FO peut être utilisée non seulement sur le fichier actif dans la Fenêtre principale, mais aussi sur tout fichier ou dossier que vous sélectionnez dans le projet actif. Pour ce faire, cliquer avec la touche de droite et sélectionner **Transformation XSL:FO**. La feuille de style XSLT assignée au dossier de projet sélectionné est utilisée.

28.8.4 Paramètres XSL / Variables XQuery

La commande **XSL/XQuery | Paramètres XSL / Variables XQuery** ouvre le dialogue Paramètres d'entrée XSLT / Variables externes XQuery (*voir la capture d'écran*). Vous pouvez saisir le nom d'un ou de plusieurs paramètres que vous souhaitez passer à la feuille de style XSLT, ou une ou plusieurs variables externes XQuery que vous souhaitez passer au document XQuery et leurs valeurs respectives. Ces paramètres sont utilisés comme suit dans XMLSpy:

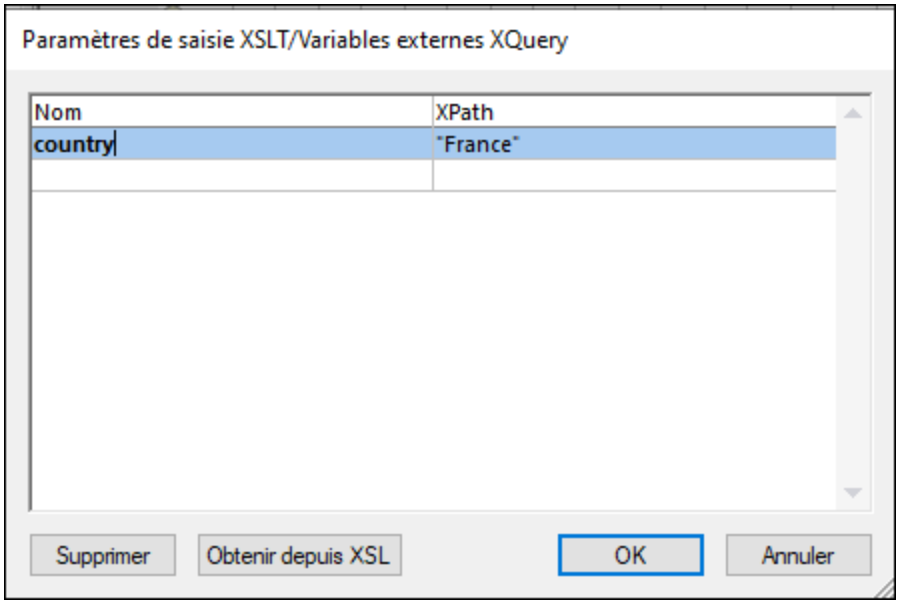
- Lorsque la commande **Transformation XSL** dans le menu XSL/XQuery est utilisée pour transformer un document XML, les valeurs de paramètre actuellement enregistrées dans le dialogue sont passées dans le document XSLT sélectionné et sont utilisées pour la transformation.
- Lorsque la commande **Exécution XQuery** dans le menu XSL/XQuery est utilisée pour traiter un document XQuery, les valeurs de variables externes XQuery actuellement enregistrées dans le dialogue sont passées dans le document XQuery pour l'exécution.

Note : les paramètres ou les variables que vous saisissez dans le dialogue Paramètres de saisie XSLT /Variables externes XQuery sont uniquement transférés sur le moteur XSLT intégré Altova. C'est pourquoi, si vous utilisez MSXML ou un autre moteur externe que vous avez configuré, ces paramètres ne sont pas transférés à ce moteur.

Note : Il ne s'agit pas d'une erreur si un paramètre XSLT ou une variable XQuery externe est définie dans le dialogue Paramètres de saisie XSLT /Variables externes XQuery mais il n'est pas utilisé dans le document XSLT/XQuery ou dans la transformation.

Utiliser les paramètres XSLT

La valeur que vous saisissez pour le paramètre peut être une expression XPath sans guillemets ou un string de texte délimité par des guillemets. Si le document actif est un document XSLT, la touche **Obtenir depuis XSL** sera activée. Cliquer sur cette touche pour insérer des paramètres déclarés dans le XSLT à l'intérieur du dialogue avec leurs valeurs par défaut. Cela vous permet d'inclure rapidement des paramètres déclarés puis de changer leurs valeurs par défaut comme requis.



Note : Une fois qu'un ensemble de valeurs de paramètres est saisi dans le dialogue, il est utilisé pour toutes les transformations ultérieures jusqu'à ce qu'il soit supprimé explicitement ou que l'application soit redémarrée. Les paramètres saisis dans le dialogue sont spécifiés au niveau de l'application, et seront passés dans le document XSLT respectif pour chaque transformation effectuée par le biais de l'IDE à partir de ce point. Cela signifie que :

- Les paramètres ne sont pas associés à un document particulier
- Chaque paramètre saisi dans le dialogue est supprimé une fois que XMLSpy a été fermé.

© 2016-2022 Altova GmbH

Exemple d'utilisation pour les paramètres XSLT

Nous avons un document XML qui contient les noms de pays et leurs capitales respectives :

```
<document>
   <countries>
      <country name="USA" capital="Washington DC"/>
      <country name="UK" capital="London"/>
      <country name="France" capital="Paris"/>
      <country name="Russia" capital="Moscow"/>
      <country name="China" capital="Beijing"/>
   </countries>
</document>
```

Le document XSLT suivant générera un document XML qui affiche un pays depuis le fichier XML ensemble avec la capitale du pays. Le pays est séelectionné en saisissant le nom en tant que valeur du paramètre appelé **country** (*affiché en surbrillance jaune ci-dessous*).

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:param name="country" select="'USA'"/>
   <xsl:template match="countries">
      <xsl:for-each select="country[@name=$country]">
         <country>
            <name><xsl:value-of select="$country"/></name>
            <capital><xsl:value-of select="@capital"/></capital>
         </country>
      </xsl:for-each>
   </xsl:template>
</xsl:stylesheet>
```

Lorsque ce document XSLT est exécuté dans le document XML recensé ci-dessus, le résultat sera :

```
<country><name>USA</name><capital>Washington DC</capital></country>
```

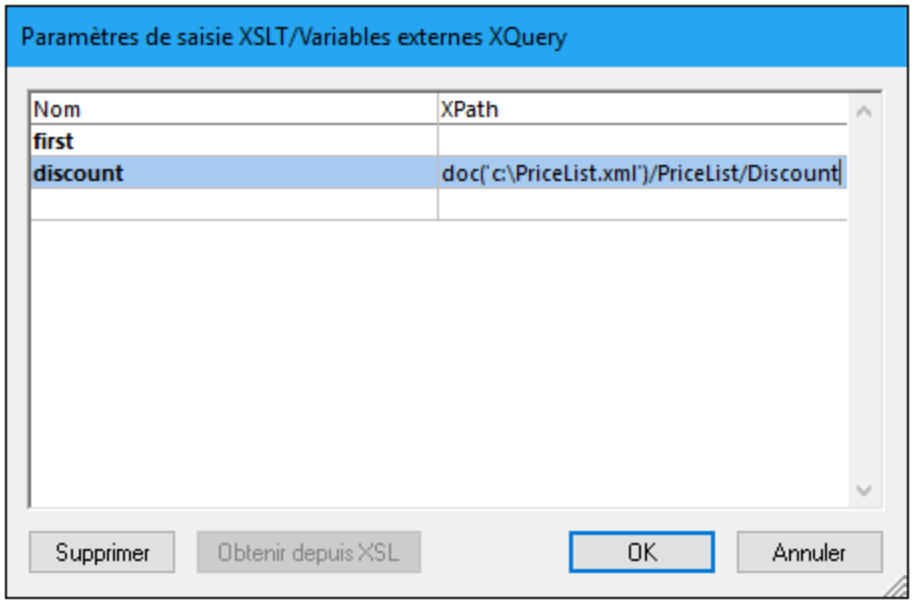
Maintenant, si vous créez un paramètre dans le dialogue Paramètre d'entrée XSLT / Variables externes XQuery appelé **country** et lui donner une valeur (*voir la capture d'écran ci-dessus*), alors cette valeur sera passée au paramètre **country** dans la feuille de style XSLT pour la transformation. De cette manière, vous pouvez passer différentes valeurs à différents paramètres lors de l'exécution.

Note :

- Si vous utilisez la commande **Transformation XSL:FO** (**XSL/XQuery | XSL:FO Transformation**), alors les paramètres saisis dans le dialogue Paramètres d'entrée XSLT / Variables externes XQuery **ne sont pas** passés à la feuille de style. Si ces paramètres doivent être utilisés dans la sortie PDF, transformez d'abord de XML à FO utilisant la commande de Transformation XSLT (**XSL/XQuery | XSL Transformation**), puis transformez FO en PDF utilisant la commande **XSL:FO Transformation** (**XSL/XQuery | XSL:FO Transformation**).
- Si vous utilisez un processeur XSLT autre que que les moteurs Altova XSLT intégrés, les paramètres que vous saisissez dans le dialogue Paramètres d'entrée ne seront pas passés au processeur externe.

Utiliser les variables XQuery externes

La valeur que vous saisissez pour des variables XQuery externes pourrait être une expression XPath sans guillemets ou un string de texte délimité par des guillemets. Le type de données de la variable externe est spécifié dans la déclaration variable dans le document XQuery.



Note : une fois qu'un ensemble de variables XQuery externe est saisi dans le dialogue, celles-ci sont utilisées pour toutes les exécutions ultérieures jusqu'à ce qu'elles soient explicitement supprimées ou que l'application est redémarrée. Les variables saisies dans le dialogue sont spécifiées au niveau de l'application, et seront passées dans le document XQuery respectif pour chaque exécution effectuée par le biais de l'IDE à partir de ce point. Cela signifie que :

- Les variables ne sont pas associées à un document particulier
- Chaque variable saisie dans le dialogue est supprimé une fois que l'application (XMLSpy) a été fermé.

Exemple d'utilisation pour des variables XQuery externes

Dans l'exemple suivant, une variable `$first` est déclarée dans le document XQuery et est ensuite utilisée dans la clause de retour de l'expression FLWOR :

```
xquery version "1.0";
declare variable $first comme xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

Cet XQuery renvoie `Peter Jones` si la valeur de la variable externe (saisie dans le dialogue Paramètres d'entrée XSLT / Variables externes XQuery) est `Peter`. Veuillez noter les points suivants :

- Le mot-clé `externe` dans la déclaration de variable dans le document XQuery indique que cette variable est une variable externe.
- Définir le type statique de la variable est optionnel. Si le type de données pour la variable n'est pas spécifiée dans la déclaration de variable, alors la valeur de variable obtient le type `xs:untypedAtomic`.

© 2016-2022 Altova GmbH

- Si une variable externe est déclarée dans le document XQuery, mais qu'aucune variable de ce nom n'est passée au document XQuery, alors une erreur est rapportée.
- Si une variable externe est déclarée et est saisie dans le dialogue Paramètres d'entrée XSLT / Variables externes XQuery, alors elle est considérée avoir l'étendue pour que le document XQuery soit exécuté. Si une nouvelle variable avec le nom est déclarée à l'intérieur du document XQuery, la nouvelle variable écrase temporairement la variable externe in-scope. Par exemple, le document XQuery ci-dessous renvoie `Paul Jones` même si la variable externe in-scope `$first` a une valeur `Peter`.

```
xquery version "1.0";
declare variable $first comme xs:string external;
let $first := "Paul"
let $last := "Jones"
return concat($first, " ", $last )
```

28.8.5 Exécution XQuery/Mise à jour



La commande **XSL/XQuery | Exécution XQuery/Mise à jour** exécute un document XQuery (1.0/3.1) ou XQuery Update (1.0/3.0). Selon que le fichier sélectionné est un fichier XQuery ou XQuery Update, soit une exécution XQuery soit une mise à jour XQuery sera exécutée. XMLSpy reconnaît le type du document (XQuery ou XQuery Update) sur la base de l'association de type de fichier du document (définie dans la section Types de fichier du dialogue Options).

Le moteur XQuery à utiliser (1.0 ou 3.1) est sélectionné automatiquement sur la base de la déclaration de la version dans le document. S'il n'y a aucune déclaration de version dans le document, la version par défaut spécifiée dans la section XQuery du dialogue Options est utilisée. La commande **Exécution XQuery/Mise à jour** peut être invoquée lorsqu'un fichier XQuery, XQuery Update ou XML est actif. Lorsqu'elle est invoquée depuis un fichier XML, elle ouvre un dialogue demandant d'associer un fichier XQuery au fichier XML. Vous pouvez aussi sélectionner un fichier par le biais de la ressource globale ou d'une URL (cliquer sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**).

Note : la commande est aussi disponible dans le menu contextuel des items de la fenêtre Projet.

Automatiser la validation avec RaptorXML 2023

RaptorXML est l'application autonome d'Altova pour la validation XML, la transformation XSLT, et la transformation XQuery. Elle peut être utilisée depuis la ligne de commande, par le biais d'une interface COM, dans des programmes Java et dans des applications .NET. Les tâches d'exécution XQuery peuvent donc être automatisées avec l'aide de RaptorXML. Par exemple, vous pouvez créer un fichier batch qui commande à RaptorXML de procéder à des exécutions XQuery sur un ensemble de documents et envoie la sortie vers un fichier de texte. Voir la documentation RaptorXML pour plus de détails.

28.8.6 Activer Back-Mapping



Cette commande, qui est aussi disponible dans la barre d'outils principale, permet de passer à la fonction de Back-Mapping.

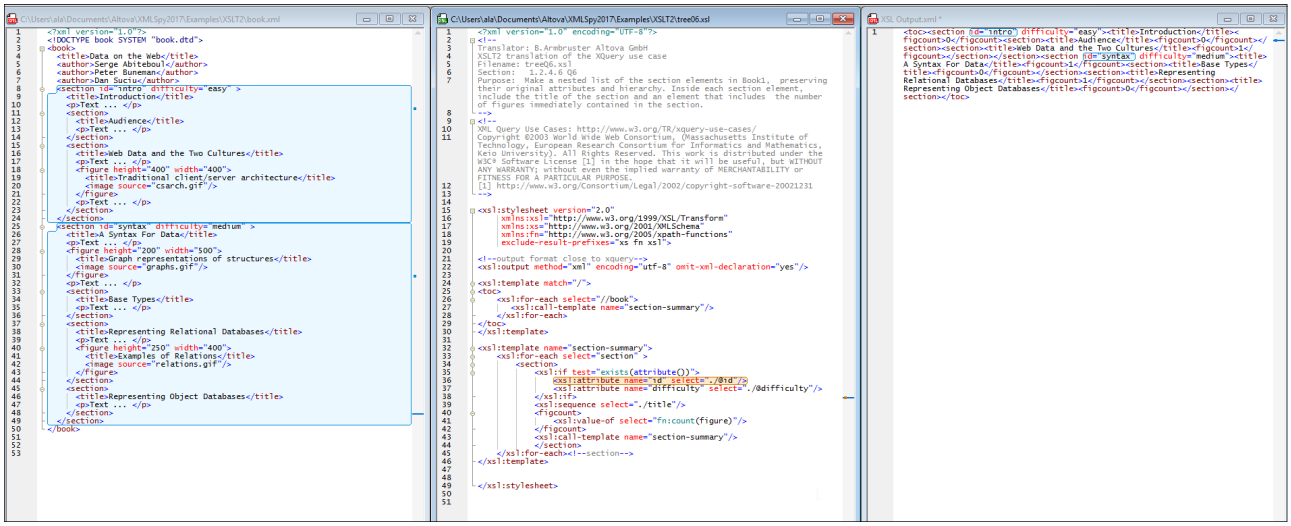
Une fois que le Back-Mapping a été activé (avec cette commande), les transformations XSLT et les exécutions XQuery sont effectuées de manière à ce que le document de résultat sera remappé sur les documents XSLT+XML ou XQuery+XML d'origine. Cela signifie que si vous cliquez sur un nœud dans le document de résultat, l'**instruction XSLT** ***et*** les **données de source XML** qui ont généré ce nœud de résultat particulier seront marquées *(voir capture d'écran ci-dessous)*. Cela est utile pour contrôler comment les transformations XSLT ou l'exécution XQuery crée les différentes parties du document de résultat. Vous pouvez également cliquer sur un nœud soit dans le document XSLT/XQuery ou le document XML pour consulter les parties correspondantes dans les deux autres documents.

Note : Les documents de résultats de tous les types sauf HTML sont ouverts dans le Mode Texte. Les documents de résultat HTML sont ouverts dans le Mode Navigateur mais vous pouvez passer au Mode Texte si vous le souhaitez. Si le document de résultat est ouvert dans le Mode Navigateur, le back-mapping sera disponible uniquement en le sélectionnant dans le document de résultat ; le back-mapping n'est pas disponible à la sélection ni dans le document XML ni dans le document XSLT/XQuery.

Note : Le Back-mapping n'est pas disponible pour les transformations depuis le Mode Authentic ou celles qui sont exécutées en tant que transformations de projet .

La capture d'écran ci-dessous montre le back-mapping d'une transformation XSLT. Les trois documents—`XML+XSLT+Result`—s'affichent dans des fenêtres verticales, dans cet ordre, l'une à côté de l'autre. L'instruction XSLT qui génère l'attribut **`section/@id`** dans le document de résultat a été cliqué. Par conséquence, tous les nœuds de résultat générés depuis cette instruction sont marqués, ainsi que les données de source XML à partir desquelles le nœud de résultat a été généré. Vous pouvez aussi cliquer sur les nœuds dans le document de résultat ou le document XML pour marquer les nœuds correspondants dans les deux autres documents.

© 2016-2022 Altova GmbH



Lorsque vous cliquez sur la commande **Activer Back-Mapping XSLT/XQuery**, un dialogue s'affiche qui vous demande si vous souhaitez **disposer les fenêtres en mosaïques** après la transformation. Si vous choisissez de le faire, les trois documents seront disposés verticalement côte à côte comme dans la capture d'écran ci-dessus.

Veuillez noter les points suivants :

- Seuls les documents XML qui sont chargés depuis un disque sont affichés ; les arborescences temporaires ne sont pas affichées.
- Dans certains cas, comme des exécutions XQuery, le document de résultat est créé sans obtenir de données depuis aucune source XML. Dans ces cas, aucun fichier XML n'est impliqué dans le back-mapping ; par conséquence, aucun n'est affiché.
- Si plusieurs fichiers XML sont utilisés en tant que sources de données, le premier qui sera rencontré dans le processus de transformation ou d'exécution sera affiché.
- Le back-mapping est plus lent et nécessite plus de mémoire que les transformations/exécutions qui ne sont pas back-mappées. Veuillez ne pas oublier cela lorsque vous travaillez avec des fichiers volumineux.
- Le menu contextuel du document de résultat (en mosaïque ou pas) contient des commandes (**Aller au nœud contextuel** et **Aller à l'instruction de source**) pour vous emmener vers les nœuds correspondants dans le document XML et XSLT/XQuery, respectivement.

La barre outils Back-Mapping

La barre outils Back-Mapping (*capture d'écran ci-dessous*) contient les icônes suivantes :



- *Marquer HTML dans le navigateur sur passage de la souris:* si le document de résultat est affiché dans le Mode Navigateur, le back-mapping est uniquement disponible en sélectionnant le contenu dans le document de résultat ; il n'est pas disponible en sélectionnant ni dans le document XML ni dans le document XSLT/XQuery. Dans le Mode Navigateur, vous pouvez sélectionner le contenu pour le back-mapping d'une de deux manières : (i) en cliquant sur du contenu dans le Mode Navigateur, ou en (ii) passant la souris sur le contenu. Utiliser cette commande à bascule pour choisir entre les deux

méthodes de sélection. Le passage de la souris est utile dans les cas où le fait de cliquer dans le contenu dans le Mode Navigateur peut entraîner un changement dans le document de résultat (par exemple en cliquant sur un bouton radio ou une liste de choix).

- *Terminer la session de back-mapping:* termine la session de back-mapping.

Terminer la session de back-mapping

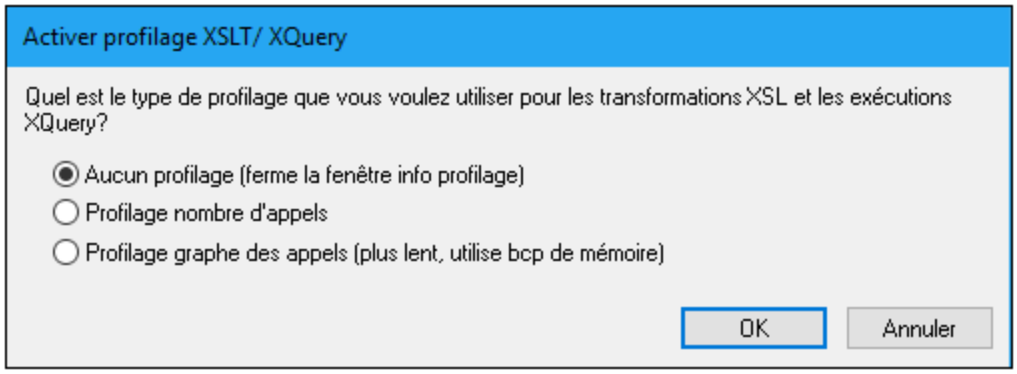
Alors qu'une session de back-mapping est en cours d'exécution, une icône **Terminer Session Back-Mapping** s'affiche dans la barre d'outils Back-Mapping (*voir ci-dessous*). Cliquer dedans pour terminer la session de back-mapping. Vous devriez utiliser la session de back-mapping uniquement pour des diagnostics. Si vous souhaitez éditer un des documents, il vaut mieux terminer la session de back-mapping avant l'édition.

Couleurs de marquage de texte

Les couleurs du back-mapping actif (contenu back-mappé actuellement sélectionné) et du back-mapping non actif (contenu back-mappé qui n'est pas sélectionné) peuvent être définies dans la catégorie *Divers* des paramètres Police (**Outils | Options | Polices et couleurs**).

28.8.7 Activer profilage XSLT/XQuery

La commande **Activer profilage XSLT/XQuery** ouvre le dialogue de profilage Activer XSLT/XQuery. Ce dialogue vous permet d'activer le Profileur, qui est un outil qui analyse le temps qu'il faut pour que les instructions soient exécutées pendant une transformation XSLT ou une exécution XQuery.



28.8.8 Assigner XSL



La commande **XSL/XQuery | Assigner XSL...** assigne une feuille de style XSLT à un document XML. Cliquer sur la commande pour ouvrir un dialogue vous permettant de spécifier le fichier XSLT que vous souhaitez assigner. Vous pouvez aussi sélectionner un fichier par le biais de la ressource globale ou d'une URL (cliquer

© 2016-2022 Altova GmbH

sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**).

Une instruction de traitement `xml-stylesheet` est insérée dans le document XML :

```
<?xml-stylesheet type="text/xsl" href="C:\workarea\recursion\recursion.xslt"?>
```

Note : Vous pouvez rendre le chemin du fichier assigné relatif en cliquant sur la case **Rendre chemin relatif à...**.

28.8.9 Assigner XSL-FO

La commande **XSL/XQuery | Assigner XSL:FO** assigne une feuille de style XSLT pour la transformation en FO dans un document XML. La commande ouvre un dialogue pour vous permettre de spécifier le fichier XSL ou XSLT que vous souhaitez assigner et insère l'instruction de traitement requise dans votre document XML.

Vous pouvez rendre relatif le chemin du fichier assigné en cochant la case *Rendre chemin relatif à*. Vous pouvez aussi sélectionner un fichier par le biais d'une ressource globale ou une URL (cliquer sur la touche **Chercher**) ou un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**).

Note : un document XML peut avoir deux fichiers XSLT qui lui sont assignés : un fichier pour les transformations XSLT standard, un deuxième fichier pour une transformation XSLT en FO.

28.8.10 Assigner fichier échantillon XML



La commande **XSL/XQuery | Assigner fichier échantillon XML** assigne un fichier XML à un document XSLT. La commande insère une instruction de traitement nommant un fichier XML qui sera traité avec ce fichier XSLT lorsque la Transformation XSL sera exécutée sur le fichier XSLT :

```
<?altova_samplexml C:\workarea\html2xml\article.xml?>
```

Note : Vous pouvez rendre le chemin du fichier assigné relatif en cliquant sur la case Rendre chemin relatif à.... Vous pouvez aussi sélectionner un fichier par le biais de la ressource globale ou d'une URL (cliquer sur la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**).

28.8.11 Aller au XSL



La commande **XSL/XQuery | Aller au XSL** ouvre le document XSLT associé. Si votre document XML contient une instruction de traitement de feuille de style (par ex. une assignation XSLT) comme :

```
<?xml-stylesheet type="text/xsl" href="Company.xsl"?>
```

alors, la commande **Aller au XSL** ouvre le document XSLT dans XMLSpy.

28.8.12 Aller à l'instruction de source

Si vous choisissez un nœud dans le document de résultat d'un back-mapping, cliquer sur **Aller à l'instruction de source** pour emmener le curseur vers l'instruction de source contenue dans le document XSLT ou XQuery qui a généré le nœud sélectionné dans le document de résultat. Vous pourriez ensuite, par exemple éditer directement l'instruction si vous le souhaitez.

Voir la description de la commande de menu **XSL | Activer Back-Mapping** pour une description du back-mapping.

28.8.13 Aller au nœud de contexte

Si vous choisissez un nœud dans le document de résultat d'un back-mapping, cliquer sur **Aller au nœud de contexte** pour emmener le curseur vers le nœud de données contenu dans le document XML qui a généré le contenu du nœud sélectionné dans le document de résultat. Vous pourriez ensuite, par exemple éditer le nœud de données si vous le souhaitez.

Voir la description de la commande de menu **XSL | Activer Back-Mapping** pour une description du back-mapping.

28.8.14 Démarrer Débogueur / OK



Alt+F11

La commande **XSL/XQuery | Démarrer Débogueur / OK** démarre ou poursuit le traitement du document XSLT/XQuery jusqu'à la fin. Si des points d'arrêt ont été définis, le traitement s'interrompra à ce moment. Si des points de trace ont été définis, la sortie pour ces instructions sera affichée dans la fenêtre Trace lorsque le nœud de fermeture de la déclaration contenant le point de trace a été atteint. Si la session de débogage n'a

© 2016-2022 Altova GmbH

pas été lancée, cette touche démarrera la session et s'arrêtera au premier nœud à traiter. Si la session est exécutée, le document XSLT/XQuery sera traité jusqu'à la fin ou jusqu'à la rencontre du prochain point d'arrêt.

28.8.15 Arrêter Débogueur XSLT



La commande **XSL/XQuery | Arrêter Débogueur** arrête le débogueur. Il s'agit d'une opération différente de l'arrêt de la **session** de débogage dans laquelle le débogueur est exécuté. Cela est utile si vous souhaitez éditer un document en pleine session de débogage ou si vous souhaitez utiliser d'autres fichiers pendant la même session de débogage. Une fois avoir arrêté le débogueur, vous devez redémarrer le débogueur pour recommencer depuis le début du document XSLT/XQuery.

28.8.16 Redémarrer Débogueur



La commande **XSL/XQuery | Redémarrer Débogueur** vide la fenêtre de sortie et redémarre la session de débogage avec les fichiers sélectionnés actuellement.

28.8.17 Terminer session débogueur



La commande **XSL/XQuery | Arrêter session de débogage** termine la session de débogage et vous ramène au mode XMLSpy normal qui était actif avant d'avoir commencé la session de débogage. Selon les paramètres que vous avez définis dans le dialogue Paramètres de Débogueur XSLT/XQuery, les documents de sortie qui ont été ouverts pour la session de débogage resteront ouverts ou pas.

28.8.18 Pas à pas détaillé

 **F11**

La commande **XSL/XQuery | Pas à pas détaillé** passe en étapes individuelles à travers tous les nœuds et expressions XPath dans la feuille de style. Cette commande est aussi utilisée pour redémarrer le débogueur après son arrêt.

28.8.19 Pas à pas sortant



Shift+F11

La commande **XSL/XQuery | Pas à pas sortant** sort du nœud actuel pour passer au frère suivant du nœud parent ou au nœud suivant dans le niveau plus élevé que celui du nœud parent.

28.8.20 Pas à pas principal



Ctrl+F11

La commande **XSL/XQuery | Pas à pas principal** passe au-dessus du nœud actuel vers le nœud suivant du même niveau, ou au nœud suivant du niveau supérieur à celui du nœud actuel. Cette commande est aussi utilisée pour redémarrer le débogueur après son arrêt.

28.8.21 Afficher nœud d'exécution actuel



La commande **XSL/XQuery | Afficher nœud d'exécution actuel** affiche/sélectionne le nœud d'exécution actuel dans le document XSLT/XQuery et le nœud contextuel correspondant dans le document XML. Cela est utile si vous avez cliqué dans d'autres onglets qui affichent ou marquent un code spécifique dans la feuille de style XSLT ou le fichier XML, et que vous souhaitez retourner à l'endroit où vous vous trouviez auparavant.

28.8.22 Insérer/Supprimer Point d'arrêt



F9

La commande **XSL/XQuery | Insérer/Supprimer Point d'arrêt** insère ou supprime un point d'arrêt à la position de curseur actuelle. Les points d'arrêt inline peuvent être définis pour des nœud aussi bien dans les

© 2016-2022 Altova GmbH

XSLT/XQuery que les documents XML, et déterminent où le traitement s'interrompra. Une ligne en pointillés rouge apparaît au-dessus du nœud lorsque vous définissez un point d'arrêt. Les points d'arrêt ne peuvent pas être définis sur des nœuds fermants, et les points d'arrêt sur les attributs dans des documents XSLT seront ignorés. Cette commande est aussi disponible en cliquant avec la touche de droite à l'emplacement du point d'arrêt.

28.8.23 Insérer/Supprimer Point de trace



Shift+F9

La commande **XSL/XQuery | Insérer/Supprimer Point de trace** insère ou supprime un point de trace à la position de curseur actuelle dans un document XSLT/XQuery. Pour des instructions contenant un point de trace, pendant le débogage, la valeur de l'instruction est affichée dans la fenêtre Trace lorsque le nœud de cette instruction est atteint. Une ligne en pointillés bleus apparaît au-dessus du nœud lorsque vous définissez un point de trace. Les points de trace ne peuvent pas être définis sur des nœuds fermants. Cette commande est aussi disponible en cliquant avec la touche de droite à l'emplacement du point de trace.

28.8.24 Activer/Désactiver Point d'arrêt



Ctrl+F9

La commande **XSL/XQuery | Activer/Désactiver Point d'arrêt** active ou désactive des points d'arrêt déjà définis. Le marquage de point d'arrêt passe au gris lorsque le point d'arrêt est désactivé. Le débogueur ne s'arrête pas aux points d'arrêt désactivés. Pour désactiver/activer un point d'arrêt, placer le curseur dans ce nom de nœud et cliquer sur la commande **Activer/Désactiver Point d'arrêt**. Cette commande est aussi disponible en cliquant avec la touche de droite à l'emplacement où vous souhaitez activer/désactiver le point d'arrêt.

28.8.25 Activer/Désactiver Point de trace



Ctrl+Shift+F9

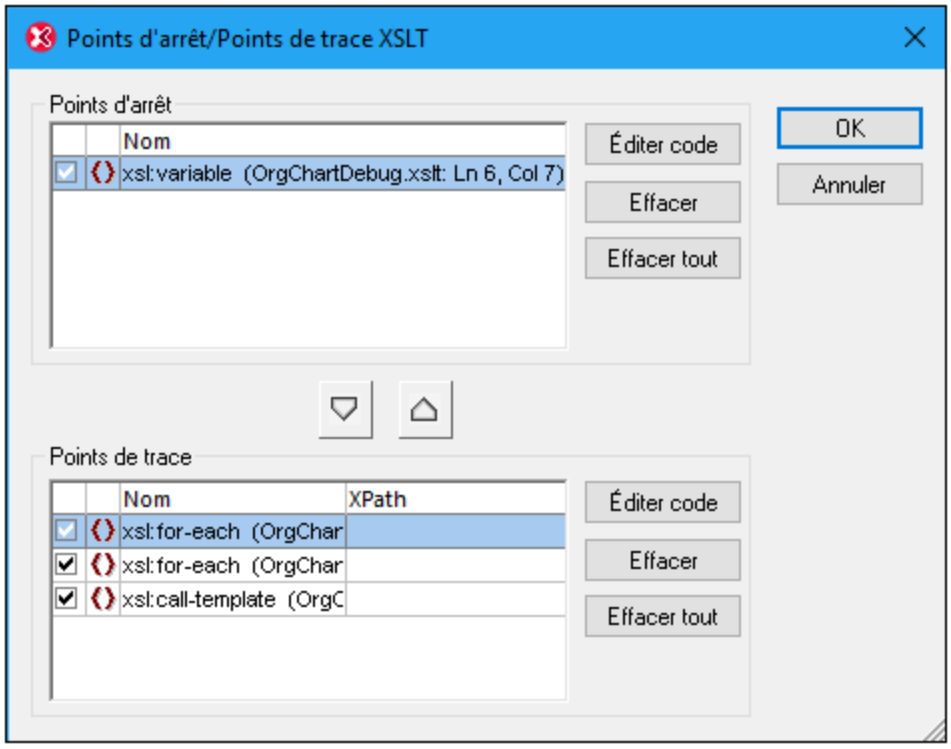
La commande **XSL/XQuery | Activer/Désactiver Point de trace** active ou désactive des points de trace déjà définis. Le marquage de point de trace bleu passe au gris lorsque le point de trace est désactivé. Aucune sortie n'est affichée pour les instructions avec des points de trace désactivés. Pour activer/désactiver un point de

trace, placer le curseur dans ce nom de nœud et cliquer sur la commande **Activer/Désactiver Point de trace**. Cette commande est aussi disponible en cliquant avec la touche de droite à l'emplacement où vous souhaitez activer/désactiver le point de trace.

28.8.26 Points d'arrêt/ Points de trace



La commande **XSL/XQuery | Points d'arrêt/Points de trace** ouvre le dialogue Points d'arrêt / Points de trace XSLT. Ceci affiche une liste de tous les points d'arrêt et points de trace actuellement définis (y compris les points d'arrêt et point de trace désactivés) dans tous les fichiers de la session de débogage actuelle.



Les fonctions suivantes sont disponibles :

- Les cases à cocher indiquent si un point d'arrêt ou un point de trace est activé (coché) ou désactivé.
- Vous pouvez mettre en surbrillance un point d'arrêt ou un point de trace à la fois.
- Supprimer un point d'arrêt ou point de trace en surbrillance en cliquant sur le bouton **Supprimer** correspondant.
- Supprimer tous les points d'arrêt ou points de trace en cliquant sur le bouton **Supprimer tout** correspondant.
- La touche **Éditer code** vous amène directement à ce point d'arrêt/point de trace dans le fichier.
- Cliquez sur  pour déplacer le point d'arrêt en surbrillance vers le volet points de trace.

© 2016-2022 Altova GmbH

- Cliquez sur  pour déplacer le point de trace en surbrillance vers le volet Points d'arrêt.
- Dans la colonne XPath du volet Points de trace, vous pouvez définir un XPath pour chaque point de trace. Ceci vous permet de spécifier une condition qui doit être satisfaite pour que le point de trace puisse être appliqué.

28.8.27 Déboguer les fenêtres

Placer le curseur sur la commande **XSL/XQuery | Déboguer les fenêtres** pour ouvrir un sous-menu contenant la liste des différentes Fenêtres d'information du Débogueur XSLT/XQuery. Choisir une Fenêtre d'information depuis cette liste pour afficher/dissimuler cette Fenêtre d'information dans l'interface Débogueur XSLT/XQuery. Cette commande peut uniquement être utilisée lorsqu'une session de débogage est en cours.

28.8.28 Paramètres de débogage



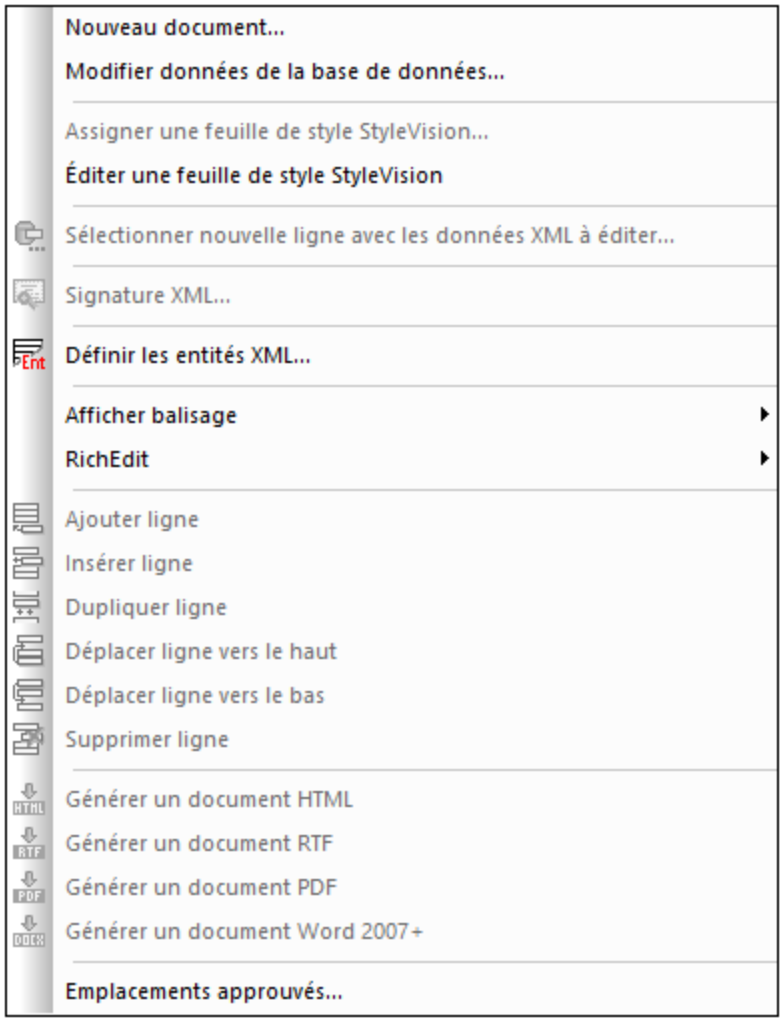
La commande **XSL/XQuery | Paramètres de débogage** ouvre le dialogue Paramètres de débogage, qui vous permet de définir les options d'utilisateur pour le Débogueur. Voir la section Débogueur XSLT/XQuery pour plus de détails.

28.9 Menu Authentic

Authentic View vous permet d'éditer des documents XML **basés sur des StyleVision Power Stylesheet (fichiers .sps) créés dans le produit Altova StyleVision !** Ces feuilles de style contiennent des informations qui permettent à un fichier XML d'être affichés graphiquement dans Authentic View. Outre le fait de contenir des informations d'affichage, les StyleVision Power Stylesheet permettent aussi d'écrire des données dans le fichier XML. Ces données sont traitées dynamiquement à l'aide des capacités disponibles dans les feuilles de style XSLT et la sortie est instantanément produite dans Authentic View.

De plus, les StyleVision Power Stylesheet peuvent être créées pour afficher un mode XML éditable d'une base de données. La StyleVision Power Stylesheet contient des informations pour se connecter à la base de données, affichant des données provenant de la base de données dans Authentic View, et enregistre des données dans la base de données.

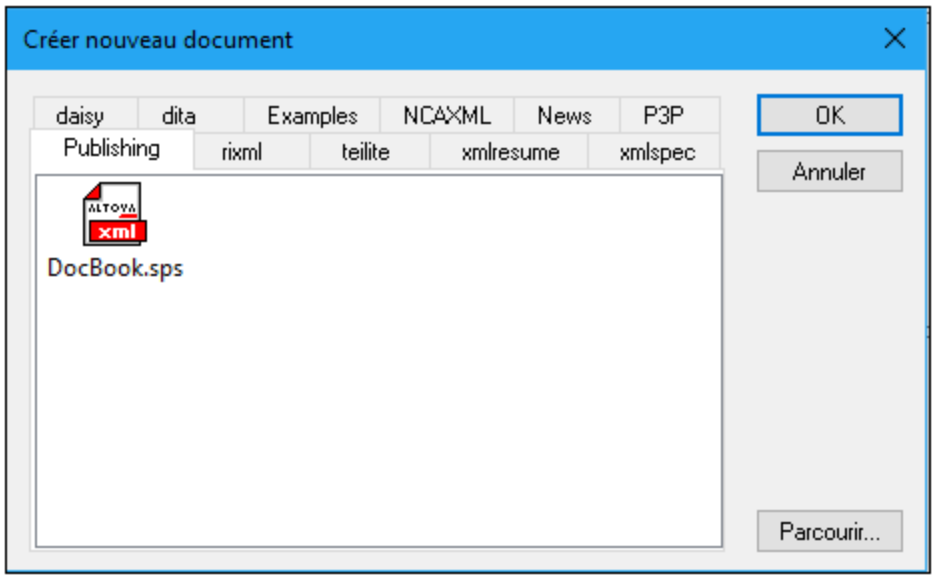
Le menu **Authentic** contient des commandes pertinentes à l'édition des documents XML dans Authentic View. Pour voir un tutoriel de Authentic View, consulter la section Tutoriels Mode Authentic.



 © 2016-2022 Altova GmbH

28.9.1 Nouveau document

Cette commande vous permet d'ouvrir un nouveau modèle de document XML dans Authentic View. Le modèle de document XML est basé sur une StyleVision Power Stylesheet (fichier .sps), et peut être ouvert en choisissant le StyleVision Power Stylesheet (fichier SPS) dans le dialogue Créer nouveau document (*capture d'écran ci-dessous*). En sélectionnant une SPS et en cliquant sur **OK**, le modèle de document XML défini pour ce fichier SPS est ouvert dans Authentic View.



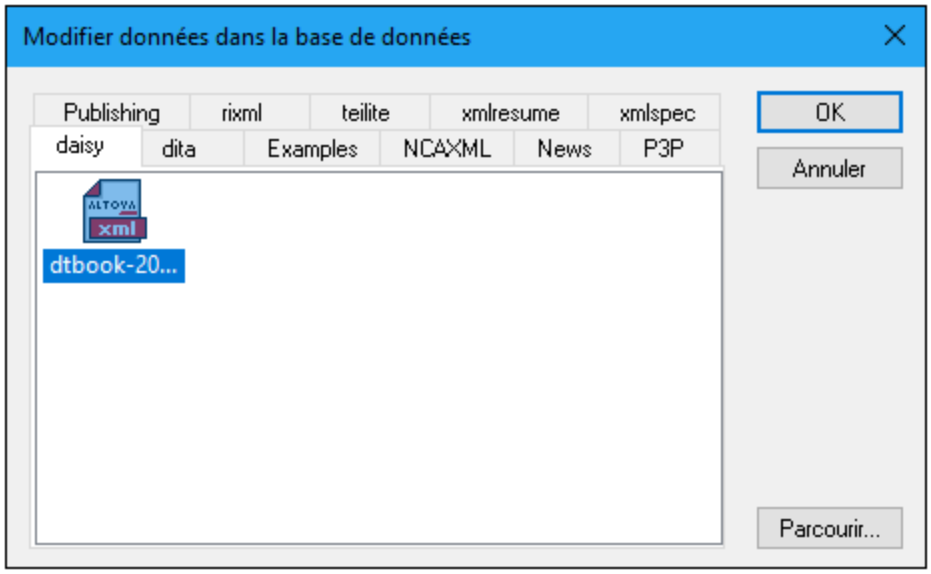
Le dialogue Créer nouveau document propose une variété de modèles de documents XML qui sont basés sur des DTD ou des schémas populaires. En alternative, vous pouvez aussi chercher un fichier SPS taillé sur mesure auquel un fichier XML modèle a été attribué. Les fichiers SPS sont créés en utilisant Altova StyleVision, une application qui permet de concevoir des modèles de document XML basés sur un schéma DTD ou XML. Une fois avoir conçu la SPS exigée dans StyleVision, un fichier XML est attribué (dans StyleVision) en tant que fichier XML modèle dans la SPS. Les données dans ce fichier XML fournissent les données de démarrage du nouveau modèle de contenu qui est ouvert dans le Authentic View de XMLSpy.

Le nouveau modèle de document XML aura donc des propriétés de présentation de document définies dans la SPS et les données du fichier XML qui ont été choisies en tant que fichier XML de modèle. L'utilisateur Authentic View peut maintenant éditer les modèles de document XML dans une interface graphique WYSIWYG, et l'enregistrer en tant que modèle XML.

28.9.2 Modifier les données de la base de données

La commande **Authentic | Modifier les données de la base de données** vous permet d'ouvrir un affichage éditable d'une base de données (BD) dans Authentic View. Toutes les informations concernant la connexion à la BD et comment afficher la BD et accepter les changements de la BD dans Authentic View sont contenues dans une StyleVision Power Stylesheet. La commande **Éditer les données de base de données** ouvre une StyleVision Power Stylesheet, configure une connexion vers la BD et affiche les données de la base de données (via un filtre XML) dans Authentic View.

Lorsque vous cliquez sur la commande **Éditer les données de base de données**, le dialogue Éditer les données de base de données (*voir la capture d'écran ci-dessous*) apparaît.



Chercher le fichier SPS requis et le sélectionner. Il se connecte à la BD et ouvre un affichage éditable de la BD dans Authentic View. Le design du mode BD affiché dans Authentic View est contenu dans la StyleVision Power Stylesheet.

Note : si, avec la commande **Éditer les données de base de données**, vous tentez d'ouvrir une StyleVision Power Stylesheet qui n'est pas basée sur une BD ou pour ouvrir une StyleVision Power Stylesheet basée sur BD qui a été créée dans une version de StyleVision avant la release StyleVision 2005, vous recevrez une erreur.

Note : les StyleVision Power Stylesheet sont créées utilisant Altova StyleVision.

28.9.3 Assigner une feuille de style StyleVision

La commande **Assigner une feuille de style StyleVision** assigne une StyleVision Power Stylesheet (SPS) à un **document XML** pour permettre l'affichage et l'édition de ce document XML dans Authentic View. La StyleVision Power Stylesheet qui doit être assignée au fichier XML doit être basée sur le même schéma que celui sur lequel le fichier XML se base.

Pour assigner une StyleVision Power Stylesheet à un fichier XML :

1. Activer le fichier XML et sélectionner **Assigner une feuille de style StyleVision**.
2. La commande ouvre un dialogue dans lequel vous spécifiez le fichier StyleVision Power Stylesheet que vous souhaitez assigner au XML.
3. Cliquer sur **OK** pour insérer l'instruction SPS requise dans votre document XML. Veuillez noter que vous pouvez rendre le chemin du fichier assigné relatif en cliquant sur la case *Rendre chemin relatif à*. Vous pouvez aussi sélectionner un fichier par le biais de la ressource globale ou d'une URL (cliquer sur

© 2016-2022 Altova GmbH

la touche **Chercher**) ou sur un fichier dans une des fenêtres ouvertes dans XMLSpy (cliquer sur la touche **Fenêtre**).

```
<?xml version="1.0" encoding="UTF-8"?>
<?altova_sps HTML-Orgchart.sps?>
```

Dans l'exemple ci-dessus, StyleVision Power Stylesheet est appelée `HTML_Orgchart.sps`, et se trouve dans le même répertoire que le fichier XML.

Note : Les versions précédentes des produits Altova utilisent une instruction de traitement avec une cible ou un nom de `xmlspysps`, une instruction de traitement ressemblerait donc à `<?xmlspysps HTML-Orgchart.sps?>`. Ces instructions de traitement plus anciennes sont toujours valides avec Authentic View dans les versions actuelles des produits Altova.

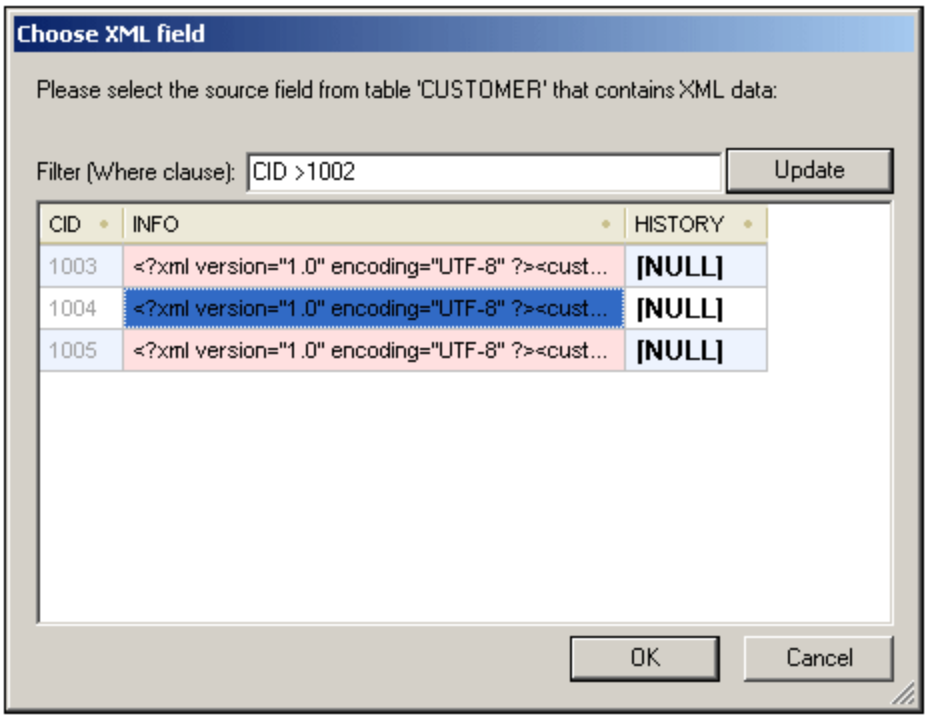
28.9.4 Éditer une feuille de style StyleVision

La commande **Éditer une feuille de style StyleVision** est uniquement disponible dans le mode Authentic (ce qui implique que StyleVision Power Stylesheet a été assignée au document XML). La commande lance StyleVision et vous permet d'éditer la StyleVision Power Stylesheet immédiatement dans StyleVision.

28.9.5 Sélectionner nouvelle ligne avec données XML à éditer

La commande **Sélectionner nouvelle ligne avec données XML à éditer** vous permet de sélectionner une nouvelle ligne depuis la table pertinente dans une BD XML (comme IBM DB2). Cette ligne apparaît dans Authentic View, peut y être éditée et puis enregistrée dans la BD.

Lorsqu'une BD XML est utilisée en tant que la source de données XML, les données XML qui sont affichées dans Authentic View sont celles du document XML contenu dans une des cellules de la colonne de données XML. La commande **Sélectionner nouvelle ligne avec données XML à éditer** vous permet de choisir un document depuis une autre cellule (ou ligne) de cette colonne XML. La sélection de la commande **Sélectionner nouvelle ligne** permet d'afficher le dialogue Choisir le champ XML (*capture d'écran ci-dessous*), qui affiche la table contenant la colonne XML.



Vous pouvez saisir un filtre pour cette table. Le filtre devrait être une clause SQL `WHERE` (uniquement la condition, sans le mot-clé `WHERE`, par exemple : `CID>1002`). Cliquer sur **Mettre à jour** pour réinitialiser le dialogue. Dans la capture d'écran ci-dessus, vous pouvez voir le résultat d'un mode filtré. Ensuite, sélectionner la cellule contenant le document XML requis et cliquer sur **OK**. Le document XML dans la cellule (ligne) sélectionnée est chargé dans Authentic View.

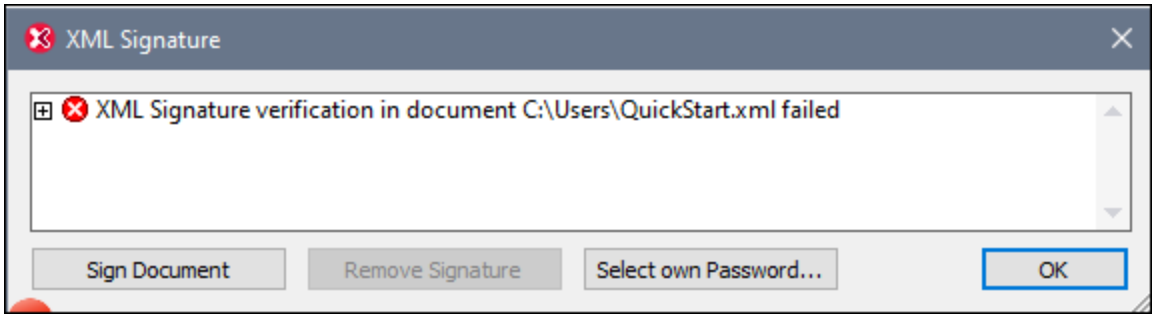
28.9.6 Signature XML

La commande **XML Signature** est disponible dans le Mode Authentic si les Signatures XML ont été activées pour la SPS associée. La commande **Signature XML** est aussi disponible en tant que l'icône de barre d'outils Signature XML  dans la barre d'outils Authentic.

Vérification et certificat/mot de passe

Cliquer sur la commande **Signature XML** pour lancer le processus de vérification de la signature. Si aucune signature n'est présente dans le document, un message s'affichera dans le dialogue de Signature XML (*voir capture d'écran ci-dessous*), et le dialogue aura une touche qui permet à l'utilisateur Authentic View de signer le document.

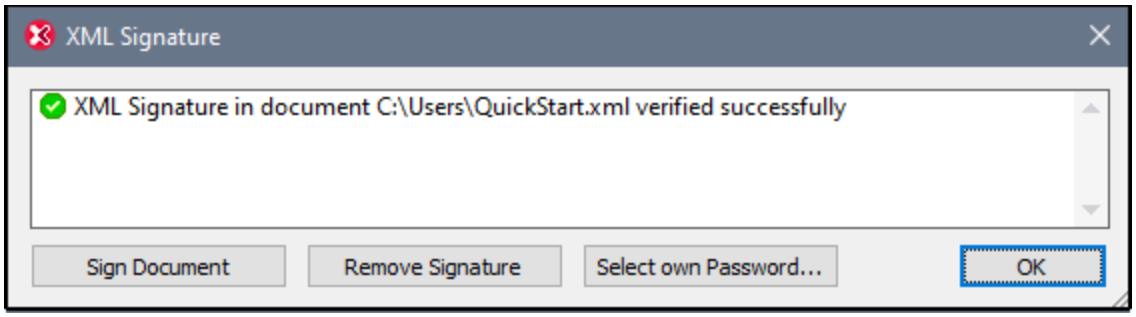
© 2016-2022 Altova GmbH



Si la touche **Choisir propre certificat** ou **Choisir propre mot de passe** se trouve dans ce dialogue, cela signifie que le Authentic View a reçu l'option de choisir un certificat ou mot de passe propre. (Le fait de savoir si un certificat ou un mot de passe doit être choisi aura été décidé par le designer de la SPS au moment de la configuration de la signature. La signature sera soit basée sur un certificat ou sur un mot de passe.) Cliquer sur un de ces boutons, s'ils sont présents dans le dialogue, activer l'utilisateur Authentic View pour charger un certificat ou saisir un mot de passe. La sélection de l'utilisateur Authentic View est stockée dans la mémoire et est valide pour la session actuelle uniquement. Si, après avoir choisi un certificat ou un mot de passe, le document ou l'application est fermée, les paramètres du certificat/mot de passe sont rétablis sur les paramètres enregistrés à l'origine avec la SPS.

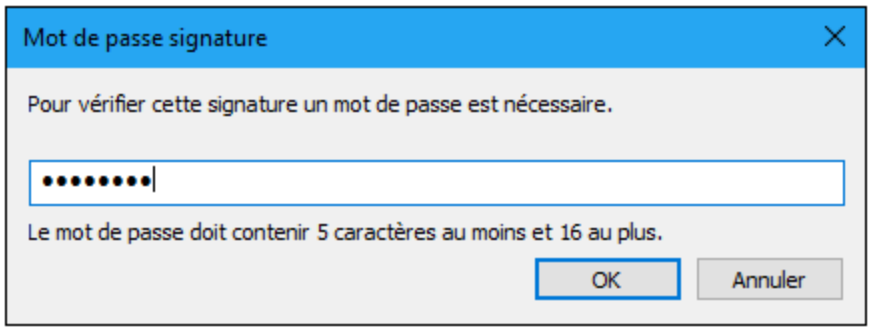
Informations de vérification et d'authentification

Si le processus de vérification est exécuté sur un document signé, deux situations sont possibles. Premièrement : si l'information d'authentification est disponible (dans la signature ou la SPS), alors le processus de vérification est exécuté directement et le résultat est affiché (*capture d'écran ci-dessous*).



L'information d'authentification est soit l'information de la clé de signature ou le mot de passe de signature. Le designer de la SPS aura spécifié si l'information de la clé du certificat est enregistrée dans la signature lorsque le document XML est signé ou, dans le cas d'une signature à base de mot de passe, si le mot de passe est enregistré dans la SPS. Dans les deux cas, l'authentification est disponible. Par conséquence le processus de vérification sera exécuté directement, sans qu'il soit nécessaire d'utiliser une entrée de la part de l'utilisateur Authentic View.

La deuxième situation possible se produit lorsque l'information d'authentification n'est pas disponible dans la signature (information de la clé du certificat) ou le fichier SPS (mot de passe). Dans cette situation, l'utilisateur Authentic View sera invité à fournir l'information d'authentification : un mot de passe (*voir capture d'écran ci-dessous*) ou l'emplacement d'un certificat. Si la SPS permet à Authentic View de sélectionner leur propre mot de passe ou certificat, cliquez sur **Sélectionner propre mot de passe (ou Certificat)** à cette fin



28.9.7 Définir les entités XML

Vous pouvez définir des entités à utiliser dans Authentic View, que votre document soit basé sur une DTD ou sur un Schéma XML. Une fois définies, ces entités sont affichées dans l'Assistant à la saisie Entités et dans le sous-menu **Insérer entité** du menu contextuel. Lorsque vous double-cliquez sur une entité dans l'Assistant à la saisie Entité, cette entité est insérée au niveau de l'insertion du curseur.

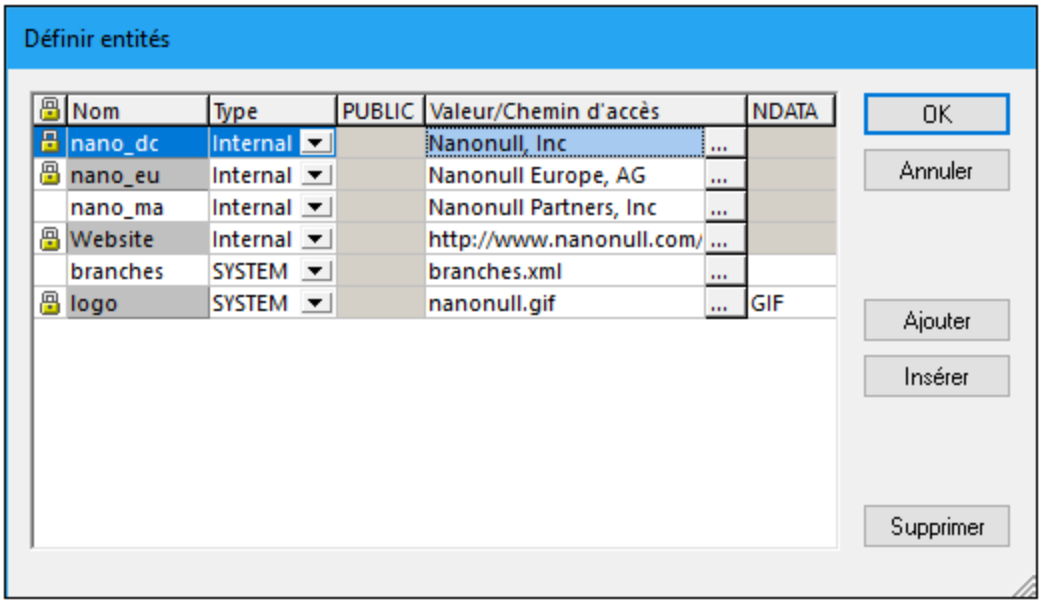
Une entité est utile si vous comptez utiliser un string de texte, un fragment XML, ou une autre ressource externe dans plusieurs endroits de votre document. Vous définissez l'entité, qui, en principe consiste en un nom bref qui représente les données requises, dans le dialogue Définir entités. Une fois avoir défini une entité, vous pouvez l'utiliser dans plusieurs endroits de votre document. Cela vous aide à économiser du temps et améliore l'entretien considérablement.

Il existe deux types généraux d'entités que vous pouvez utiliser dans votre document: une **entité parsée**, qui est une donnée XML (soit un string de texte soit un fragment d'un document XML), ou une **entité non-parsée**, qui est une donnée non-XML comme un fichier binaire (généralement un graphique, un son, ou un objet multimédia). Chaque entité possède un nom et une valeur. Dans le cas des entités parsées, l'entité est un espace réservé pour les données XML. La valeur de l'entité est soit la donnée XML elle-même, soit une URI qui pointe vers un fichier `.xml` qui contient les données XML. Dans le cas des entités non-parsées, la valeur de l'entité est une URI qui pointe vers le fichier de données non-XML.

Pour définir une entité :

1. Cliquer sur **Authentic | Définir les entités XML**. Cela permet d'ouvrir le dialogue Définir les entités.

© 2016-2022 Altova GmbH



2. Saisir le nom de votre entité dans le champ **Nom**. Il s'agit du nom qui apparaîtra dans l'Assistant à la saisie Entité.
3. Saisir le type d'entité à partir de la liste déroulante dans le champ **Type**. Trois types sont possibles. Une entité **Interne** est une entité pour laquelle le texte à utiliser est stocké dans le document XML lui-même. Le fait de choisir **PUBLIC** ou **SYSTEM** spécifie que la ressource est située en-dehors du fichier XML, et qu'elle sera localisée avec un identifiant public ou un identifiant de système, respectivement. Un identifiant de système est une URI qui donne l'emplacement de la ressource. Un identifiant public est un identifiant indépendant de tout endroit, qui permet à certains processeurs d'identifier la ressource. Si vous spécifiez aussi bien un identifiant public et de système, l'identifiant public résoudra à l'identifiant de système et l'identifiant de système sera utilisé.
4. Si vous avez sélectionné PUBLIC en tant que le Type, saisir l'identifiant public de votre ressource dans le champ PUBLIC. Si vous avez sélectionné Internal ou SYSTEM en tant que votre Type, le champ PUBLIC sera désactivé.
5. Dans le champ **Valeur/Chemin**, vous pouvez saisir un des types suivants :

   - Si le type d'entité est Internal, saisir la chaîne de texte que vous souhaitez en tant que la valeur de votre entité. Ne pas ajouter de guillemets pour délimiter l'entrée. Tous les guillemets que vous saisissez seront traités en tant que partie intégrante de la chaîne de texte.
   - Si le type d'entité est SYSTEM, saisir l'URI de la ressource ou sélectionner une ressource sur votre réseau local en utilisant la touche **Chercher**. Si la ressource contient des données parsées, il doit s'agir d'un fichier XML (le fichier doit avoir une extension **.xml**). En alternative, la ressource peut être un fichier binaire, comme un fichier GIF.
   - Si le type d'entité est PUBLIC, vous devrez ajouter en plus un identifiant de système dans ce champ.

6. L'entrée NDATA indique au processeur que cette entité ne doit pas être parsée mais doit être envoyée au processeur pertinent. Le champ NDATA doit donc être utilisé uniquement avec des entités parsées.

Fonctions de dialogue

Vous pouvez apposer, insérer et supprimer des entités en cliquant sur les touches appropriées. Vous pouvez aussi trier les entités par la valeur alphabétique de n'importe quelle colonne en cliquant sur l'en-tête de colonne

; cliquer une fois pour un tri dans l'ordre alphabétique ascendant, cliquer deux fois pour un tri dans l'ordre alphabétique descendant. Vous pouvez aussi redimensionner la fenêtre de dialogue et la largeur des colonnes.

Une fois qu'une entité a été utilisée dans le document XML, elle est verrouillée et ne peut pas être éditée dans le dialogue Définir les entités. Les entités verrouillées sont indiquées par un symbole de cadenas dans la première colonne. En verrouillant une entité, vous vous assurez que le document XML reste valide par rapport aux entités. (Le document serait invalide si une entité est référencée mais pas définie.)

Les entités doubles sont marquées.

Limitations

- Une entité contenue dans une autre entité n'est pas résolue, ni dans le dialogue, Authentic View, ni dans la sortie XSLT, et le caractère "&" d'une telle entité est affiché dans sa forme échappée, par ex. `&`.
- Les entités externes ne sont pas résolues dans Authentic View, sauf dans le cas où une entité est un fichier d'image et qu'elle est saisie en tant que la valeur d'un attribut qui a été défini dans le schéma comme étant de type `ENTITY` ou `ENTITIES`. Ce genre d'entités est résolu lorsque le document est traité avec un XSLT généré depuis la SPS.

28.9.8 Afficher balisage

La commande **Afficher balisage** a un sous-menu avec des options pour contrôler le balisage dans le document XML Authentic. Ces options sont décrites ci-dessous.



La commande **Masquer balisage** masque les symboles de balisage dans Authentic View.



La commande **Afficher petite balise** affiche les petits symboles de balisage dans Authentic View.



La commande **Afficher grande balise** affiche les grands symboles de balisage dans Authentic View.

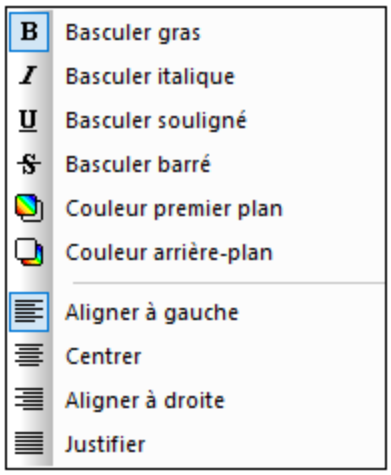


La commande **Afficher balise mixte** affiche les symboles de balisage mixtes dans Authentic View. La personne qui élabore la StyleVision Power Stylesheet peut choisir l'affichage de grandes balises, de petites balises ou sans balises pour les éléments/attributs individuels dans le document. L'utilisateur d'Authentic View voit ces balisages personnalisés dans le mode d'affichage mixte.

© 2016-2022 Altova GmbH

28.9.9 RichEdit

Passer avec la souris sur la commande **RichEdit** pour ouvrir un sous-menu contenant les commandes de marquage RichEdit (*capture d'écran ci-dessous*). Les commandes de menu de ce sous-menu sont uniquement activées dans Authentic View et lorsque le curseur est placé dans un élément qui a été créé en tant qu'un composant RichEdit dans le design SPS.



Les propriétés de style du texte du menu RichEdit seront appliquées au texte sélectionné lorsqu'une commande de marquage RichEdit est cliquée. L'utilisateur Authentic View peut, outre la police et la taille de la police spécifiée dans la barre d'outils Authentic, préciser en plus l'épaisseur de police, le style de police, la décoration de police, la couleur, la couleur d'arrière-plan et l'alignement du texte sélectionné.

28.9.10 Ajouter/Insérer/Dupliquer/Supprimer ligne



La commande **Ajouter ligne** appose une ligne à la table actuelle dans Authentic View.



La commande **Insérer ligne** insère une ligne à la table actuelle dans Authentic View.



La commande **Dupliquer ligne** duplique une ligne à la table actuelle dans Authentic View.



La commande **Supprimer ligne** supprime une ligne à la table actuelle dans Authentic View.

28.9.11 Déplacer Ligne, Supprimer Ligne

Les prochaines commandes de ligne suivantes sont activées dans Authentic View :

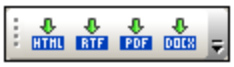
- **Déplacer Ligne vers le haut** déplace la ligne de table présente d'une ligne vers le haut dans Authentic View.
- **Déplacer Ligne vers le haut** déplace la ligne de table actuelle d'une ligne vers le haut dans Authentic View.
- **Supprimer Ligne** supprime la ligne actuelle.

28.9.12 Générer Document HTML, RTF, PDF, Word 2007+

Ces cinq documents génère les documents de sortie depuis le document XML de Authentic View stocké dans le fichier PXF :

- **Générer un document HTML**
- **Générer un document RTF**
- **Générer un document PDF**
- **Générer un document Word 2007+**
- **Générer un document Texte**

Ils sont également disponibles dans la barre d'outils du formulaire portable XML (PXF) (*voir la capture d'écran ci-dessous*).



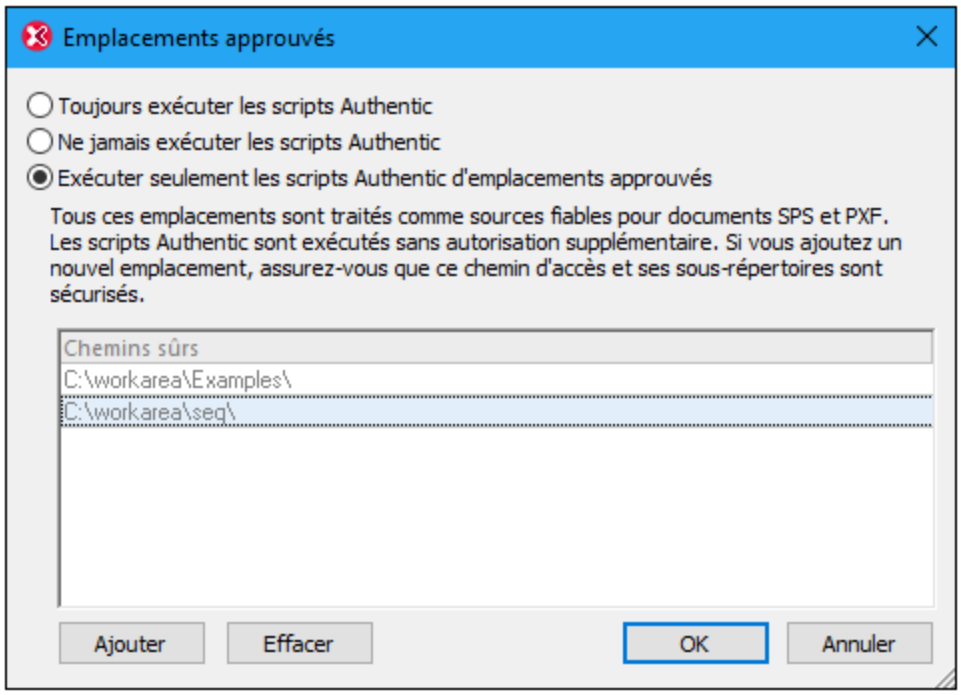
Cliquer sur les commandes individuelles ou boutons génère respectivement la sortie de HTML, RTF, PDF ou DocX.

Ces boutons sont activés quand un fichier PXF est ouvert dans Authentic View. Des commandes et boutons individuels sont activés si le fichier PXF a été configuré pour contenir la feuille de style pour ce format de sortie spécifique. Par exemple, si le fichier PXF a été configuré pour contenir les feuilles de style XSLT pour HTML et RTF, alors uniquement les commandes et boutons de la barre d'outils seront activés tandis que ceux pour la sortie texte, PDF et DocX (Word 2007+) seront désactivés.

28.9.13 Emplacements approuvés

La commande Emplacements approuvés ouvre le dialogue Emplacements approuvés (*capture d'écran ci-dessous*) dans lequel vous pouvez spécifier les paramètres de sécurité pour les scripts dans une SPS. Lorsqu'un fichier XML basé sur une SPS contenant un script est transféré sur le Mode Authentic, l'exécution du script sera autorisée ou non selon les paramètres que vous définissez dans ce dialogue.

© 2016-2022 Altova GmbH

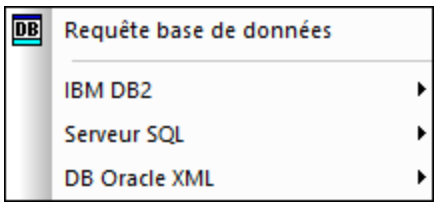


Les trois options disponibles sont :

- Les scripts Authentic sont toujours exécutés lorsqu'un fichier est ouvert dans le Mode Authentic.
- Les scripts Authentic ne sont jamais exécutés lorsqu'un fichier est ouvert dans le Mode Authentic.
- Seuls les scripts Authentic se trouvant dans des emplacements approuvés sont exécutés. La liste des emplacements (dossiers) approuvés est affichée dans le volet du bas. Utiliser la touche **Ajouter** pour chercher un dossier et l'ajouter à la liste. Pour supprimer une entrée de la liste, choisir une entrée dans la liste des Emplacements approuvés et cliquer sur **Supprimer**.

28.10 Menu DB

Le menu **DB** est le menu pour les opérations de base de données (BD). Il est affiché dans la capture d'écran ci-dessous et contient les items de menu listés ci-dessous. Les descriptions des commandes dans les sous-menu du menu BD se trouvent dans les sous-sections de cette section.



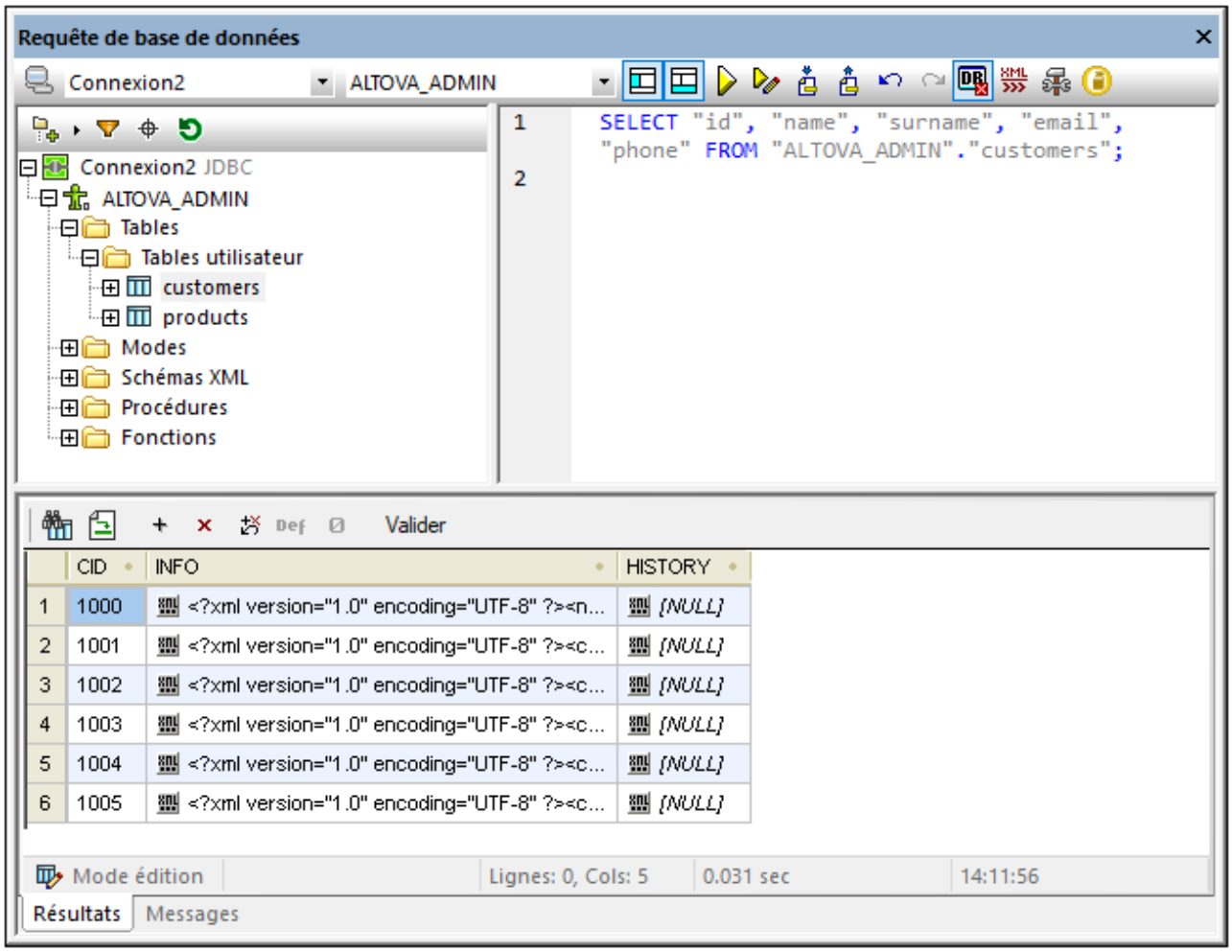
- Requête base de données, qui vous permet de requêter une variété de bases de données.
- IBM DB2, qui contient des commandes qui fournissent une prise en charge pour les fonctions spécifiques à IBM DB2.
- Serveur SQL, qui contient des commandes pour gérer les bases de données de serveurs SQL.
- Bases de données Oracle, qui contient des commandes pour travailler avec des bases de données Oracle.

Les opérations décrites dans cette section nécessitent une connexion à une base de données (pour plus d'instructions, voir Se connecter à une base de données).

28.10.1 Requête de base de données

La commande **Requête de base de données** ouvre la fenêtre Requête de base de données (*capture d'écran ci-dessous*). Une fois que la fenêtre de requête est ouverte, l'affichage de son contenu peut être activé et désactivé en cliquant sur la commande **BD | Requête de la base de données** ou sur l'icône de la barre d'outils Requête de base de données .

© 2016-2022 Altova GmbH



Aperçu de la fenêtre de la Requête de la base de données

La fenêtre de la Requête de base de données consiste en trois parties :

- Un **volet Navigateur** en haut à gauche, qui affiche des informations de connexion et des tables de base de données.
- Un **volet Requête** en haut à droite, dans lequel la requête est saisie.
- Un **volet Résultats/Messages** à onglets. Le volet de Résultats affiche le résultat de la requête dans ce que nous appelons la Grille de résultat. Le volet Messages affiche des messages à propos de l'exécution de requête, y compris des avertissements et des erreurs.

La fenêtre Requête de base de données dispose d'une barre d'outils en haut de l'écran. À ce point, veuillez noter les deux icônes de la barre d'outils juste en-dessous. Les autres icônes de la barre d'outils sont décrits dans la section volet de requête : Description et fonctions.



Active et désactive le volet Navigateur.

 Active et désactive le volet Résultats/Messages.

Aperçu du mécanisme de Requête de la base de données

Le mécanisme de Requête de la base de données est le suivant : il est également décrit en détail dans les sous-sections de cette section

1. Une connexion à la base de données est établie par le biais de la fenêtre Requête de la base de données. Les bases de données prises en charge comprennent : MS Access 2000 et 2003 ; Microsoft SQL Server ; Oracle ; MySQL ; Sybase et IBM DB2.
2. La base de données connectée ou les parties de la base de données sont affichées dans le volet Navigateur, qui peut être configuré pour s'adapter aux exigences de l'affichage.
3. Une requête rédigée dans une syntaxe appropriée à la base de données que vous souhaitez requêter est saisie dans le volet Requête, et la requête est exécutée.
4. Les résultats de la requête peuvent être consultées avec plusieurs filtres différents, édités et réenregistrés dans la BD.

28.10.1.1 Sources de données

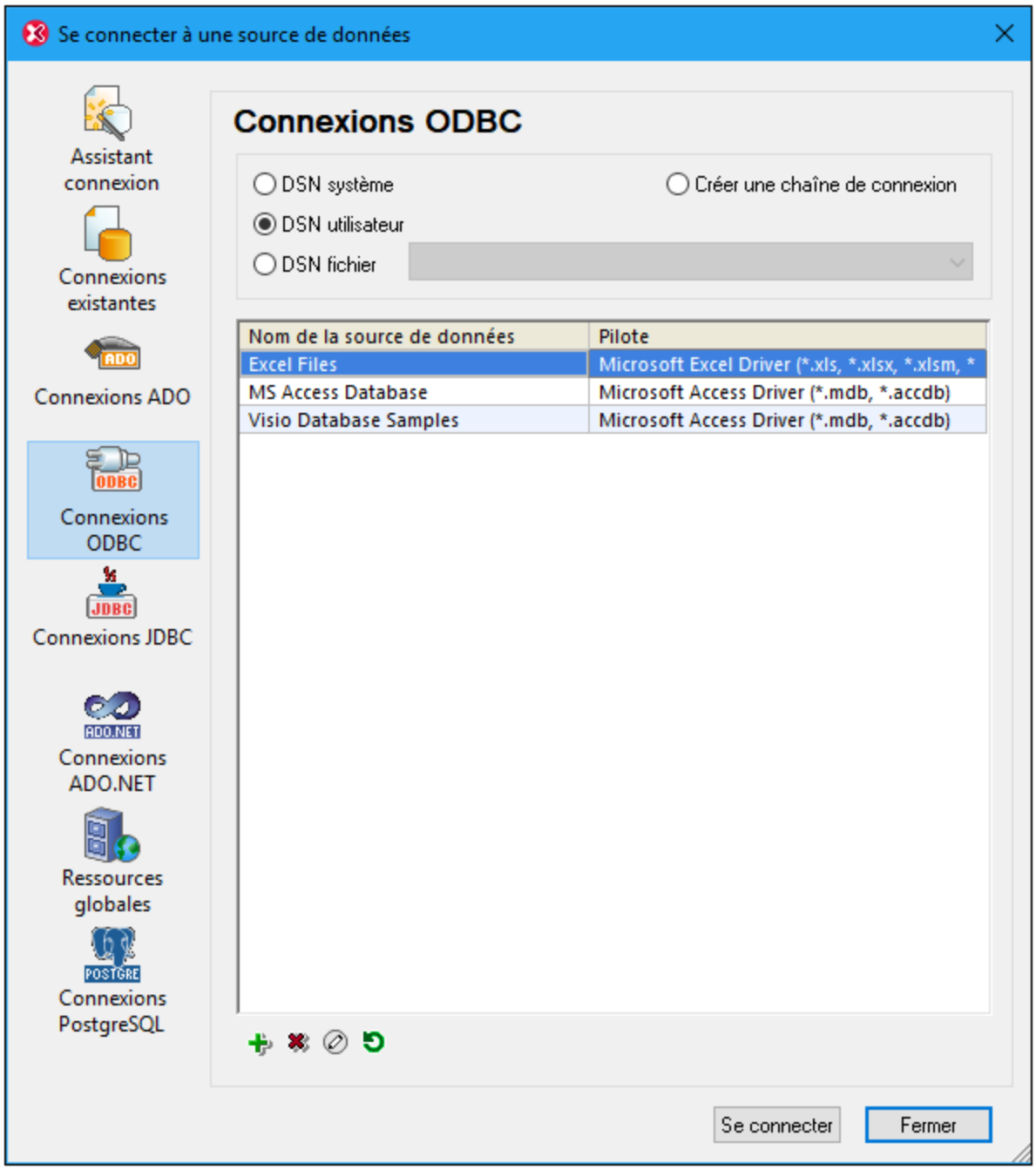
Afin de requêter une base de données, vous devez tout d'abord vous connecter à la base de données requise. Cette section décrit comment procéder :

- Se connecter à une base de données et
- choisir la source de données requise et l'objet racine depuis plusieurs connexions existantes.

Se connecter à une base de données

Lorsque vous cliquez sur la commande **Requête de base de données** pour la première fois dans une session (ou si aucune connexion de base de données n'existe), le dialogue Connexion rapide (*capture d'écran ci-dessous*) s'ouvre et vous permet de vous connecter à une base de données. Afin de pouvoir faire des connexions ultérieurement, cliquer sur l'icône Connexion rapide  dans la fenêtre Requête de base de données.

© 2016-2022 Altova GmbH



Vous trouverez une description vous expliquant comment vous connecter à une base de données par le biais du dialogue Connexion rapide dans la section Se connecter à une source de données.

La table ci-dessous énumère toutes les bases de données prises en charge. Bien qu'Altova œuvre pour prendre en charge d'autres bases de données, une connexion et un traitement des données réussis ont uniquement été testés avec les bases de données énumérées ci-dessous. Si votre application Altova est une version 64-bit, assurez-vous que vous avez accès aux pilotes de base de données 64-bit nécessaires pour la base de

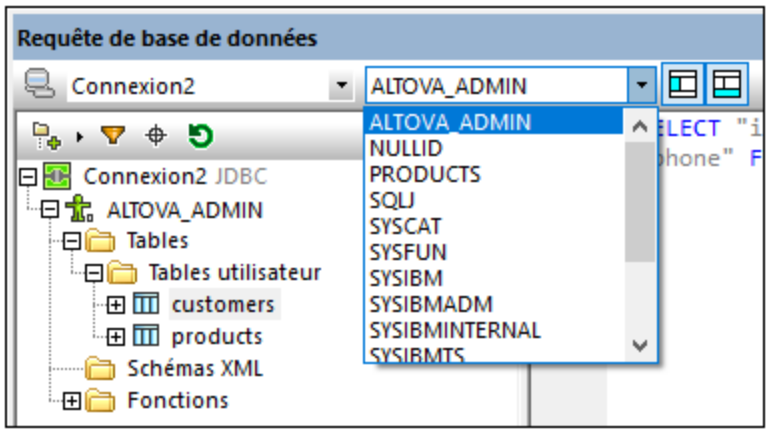
données spécifique à laquelle vous tentez de vous connecter.

Base de données	Notes
Firebird 2.x, 3.x	
IBM DB2 DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 i6.x, 7.4	Les fichiers logiques sont pris en charge et présentés en tant qu'affichage.
IBM Informix 11.70 et ultérieur	
MariaDB 10 et ultérieur	
Microsoft Access 2003 et ultérieur	Au moment de la rédaction de ce manuel (début septembre 2019), il n'a y pas de Microsoft Access Runtime disponible pour Access 2019. Vous pouvez vous connecter à une base de données Access 2019 depuis les produits Altova uniquement si Microsoft Access 2016 Runtime est installé et uniquement si la base de données n'utilise pas le type de données « Large Number ».
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005 et ultérieur Microsoft SQL Server on Linux	
MySQL 5 et ultérieur	
Oracle 9i et ultérieur	
PostgreSQL 8 et ultérieur	Les connexions PostgreSQL sont prises en charge aussi bien en tant que connexions natives qu'en tant que connexions sur base pilote par le biais d'interfaces (pilotes) comme ODBC ou JDBC. Les connexions natives ne nécessitent pas de pilotes.
Progress OpenEdge 11.6	
SQLite 3.x	Les connexions SQLite sont prises en charge en tant que connexions directes natives du fichier de base de données SQLite. Aucun pilote séparé n'est nécessaire. Dans l'affichage Authentic, les données provenant d'une base de données SQLite ne sont pas éditables. Si vous essayez d'enregistrer des données SQLite à partir de l'affichage Authentic, une boîte de messages vous informera de cette limitation connue.
Sybase ASE 15, 16	
Teradata 16	

© 2016-2022 Altova GmbH

Choisir la source de données requise

Toutes les connexions existantes et les objets racine de chacun d'entre eux sont listés, respectivement, dans deux listes de choix dans la barre d'outils de la fenêtre de Requête de la base de données (*capture d'écran ci-dessous*).



Dans la capture d'écran ci-dessus, la base de données portant le nom `StyleVision DB` a été sélectionnée. Parmi les objets de racine disponibles pour cette base de données, l'objet de racine `ALTOVA_USER` a été sélectionné. La base de données et l'objet racine sont ensuite affichés dans le volet Navigateur.

28.10.1.2 Volet Navigateur : consulter les objets BD

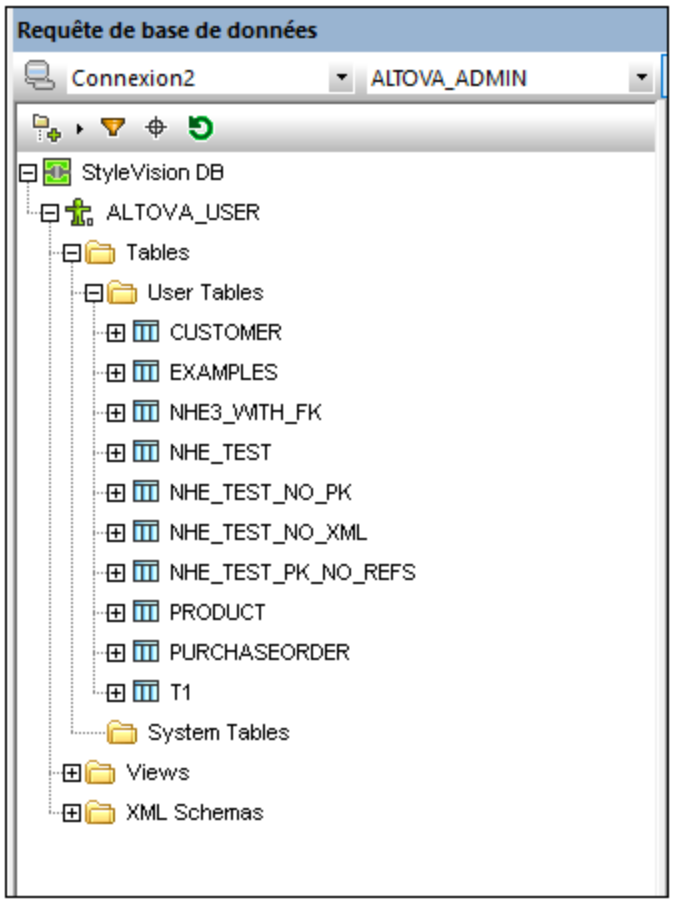
Le volet Navigateur fournit un aperçu des objets dans la base de données sélectionnée. Cet aperçu comprend des informations de contraintes de base de données, comme par exemple si une colonne est un clé primaire ou étrangère. Dans les bases de données IBM DB2 version 9, le Navigateur montre en plus des schéma XML enregistrés dans un dossier séparé.

Cette section décrit les éléments suivants :

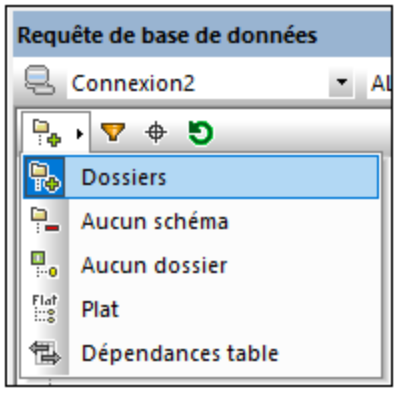
- Les mises en page disponibles dans le volet Navigateur.
- Comment filtrer des objets de base de données.
- Comment trouver des objets de base de données.

Mises en page du volet Navigateur

La mise en page par défaut de Dossiers affiche les objets de base de données de manière hiérarchique. Selon l'objet sélectionné, des options de menu contextuel différentes sont disponibles lorsque vous cliquez sur un item avec la touche de droite.



Pour sélectionner une mise en page pour le Navigateur, cliquer sur l'icône de mise en page dans la barre outils du volet Navigateur et choisir la mise en page depuis la liste déroulante (*capture d'écran ci-dessous*). Veuillez noter que l'icône change avec la mise en page sélectionnée.



Les mises en page disponibles sont :

- *Dossiers :* Organise les objets de base de données dans des dossiers sur la base d'un type d'objet dans une arborescence hiérarchique, il s'agit du paramètre par défaut.

© 2016-2022 Altova GmbH

- *Aucun schéma :* Semblable à la mise en page des Dossiers, sauf qu'il n'y a pas de dossiers de schéma de base de données ; les tables ne sont donc pas catégorisées par un schéma de base de données.
- *Aucun dossier :* Affiche des objets de base de données dans une hiérarchie sans utiliser de dossiers.
- *Plat:* Divise les objets de base de données par type dans le premier niveau hiérarchique. Par exemple, au lieu de colonnes contenues dans la table correspondante, toutes les colonnes sont affichées dans un dossier Colonnes séparé.
- *Dépendances de table :* Catégorise les tables conformément à leurs relations avec les autres tables. Il existe des catégories pour les tables avec des clés étrangères, des tables référencées par des clés étrangères et des tables qui n'ont pas de relations avec d'autres tables.

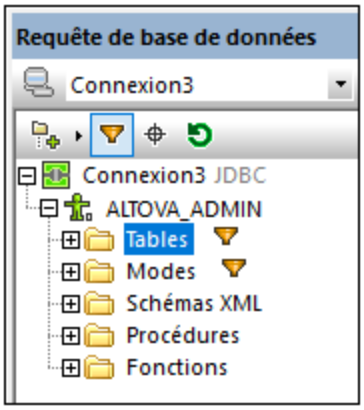
Pour trier les tables en tables Utilisateur et Système, passer à la mise en page Dossiers, Aucun schéma ou Plat, puis cliquer avec la touche de droite sur le dossier Tables et sélectionner **Trier dans les tables Utilisateur et Système**. Les tables sont triées alphabétiquement dans les dossiers Tables utilisateur et Tables système.

Filtrer les objets de base de données

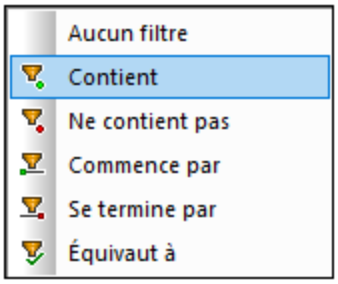
Dans le volet Navigateur (dans toutes les mises en page sauf Aucun dossier et Dépendances de table), les schémas, tables et modes peuvent être filtrés par un nom ou une partie de nom. Les objets sont filtrés pendant que vous saisissez les caractères, et le filtre est insensible à la casse par défaut.

Pour filtrer les objets dans le Navigateur, procéder comme suit :

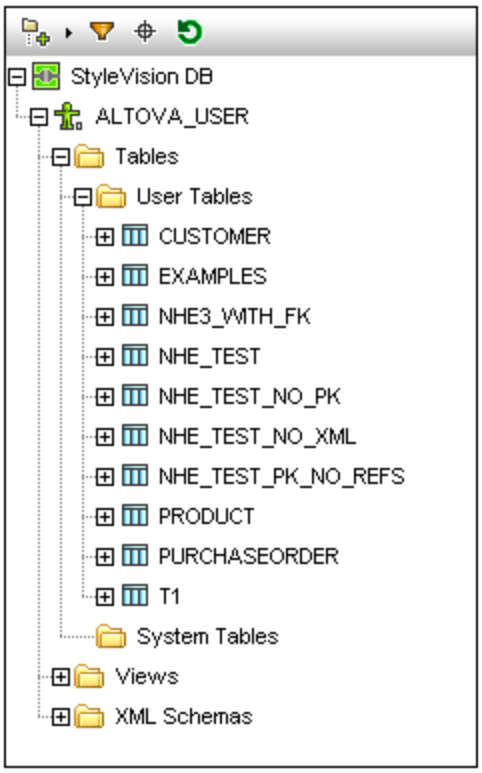
1. Cliquer sur l'icône Filtrer les contenus de dossier dans la barre d'outils du volet Navigateur. Les icônes de filtre apparaissent à côté des dossiers Tables et Modes dans la mise en page actuellement sélectionnée (*capture d'écran ci-dessous*).



2. Cliquer sur l'icône de filtrer à côté du dossier que vous souhaitez filtrer et choisir l'option de filtre depuis le menu qui s'ouvre, par exemple `Contains`.



3. Dans le champ de saisie qui apparaît, saisir le string du filtre (dans la capture d'écran ci-dessous, le string de filtre dans le dossier `Tables` est `NHE`). Le filtre est appliqué pendant que vous saisissez.

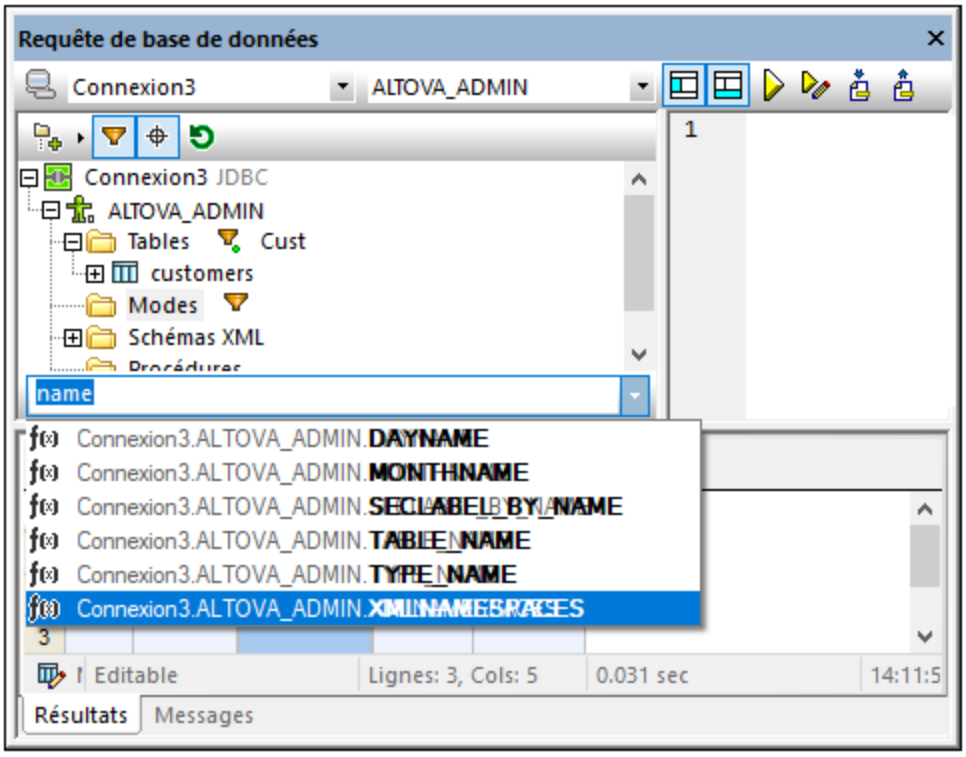


Trouver des objets de base de données

Pour trouver un item de base de données spécifique par son nom, vous pouvez utiliser le Localisateur d'objet du volet de Navigateur. Il fonctionne comme suit :

1. Dans la barre d'outils du volet Navigateur, cliquer sur l'icône du Localisateur d'objet. Une liste déroulante apparaît en bas du navigateur.
2. Saisir le string de recherche dans le champ de saisie de cette liste, par exemple `name` (*capture d'écran ci-dessous*). Cliquer sur la flèche pour afficher tous les objets contenant le string de recherche.

© 2016-2022 Altova GmbH



3. Cliquer sur l'objet dans la liste pour le voir dans le Navigateur.

28.10.1.3 Volet Requête : description et fonctions

Le volet Requête est un éditeur SQL intelligent pour la saisie de requêtes dans la base de données sélectionnée. Une fois avoir saisi la requête, cliquer sur la commande Exécuter de la fenêtre Requête de base de données pour exécuter la requête et afficher le résultat et les messages d'exécution dans le Volet Résultats/Messages. Vous trouverez une description du travail avec les requêtes dans la section suivante, Volet Requête : travailler avec des requêtes. Dans cette section, nous décrivons les fonctions principales du panneau Requête :

- Icônes de l'Éditeur SQL dans la barre d'outils de la Requête de base de données
- Options Éditeur SQL
- Remplissage automatique des déclarations SQL
- Définition des régions dans un script SQL
- Insertion des commentaires dans un script SQL
- Utilisation des signets

Icônes Éditeur SQL dans la barre d'outils de la Requête de base de données

Les icônes suivantes dans la barre d'outils de la fenêtre Requête de base de données sont utilisées lors du travail avec l'Éditeur SQL :

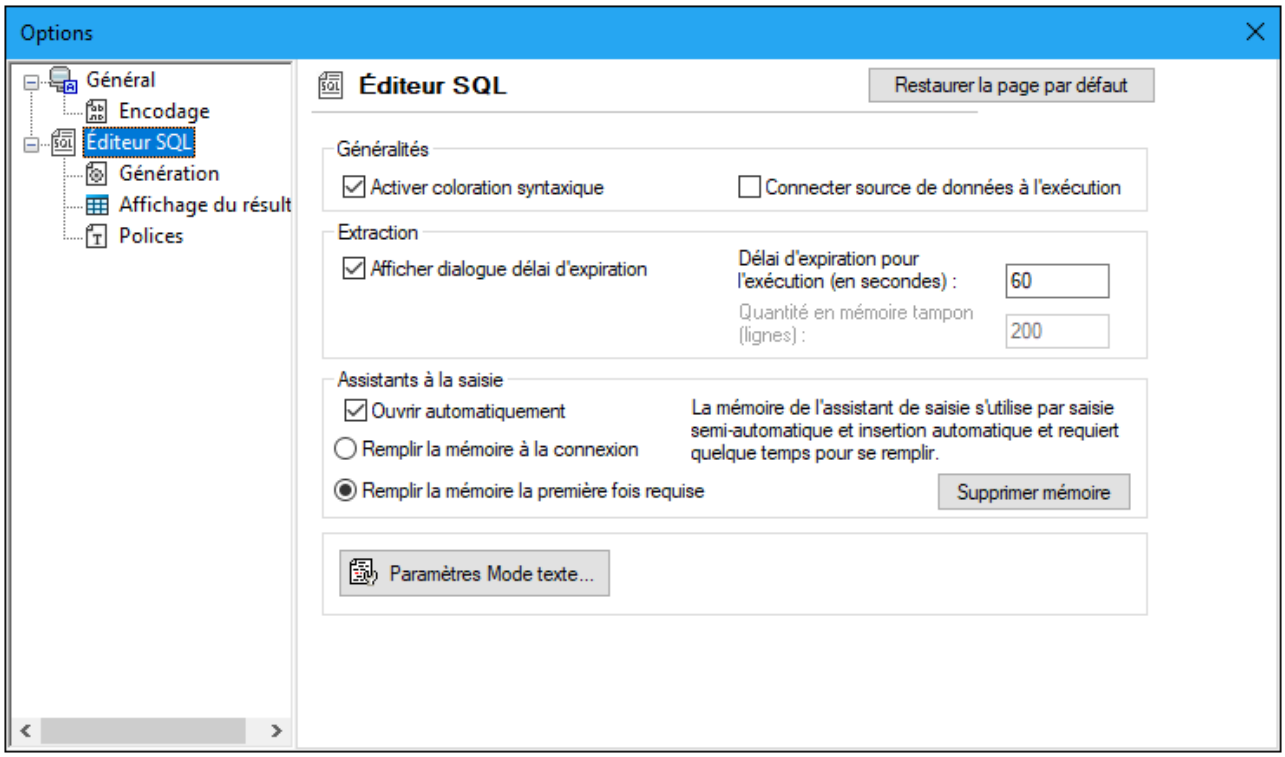
	**Exécuter**	Exécute la déclaration SQL sélectionnée actuellement. Si le

		script contient plusieurs déclarations et qu'aucune n'est sélectionnée, toutes les déclarations seront exécutées.
	**Exécuter avec édition des données**	Identique à la commande Exécuter, sauf que les résultats (dans l'onglet Résultats) sont éditables.
	**Importer fichier SQL**	Ouvre un fichier SQL dans l'Éditeur SQL.
	**Exporter fichier SQL**	Enregistre les requêtes SQL dans un fichier SQL.
	**Annuler**	Annule un nombre illimité d'éditions dans l'Éditeur SQL.
	**Rétablir**	Rétablit un nombre illimité d'éditions dans l'Éditeur SQL.
	**Masquer Requête BD à l'ouverture de XML**	Définit si la fenêtre Requête de BD doit être masquée lorsqu'un document XML est ouvert pour l'édition.
	**Validation automatique à enregistrement XML**	Lorsqu'un document XML édité est enregistré dans XMLSpy, les changements sont validés dans la BD si cette option est activée. Sinon, les changements doivent être validés explicitement dans le volet Résultats.
	**Options**	Ouvrir le dialogue Options de l'Éditeur SQL.
	**Ouvrir Script SQL dans DatabaseSpy**	Ouvre le script SQL dans le produit DatabaseSpy d'Altova.

Options

Cliquer sur l'icône **Options** dans la barre d'outils Requête de base de données pour ouvrir le dialogue Options (*capture d'écran ci-dessous*). Un page de paramètres peut être sélectionnée dans le volet de gauche et les options sur cette page peuvent être sélectionnées. Cliquer sur la touche **Restaurer la page par défaut** de manière à réinitialiser les options sur cette page à leurs paramètres d'origine.

© 2016-2022 Altova GmbH



Les paramètres principaux sont les suivants :

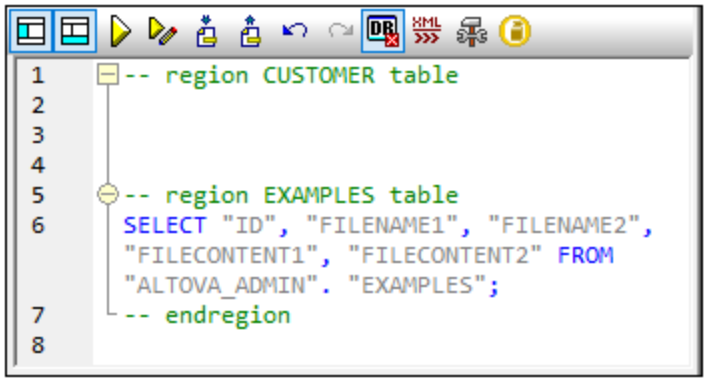
- ***Général | Encodage :*** Options pour définir l'encodage des nouveaux fichiers SQL, des fichiers SQL existants pour lesquels l'encodage ne peut pas être détecté et pour le réglage du Byte Order Mark (BOM). (Si l'encodage des fichiers SQL existants peut être détecté, les fichiers sont ouverts et enregistrés sans changer l'encodage.)
- ***Éditeur SQL :*** les options pour activer et désactiver l'exécution de la coloration syntaxique et les connexions de source de données. Un timeout peut être défini pour l'exécution de la requête, et un dialogue pour modifier le timeout peut aussi être affiché si le délai spécifique est dépassé. Les assistants à la saisie réfèrent aux assistants à la saisie qui apparaissent en tant que partie de la fonction de remplissage automatique. Lorsque vous saisissez des déclarations SQL, l'éditeur affiche une liste de suggestions de remplissage automatique sensible au contexte. Ces suggestions peuvent être définies pour apparaître automatiquement. Si l'affichage automatique est éteint, vous pouvez demander une suggestion de remplissage automatique dans l'Éditeur SQL en appuyant sur **Ctrl+barre espace**. Le tampon pour les informations de l'assistant à la saisie peut être rempli soit lors de la connexion à la source de donnée, soit la première fois qu'elles sont nécessaires. La touche Paramètres Mode Texte ouvre la fenêtre des options du Mode Texte de XMLSpy.
- ***Éditeur SQL | Génération SQL :*** l'application génère des déclarations SQL lorsque vous glissez des objets depuis le volet Navigation dans le volet Requête. Les options pour la génération de déclaration SQL peuvent être définies dans l'onglet de génération SQL. Utiliser le volet *Base de données* pour sélectionner un type de base de données et définir les options de génération de déclaration individuellement pour les différents types de base de données avec lesquels vous travaillez. Cocher la case *Appliquer à toutes les bases de données* pour régler les options actuellement sélectionnées pour toutes les bases de données. Les options contiennent l'ajout de point-virgule aux déclaration et d'entourer les identifieurs avec des caractères d'échappement. Si la case à cocher *Ajouter points-virgules à la fin de la déclaration* est activée, Un point-virgule sera ajouté lorsque vous générez une déclaration SQL dans l'Éditeur SQL. Veuillez noter que l'édition des données dans les bases de

données Oracle et les bases de données IBM iSeries et DB2 via une connexion JDBC est uniquement possible si cette case est décochée.

- ***Éditeur SQL | Affichage du résultat :*** Options pour configurer l'onglet Résultat.
- ***Éditeur SQL | Polices :*** Options pour configurer le style de police du texte dans l'éditeur de texte et dans le mode de Résultat.

Définition des régions dans un script SQL

Les Régions sont des sections dans des scripts SQL qui sont marquées et déclarées comme constituant une unité. Les régions peuvent être réduites et agrandies pour dissimuler ou montrer des parties du script. Il est également possible d'imbriquer des régions dans d'autres régions. Les régions sont délimitées par des commentaires `--region` et `--endregion`, respectivement, avant et après la région. En option, les régions peuvent recevoir un nom qui est saisi après le délimiteur `-- region` (*voir capture d'écran ci-dessous*).



Pour insérer une région, sélectionner les déclarations à effectuer dans une région, cliquer avec la touche de droite et choisir **Insérer Région**. La région agrandissable/réduisible est créée. Ajouter un nom si vous le souhaitez. Dans la capture d'écran ci-dessus, veuillez noter la numérotation de ligne. Pour supprimer une région, supprimer les deux délimiteurs `--region` et `--endregion`.

Insertion des commentaires dans un script SQL

Le texte contenu dans un script SQL peut être décommenté. Ces portions du script sont sautées lorsque le script est exécuté.

- Pour décommenter un bloc, marquer le bloc, cliquer avec la touche de droite et sélectionner **Insérer/Supprimer commentaire de bloc**. Pour supprimer le commentaire de bloc, marquer le commentaire, cliquer avec la touche de droite et sélectionner **Insérer/Supprimer commentaire de bloc**.
- Pour décommenter une ligne ou une partie de ligne, placer le curseur à l'endroit où le commentaire de ligne devrait démarrer, cliquer avec la touche de droite et sélectionner **Insérer/Supprimer commentaire de ligne**. Pour supprimer le commentaire de ligne, marquer le commentaire, cliquer avec la touche de droite et sélectionner **Insérer/Supprimer commentaire de ligne**.

Utilisation des signets

Les signets peuvent être insérés à des lignes spécifiques et vous pouvez ensuite parcourir les signets dans le document. Pour insérer un signet, placer le curseur dans la ligne à marquer, cliquer avec la touche de droite et sélectionner **Insérer/Supprimer le signet**. Pour vous rendre au signet suivant ou précédent, cliquer avec la touche de droite et sélectionner **Aller au signet suivant** ou **Aller au signet précédent**, respectivement. Pour

© 2016-2022 Altova GmbH

supprimer un signet, placer le curseur dans la ligne où vous souhaitez supprimer le signet, cliquer avec la touche de droite et sélectionner **Insérer/Supprimer signet**. Pour supprimer tous les signets, cliquer avec la touche de droite et sélectionner **Supprimer tous les signets**.

28.10.1.4 Volet Requête : travailler avec des requêtes

Une fois vous être connecté à une base de données, un script SQL peut être saisi dans l'Éditeur SQL et être exécuté. Cette section décrit :

- Comment un script SQL est saisi dans l'Éditeur SQL.
- Comment le script est exécuté dans la fenêtre de Requête de la base de données.

Les icônes suivantes sont mentionnées dans cette section :

	**Exécuter la requête**	Exécute une déclaration SQL sélectionnée actuellement. Si le script contient plusieurs déclarations et qu'aucune n'est sélectionnée, toutes seront exécutées.
	**Exécuter pour l'édition des données**	Identique à la commande Exécuter, sauf que les résultats (dans l'onglet Résultats) sont éditables.
	**Importer fichier SQL**	Ouvre un fichier SQL dans l'Éditeur SQL.

Créer des déclarations SQL et des scripts dans l'Éditeur SQL

Les méthodes de GUI suivantes peuvent être utilisées pour créer des déclarations SQL ou des scripts :

- *Glisser et déposer :* glisser un objet depuis le volet Navigateur dans l'Éditeur SQL. Une déclaration SQL est générée pour requêter la base de données pour cet objet.
- *Menu contextuel :* cliquer avec la touche de droite sur un objet dans le volet Navigateur et choisir **Afficher dans Éditeur SQL | Sélectionner**.
- *Saisie manuelle :* saisir les déclarations SQL directement dans l'Éditeur SQL. La fonction de remplissage automatique peut aider avec l'édition.
- *Importer un script SQL :* cliquer sur l'icône **Importer fichier SQL** dans la barre d'outils de la fenêtre Requête de base de données.

Exécuter des déclarations SQL

Si le script SQL dans l'Éditeur SQL a plus d'une déclaration SQL, choisir la déclaration à exécuter et cliquer sur l'icône **Exécuter** ou **Exécuter avec édition des données** dans la barre d'outils de la fenêtre Requête de la base de données. Si aucune déclaration n'est sélectionnée dans le script SQL, toutes les déclarations dans le script seront exécutées. Les données de la base de données sont extraites et affichées en tant que grille dans l'onglet Résultats. Si vous avez sélectionné **Exécuter avec édition des données**, les données extraites dans la Grille de résultat peuvent être éditées. Des messages concernant l'exécution sont affichés dans l'onglet Messages.

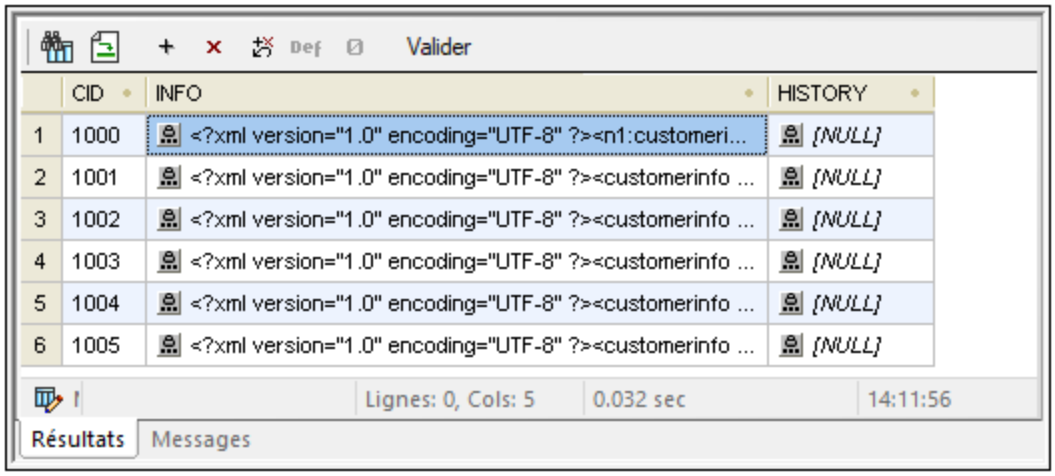
28.10.1.5 Résultats et Messages

Les volets Résultats/Messages ont deux onglets :

- Le volet Résultats affiche les données qui sont extraites par la requête.
- Le Volet Messages affiche les messages concernant l'exécution de la requête.

Onglet Résultats

Les données extraites par la requête sont affichées sous la forme d'une grille dans l'onglet Résultats (*capture d'écran ci-dessous*). Lorsque les résultats de requête ont été générés à l'aide de la commande **Exécuter Requête**, les documents XML dans l'onglet Résultats sont indiqués par l'icône XML (*capture d'écran ci-dessous*). Si la commande **Exécuter pour l'édition des données** a été utilisée, les documents XML sont montrés avec l'icône XML éditable.

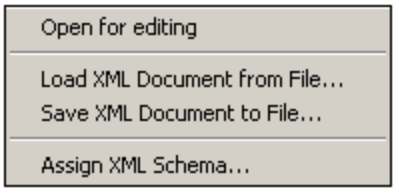


Les opérations suivantes peuvent être effectuées dans l'onglet Résultats, par le biais du menu contextuel qui s'ouvre lorsque vous cliquez dans l'emplacement approprié avec la touche de droite dans l'onglet Résultats:

- *Trier dans une colonne :* cliquer avec la touche de droite n'importe où dans la colonne sur laquelle les enregistrements doivent être triés, puis sélectionner **Trier | Croissant/Décroissant/Réinitialiser défaut**.
- *Copier dans le presse-papiers :* cela consiste en deux étapes : (i) choisir la plage des données ; et (ii) copier la sélection. Les données peuvent être sélectionnées de plusieurs manières : (i) en cliquant sur un en-tête de colonne ou un numéro de ligne pour sélectionner la colonne ou la ligne, respectivement ; (ii) sélectionner des cellules individuelles (utiliser les touches **Shift** et/ou **Ctrl** pour sélectionner des cellules multiples) ; (iii) cliquer avec la touche de droite sur une cellule et choisir **Sélection | Ligne/Colonne/Tout**. Une fois avoir effectué la sélection, cliquer avec la touche de droite et sélectionner **Copier cellules sélectionnées**. Cela copie la sélection dans le presse-papiers, à partir de là elle peut être collée dans une autre application.
- *Ajouter une nouvelle ligne :* si la requête a été exécutée pour l'édition, cliquer avec la touche de droite n'importe où dans le volet Résultats pour accéder à la commande **Ajouter ligne**.
- *Supprimer une ligne :* si la requête a été exécutée pour l'édition, cliquer avec la touche de droite n'importe où dans une ligne pour accéder à la commande **Supprimer ligne**.
- *Éditer des enregistrements :* si la requête a été exécutée pour l'édition, des champs individuels peuvent être édités. Pour valider les changements, cliquer sur la touche **Valider** dans la barre d'outils de l'onglet Résultats.

© 2016-2022 Altova GmbH

- *Éditer des enregistrements XML :* Cette fonction est prise en charge pour les bases de données IBM DB2, SQLServer, PostgreSQL (8.3 et plus) et Oracle (9 et plus) et uniquement pour les tables de BD qui ont une clé primaire. Si la requête a été exécutée pour l'édition et qu'un champ éditable est un champ XML, le fait de cliquer sur l'icône XML Éditable dans la Grille de résultats ouvre le menu Éditer XML (*capture d'écran ci-dessous*). Un champ XML peut aussi être ouvert pour l'édition des données en cliquant avec la touche de droite sur le champ XML se trouvant dans le volet Dossiers et en sélectionnant la commande **Éditer données**.



La commande **Ouvrir pour éditer** ouvre le document XML dans une fenêtre XMLSpy et l'icône XML Éditable se transforme en , les trois points sont maintenant rouges. Lorsque ce document est enregistré et si l'icône Validation automatique des modifications XML dans la barre d'outils Requête de base de données a été sélectionnée lorsque le document a été ouvert, les changements du document XML sont validés automatiquement dans la base de données. Sinon, les modifications enregistrées devront être validées avec la touche Valider du volet Résultats. (Veuillez noter que pour passer entre la fenêtre de document XML et la fenêtre de Requête de base de données, vous devez cliquer sur la commande **BD | Requête de base de données**.) La commande **Charger le Document XML depuis le fichier** charge un document XML externe dans le champ sélectionné de la base de données. La commande **Enregistrer Document XML sous fichier** enregistre le document XML du champ sélectionné de la base de données vers un emplacement de fichier de votre choix. La commande Attribuer Schéma XML ouvre le dialogue Choisir Schéma XML dans lequel vous pouvez sélectionner un Schéma XML à attribuer au document XML. Cette attribution est enregistrée dans la base de données. L'attribution de Schéma XML est expliquée en détail dans la section IBM DB2 | Assigner Schéma XML.

- *Définir NUL, Définir défaut, Annuler les modifications pour cette cellule :* si la requête a été exécutée pour l'édition, cliquer avec la touche de droite dans une cellule permet l'accès aux commandes permettant de définir une valeur `NULL` ou, si définie, une valeur par défaut de colonne pour cette cellule. Les changements effectués dans une cellule peuvent être annulés avec la commande **Annuler les changements pour cette cellule** ; la valeur éditée actuellement est remplacée par la valeur actuelle dans la BD.

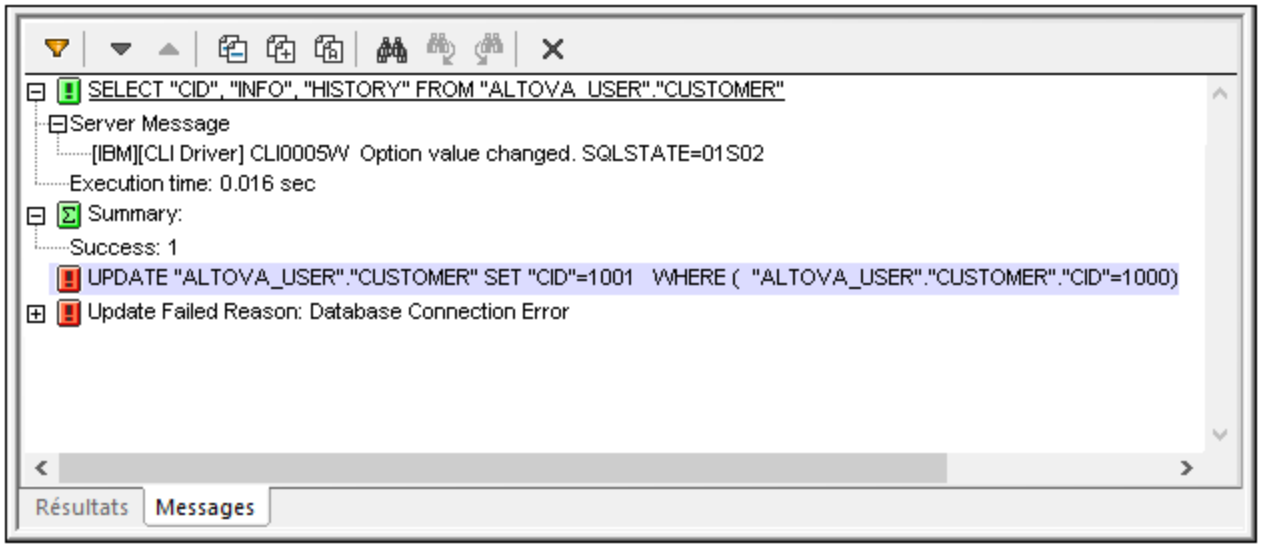
L'onglet Résultats dispose des icônes de barre d'outils suivantes :

	Aller à l'instruction	Marque l'instruction dans l'Éditeur SQL qui a produit le résultat actuel.
	Trouver	Trouve un texte dans le volet Résultats. Le contenu du document XML est aussi parcouru.
	Ajouter nouvelle ligne	Ajoute une nouvelle ligne à la Grille Résultat.
	Supprimer ligne	Supprime la ligne actuelle dans la Grille Résultat.
	Annuler les changements de la Grille de résultat	Annule toutes les modifications effectuées dans la Grille Résultat.

Commit	**Valider**	Valide les modifications effectuées dans la Grille Résultat dans la base de données.

Onglet Messages

L'onglet Messages fournit l'information concernant l'instruction SQL précédemment exécutée et rapporte des erreurs ou des messages d'avertissement.



La barre d'outils de l'onglet Messages contient des icônes qui vous permettent de personnaliser l'affichage, de le parcourir et de copier les messages dans le presse-papiers. L'icône **Filtrer** permet d'activer ou de désactiver l'affichage de types de message particuliers. Les icônes **Suivant** et **Précédent** déplacent la sélection vers le haut ou le bas de la liste, respectivement. Les messages peuvent aussi être copiés avec ou sans leurs composants enfant sur le presse-papiers, leur permettant d'être collés dans des documents. La fonction **Trouver** vous permet de spécifier un terme de recherche et d'effectuer une recherche dans la liste. Enfin, l'icône **Effacer** efface les contenus du volet Messages.

Note : Ces icônes de barre d'outils sont aussi disponibles en tant que commandes de menu contextuel.

28.10.2 IBM DB2

L'item de menu **IBM DB2** déroule un sous-menu contenant des commandes (i) pour enregistrer et désenregistrer des schémas avec une base de données IBM DB2 (**Gérer Schémas XML**), et (ii) pour attribuer des schémas pour une validation de fichier XML (**Attribuer Schéma XML**).

Ces deux mécanismes exigent que vous vous connectiez à la base de données IBM DB2 requise. Pour un exemple de connexion, voir l'Exemple : IBM DB2 (ODBC). Dans cette section, l'attention est portée sur la gestion des schémas dans une base de données IBM DB2 et comment attribuer des Schémas XML à un fichier XML BD.

Note : La grille de résultat de la fenêtre de requête de base de données fournit une fonction importante pour travailler avec les fichiers XML dans les bases de données IBM DB2. Cette fonction comprend la

© 2016-2022 Altova GmbH

capacité d'ouvrir des fichiers pour l'édition, le chargement de fichiers XML dans des cellules BD en tant que fichiers XML, enregistrer ces fichiers XML BD en externe, et d'attribuer des Schémas XML à des fichiers XML BD.

28.10.2.1 Gérer les Schémas XML

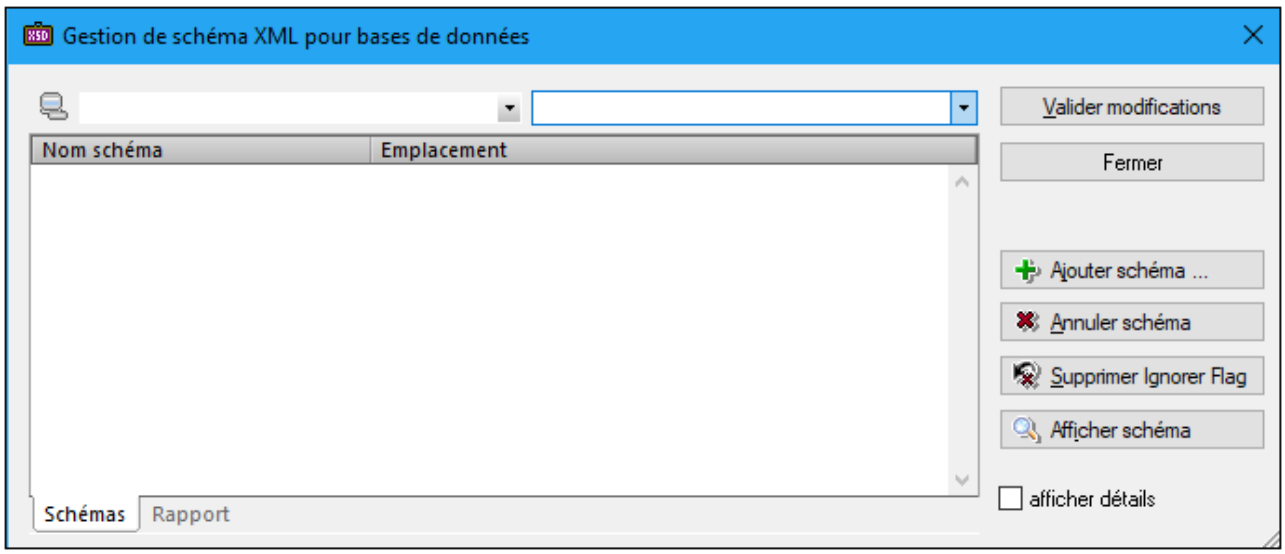
La fonction **Gérer les schéma XML** permet d'ajouter et d'extraire des schémas dans des schémas de bases de données individuelles dans un base de données IBM DB2. Pour gérer les schémas, vous devez procéder comme suit :

- Connectez-vous à la base de données IBM DB2
- Choisir le schéma de base de données pour lequel les schémas doivent être ajoutés ou abandonnés
- Exécuter les actions de gestion de schéma.

Ces étapes sont décrites en détail ci-dessous.

Se connecter à une base de données IBM DB2

Cliquer sur la commande **Gérer les schémas XML** pour ouvrir la Gestion de Schéma XML pour le dialogue Bases de données (*capture d'écran ci-dessous*).

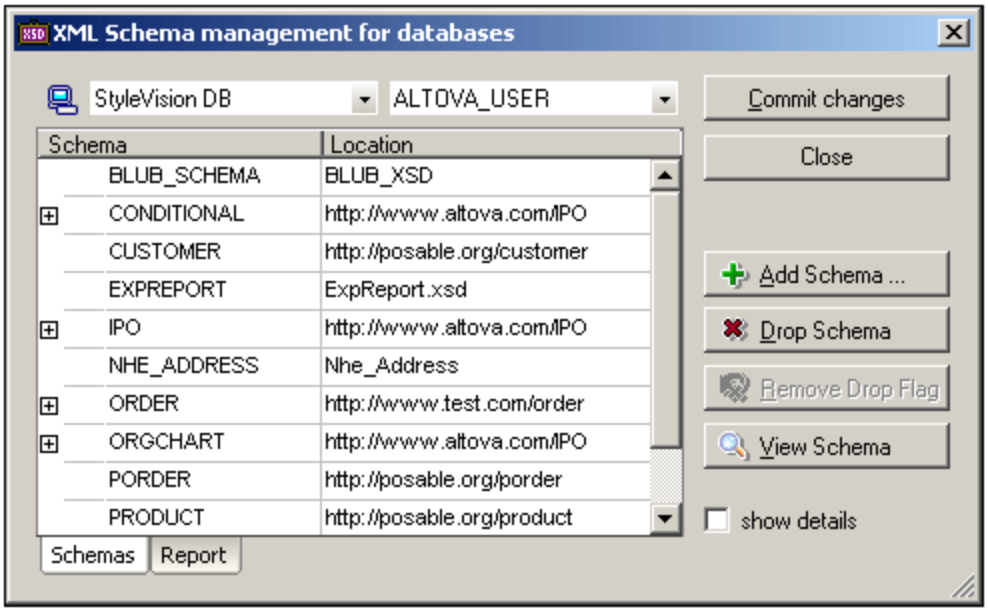


La première chose à faire, s'il n'y a pas de connexion à la base de données requise est de vous y connecter. Si une connexion existe déjà, elle apparaît dans la liste de choix de la Base de données. Pour lancer le processus de connexion, cliquer sur l'icône Connexion rapide dans le dialogue. Le dialogue Connexion rapide s'ouvre, avec lequel vous pouvez établir la connexion à la base de données (pour plus de détails, voir Se connecter à une base de données).

Afficher la liste des schémas XML

Une fois que la connexion à la base de données IBM DB2 a été établie, la base de données est listée dans la liste de choix à gauche (*voir capture d'écran ci-dessous*). Si plus d'une connexion est ouverte actuellement,

vous pouvez sélectionner la base de données requise dans cette liste de choix. Dans la capture d'écran ci-dessous, la base de données StyleVision DB a été sélectionnée.



La liste de choix de droite liste tous les schémas de base de données de la base de données IBM DB2 actuellement sélectionnée. Lorsqu'un schéma de base de données est sélectionné dans cette liste de choix, tous les Schémas XML enregistrés pour le schéma de base de données sélectionné sont affichés dans le volet principal. Dans la capture d'écran ci-dessus, tous les Schémas XML enregistrés avec le schéma de base de données `Altova_User` sont listés, avec leurs emplacements. Cocher la case Afficher détails pour afficher des informations supplémentaires dans le volet principal.

Gérer les Schémas XML

La liste des schémas dans le volet principal représente les schémas enregistrés pour le schéma de base de données sélectionné. Une fois que la liste des Schémas XML est affichée, vous pouvez ajouter des schémas à la liste ou supprimer des schémas depuis cette liste.

Pour ajouter un schéma, cliquer sur la touche **Ajouter**, chercher le fichier de schéma requis et le sélectionner. Le fichier de schéma sélectionné est ajouté à la liste dans le volet principal. Cliquer sur la touche **Valider les modifications** pour enregistrer le schéma récemment ajouté avec le schéma de base de données.

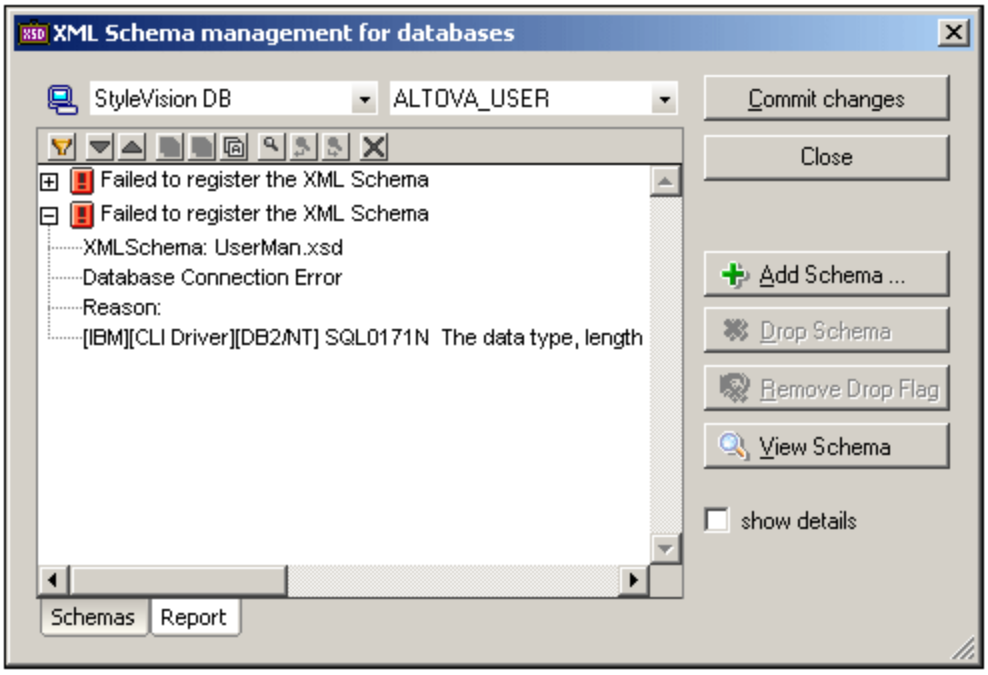
Pour annuler un schéma, sélectionner le schéma dans le volet principal et cliquer sur la touche **Annuler schéma**. Un flag Annuler est attribué au schéma, indiquant qu'il est programmé pour l'annulation lors de la prochaine validation. Le flag Annuler peut être supprimé en sélectionnant le schéma marqué du flag et en cliquant sur la touche **supprimer flag Annuler**. Lorsque la touche **Valider modifications** est cliquée, tous les schémas qui ont été marqués à l'annulation seront désenregistrés depuis le schéma de la base de données.

Cliquer sur la touche Afficher Schéma pour ouvrir le schéma dans XMLSpy. Pour fermer le dialogue Gestion de schéma XML, cliquer sur la touche **Fermer**.

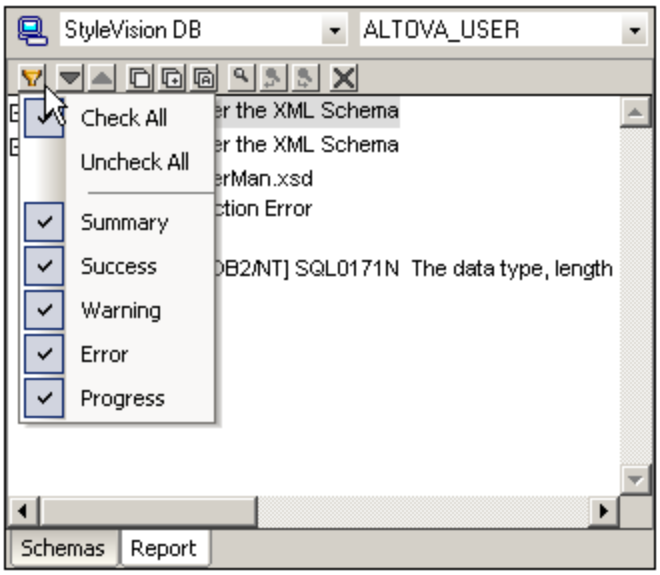
© 2016-2022 Altova GmbH

Rapports

Lorsque la touche **Valider modifications** est cliquée, la base de données est modifiée conformément aux modifications que vous avez effectuées. Un rapport de l'action de validation est affiché dans le volet Valider (*capture d'écran ci-dessous*), vous permettant d'évaluer le succès de l'action et d'effectuer un débogage le cas échéant. Chaque rapport sera affiché en-dessous du rapport précédent.



Le volet Rapport comporte une barre outils contenant des icônes qui vous permettent de personnaliser l'affichage de la liste de rapports, de parcourir la liste, de copier des messages de rapport, de chercher du texte et d'effacer le volet (*voir capture d'écran ci-dessous*).



L'icône **Filtre** permet l'affichage de types particuliers de messages à activer et désactiver. Les icônes **Suivant** et **Précédent** déplacent la sélection vers le haut ou le bas de la liste, respectivement. Vous pouvez aussi copier des messages avec ou sans leurs composants enfants dans le presse-papiers, vous pour les coller dans des documents. La fonction **Chercher** permet de spécifier un terme de recherche et puis de chercher ce terme dans la liste de haut en bas ou inversement. Enfin, l'icône **Supprimer** efface les contenus du volet Rapport.

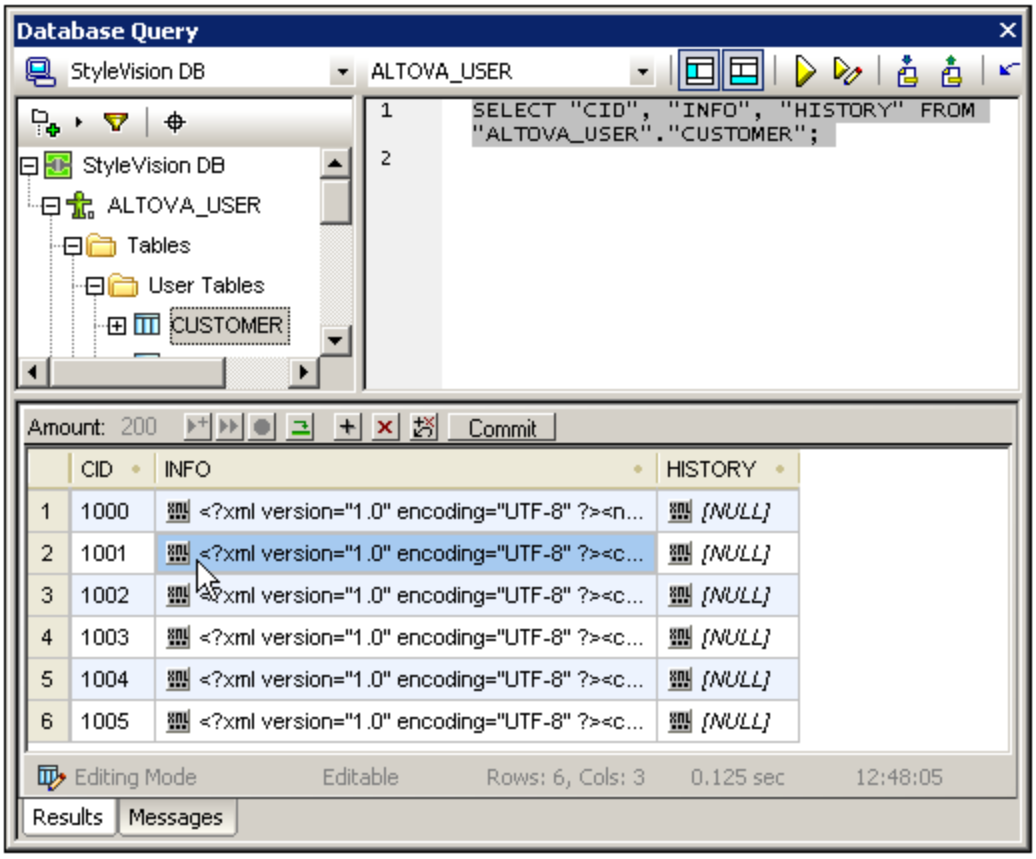
28.10.2.2 Attribuer un Schéma XML

La fonction Attribuer un Schéma XML attribue un schéma à un fichier XML ouvert pour l'édition par le biais de la Grille de Résultat de la fenêtre Requête de base de données. Une fois que l'attribution est effectuée, le fichier XML peut être validé par rapport au schéma attribué. L'attribution est rédigée sur la BD lorsque le fichier XML est enregistré sous XMLSpy.

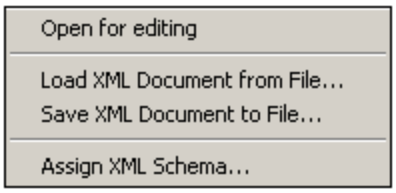
Ouvrir un fichier XML de BD pour l'édition

Dans la fenêtre de Requête de base de données, lorsqu'une requête est adressée à une BD XML et que la requête est exécutée pour l'édition de données, la Grille de Résultats située en bas de la fenêtre de Requête de base de données fournit l'accès aux fichiers XML dans la base de données pour qu'ils puissent être éditées (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



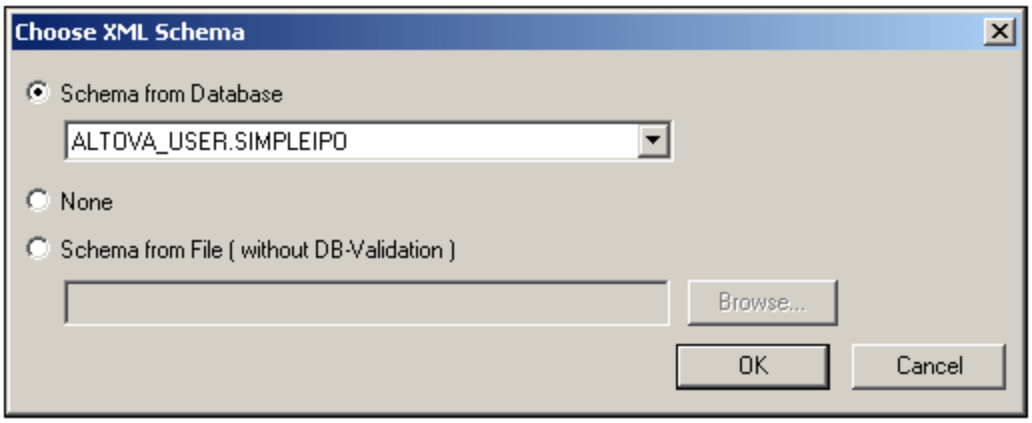
Cliquer sur l'icône XML pour ouvrir le menu suivant.



Choisir la commande **Ouvrir pour éditer** pour ouvrir le document XML dans XMLSpy, où il peut être édité.

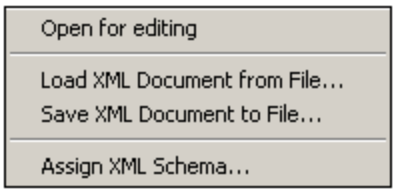
Attribuer un schéma au fichier XML de BD

Ce n'est que lorsque le fichier XML de BD est ouvert pour être édité dans XMLSpy que la commande **IBM DB2 | Attribuer un schéma XML** est activée. Une fois que le document XML est activé dans XMLSpy, cliquer sur la commande **Attribuer Schéma XML** pour ouvrir le dialogue Choisir Schéma XML (*capture d'écran ci-dessous*).



Un schéma peut être choisi à partir de ceux stockés dans la base de données (ils sont listés dans la liste déroulante Schéma de la liste de choix Base de données), ou bien à partir des fichiers externes qui peuvent être parcourus. Cliquer sur **OK** pour attribuer le schéma au fichier XML. Veuillez noter que l'attribution n'est pas écrite dans le fichier XML. Lorsque le fichier XML est enregistré dans XMLSpy—et si l'icône Validation automatique des modifications XML  située dans la barre d'outils Requête base de donnée a été sélectionnée lors de l'ouverture du document, alors l'attribution du schéma est enregistrée dans la base de données. Veuillez noter que l'attribution de schéma est rédigée dans la base de données et **pas** dans le fichier XML.

Note : Le menu Éditer XML dans la Grille de résultats de la fenêtre Requête de base de données dispose également d'une commande **Attribuer Schéma XML** (*voir capture d'écran ci-dessous*), qui attribue aussi un schéma au fichier XML BD.



La différence entre les deux commandes Attribuer Schéma XML est que la commande dans le menu B**D | IBM DB2** vous permet d'attribuer une Schéma XML pendant que vous éditez le fichier XML vous permettant par là même de modifier les attributions du schéma, tout en éditant le document XML et en validant le document XML immédiatement.

28.10.3 Serveur SQL

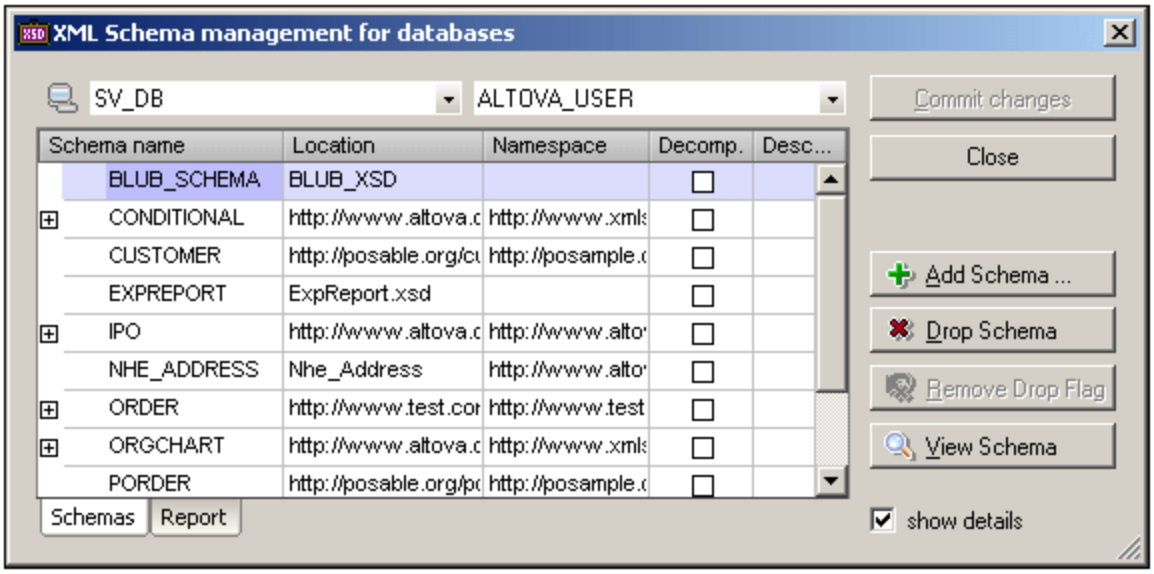
L'item de menu **Serveur SQL** déroule un sous-menu contenant la commande Gérer les Schémas XML.

28.10.3.1 Gérer les Schémas XML

La gestion de Schéma XML pour les bases de données vous permet d'ajouter et de supprimer des Schémas XML depuis l'archivage de schéma d'une base de données XML. Une fois s'être connecté à la base de données, XMLSpy fournit la gestion de Schéma XML pour le dialogue Bases de données, dans lequel les

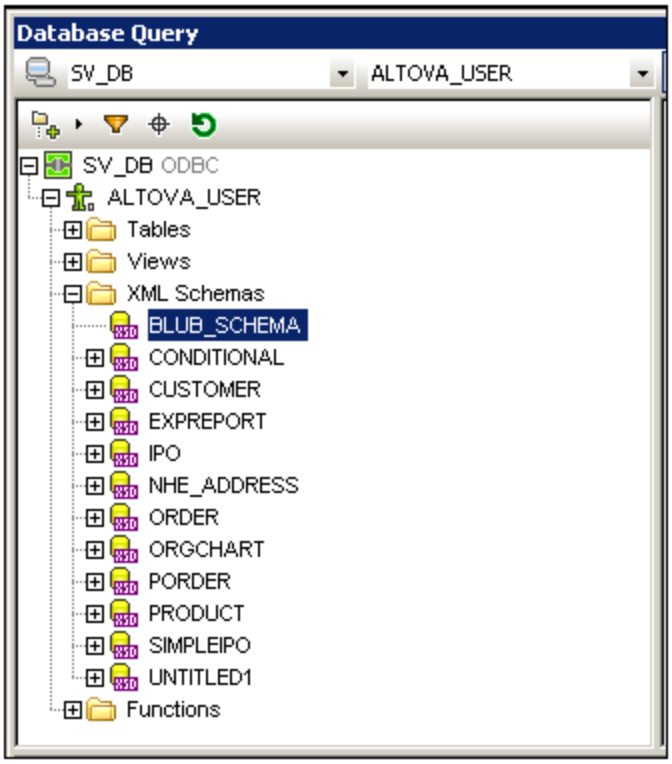
 © 2016-2022 Altova GmbH

Schémas XML peuvent être gérés.



La boîte de dialogue fournit une icône Quick Connect qui appelle Quick Connect wizard á se connecter à la source de données. Si plus d'une connexion existe actuellement, la connexion requise peut être sélectionnée depuis la liste de choix située dans le côté gauche. L'objet racine requis peut ensuite être sélectionné depuis la liste de choix située à droite. Tous les Schémas XML se trouvant actuellement dans l'archivage pour cet objet racine sont affichés dans la fenêtre du dialogue. Le nom, l'emplacement et l'espace de nom de chaque schéma sont listés.

Notez que les schémas stockés peuvent aussi être consultés dans la fenêtre de requête de la base de données *(voir la capture d'écran ci-dessous)*, mais ils ne peuvent pas être gérés ici. Pour gérer les schémas, utilisez le dialogue Gestions des Schémas XML pour les bases de données.

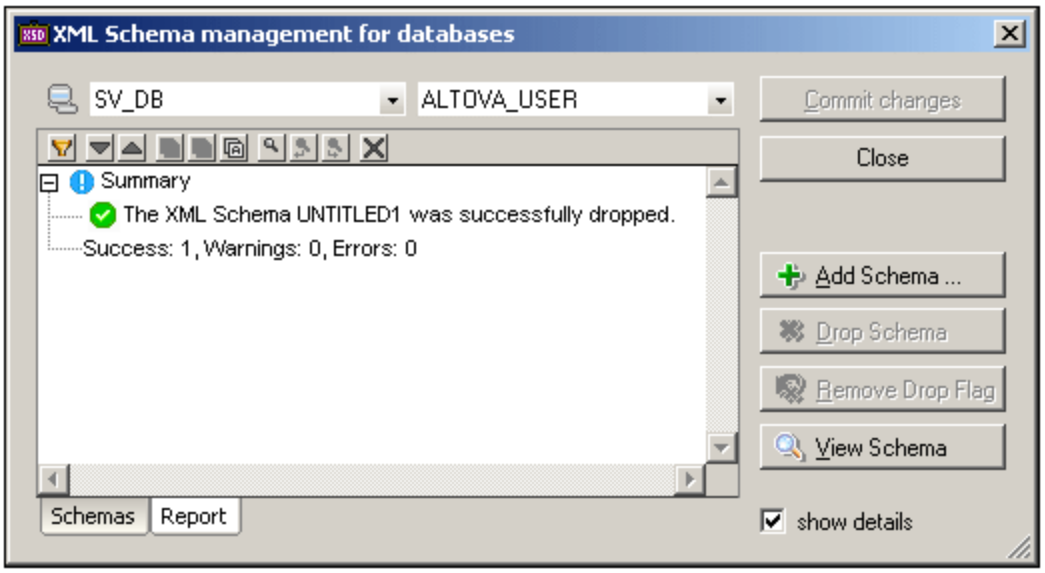


Dans le dialogue Gestion des Schémas XML, vous pouvez faire les choses suivantes :

- Ajouter un schéma utilisant le bouton **Ajouter Schéma**. Le schéma sélectionné sera apposé à la liste et marqué pour l'addition.
- Marquer des schémas dans la liste pour suppression avec la touche **Annuler schéma**. Le Drop flag peut être supprimé avec le bouton **Remove Drop Flag** .
- Ouvrir un schéma sélectionné dans le Mode Schéma en cliquant sur la touche **Afficher schéma**.
- Valider l'ajout et annuler les changements (suppression) avec la touche **Valider modifications**.

Une fois que les changements ont été archivés, un rapport de l'action d'archivage peut être consultée dans l'onglet Rapport (capture d'écran ci-dessous).

© 2016-2022 Altova GmbH



28.10.4 Oracle XML DB

XMLSpy vous permet de vous connecter et de requêter des bases de données DB XML Oracle.

Les fonctions des bases de données suivantes sont prises en charge :

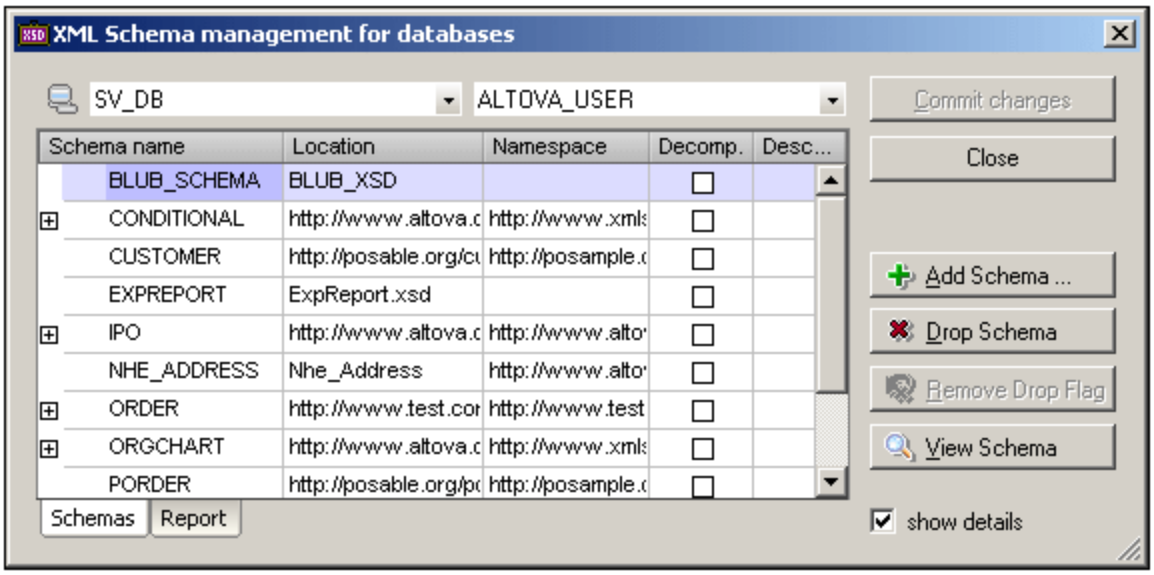
- Ajouter (et enregistrer) un Schéma XML à la DB XML Oracle. Le client DB XML Oracle doit être installé pour que vous puissiez enregistrer les Schémas XML par le biais de XMLSpy.
- Ouvrir et supprimer les schémas
- Requêter la base de donnés à l'aide des déclarations XPath (DBUri)
- Chercher des documents XML (avec WebDAV)
- Créer un document XML sur la base d'un schéma enregistré dans la base de données

Processus d'installation général :

- Télécharger et installer XMLSpy
- Installer serveur Oracle (le cas échéant)
- Créer une base de données Oracle

28.10.4.1 Gérer les Schémas XML

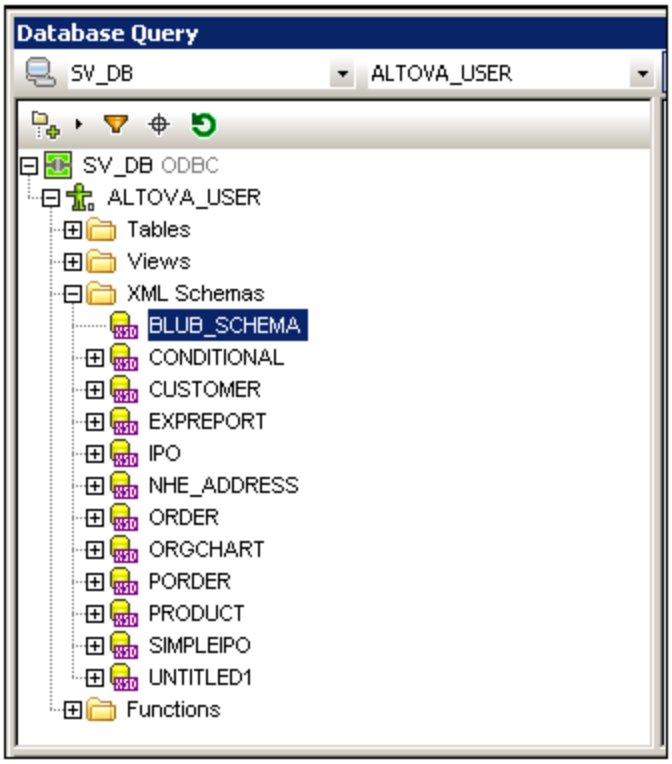
La gestion de Schéma XML pour les bases de données vous permet d'ajouter et de supprimer des Schémas XML depuis l'archivage de schéma d'une base de données XML. Une fois s'être connecté à la base de données, XMLSpy fournit la gestion de Schéma XML pour le dialogue Bases de données, dans lequel les Schémas XML peuvent être gérés.



La boîte de dialogue fournit une icône Quick Connect qui appelle Quick Connect wizard á se connecter à la source de données. Si plus d'une connexion existe actuellement, la connexion requise peut être sélectionnée depuis la liste de choix située dans le côté gauche. L'objet racine requis peut ensuite être sélectionné depuis la liste de choix située à droite. Tous les Schémas XML se trouvant actuellement dans l'archivage pour cet objet racine sont affichés dans la fenêtre du dialogue. Le nom, l'emplacement et l'espace de nom de chaque schéma sont listés.

Notez que les schémas stockés peuvent aussi être consultés dans la fenêtre de requête de la base de données *(voir la capture d'écran ci-dessous)*, mais ils ne peuvent pas être gérés ici. Pour gérer les schémas, utilisez le dialogue Gestions des Schémas XML pour les bases de données.

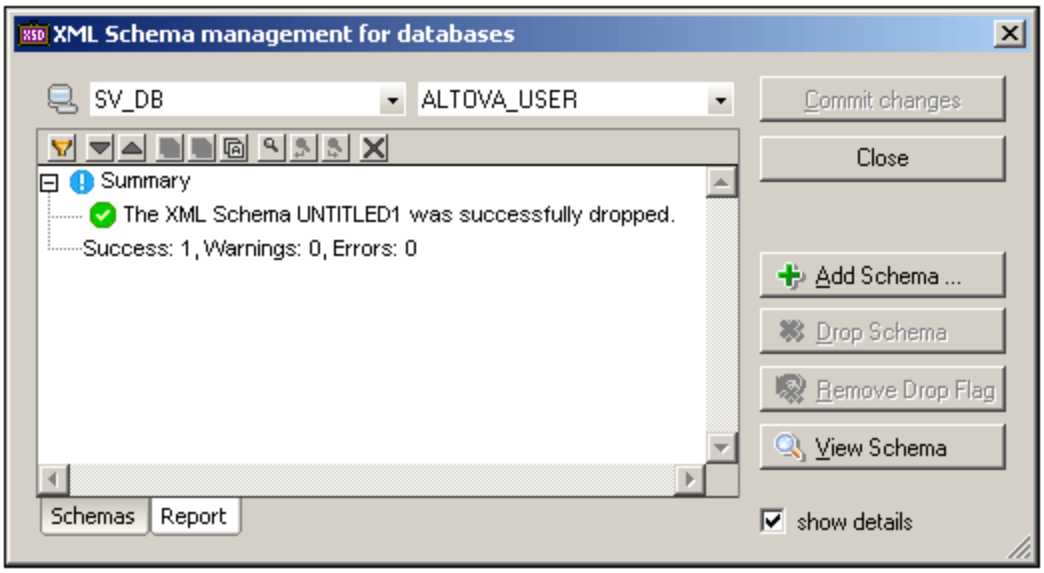
© 2016-2022 Altova GmbH



Dans le dialogue Gestion des Schémas XML, vous pouvez faire les choses suivantes :

- Ajouter un schéma utilisant le bouton **Ajouter Schéma**. Le schéma sélectionné sera apposé à la liste et marqué pour l'addition.
- Marquer des schémas dans la liste pour suppression avec la touche **Annuler schéma**. Le Drop flag peut être supprimé avec le bouton **Remove Drop Flag** .
- Ouvrir un schéma sélectionné dans le Mode Schéma en cliquant sur la touche **Afficher schéma**.
- Valider l'ajout et annuler les changements (suppression) avec la touche **Valider modifications**.

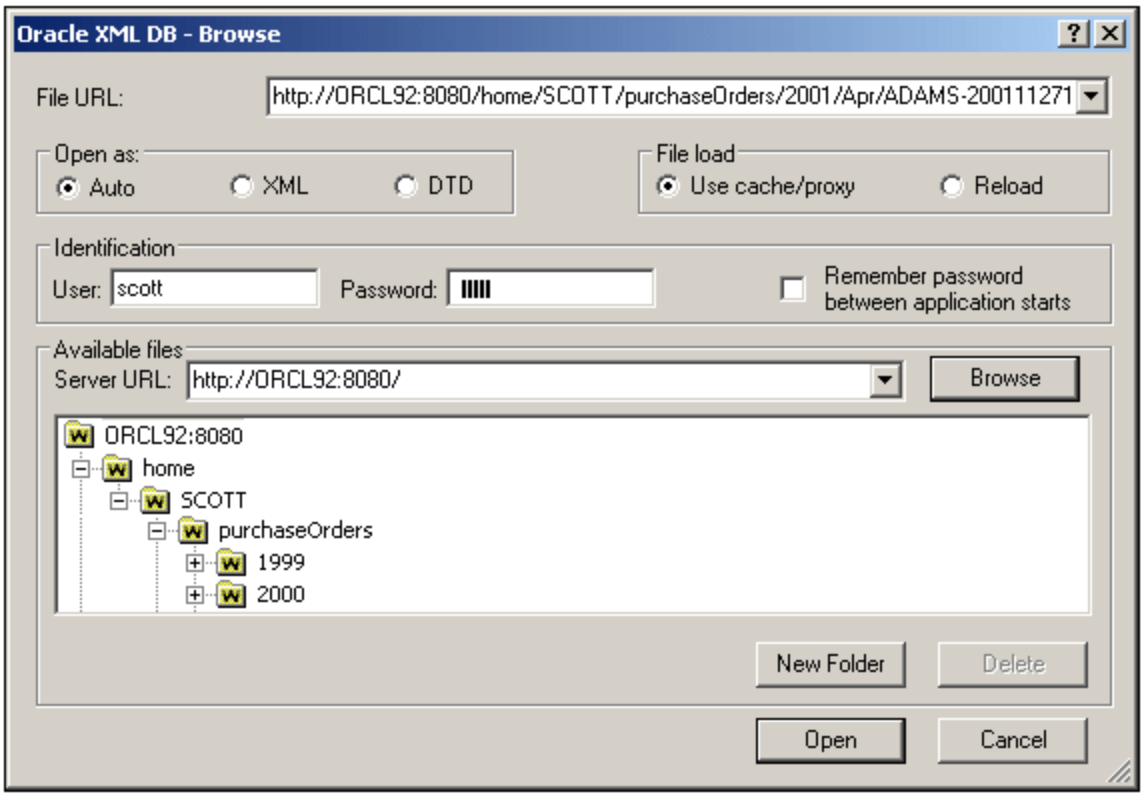
Une fois que les changements ont été archivés, un rapport de l'action d'archivage peut être consultée dans l'onglet Rapport (capture d'écran ci-dessous).



28.10.4.2 Parcourir les document XML Oracle

Cette commande vous permet de parcourir les documents XML disponibles sur votre serveur. Les détails de serveur sont automatiquement remplis si vous avez cherché la base de données auparavant ou si vous avez listé les schémas. Si ce n'est pas le cas, il faudra les saisir manuellement.

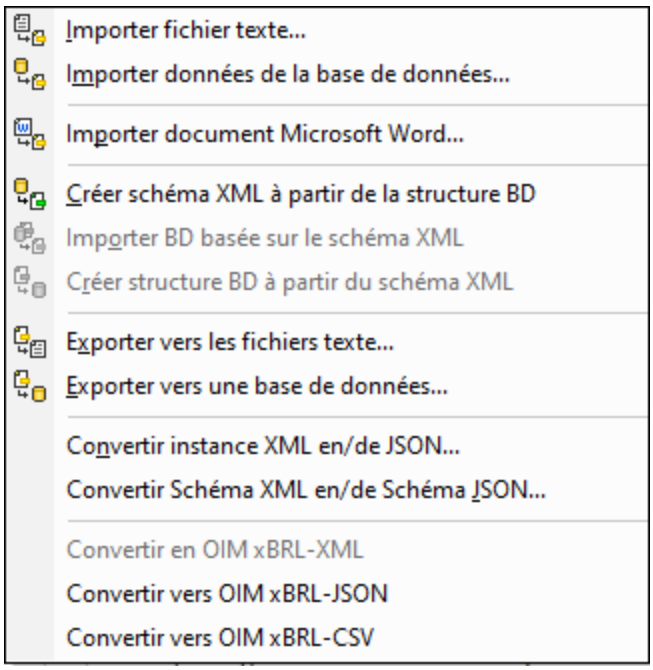
© 2016-2022 Altova GmbH



Utiliser le mode arborescent pour trouver les fichiers XML spécifiques. Double-cliquer sur un fichier dans le mode arborescent pour l'ouvrir. Vous pouvez aussi cliquer sur un fichier puis cliquer sur Ouvrir pour obtenir le même résultat. La touche **Nouveau dossier** ajoute un nouveau dossier, la touche **Supprimer** supprime le fichier XML actuellement sélectionné.

28.11 Menu Conversion

Le menu **Convertir** menu (*voir la capture d'écran ci-dessous*) fournit une fonctionnalité de données puissante d'échange de données entre les formats de données :



- Importer et exporter du texte, traitement de texte, base de données et fichiers XML.
- Importer des données de base de données basées sur un schéma XML existant.
- Créer un schéma XML basé sur la structure d'une base de données existante.
- Créer une structure de base de données, basée sur un schéma XML existant.
- Convertir entre des instances XML et des instances JSON et entre des schémas XML et des schémas JSON.
- Convertir des données XBRL en des représentations OIM de xBRL-JSON et xBRL-CSV, et convertir toute représentation OIM (xBRL-XML, xBRL-JSON et xBRL-CSV) en une autre.

28.11.1 Importer fichier texte

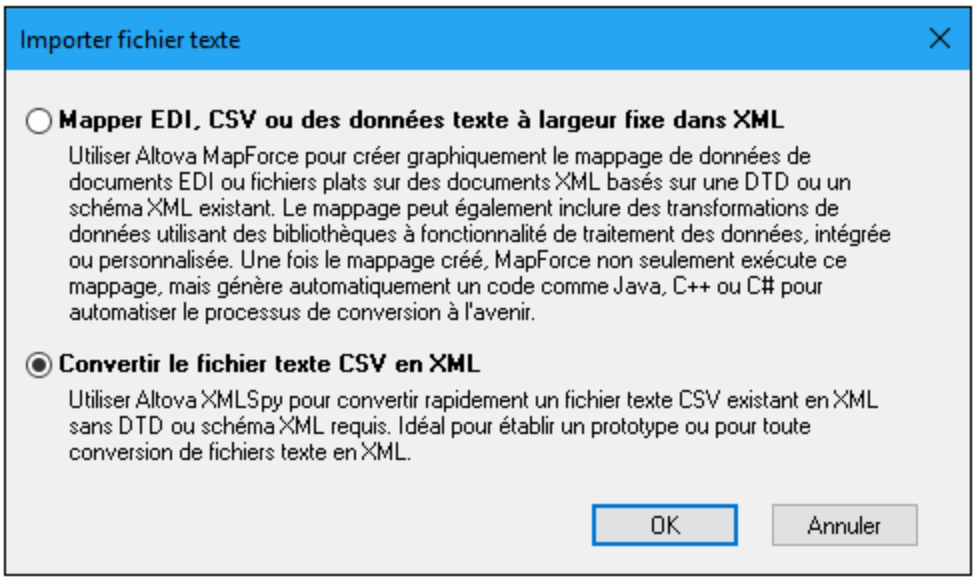


Cette commande vous permet d'**importer** tout fichier de **texte structuré** dans XMLSpy et de le convertir immédiatement en format XML. Cela est utile lorsque vous souhaitez importer des données anciennes provenant de vieux systèmes. Les étapes pour importer les données dans un fichier de texte en tant que document XML sont décrites ci-dessous.

1. Sélectionnez l'item de menu Convertir | Importer fichier texte.

© 2016-2022 Altova GmbH

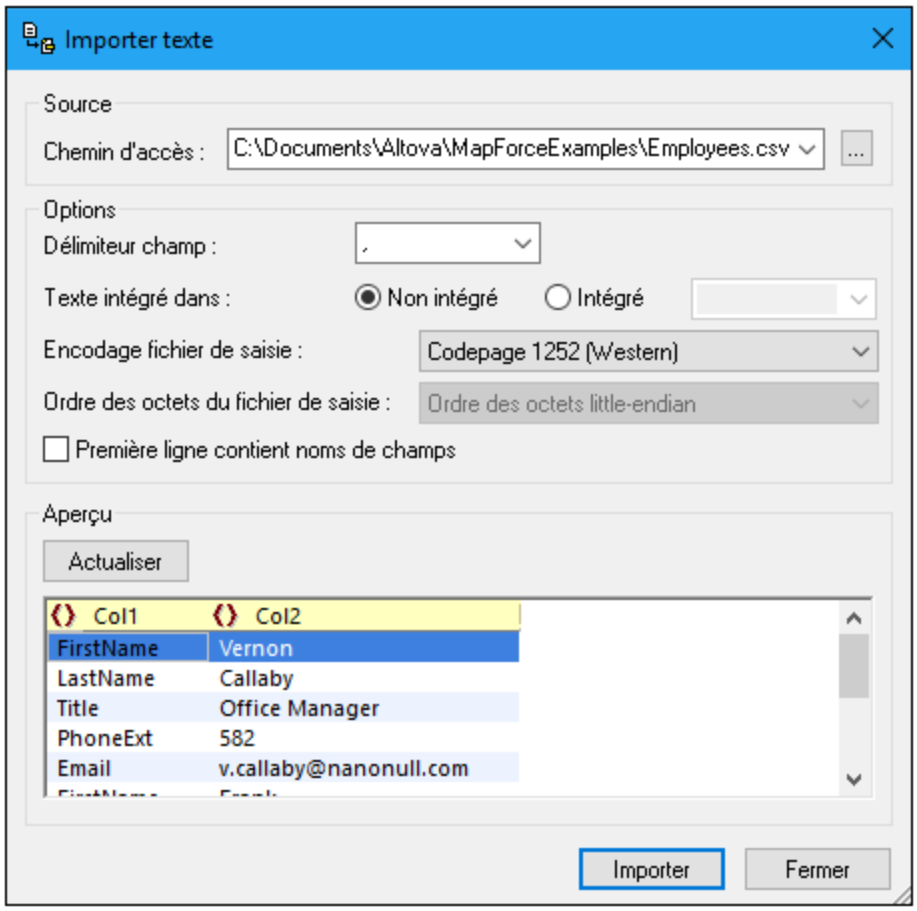
2. Dans le dialogue qui apparaît (*voir la capture d'écran ci-dessous*), sélectionnez une des deux options. (Pour que l'option de mappage fonctionne, Altova MapForce doit être installé.)



3. Cliquer sur OK. Le dialogue Importer Texte apparaît.
4. Sélectionnez les options d'importation de texte que vous souhaitez (décrites dans la prochaine section) et cliquez sur Importation. Les données importées sont converties dans un document XML qui est affiché dans le Mode Grille.

Options Importer Texte

Les options d'importation de texte sont spécifiées dans le dialogue Importer Texte (*voir la capture d'écran ci-dessous*) et sont décrites ci-dessous. Voir la section précédente pour des informations sur comment accéder au dialogue.



Chemin d'accès
Saisir le chemin d'accès pour importer dans le champ de saisie de chemin ou choisir le fichier en utilisant la touche Parcourir à droite du champ de saisie. Une fois que le fichier a été sélectionné, un Mode Grille du fichier XML est affiché dans le panneau d'aperçu. Tout changement effectué dans les options sélectionnées dans ce dialogue sera immédiatement réfléchi dans l'aperçu.

Délimiteur
Afin de pouvoir importer un fichier de texte, vous devez spécifier le délimiteur de champ utilisé pour séparer les colonnes ou les champs dans le fichier. XMLSpy détectera automatiquement des séparateurs de ligne communs (CR, LF, ou CR+LF).

Guillemets de string
Les fichiers de texte exportés depuis les systèmes anciens contiennent parfois des valeurs textuelles entre guillemets pour mieux les distinguer des valeurs numériques. Si c'est le cas, vous pouvez spécifier le type de guillemets que vous utilisez dans votre fichier et les retirer automatiquement lorsque les données sont importées.

Encodage
Les données sont converties en Unicode (la base de tous les documents XML), donc vous devez spécifier dans quel ensemble de caractère le fichier est actuellement encodé. En ce qui concerne les systèmes Window US ou d'Europe occidentale, il s'agira probablement de Codepage 1252, aussi référé en tant qu'encodage ANSI.

© 2016-2022 Altova GmbH

Tri d'octets
Si vous importez des fichiers Unicode 16-bit ou 32-bit (UCS-2, UTF-16, ou UCS-4), vous pouvez aussi passer entre l'ordre de byte little-endian et big-endian.

Première ligne contient des noms de colonnes
Il est très commun que les fichiers de texte contiennent des noms de champ dans la première ligne du fichier. Si c'est le cas, cochez cette case.

Aperçu
Dans le volet Aperçu, vous pouvez renommer les en-tête de colonne en cliquant dans un nom et en l'éditant. Les en-têtes de colonne seront les noms de l'élément ou de l'attribut dans le document XML. Vous pouvez aussi sélectionner si vous souhaitez qu'une colonne doit être un élément ou un attribut dans le document XML, ou s'il doit être importé dans le document XML. Cliquez sur l'icône de type colonne dans chaque en-tête de colonne pour passer à travers ces options. Par exemple, dans la capture d'écran ci-dessus, la colonne *Longitude en secondes* (*LonS*) ne sera pas importée.

28.11.2 Importer données de base de données



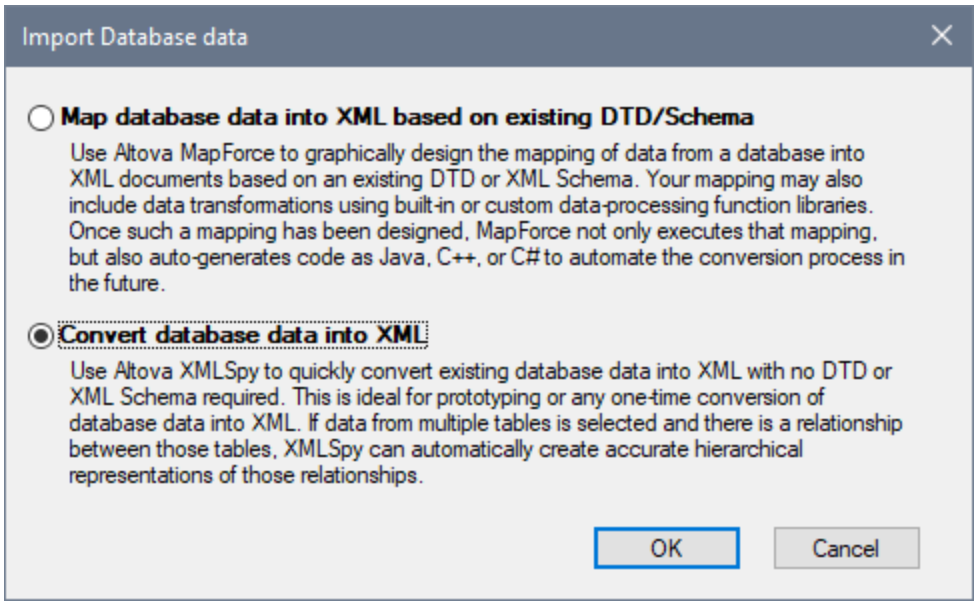
La commande **Importer données de base de données** vous permet d'importer des données depuis n'importe quelle variété de bases de données dans un fichier XML. Le mécanisme d'importation comprend deux étapes :

1. Une connexion est établie à la base de données. Pour plus d'instructions, voir Se connecter à une base de données.
2. Les données à importer sont sélectionnées.

Sélectionner la commande d'importation

Afin d'importer des données de base de données, procédez comme suit:

1. Lorsque vous cliquez sur la commande Importer données de base de données, le dialogue Importer données de base de données (*voir la capture d'écran ci-dessous*) apparaît.



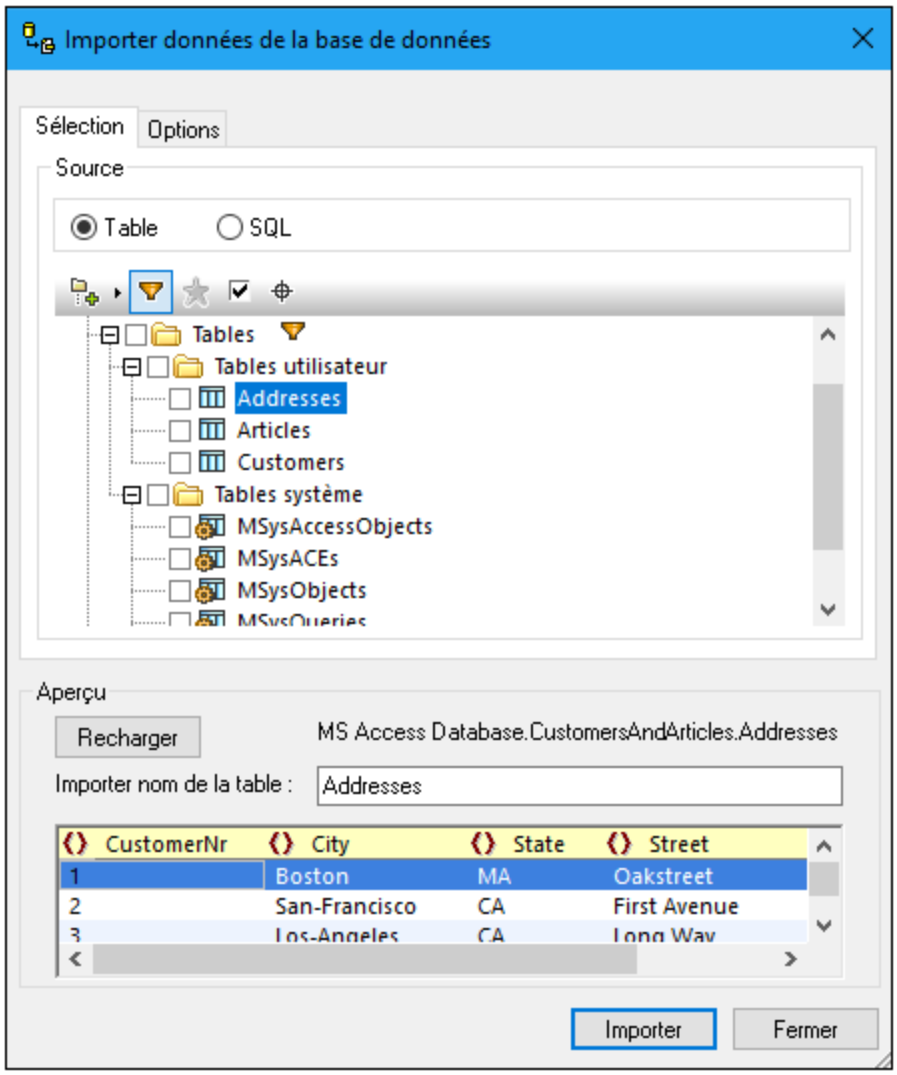
2. Sélectionnez *Convertir données de base de données dans XML* et cliquez sur **OK**. (Pour que l'option de mappage fonctionne, Altova MapForce doit être installé.)
3. Dans le dialogue Se connecter à une source de données, vous établissez une connexion à la base de données. Pour plus d'instructions, voir Se connecter à une base de données.
4. Une fois que la connexion à la base de données a été établie, le dialogue Importer des données de base de données affiche les onglets et les fenêtres qui vous permettent de sélectionner les données de base de données à importer. Ces options sont décrites ci-dessous. Une fois avoir terminé, cliquez sur la touche **Importer** pour importer les données de base de données comme document XML.

Sélection des données et options d'importation

Le dialogue Importer données de base de données pour définir la sélection et les options d'importation consiste en deux parties (*affichées séparément dans les captures d'écran ci-dessous*):

- une partie supérieure avec deux onglets : (i) Sélection, et (ii) Options.
- une partie inférieure, qui est une fenêtre d'aperçu affichant les données conformément à la sélection des données et aux options d'importation.

© 2016-2022 Altova GmbH



Méthode de sélection des données

Dans l'onglet *Sélection* (*capture d'écran ci-dessus*), le volet *Source* (*capture d'écran ci-dessous*) affiche soit une représentation des tables de la base de données ou une instruction SQL éditable pour sélectionner les tables requises, chaque affichage étant sélectionné en cliquant sur le bouton radio respectif.



Options der sélection de table

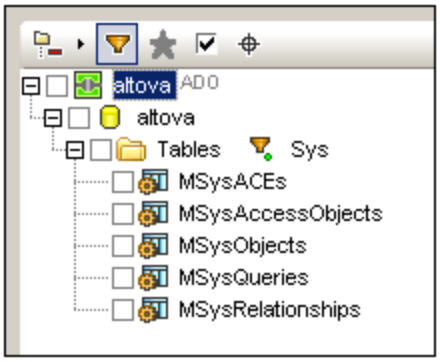
Dans l'aperçu Table, vous pouvez sélectionner les tables dans la base de données que vous souhaitez importer en cochant la case de la table (*capture d'écran ci-dessus*). Les contenus de la table peuvent ensuite être affichés dans le volet Aperçu en cliquant sur le bouton **Aperçu**.

La sélection de table fournit des commandes de sélection via les icônes dans une barre d'outils (*capture d'écran ci-dessous*).



Les icônes sont, de gauche :

- *Mise en page des dossiers :* qui vous permet d'organiser les objets de base de données comme suit : (i) des dossiers basés sur le type d'objet ; (ii) des dossiers basés sur un type d'objet, mais sans dossiers de schéma; (iii) dans une hiérarchie, mais sans dossiers; et (iv) des catégories de tables, basées sur leurs relation avec d'autres tables.
- *Contenus de dossier de filtre :* applique un filtre à un dossier sélectionné, permettant aux objets de dossier d'être filtrés. Par exemple, dans la capture d'écran ci-dessous, le filtre a été défini pour afficher des objets qui contiennent le texte `Sys` dans son nom. Cliquer sur l'icône fait apparaître un menu avec une liste de possibilités de filtre.

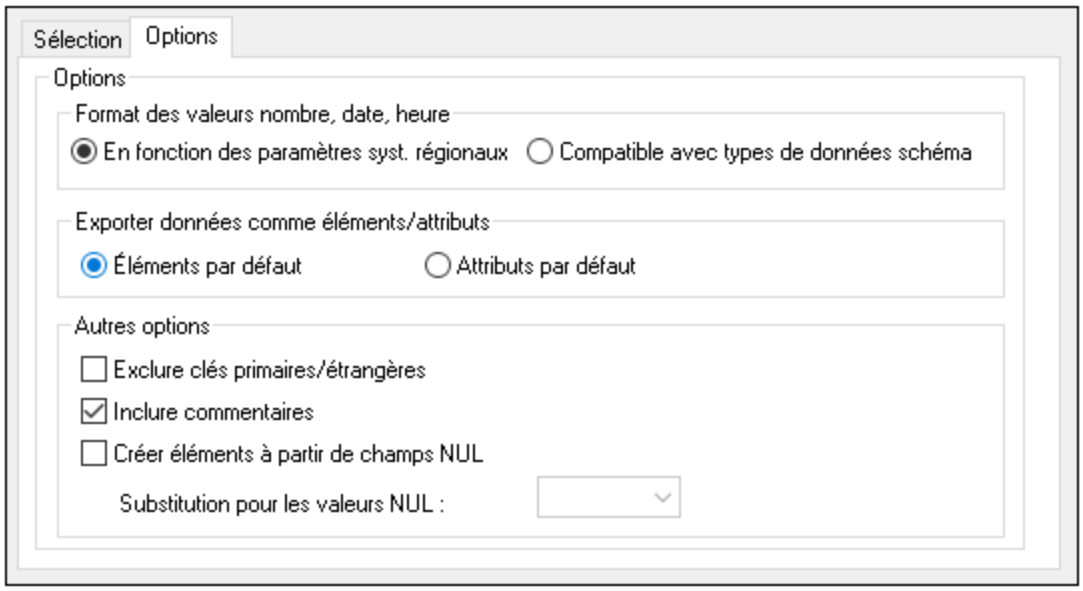
- *Afficher les favoris* Filtre les objets affichés aux favoris.

© 2016-2022 Altova GmbH

- *Afficher uniquement les objets cochés* : Filtre les objets affichés aux objets cochés.
- *Localisateur d'objet :* Affiche un champ de texte qui se comporte comme un champ d'entrée Recherche. Vous pouvez saisir un string de texte et la liste déroulante affichera des objets dont les noms contiennent ce string de texte. Sélectionner un de ces objets depuis la liste déroulante pour marquer cet objet dans l'arborescence.

Onglet Options

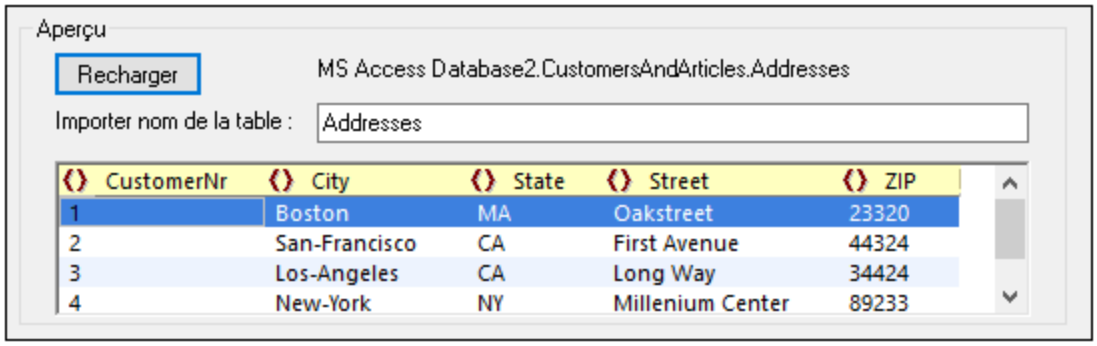
Dans l'onglet Option (*capture d'écran ci-dessous*), vous pouvez spécifier comment importer le numéro, la date et les valeurs horaires ; si les données sont importées en tant qu'élément ou attributs ; et si les commentaires et les champs `NULL` doivent être inclus dans l'importation.



Lorsque les champs `NULL` sont activés pour l'importation, vous pouvez saisir une valeur de substitution XML.

Volet Aperçu

Le volet Aperçu (*capture d'écran ci-dessous*) affiche la structure de la table actuellement sélectionnée dans l'onglet Sélection. Lorsqu'une nouvelle table est sélectionnée dans l'onglet Sélection, cliquez sur la touche **Aperçu** dans le volet Aperçu pour afficher la table. Cliquez sur la touche **Recharger** pour réinitialiser l'aperçu.



Il est possible de spécifier qu'un champ puisse être importé en tant qu'un élément ou attribut, ou ne pas être

importé, en cliquant sur le symbole à gauche du nom de la colonne. Vous pouvez cliquer sur dans les options élément, attribut et ignorer. Dans la capture d'écran ci-dessous, le champ `city`, par exemple, a été configuré pour être importé en tant qu'un élément, alors que le champ `street` a été configuré pour être importé en tant qu'un attribut.

Conversions de type de données

Les informations concernant les types de données de base de données en types de données de Schéma XML sont listées dans les Annexes.

28.11.3 Importer Document Microsoft Word



Cette commande vous permet d'importer directement un document Microsoft Word et sa conversion en XML. En sélectionnant cette commande, la boîte de dialogue Ouvrir apparaît dans laquelle vous sélectionnez le document Word que vous souhaitez importer. XMLSpy génère automatiquement un document XML avec des feuilles de style CSS incluses. Chaque paragraphe de document Word génère un élément XML, dont le nom est dérivé du style de paragraphe correspondant dans le document Word.

28.11.4 Créer le Schéma XML depuis la structure BD



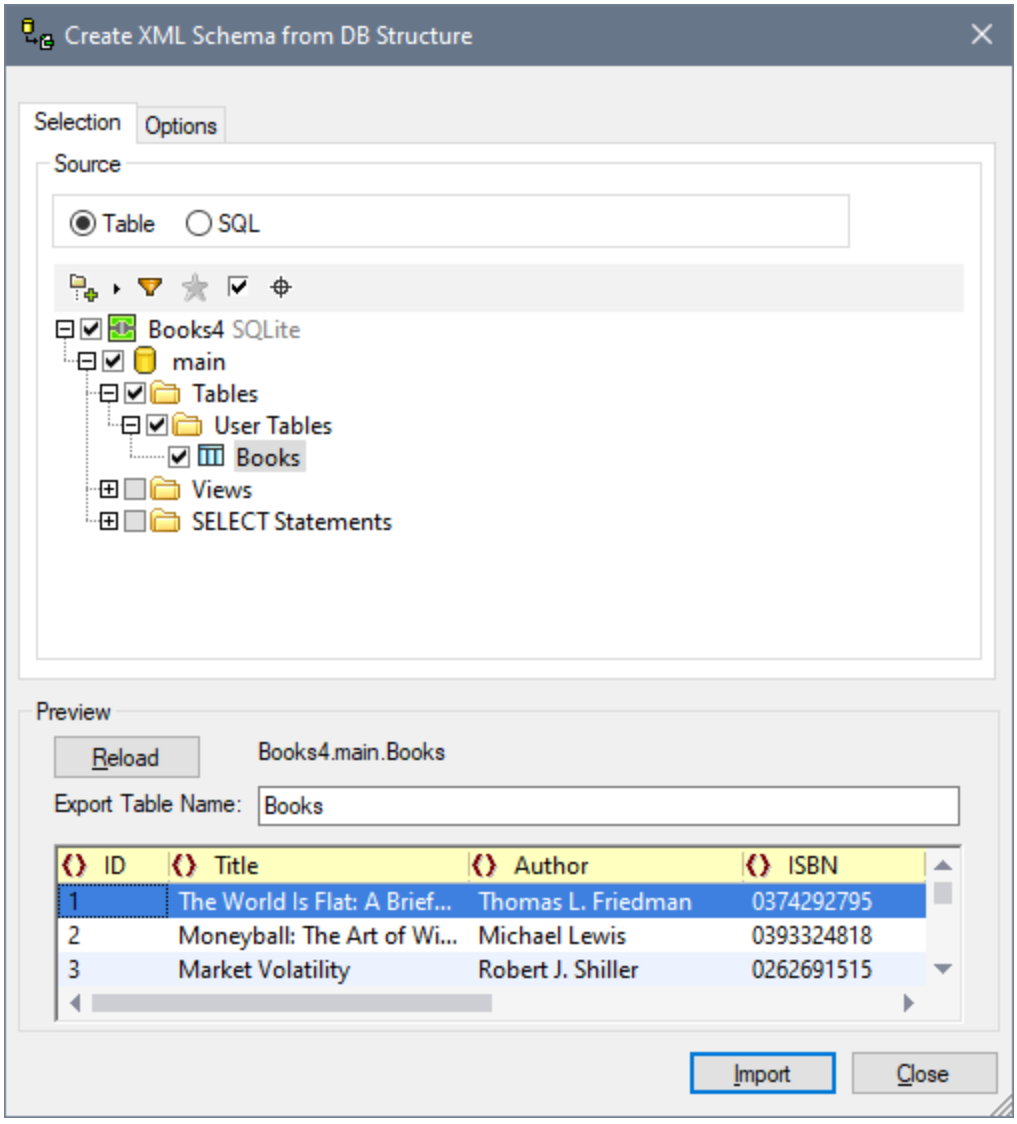
La commande **Créer un Schéma XML depuis la Structure BD** vous permet de créer un Schéma XML depuis la structure d'une variété de bases de données. Le mécanisme de création de Schéma XML implique deux étapes :

1. Une connexion est établie à la base de données. Pour plus d'instructions, voir Se connecter à une base de données.
2. Les options pour la sélection de données de la base de données et Schéma XML sont spécifiées. Celles-ci sont décrites ci-dessous.

Sélectionner la structure de données à importer

Après avoir établi une connexion à la base de données, le dialogue Créer Schéma XML depuis la structure de BD est affiché (*voir la capture d'écran ci-dessous*). Ici, vous pouvez sélectionner la structure de base de données à importer. Après avoir réalisé cette étape, cliquez sur Importer pour créer un Schéma XML qui défini cette structure pour un document XML.

© 2016-2022 Altova GmbH



Méthode de sélection des données

Dans l'onglet *Sélection* (*capture d'écran ci-dessus*), le volet *Source* (*capture d'écran ci-dessous*) affiche soit une représentation des tables de la base de données ou une instruction SQL éditable pour sélectionner les tables requises, chaque affichage étant sélectionné en cliquant sur le bouton radio respectif.



Options der sélection de table

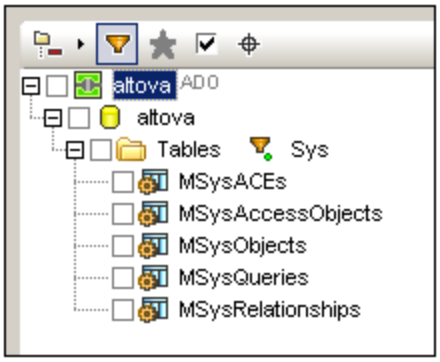
Dans l'aperçu Table, vous pouvez sélectionner les tables dans la base de données que vous souhaitez importer en cochant la case de la table (*capture d'écran ci-dessus*). Les contenus de la table peuvent ensuite être affichés dans le volet Aperçu en cliquant sur le bouton **Aperçu**.

La sélection de table fournit des commandes de sélection via les icônes dans une barre d'outils (*capture d'écran ci-dessous*).



Les icônes sont, de gauche :

- *Mise en page des dossiers :* qui vous permet d'organiser les objets de base de données comme suit : (i) des dossiers basés sur le type d'objet ; (ii) des dossiers basés sur un type d'objet, mais sans dossiers de schéma; (iii) dans une hiérarchie, mais sans dossiers; et (iv) des catégories de tables, basées sur leurs relation avec d'autres tables.
- *Contenus de dossier de filtre :* applique un filtre à un dossier sélectionné, permettant aux objets de dossier d'être filtrés. Par exemple, dans la capture d'écran ci-dessous, le filtre a été défini pour afficher des objets qui contiennent le texte `Sys` dans son nom. Cliquer sur l'icône fait apparaître un menu avec une liste de possibilités de filtre.



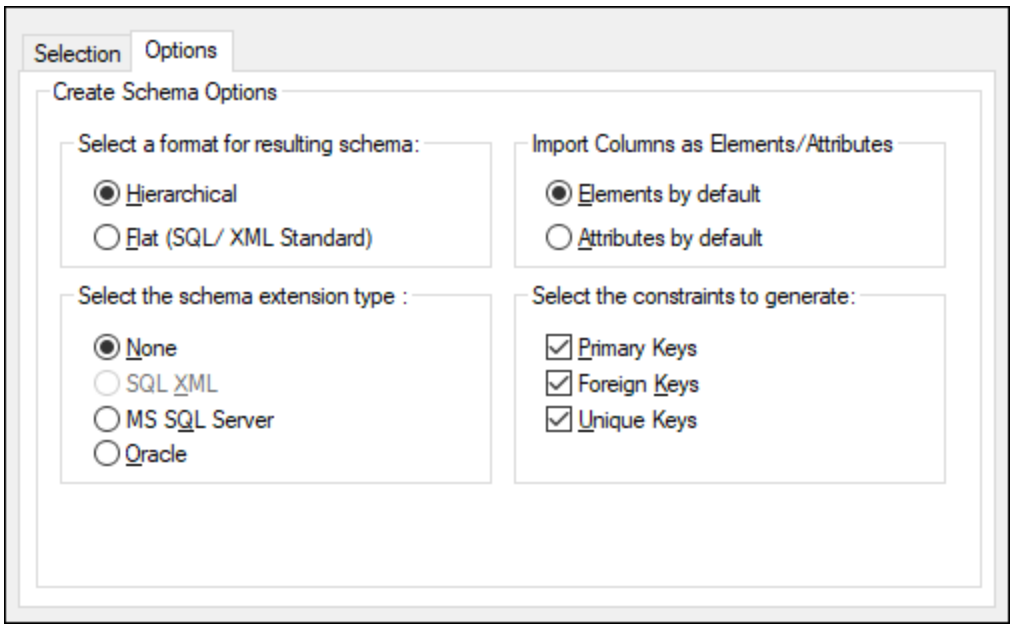
- *Afficher les favoris* Filtre les objets affichés aux favoris.

© 2016-2022 Altova GmbH

- *Afficher uniquement les objets cochés* : Filtre les objets affichés aux objets cochés.
- *Localisateur d'objet :* Affiche un champ de texte qui se comporte comme un champ d'entrée Recherche. Vous pouvez saisir un string de texte et la liste déroulante affichera des objets dont les noms contiennent ce string de texte. Sélectionner un de ces objets depuis la liste déroulante pour marquer cet objet dans l'arborescence.

Onglet Options

Dans l'onglet *Options* (*voir la capture d'écran ci-dessous*), vous pouvez spécifier le format du schéma, son type d'extension, si les colonnes doivent être importées comme éléments ou attributs et les contraintes de base de données qui devraient être générées dans le schéma.



Format de schéma
Vous pouvez sélectionner soit un format plat (Standard SQL/XML), soit un format de schéma hiérarchique.

- Le modèle de schéma *modèle de schéma flat* est basé sur une spécification ISO-ANSI SQL/XML INCITS/ISO/IEC 9075-14-2008. La spécification SQL/XML définit comment mapper les bases de données en XML. Les relations sont définies dans les schémas en utilisant des contraintes d'identité ; il n'y a pas de références aux éléments. Ainsi, le schéma présente une structure plate qui ressemble à un aperçu arborescent de la base de données. La spécification peut être achetée dans la boutique ANSI. Pour plus d'information, voir www.iso.org.
- *Le modèle de schéma* hiérarchique affiche les dépendances de table visuellement, dans un type d'aperçu arborescent où des tables dépendantes sont affichées en tant qu'éléments enfants indentés dans le modèle de contenu. Les dépendances de table sont aussi affichées dans l'onglet Contraintes d'identité.

Les tables sont listées en tant qu'éléments globaux dans le schéma, et les colonnes sont les éléments ou les attributs de ces éléments globaux (l'utilisateur décide s'il veut mapper les colonnes en tant qu'éléments ou en tant qu'attributs). Les relations sont créées de manière hiérarchique pour qu'un champ de clé étrangère dans une table soit de fait une référence à l'élément global qui représente cette table.

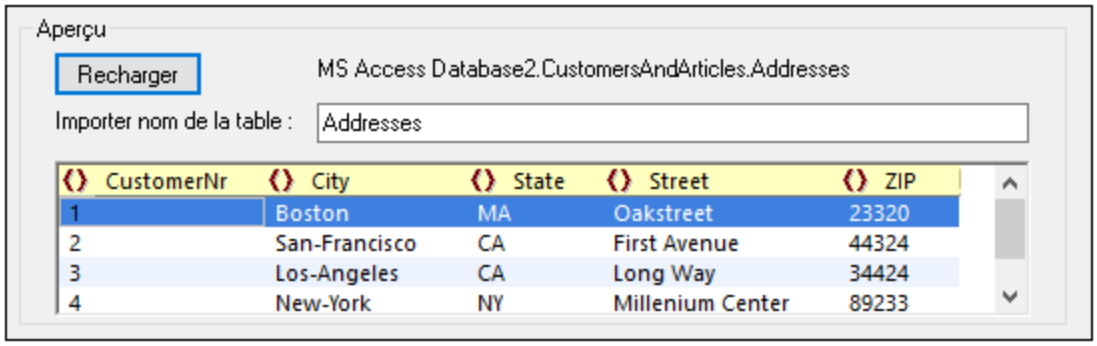
Type d'extension de schéma
L'information d'extension de schéma est une information supplémentaire lue par la base de données qui est, ensuite, incorporée dans le schéma, soit comme des données d'annotation, soit des attributs. Il existe quatre types d'options d'extension lors de la génération de schémas : (i) aucune information d'extension ; (ii) extensions SQL/XML ; (iii) extensions serveur SQL MS ; et (iv) extensions Oracle. Celles-ci sont décrites ci-dessous :

- *Aucun :* Aucune information supplémentaire n'est fournie par la base de données.
- *SQL XML :* Les extensions SQL/XML ne sont pas insérées lorsque des schémas sont générés dans un format plat. L'information d'extension est stockée dans les annotations et est décrite dans la spécification SQL/XML (INCITS/ISO/IEC 9075-14-2008).
- *Serveur MS SQL :* Sélectionner Microsoft SQL Server génère des extensions de serveur SQL. Voir SQL Server Books Online pour les ressources et l'information de MSDN à propos de l'annotation des schémas XSD. Les éléments liés à l'annotation suivants sont générés dans le schéma: `sql:relation`, `sql:field`, `sql:datatype`, `sql:mapped`.
- *Oracle* Les extensions Oracle sont sélectionnées par défaut en travaillant avec une base de données Oracle. Une information de base de données supplémentaire est stockée en tant qu'attributs. L'information détaillée peut être trouvée dans la documentation en ligne d'Oracle. Les sous-ensembles suivants des attributs sont générés actuellement : `SQLName`, `SQLType`, `SQLSchema`.

Note : Bien que les extensions SQL Server et Oracle peuvent être générées pour leurs bases de données respectives, elles ne sont pas limitées. Cela peut être utile lors du travail avec une troisième base de données et si vous souhaitez générer un schéma capable de travailler plus tard soit avec le serveur SQL ou Oracle.

Volet Aperçu

Le volet Aperçu (*capture d'écran ci-dessous*) affiche la structure de la table actuellement sélectionnée dans l'onglet Sélection. Lorsqu'une nouvelle table est sélectionnée dans l'onglet Sélection, cliquez sur la touche **Aperçu** dans le volet Aperçu pour afficher la table. Cliquez sur la touche **Recharger** pour réinitialiser l'aperçu.



Il est possible de spécifier qu'un champ puisse être importé en tant qu'un élément ou attribut, ou ne pas être importé, en cliquant sur le symbole à gauche du nom de la colonne. Vous pouvez cliquer sur dans les options élément, attribut et ignorer. Dans la capture d'écran ci-dessous, le champ `city`, par exemple, a été configuré pour être importé en tant qu'un élément, alors que le champ `street` a été configuré pour être importé en tant qu'un attribut.

© 2016-2022 Altova GmbH

Conversions de type de données

Les informations concernant les types de données de base de données en types de données de Schéma XML sont listées dans les Annexes.

28.11.5 Importation BD basée sur un Schéma XML



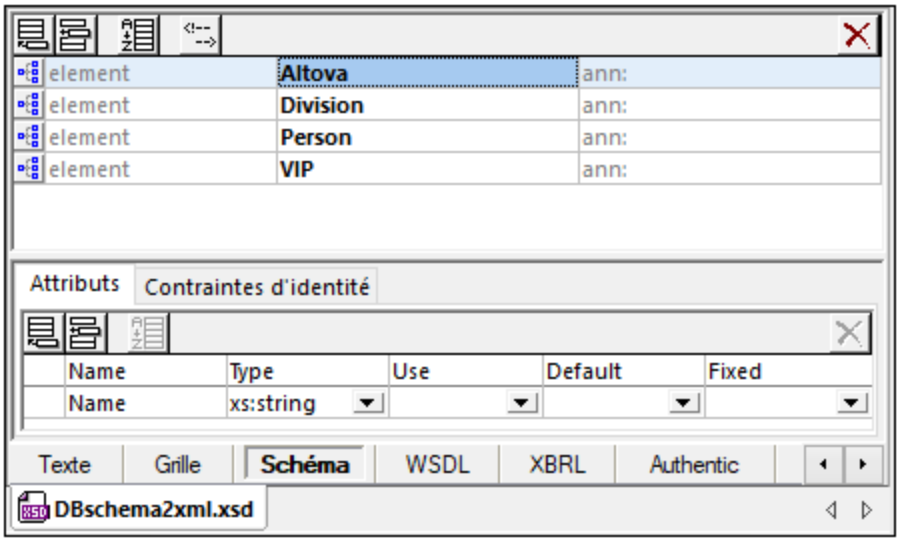
La commande **Importation BD basée sur le Schéma XML** crée un document XML qui est valide conformément à un Schéma XML donné et qui contient des données importées depuis une base de données. En ce qui concerne cette fonction, les bases de données suivantes sont prises en charge :

- Microsoft Access 2000 et 2003
- Microsoft SQL Server
- Oracle
- MySQL
- Sybase
- IBM DB2

Les données à importer sont déterminées par la table qui est sélectionnée dans la base de données. Connectez-vous à la base de données avec le Schéma XML requis (celui sur lequel vous souhaitez baser l'importation) en tant que le document actif dans le Mode Schéma. Ensuite sélectionnez les tables que vous souhaitez importer et cliquez sur Importer. Les données sont importées dans un document XML et le document a la structure du Schéma XML qui était actif lorsque les données ont été importées.

Dans l'exemple ci-dessous, les données provenant de la base de données MS Access sont importées avec un Schéma XML actif dans le Mode Schéma. Suivez les étapes ci-dessous pour effectuer une importation :

1. Cela permet d'ouvrir le fichier de schéma dans le Mode Schéma (*voir la capture d'écran ci-dessous*).



2. Choisir la commande de menu **Importation BD basée sur le Schéma XML**. Ceci ouvre le dialogue Se connecter à la source de données.
3. Sélectionnez l'option Microsoft Access (ADO), puis cliquez sur **Suivant**.
4. Cliquez sur **Browse** et sélectionnez le fichier de base de données. Puis cliquer sur **Suivant**.
5. Dans l'importation BD basée sur le dialogue Schéma XML qui apparaît, allez à l'onglet Tables, sélectionnez une ou plusieurs tables que vous souhaitez importer (par exemple `Altova`), puis cliquez sur **Importer**. La table est importée dans un document XML qui est affiché dans le Mode Grille.

Conversions de type de données

Les informations concernant les types de données de base de données en types de données de Schéma XML sont listées dans les Annexes.

28.11.6 Créer une structure BD depuis le Schéma XML

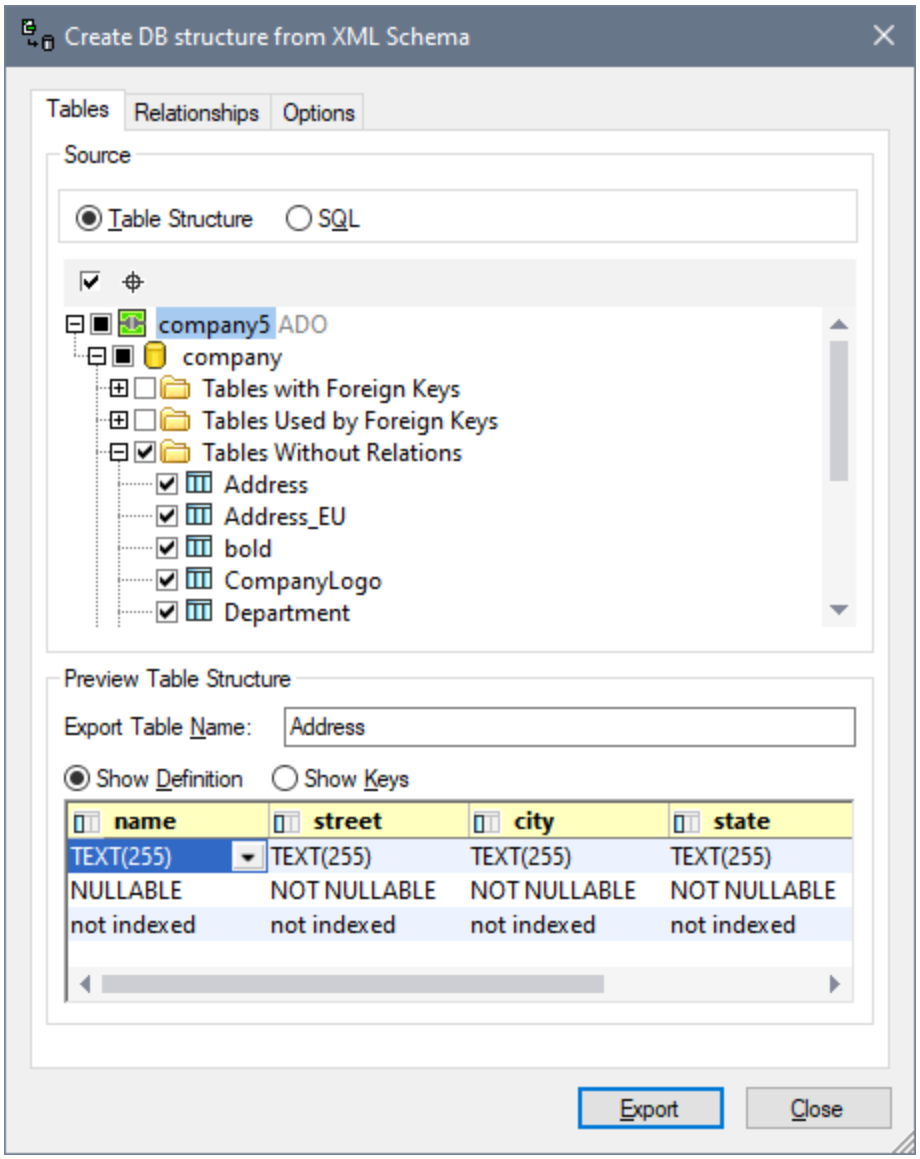


XMLSpy vous permet de créer une base de données vide (ou base de données squelette) basée sur un fichier de schéma existant. La méthode décrite ci-dessous est généralement la même pour chaque type de base de données.

1. Cela permet d'ouvrir le fichier de schéma dans le Mode Schéma/WSDL.
2. Sélectionner la commande de menu **Conversion | Créer la structure BD depuis le Schéma XML**. Cela ouvre le dialogue Se connecter à une source de données, qui vous permet de vous connecter à une base de données (BD).
3. Utiliser les étapes décrites dans la section Se connecter à une source de données pour vous connecter à la base de données requise. Par exemple, pour se connecter à la base de données Microsoft Access, sélectionnez le bouton radio Microsoft Access et continuez le processus pour sélectionner une base de données. Vous pouvez utiliser une base de données existante ou en créer une nouvelle dans laquelle la structure de schéma sera contenue.

© 2016-2022 Altova GmbH

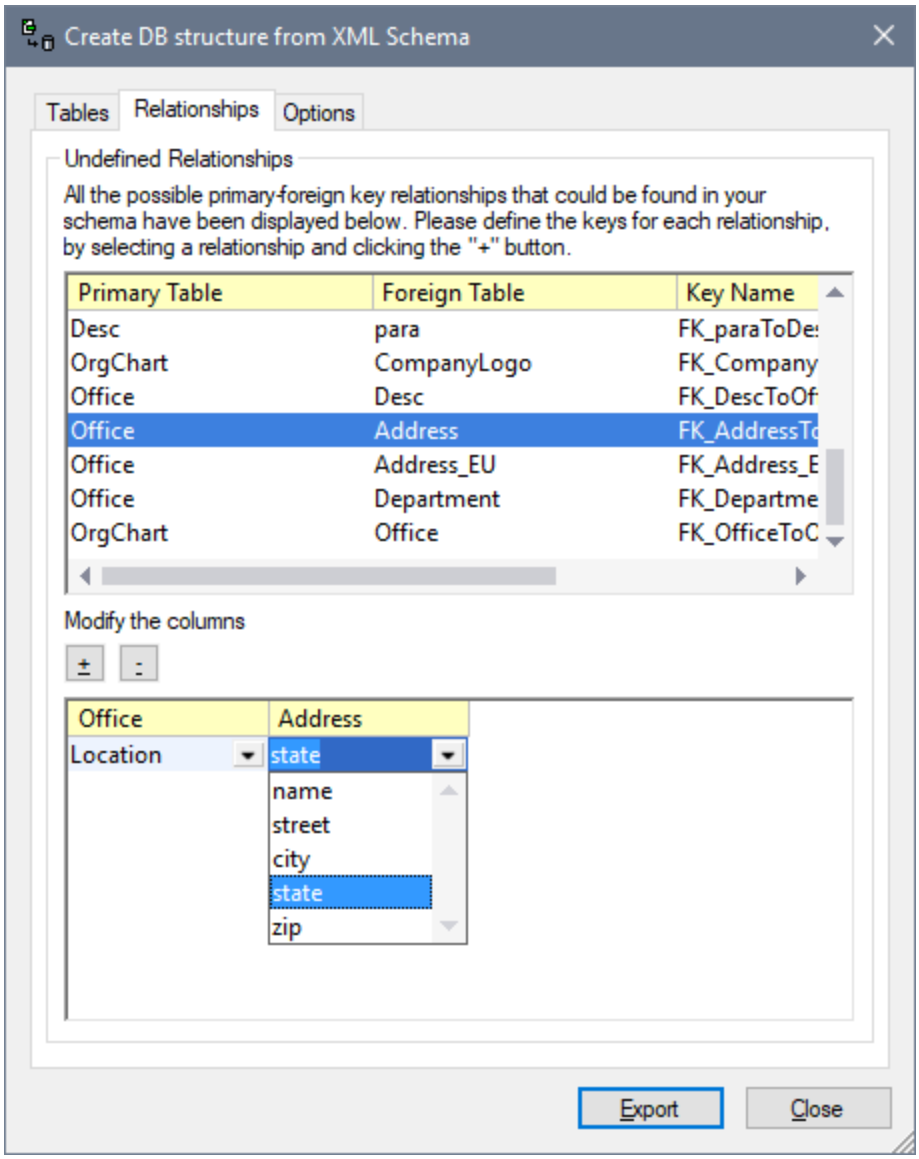
4. Dans le dialogue Créer structure BD depuis un schéma XML, les tables sont créées depuis le schéma et affichées dans un format arborescent à l'endroit où ils se produiront dans la BD. Par exemple, dans la capture d'écran ci-dessous, la table Address est créée, puis sélectionnée pour une exportation. Les tables qui ne doivent pas être exportées doivent être désélectionnées (en décochant la case ou en sélectionnant l'item approprié depuis le menu contextuel pour cette table).



Créer des tables BD avec des relations

Si le Schéma XML à partir duquel la structure BD est générée a des relations définies sous la forme de contraintes d'identité, ces relations sont créées automatiquement dans la structure, alors ces relations sont créées automatiquement dans la structure de BD générée et sont affichées dans la structure de table. Les tables avec des relations sont listées sous les sections : Tables avec ForeignKeys et Tables utilisées par ForeignKeys. Les tables sans relations sont listées dans la section Tables indépendantes.

Dans l'onglet Relations, vous pouvez créer et modifier les relations de table. L'onglet liste toutes les relations clé primaire/clé étrangère possibles (*capture d'écran ci-dessous*).



Pour créer une relation, procédez comme suit :

1. Choisir une des relations clé primaire/clé étrangère possibles .
2. Dans le volet inférieur du dialogue, cliquez sur la touche Plus pour créer une relation.
3. Sélectionner les colonnes requises dans chacune des deux tables depuis les listes déroulantes respectives.

Vous pouvez aussi supprimer une relation en la sélectionnant et en cliquant sur la touche Moins.

© 2016-2022 Altova GmbH

Notes concernant la structure et la connexion de la base de données

La structure du schéma, définie par les contraintes d'identité, est réfléchie par les contraintes d'identité, et est reflétée dans la base de données résultante. La table ci-dessous affiche le type de base de données créé, les restrictions et les méthodes de connexion, lorsque vous utilisez la commande de menu **Créer une StructureBD depuis le Schéma XML** .

	Directement	**Utilisant ODBC**	**Utilisant ADO**
Microsoft Access (2000 et 2003)	OK *	OK	OK
Serveur MS SQL	OK *	OK	OK
Oracle	OK *	OK	OK
MySQL	-	OK *	OK +
Sybase	-	OK *	OK
IBM DB2	-	OK *	OK

* *Méthode de connexion recommandée pour chaque base de données.*
\+ *MySQL: Lors de la création de la connexion ADO basée sur ODBC, il est recommandé d'utiliser soit le DSN utilisateur soit le DSN système.*
\- *N'est pas pris en charge*

XMLSpy mappera les schémas formatés de manière hiérarchique et plate. XMLSpy reconnaît les deux formats automatiquement.
Le format plat est mappé dans SQL de deux manières différentes.

- Serveur de BD SQL, BD Oracle, ou BD Sybase :
  Un schéma qui a été généré dans un format plat, pour une des bases de données ci-dessus, verra son nom de catalogue de schéma extrait et utilisé dans le script SQL généré en tant que le nom de BD. Cela signifie que le script SQL résultant sera exécuté sur une BD cible dont le nom doit être identique à celui du nom de catalogue du schéma.
- Access (2000 ou 2003), MySQL, ou BD DB2 :
  Un schéma qui a été généré dans un format plat, pour une des bases de données ci-dessus, ignorera le nom de catalogue de schéma lorsque le script SQL est généré. Cela signifie que le script SQL résultant sera exécuté sur une base de données cible pour laquelle une connexion a été établie.

Conversions de type de données

Vous trouverez des informations concernant la conversion des types de données de Schéma XML en types de données de base de données dans les Annexes.

28.11.7 Exporter vers les fichiers de texte



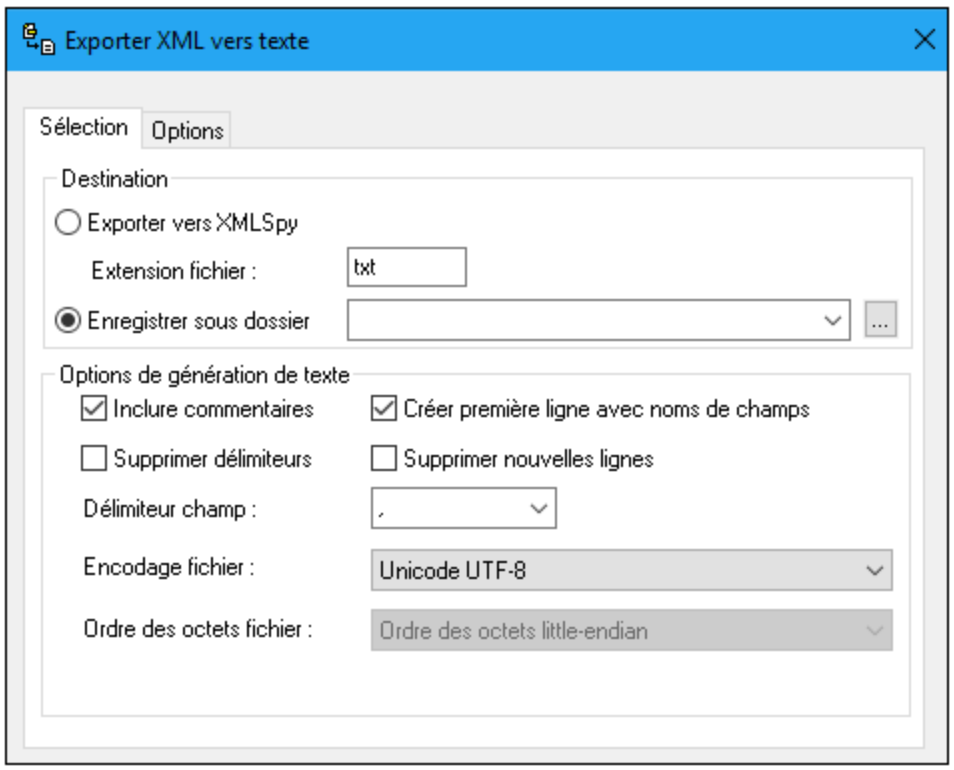
La commande **Convertir | Exporter vers des fichiers de texte** exporte les données XML en formats de texte pour l'échange avec des bases de données ou des systèmes hérités. En cliquant sur cette commande, le dialogue Exporter XML vers du texte s'ouvre. Elle comprend deux parties (*affichées dans des captures d'écran séparées ci-dessous*) :

- une partie supérieure avec deux onglets : les options (i) Sélection et (ii) Exporter.
- une partie inférieure, qui est une fenêtre d'aperçu.

Une fois avoir sélectionné les options souhaitées dans ce dialogue (*décrit ci-dessous*), cliquez sur la touche **Exporter** pour exporter les fichiers de texte.

Sélection

Dans l'onglet Sélection (*capture d'écran ci-dessous*), vous pouvez choisir la destination du fichier à exporter et les options de génération de texte.



Destination : Le fichier exporté peut être enregistré directement dans un dossier. L'extension de fichier peut être spécifiée. Le nom des fichiers sera celui des éléments (dans le fichier XML) qui sera exporté. En alternative, des fichiers sans titre peuvent être exportés vers XMLSpy. Ces fichiers seront affichés dans la GUI, et peuvent être enregistrés plus tard.

© 2016-2022 Altova GmbH

Inclure commentaires : Activez cette option pour inclure un commentaire généré par XMLSpy dans le fichier XML exporté. Le commentaire contiendra la requête SQL utilisée pour sélectionner les données de même que la liste qui contient un listitem pour chaque en-tête de colonne dans la table de base de données.

Créer une première ligne avec des noms de champ : Lorsqu'activées, les tables exportées incluent les noms des colonnes depuis la base de données. Autrement, les noms de colonne ne seront pas inclus dans le fichier de texte exporté.

Supprimer les délimiteurs : Supprime les délimiteurs qui sont contenus dans les valeurs de texte dans les données exportées. Définir le délimiteur que vous souhaitez en utilisant la liste de choix Délimiteur dans cet onglet. Par exemple, si cette option est activée et le délimiteur sélectionné est l'apostrophe, lorsque vous exportez la valeur XML `Ba'ker`, le string sera `Baker` dans le texte exporté.

Supprimer newslines : Supprime les newslines des données exportées.

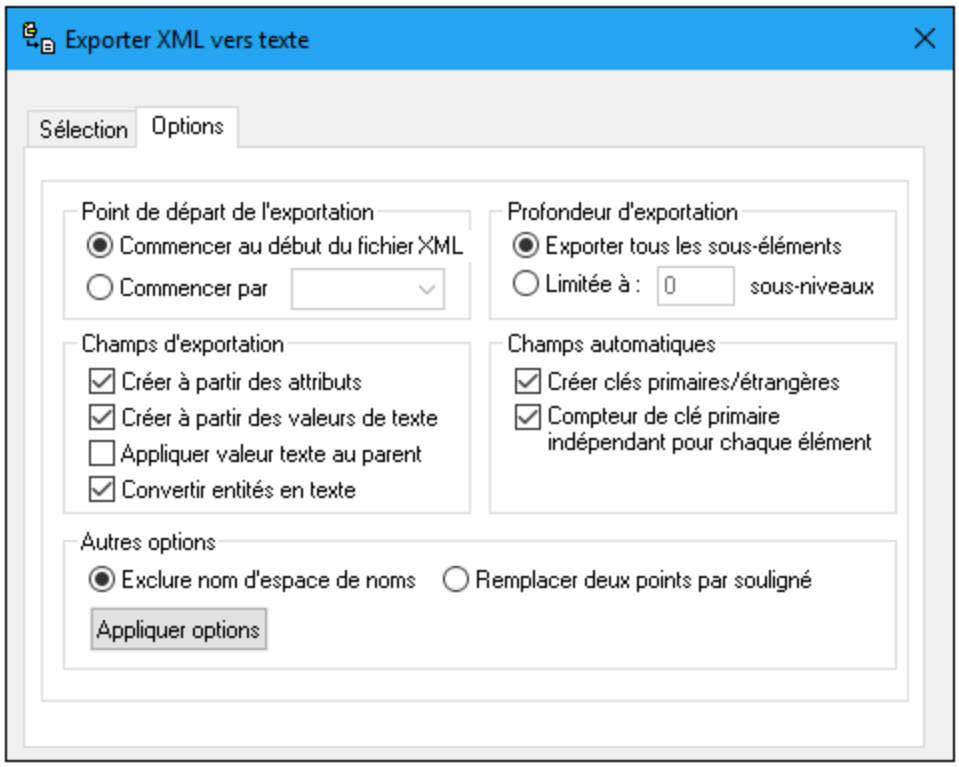
Délimiteur : Sélectionnez le caractère depuis la liste déroulante que vous souhaitez avoir supprimé pendant l'exportation. En alternative, saisissez le string de caractère désiré.

Encodage : Sélectionnez depuis la liste déroulante l'encodage désiré pour des fichiers qui sont générés pendant l'exportation.

Tri d'octets : si vous exportez des fichiers Unicode 16-bit ou 32-bit (UCS-2, UTF-16, ou UCS-4), vous pouvez aussi passer entre l'ordre de byte little-endian et big-endian.

Options d'exportation

Les options d'exportation supplémentaires, décrites ci-dessous, peuvent être spécifiées dans l'onglet Options (*capture d'écran ci-dessous*):



Point de départ de l'exportation : Vous pouvez choisir d'exporter tout le document XML ou restreindre votre exportation à la hiérarchie de données commençant par l'élément sélectionné actuellement. Le nombre de sous-niveaux en-dessous du point de départ qui sera exporté est spécifié dans l'option *Export Depth*.

Profondeur d'exportation : Spécifie le nombre de sous-niveaux en-dessous du point de départ qui sera exporté.

Champs d'exportation : selon vos données XML, il se peut que vous souhaitiez n'exporter que des éléments, des attributs ou le contenu textuel de vos éléments. Veuillez noter que vous pouvez aussi désélectionner l'exportation des éléments individuels dans la fenêtre Aperçu.

Champs automatiques : XMLSpy produira un fichier de sortie ou une table pour chaque type d'élément sélectionné. Vous pouvez choisir de créer automatiquement des paires de clé primaire/étrangères pour lier vos données dans le modèle relationnel ou définir une clé primaire pour chaque élément.

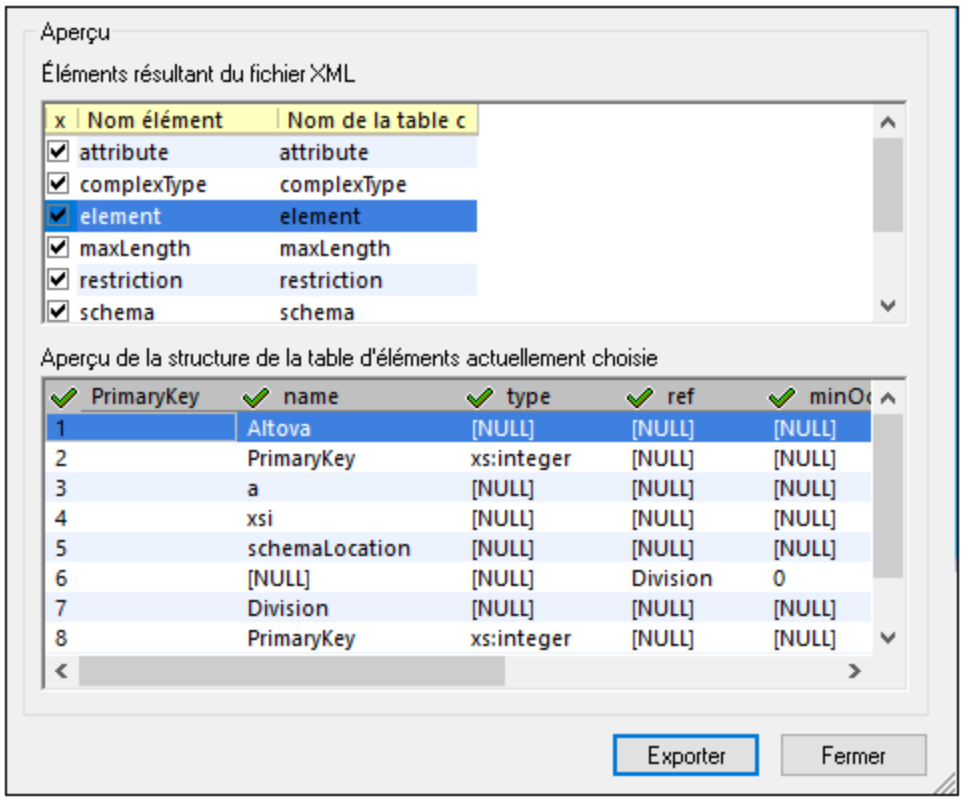
Exclure le nom d'espace de nom : Ensemble avec le bouton radio *Replace Colon With Underscore*, il s'agit d'un choix soit/soit. Spécifie si les préfixes d'espace de noms des éléments et des attributs doivent être exclus ou si le double point dans le préfixe d'espace de noms doit être remplacé par un caractère de soulignement.

Options d'application : Après avoir défini des options, cliquez sur ce bouton pour appliquer les options. L'aperçu dans le volet d'aperçu sera mis à jour avec les nouvelles options.

Fenêtre Aperçu

La fenêtre Aperçu (*capture d'écran ci-dessous*) est affichée en-dessous des onglets de Sélection et Options.

© 2016-2022 Altova GmbH



Le volet *Éléments résultant du fichier XML* affiche les noms de nœud qui seront exportés et le nom dans le fichier généré. Vous pouvez sélectionner/désélectionner les nœuds qui seront exportés. Lorsqu'un élément est sélectionné, un aperçu de sa structure est affiché dans un deuxième volet ci-dessous. Dans ce volet, cliquez à gauche d'un nom d'en-tête de colonne pour activer et désactiver l'exportation de cette colonne. Dans la capture d'écran ci-dessus, la dernière colonne (`FK_Office`) a été désactivée.

28.11.8 Exporter vers une base de données



La commande **Convertir | Exporter vers une base de données** exporte des données XML à la base de données. En cliquant sur cette commande, l'Assistant de connexion démarre et vous permet de configurer une connexion vers la base de données que vous souhaitez mettre à jour. Une fois qu'une connexion a été établie, le dialogue Exporter les données vers la base de données s'ouvre. Elle consiste en deux parties (*affichées séparées dans les captures d'écran ci-dessous*) :

- une partie supérieure avec deux onglets : les options (i) Sélection et (ii) Exporter.
- une partie inférieure, qui est une fenêtre d'aperçu.

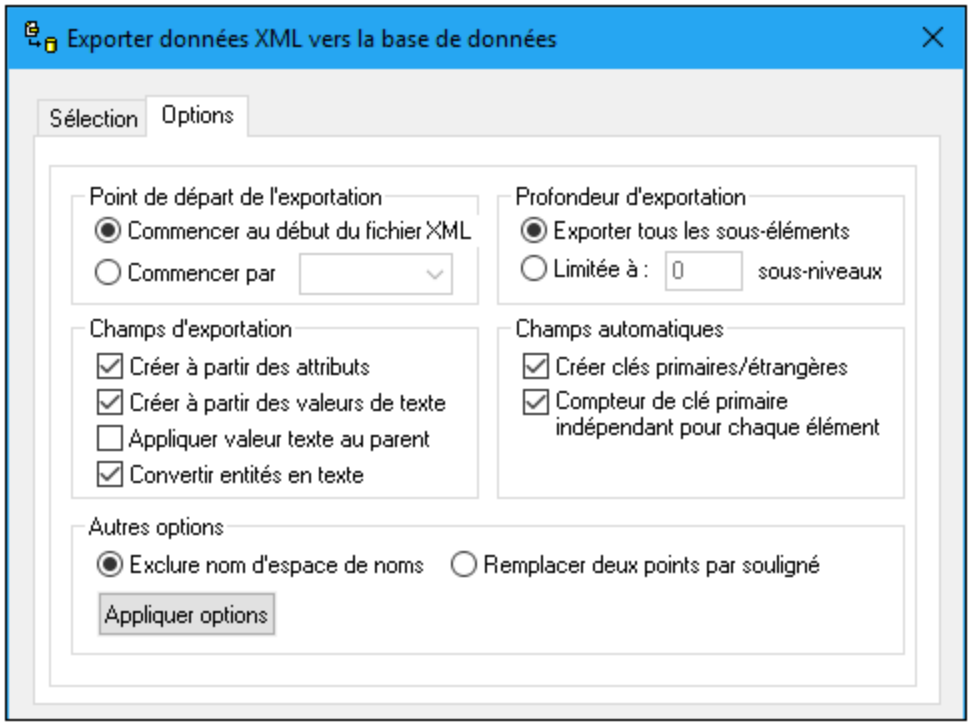
Une fois que vous avez sélectionné les options désirées dans ce dialogue (*décrites ci-dessous*), cliquez sur la touche **Exporter** pour exporter vers la base de données.

Sélection

Dans l'onglet Sélection, vous pouvez sélectionner la base de données de destination et les options de génération de table. Le champ de destination sélectionne la connexion vers la base de données. Vous devez sélectionner si les données sont créées comme nouvelles tables, mises à jour dans des tables existantes ou si d'abord elles tentent d'abord de mettre à jour une table existante, puis créent une nouvelle table si une mise à jour n'est pas possible. Vous pouvez aussi définir une action stop basée sur le nombre d'erreurs, et, en option, un journal de script SQL.

Options d'exportation

Les options d'exportation, qui sont décrites ci-dessous, peuvent être spécifiées dans l'onglet Options (*capture d'écran ci-dessous*) :



Point de départ de l'exportation : Vous pouvez choisir d'exporter tout le document XML ou restreindre votre exportation à la hiérarchie de données commençant par l'élément sélectionné actuellement. Le nombre de sous-niveaux en-dessous du point de départ qui sera exporté est spécifié dans l'option *Export Depth*.

Profondeur d'exportation : Spécifie le nombre de sous-niveaux en-dessous du point de départ qui sera exporté.

Champs d'exportation : selon vos données XML, il se peut que vous souhaitiez n'exporter que des éléments, des attributs ou le contenu textuel de vos éléments. Veuillez noter que vous pouvez aussi désélectionner l'exportation des éléments individuels dans la fenêtre Aperçu.

Champs automatiques : XMLSpy produira un fichier de sortie ou une table pour chaque type d'élément sélectionné. Vous pouvez choisir de créer automatiquement des paires de clé primaire/étrangères pour lier vos données dans le modèle relationnel ou définir une clé primaire pour chaque élément.

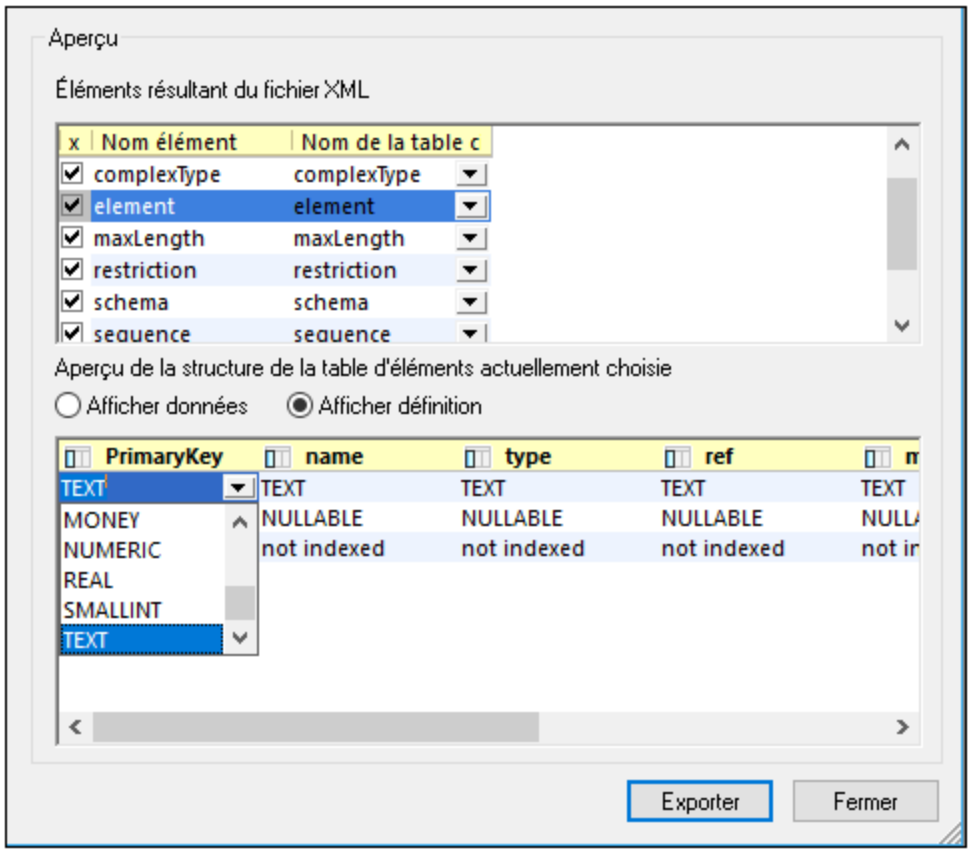
© 2016-2022 Altova GmbH

Exclure le nom d'espace de nom : Ensemble avec le bouton radio *Replace Colon With Underscore*, il s'agit d'un choix soit/soit. Spécifie si les préfixes d'espace de noms des éléments et des attributs doivent être exclus ou si le double point dans le préfixe d'espace de noms doit être remplacé par un caractère de soulignement.

Options d'application : Après avoir défini des options, cliquez sur ce bouton pour appliquer les options. L'aperçu dans le volet d'aperçu sera mis à jour avec les nouvelles options.

Fenêtre Aperçu

La fenêtre Aperçu (*capture d'écran ci-dessous*) est affichée en-dessous des onglets de Sélection et Options.



Le volet *Éléments résultant du fichier XML* affiche les noms de nœud dans le document XML qui seront exportés et le nom dans le fichier généré. Vous pouvez sélectionner/désélectionner les nœuds qui seront exportés. Lorsqu'un élément est sélectionné, un aperçu de sa structure dans le fichier généré est affiché dans un deuxième volet ci-dessous. Cet aperçu peut passer entre un aperçu de : (i) données dans la structure générée (*Afficher les données*) ; ou des (ii) définitions de chaque colonne dans la structure générée (*Afficher la définition*). La capture d'écran ci-dessus montre les définitions de la colonne.

Dans ce second volet, cliquez à gauche d'un nom de la colonne pour passer par un des quatre paramètres de la colonne : (i) Inclure dans la structure de la table ; (ii) Contrainte unique ; (iii) Contrainte de clé primaire ; (iv) Exclure de la structure de la table. Dans la capture d'écran ci-dessus, la colonne `Emplacement` a une contrainte Unique, tandis que la colonne `Fax` a déjà été exclue de la structure de table. Toutes les autres colonnes sont incluses dans la structure de table.

Lorsque la structure de table de l'élément montre les définitions de champ (*Afficher définition*), les définitions peuvent être éditées en sélectionnant la définition et en choisissant une option depuis la liste de choix de la définition (*voir capture d'écran ci-dessus*).

28.11.9 Convertir instance XML en/de JSON

Si le document actif est un document XML, cette commande génère un document JSON depuis le document XML. Si le document actif est un document JSON, la commande génère un document XML depuis le document JSON. Le document généré est ouvert dans une nouvelle fenêtre et peut ensuite être enregistré dans n'importe quel endroit. Les options de conversion sont décrites ci-dessous. Pour plus d'iinformation sur la prise en charge de JSON et de l'édition JSON dans XMLSpy, voir la section JSON et JSON Schema.

Conversions d'échantillon

Ci-dessous, vous trouverez un exemple d'un document de source XML et en-dessous le document JSON généré par la commande **Convertir Instance XML vers/depuis JSON**.

Document XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Person first="Jim" last="James">
  <Address>
    <street>4 New Street</street>
    <city>New York</city>
    <state>NY</state>
    <code>10123</code>
  </Address>
  <Tel type="home">
      123 123-1234
  </Tel>
  <Tel type="office">
      123 987-9876
  </Tel>
</Person>
```

Document JSON

```
{
       "XML": {
              "version": 1,0,
              "encoding": "UTF-8"
       },
       "Person": {
              "first": "Jim",
              "last": "James",
              « Adresse » : {
                     "street": "4 New Street",
                     "city": "New York",
                     "state": "NY",
```

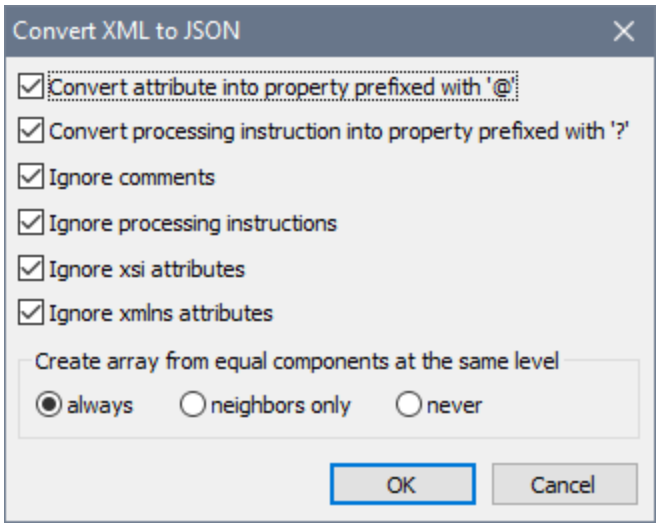

© 2016-2022 Altova GmbH

```
                "code": 10123
            },
            "Tel": [ { "type": "home",
                "Text": "\r      123 123-1234\r  "}, { "type": "office",
                "Text": "\r      123 987-9876\r  "} ]
        }
}
```

Pour convertir un document JSON en XML, rendre le document JSON actif et cliquer sur la commande **Convertir l'instance XML de/vers JSON**.

Options de conversion XML en JSON

Lorsque vous cliquez sur la commande **Convertir instance XML en/de JSON** pour convertir un document d'instance XML vers un document d'instance JSON, le dialogue Convertir XML en JSON (*capture d'écran ci-dessous*) apparaît. Vous pouvez sélectionner si vous souhaitez convertir en commentaires JSON, JSON5 ou JSON. Ensuite, définir les options de conversion que vous souhaitez et cliquez sur **OK**. Un document d'instance JSON sera généré depuis l'instance XML et le document JSON généré sera ouvert dans une nouvelle fenêtre.



La première des deux options est de définir si vous souhaitez ajouter des préfixes aux noms de propriété JSON afin d'éviter des conflits avec des éléments se trouvant dans le même niveau. Les deux listes ci-dessous expliquent la procédure. L'*attribut* XML `somenode` a été converti en une propriété JSON `@somenode`. De cette manière, vous éviterez tout conflit avec la propriété JSON `somenode` (créée depuis l'*élément* XML `somenode`).

Instance XML

```
<root somenode="value">
    <somenode>content</somenode>
</root>
```

Instance JSON

```
{
```

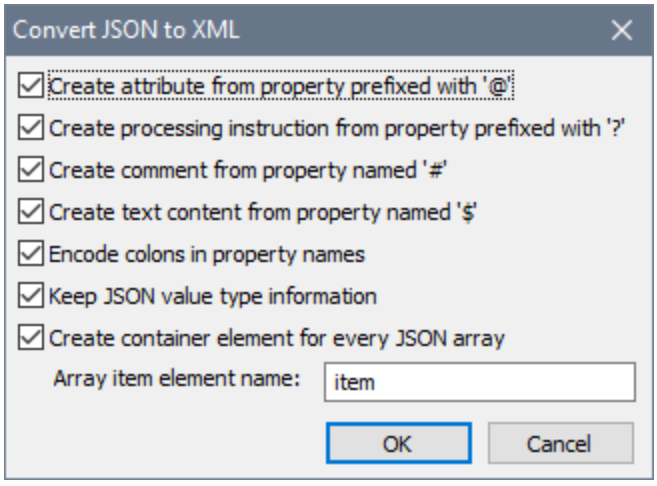
```
    "root": {
        "@somenode": "value",
        "somenode": "content"
    }
}
```

Les options suivantes vous permettent de spécifier si certains types de nœuds XML doivent être convertis ou pas. Si des commentaires XML sont inclus, ils reçoivent le nom `"#"`. Les nœuds de texte (qui se produisent généralement dans des éléments à contenu mixte) reçoivent le nom `"$"`. Si un nœud XML a un préfixe d'espace de noms, le nom JSON correspondant sera créé avec ce préfixe d'espace de noms. Si des éléments portant le même nom existe au même niveau, ils seront considérés comme des composants égaux. De manière semblable, les nœuds tels les commentaires, les instructions de traitement et le texte() au même niveau sont également des composants égaux. Si des composants égaux sont présents au même niveau, vous pourrez choisir si vous souhaitez créer un tableau ou pas. Les options sont soit créer un tableau à partir de tous les composants égaux, seulement à partir de composants égaux voisins, soit de ne pas créer de tableau du tout.

L'option *Utiliser info de type Schéma XML* permet des conversions à être faites sur la base du type de Schéma XML du nœud source. Par exemple, si un nœud est défini dans le schéma comme étant de type `xs:string`, la propriété d'objet JSON cible sera de type JSON `string` et sera entouré de guillemets. Cela est utile si, par exemple, un nombre est stocké en tant que string dans le nœud XML de source et la conversion en JSON doit aussi être fidèle en terme de type.

Options de conversion JSON en XML

Lorsque vous cliquez sur la commande **Convertir instance XML en/de JSON** pour convertir un document a instance XML vers un document d'instance JSON, le dialogue Convertir XML en JSON (*capture d'écran ci-dessous*) apparaît. Définir les options de conversion que vous souhaitez et cliquer sur **OK**. Un document d'instance XML sera généré depuis l'instance JSON et le document XML généré s'ouvrira dans une nouvelle fenêtre.



Veuillez noter les points suivants :

- Les propriétés d'objet JSON sont converties en éléments XML. Les premières options dans le dialogue vous permettent de choisir si quelques types de propriétés sont crées ou non.

© 2016-2022 Altova GmbH

- *Encoder les deux-points dans les noms de propriété :* Si sélectionnés, les deux-points dans les noms JSON sont encodés et pas créés comme deux-points. Si non sélectionnés, les deux-points sont laissés tels quels.
- *Garder l'information de type de valeur JSON :* Si sélectionnée, l'information de type JSON d'une propriété est créée comme paire d'attribut-valeur de l'élément correspondant.
- *Créer l'élément de conteneur pour chaque table JSON :* L'élément de conteneur dans le document XML obtiendra le nom de l'objet de la table JSON. Les items de l'array JSON sont créés en tant qu'éléments XML dans ce conteneur. Ils reçoivent les noms que vous avez spécifiés dans la fenêtre de saisie *Nom d'élément d'item de tableau*.

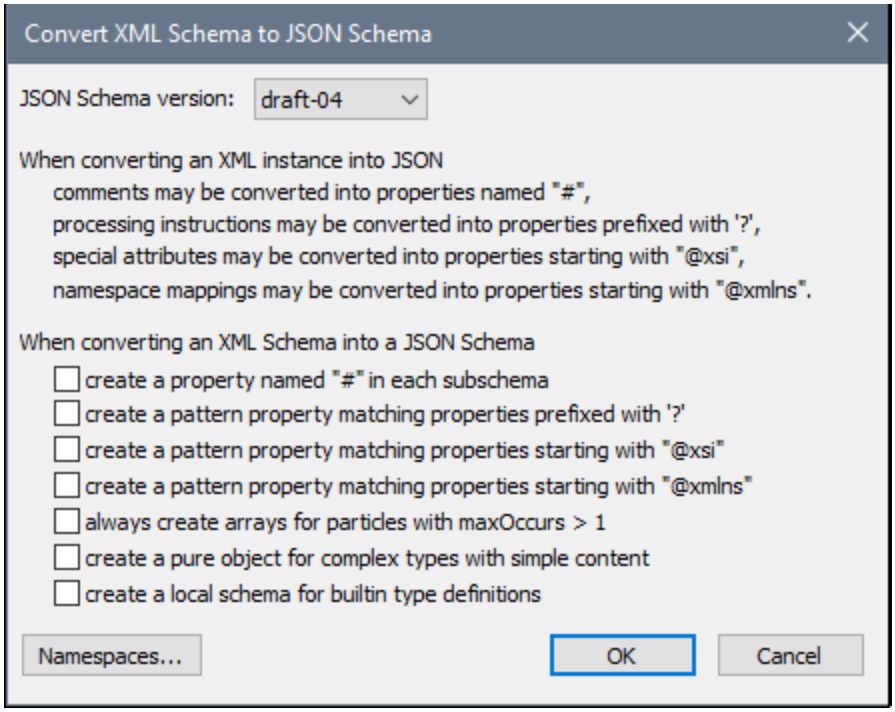
28.11.10 Convertir Schéma XML en/de Schéma JSON

Si le document actif est un schéma JSON, la commande génère un Schéma XML depuis le schéma JSON. Le document généré s'ouvre dans une nouvelle fenêtre et peut ensuite être enregistré n'importe où. Le document généré est ouvert dans une nouvelle fenêtre et peut ensuite être enregistré dans n'importe quel endroit. Les options de conversion sont décrites ci-dessous. Pour plus d'information sur la prise en charge de JSON et de l'édition JSON dans XMLSpy, voir la section JSON et JSON Schema.

Options de conversion Schéma XML en Schéma JSON

Lorsque vous cliquez sur la commande **Convertir Schéma XML en/de Schéma JSON** pour convertir un document de Schéma XML en un schéma JSON, le dialogue Convertir Schéma XML en schéma JSON (*capture d'écran ci-dessous*) apparaît. Sélectionner la version de Schéma JSON et cliquer sur **OK**. Un schéma JSON sera généré depuis le Schéma XML et le document généré s'ouvrira dans une nouvelle fenêtre.

La stratégie de conversion générale est la suivante : (i) les types simples de Schema XML sont mappés dans les types simples de Schema JSON (comme string et number) ; (ii) les types complexes de Schema XML sont mappés dans des objets JSON.



La partie supérieure du dialogue fournit des informations concernant la méthode de conversion de certains composants de Schéma XML. La partie inférieure du dialogue propose les options suivantes :

- *Créer une propriété appelée "#" dans chaque sous-schéma :* Si sélectionnée, une propriété avec ce nom est créée dans chaque définition de schéma JSON.
- *Créer des propriétés de motif correspondant à des propriétés correspondantes préfixées avec '?', "@xsi", "@xmlns":* Spécifie, pour chacun de ces préfixes, une propriété de motif pour faire correspondre les propriétés avec des noms qui possèdent ces préfixes. Pour plus d'informations concernant les propriétés de motif, voir la section Objets et propriétés JSON.
- *Toujours créer des arrays pour des particules avec maxOccurs > 1:* Dans XML Schéma, les particules sont les éléments de modèles de contenu complexe. Si le nombre d'occurrences est plus que un (1), les particules sont définies en tant qu'un tableau dans JSON Schéma. Dans le cas contraire, ils sont définis en tant que propriétés d'un objet JSON.
- *Créer un objet pur pour des types complexes avec du contenu simple :* Le type complexe de Schéma XML avec du contenu simple est un type qui autorise les attributs et le contenu de texte mais pas d'éléments enfant. Si l'option *Créer objet pur* a été sélectionnée, le type complexe est converti en un objet JSON. Les attributs du type sont convertis dans des propriétés de l'objet JSON, où les noms de propriété sont préfixés avec `@`.. Pour le contenu de texte du type, une propriété nommée `$` est générée. Si l'option *Créer un objet pur* n'est pas sélectionnée, le type complexe est converti en un objet qui peut contenir d'autres objets et des types simples JSON comme des string et des nombres.
- *Créer un schéma local pour les définitions de type built-in :* Si sélectionnée, cette option crée la définition type dans l'objet lui-même. Autrement, la définition de type est une référence à un objet séparé. Les deux cas de sortie sont affichés dans les fragments de code de Schéma JSON ci-dessous.

  La définition du type est référencée :

```
"properties": {
```


© 2016-2022 Altova GmbH

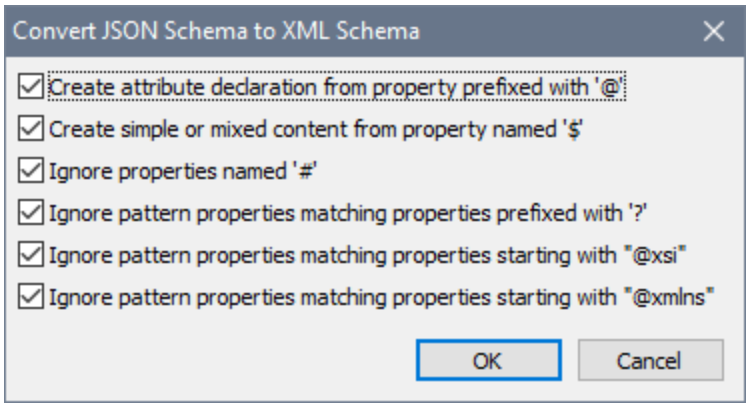
```
    "AccountManager": {
    "$ref": "#/definitions/xs:string"
    }
}
"xs:string": {
    "type": "string"
}
```

La définition du type est locale :

```
"properties": {
    "AccountManager": {
    "type": "string"
    }
}
```

Options de conversion Schéma JSON en Schéma XML

Lorsque vous cliquez sur la commande **Convertir Schéma XML de/vers Schéma JSON** pour convertir un document de Schéma JSON en un Schéma XML, le dialogue Convertir JSON en un Schéma XML (*capture d'écran ci-dessous*) apparaît. Définir les options de conversion que vous souhaitez et cliquer sur **OK**. Un schéma XML sera généré depuis le schéma JSON et le document généré s'ouvrira dans une nouvelle fenêtre.



Vous pouvez sélectionner les options suivantes :

- Si les noms de propriété JSON qui commencent avec `'@'` et `'$'` sont créés ou pas. Ils seraient créés, respectivement en tant que nœuds d'attribut et nœuds de texte.
- Si les propriétés nommées `'#'` sont créées en tant que nœuds de commentaire XML ou pas.
- Si les propriétés de motif qui correspondent aux propriétés préfixées avec `'?'`, `"@xsi"`, et/ou `"@xmlns"` sont ignorées ou pas. Si elles ne sont pas ignorées, les propriétés préfixées avec `'?'`, `"@xsi"`, et `"@xmlns"` sont converties, respectivement, en instructions de traitement, attributs préfixés `xsi:` et attributs préfixés `xmlns:`.

28.11.11 Convert to OIM xBRL-XML

Convertir les formats de données suivants en OIM xBRL-XML.

- OIM xBRL-JSON
- OIM xBRL-CSV

Faites du fichier que vous souhaitez transformer le fichier actif, puis sélectionnez la commande.

Le fichier OIM xBRL-XML sera généré, ouvert dans une nouvelle fenêtre et validé. Vous pouvez enregistrer le fichier généré dans un emplacement de votre choix.

28.11.12 Convert to OIM xBRL-JSON

Convertir les formats de données suivants en OIM xBRL-JSON.

- fichier de données XBRL
- OIM xBRL-XML
- OIM xBRL-CSV

Faites du fichier que vous souhaitez transformer le fichier actif, puis sélectionnez la commande.

Le fichier OIM xBRL-JSON sera généré, ouvert dans une nouvelle fenêtre et validé. Vous pouvez enregistrer le fichier généré dans un emplacement de votre choix.

28.11.13 Convert to OIM xBRL-CSV

Convertir les formats de données suivants en OIM xBRL-CSV.

- fichier de données XBRL
- OIM xBRL-XML
- OIM xBRL-JSON

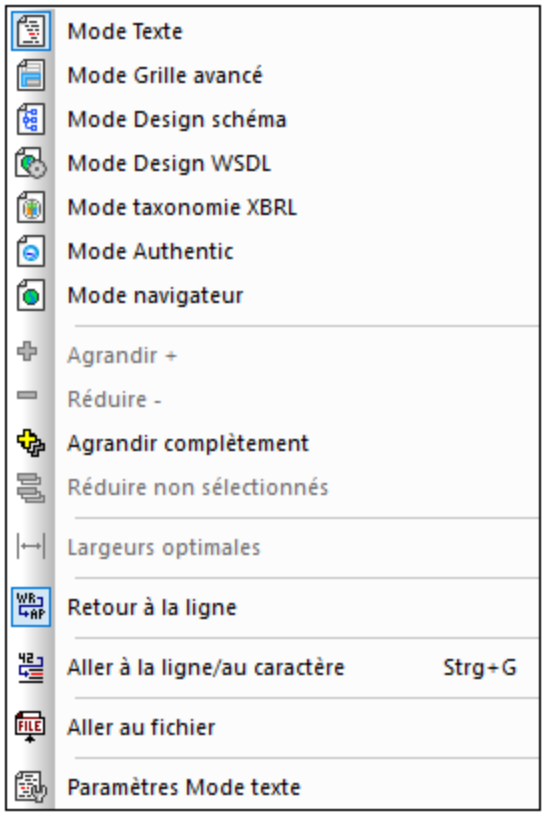
Faites du fichier que vous souhaitez transformer le fichier actif, puis sélectionnez la commande.

Vous serez invité à donner un nom au fichier sous lequel vous pourrez enregistrer un fichier JSON. Le fichier JSON contiendra les références du/des fichier(s) de données CSV qui sera/seront généré(s). Après avoir spécifié le fichier JSON et cliqué sur Enregistrer, les fichiers JSON et CSV seront générés et les résultats de validation du fichier généré seront affichés.

© 2016-2022 Altova GmbH

28.12 Menu Mode

Le menu **Mode** (*capture d'écran ci-dessous*) contrôle l'affichage de la fenêtre Principale active et vous permet de changer la manière dont est affiché le document.



Cette section fournit une description des commandes dans le menu **Mode**.

28.12.1 Mode Texte



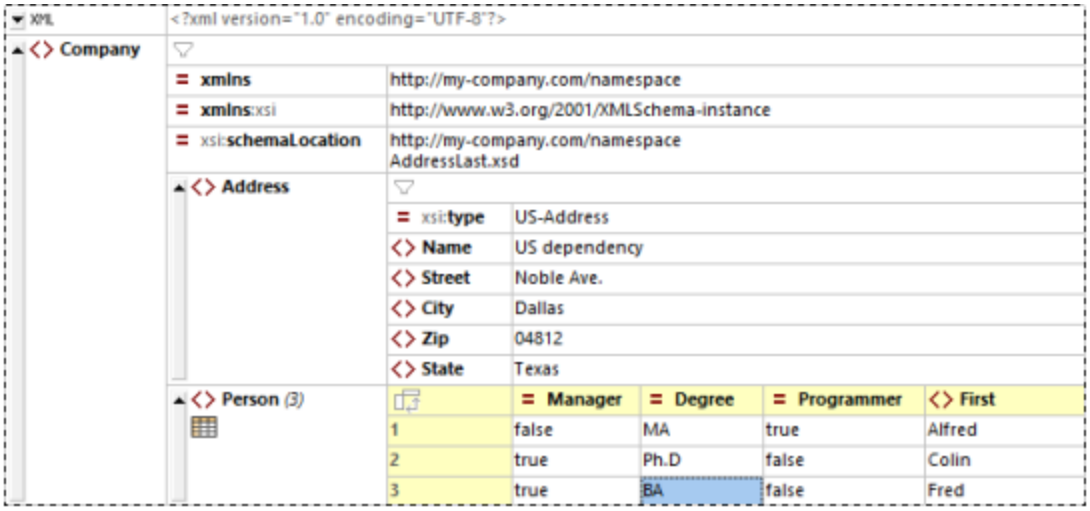
Cette commande permet de faire passer le mode actuel du document dans le Mode Texte qui vous permet d'éditer le document dans sa forme de texte. Elle prend en charge un nombre de fonctions d'édition de texte avancées, décrite en détail dans la section Mode Texte de ce document.

Note : Vous pouvez configurer des aspects du Mode Texte dans des onglets variés du dialogue Options (**Outils | Options**).

28.12.2 Mode Grille avancé



Cette commande bascule le document actuel vers le Mode Grille. Si le mode précédent est le Mode Texte, le document est automatiquement vérifié pour une bonne formation.



28.12.3 Mode Design Schéma



Cette commande fait passer le document actuel, s'il s'agit d'un document Schéma XML, dans le Mode Design Schéma. Pour une description détaillée des mécanismes disponibles dans ce mode, voir la section Mode Schéma de cette documentation.

28.12.4 Mode Design WSDL



Cette commande fait passer le document actuel, s'il s'agit d'un document WSDL (avec une extension de fichier `.wsdl`) dans le Mode Design WSDL. Ce mode est décrit en détail dans la section Mode WSDL de cette documentation.

© 2016-2022 Altova GmbH

28.12.5 Mode Taxonomie XBRL



Cette commande fait passer le document actuel au Mode Taxonomie XBRL si le document est un document de taxonomie XBRL (avec une extension de fichier `.xsd`). Veuillez noter que les documents d'instance XBRL, qui sont des fichiers XML et présentent des suffixes `.xml`, doivent être édités comme des fichiers XML normaux dans d'autres modes d'édition et ne peuvent pas être édités dans un Mode de taxonomie XBRL. Pour plus d'informations, voir la section Mode XBRL de cette documentation.

28.12.6 Mode Authentic

Cette commande fait passer le document actuel à Authentic View.

Authentic View vous permet d'éditer des documents XML basés sur des modèles StyleVision Power Stylesheet crées dans l'application StyleVision d'Altova. Ces modèles (feuilles de style StyleVision ou fichiers SPS) affichent des documents XML dans un format graphique qui permet une édition plus simple du document XML (plus simple que de l'éditer dans un format de texte avec des balises).

Si un document XML est associé avec un fichier SPS (**Authentic | Assigner une feuille de style StyleVision**), le document XML peut être consulté dans Authentic View. Vous pouvez aussi ouvrir un fichier SPS en tant qu'un nouveau modèle vide dans Authentic View, des deux manières suivantes :

- Choisir la commande **Fichier | Nouveau** puis cliquer sur la touche **Sélectionner une feuille de style StyleVision**.
- Choisir la commande **Authentic | Nouveau document** puis chercher le fichier SPS.

Voir la documentation Mode Authentic et StyleVision pour plus d'information.

28.12.7 Mode Navigateur



Cette commande fait passer le document actuel au Mode Navigateur. Un navigateur XML rend le document XML en utilisant les informations provenant des feuilles de style CSS et/ou XSL disponibles.

Au passage dans le mode Navigateur, la validité du document est tout d'abord contrôlée si l'option *Valider sur enregistrement* dans la section Fichier du dialogue Options (**Outils | Options**) a été cochée. Pour plus d'informations, voir la section Mode Navigateur de cette documentation.

28.12.8 Agrandir



Cette commande (*raccourci :* pavé numérique **'+'**) est activée dans le Mode Grille et augmente l'élément sélectionné d'un niveau. L'élément reste sélectionné après l'expansion, vous pouvez donc agrandir l'élément de plusieurs niveaux en cliquant le nombre de fois nécessaire sur la touche de raccourci **'+'**.

28.12.9 Réduire



Cette commande (*raccourci :* pavé numérique **'-'**) est activée dans le Mode Grille et réduit l'élément sélectionné d'un niveau. Vous pouvez agrandir ou réduire tout élément en cliquant sur la barre grise située à gauche de l'élément.

28.12.10 Agrandir complètement



Cette commande (*raccourci :* * ou x sur le pavé numérique) est activée dans le Mode Grille et dans le Mode Texte si la marge pliante du Mode Texte est active. Elle agrandit tous les nœuds descendants de l'élément sélectionné.

28.12.11 Réduire non sélectionnés



Cette commande (*raccourci* : **Ctrl** + pavé numérique **'-'**) est activée dans le Mode Grille et garde l'item sélectionné non réduit tout en réduisant tous les autres items. Cela permet de maximiser le contrôle d'un élément et de ses enfants tout en réduisant l'attention sur les autres nœuds.

© 2016-2022 Altova GmbH

28.12.12 Largeurs optimales



Cette commande est activée dans le Mode Grille et ajuste les largeurs de toutes les colonnes dans le Mode Grille de manière à ce que chaque colonne ait une largeur parfaitement ajustée à la plus longue ligne de texte dans toutes les cellules. Une largeur optimale maximum peut être spécifiée dans la section Mode du dialogue Options (**Outils | Options**). Veuillez noter que les largeurs optimales sont calculées sur la base des cellules visibles des colonnes. Cela permet l'optimisation du mode lorsque des éléments individuels sont réduits ou agrandis.

28.12.13 Retour à la ligne

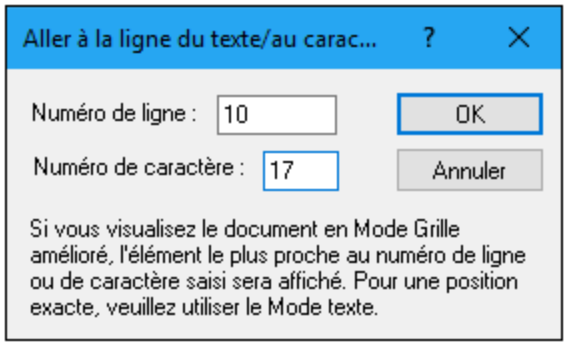


Cette commande active ou désactive le retour à la ligne dans le Mode Texte. Lorsque le retour à la ligne est activé, le texte sera coupé au bord de la fenêtre.

28.12.14 Aller à la ligne/au caractère



Cette commande (*raccourci :* **Ctrl+g**) est activée dans le Mode Texte et le Mode Grille. Elle ouvre un dialogue (*capture d'écran ci-dessous*) dans lequel vous pouvez saisir le numéro de la ligne et le numéro du caractère où vous souhaitez aller. Dans le Mode Texte, le curseur sautera à la position que vous avez saisie. Dans le Mode Grille, le nœud le plus proche de la ligne et/ou du numéro du caractère que vous avez saisi sera marqué.



Cette fonction est utile lorsque vous souhaitez vous rendre rapidement à un emplacement, par exemple, lorsque l'emplacement d'une erreur est indiqué dans un message d'erreur.

28.12.15 Aller au fichier

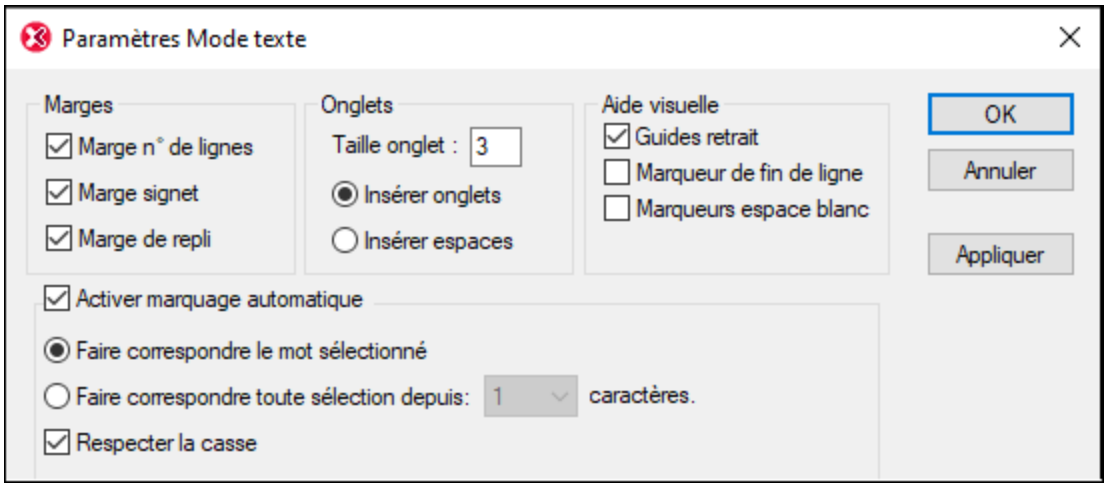


Cette commande est activée dans le Mode Texte et le Mode Grille. Lorsque le curseur est placé dans le texte qui référence un fichier (dans le Mode Texte) ou dans un nœud (dans le Mode Grille) qui contient du texte référençant un fichier, le document référencé est ouvert. Il ouvre le document référencé, depuis le fichier que vous éditez actuellement.

28.12.16 Paramètres mode Texte



La commande **Paramètres Mode Texte** est activée dans le Mode Texte. Elle ouvre le dialogue Paramètres mode Texte (*capture d'écran ci-dessous*), dans lequel vous pouvez configurer le Mode Texte. Une icône de raccourci  pour ouvrir le dialogue est disponible dans la barre d'outils Texte.



Marges

Dans le volet Marges, les marges de numéro de ligne, de signet et de replis peuvent être activées et désactivées. Chacune d'entre elles est une marge séparée dans le Mode Texte et affiche respectivement : l'icône du (i) numéro de lignes, (ii) des signets et du (ii) pli de source pour agrandir/réduire les nœuds. Les paramètres du volet Marges déterminent si les marges sont affichées dans le Mode Texte ou non. Les commandes de signet se trouvent dans le menu Éditer. Vous pouvez agrandir et réduire des nœuds dans le mode Texte uniquement si vous avez activé le paramètre *Marge de repli*.

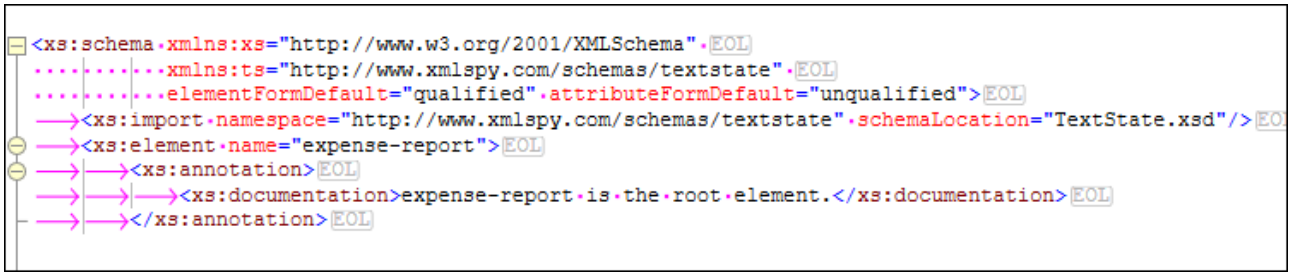
© 2016-2022 Altova GmbH

Onglets

Le volet Onglet vous permet de définir la taille d'onglet en termes d'espaces. Le bouton radio en-dessous du paramètre *Taille d'onglet* détermine si des documents sont affichés avec un retrait d'onglet ou d'espace lorsque l'option pretty-print-avec-retrait est activée dans la section Mode du dialogue Options (**Outils | Options**).

Aide visuelle

Le volet Aide visuelle contient des paramètres pour activer les guides de retrait (lignes verticales pointillées qui montrent le retrait du texte, *voir capture d'écran ci-dessous*), marqueurs de fin de ligne et marqueurs d'espace blanc (tabulateurs et espaces). (Les tabulateurs sont indiqués par des flèches, alors que les espaces sont indiqués par des pointillés (*tous les deux en rose dans la capture d'écran ci-dessous*). Les couleurs des marqueurs d'espace blanc peuvent être personnalisés dans les options de Mode Texte du dialogue Options.)



Activer marquage automatique

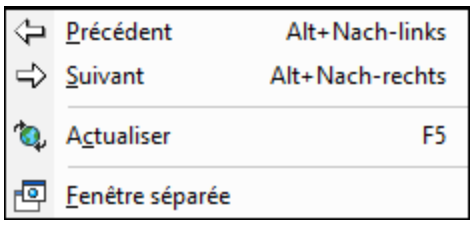
Si le marquage est activé, toutes les occurrences d'une sélection dans le Mode Texte sont marquées. Ce qui constitue une sélection peut être défini par le biais des options dans ce volet. Une sélection peut être définie pour constituer un mot entier ou un nombre fixe de caractères, la casse de texte comptant ou ne comptant pas pour une correspondance. Pour une sélection de caractère, vous pouvez spécifier le nombre minimum de caractères à faire correspondre (par exemple, deux caractères ou plus). Dans le Mode Texte, toutes les occurrences des séquences de caractère qui correspondent à votre sélection seront marquées. En ce qui concerne les recherches de mot, les noms d'élément, les noms d'attribut, les valeurs d'attribut dans guillemets et les crochets de balises d'éléments sont considérés comme étant des mots séparés.

Mappage touches

Le mappage de touches est une liste de raccourcis XMLSpy et de leurs commandes associées.

28.13 Menu Navigateur

Les commandes de menu **Navigateur** sont activées dans le mode Navigateur uniquement. Les commandes **Précédent** et **Suivant** , toutefois, sont en outre activés dans le mode Schéma.



Précédent, Suivant

La commande **Suivant** (*raccourci :* **Alt + Flèche de gauche**) est activée dans le Mode Navigateur et dans le Mode Schéma.

- Dans le Mode Navigateur, la commande Précédent affiche la page consultée précédemment. La touche **Retour arrière** obtient le même effet. La commande est utile lorsque vous cliquez sur un lien dans votre document XML puis souhaitez retourner à votre document XML.
- Dans le Mode Schéma, la commande Précédent vous amène au composant ou au mode précédemment consulté. Elle peut vous ramener jusqu'aux 500 dernières positions consultées.

La commande **Suivant** (*raccourci :* **Alt + Flèche de Droite**) est activée dans le Mode Navigateur et dans le Mode Schéma.

- Dans le Mode Navigateur, la commande vous déplace à travers les pages consultées précédemment.
- Dans le Mode Schéma, la commande vous amène au composant ou au mode précédemment consulté. Elle peut vous ramener jusqu'aux 500 dernières positions consultées.

Actualiser

La commande **Actualiser (F5)** est activée dans le Mode Navigateur et met à jour le Mode Navigateur en rechargeant le document actuel et les documents liés au document actuel (comme des feuilles de style CSS et XSL, et des DTD).

Fenêtre séparée

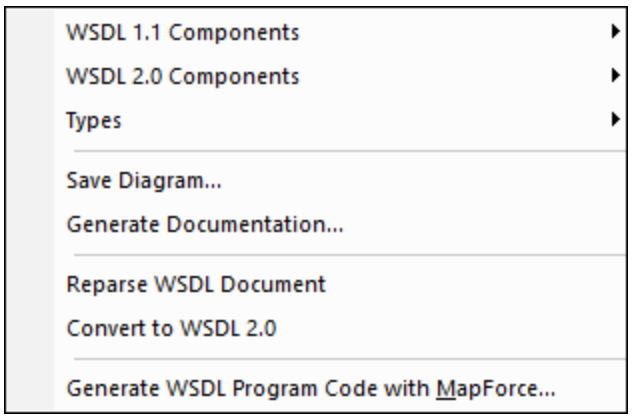
La commande **Fenêtre séparée** est activée dans le Mode Navigateur et décroche le Mode Navigateur du document des autres modes. En tant que fenêtre séparée, le Mode Navigateur peut être affiché côte-à-côte avec un mode d'édition du document.

Pour actualiser le Mode Navigateur séparé après avoir effectué une modification dans un mode d'édition, appuyer **F5** dans le mode d'édition. Pour réancrer une fenêtre en mode Navigateur séparée dans la fenêtre d'application, activez la fenêtre du mode Navigateur et cliquez sur la commande **Fenêtre séparée**.

© 2016-2022 Altova GmbH

28.14 Menu WSDL

Les commandes dans le menu **WSDL** sont disponibles lorsque vous consultez un document WSDL dans le Mode WSDL, qui est l'éditeur graphique pour créer et éditer des documents WSDL.

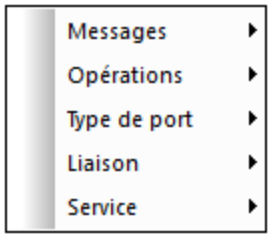


Pour une description du Mode WSDL, voir la section Mode WSDL. Pour vous lancer dans WSDL, voir le Tutoriel WSDL.

Voir aussi : Vous trouverez plus d'informations concernant le travail avec des documents WSDL dans les sections, Mode WSDL et Tutoriel WSDL.

28.14.1 Composants WSDL 1.1

Passer avec la souris au-dessus de l'item de menu **Composants WSDL 1.1** pour ouvrir un sous-menu (*capture d'écran ci-dessous*) à partir duquel vous pouvez sélectionner plusieurs commandes pour éditer les composants WSDL 1.1.



Chaque item du menu WSDL 1.1 (*capture d'écran ci-dessus*) déroule son propre sous-menu, à partir duquel les commandes se rapportant à ce composant peuvent être sélectionnées. Les commandes dans chacun de ces sous-menus sont décrites dans les sous-sections de cette section.

28.14.1.1 Messages

Insérer message

Ajoute un nouveau message au document WSDL. L'item Messages dans l'assistant à la saisie Aperçu s'ouvre et le message récemment créé y est marqué.

Supprimer message

Supprimer le message sélectionné depuis l'élément d'entrée ou de sortie.

Ajouter partie de message (paramètre)

Ajoute une partie de message (paramètre) au message sélectionné.

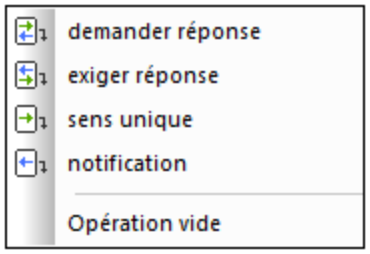
Supprimer partie de message (paramètre)

Supprime une partie de message (paramètre) depuis le message sélectionné.

28.14.1.2 Opérations

Ajouter une opération

Ajoute une nouvelle opération au Type de Port sélectionné. Le type de l'opération à apposer peut être sélectionné depuis le sous-menu (*capture d'écran ci-dessous*) de la commande de menu **Ajouter Opération**.



Supprime une opération

Supprime l'opération Type de Port sélectionnée.

Ajoute un élément d'entrée

Ajoute un élément d'entrée de Type de Port sélectionné.

Ajoute un élément de sortie

Ajoute un élément de sortie de Type de Port sélectionné.

© 2016-2022 Altova GmbH

Ajoute un élément de faute

Ajoute un élément de faute de Type de Port sélectionné.

Supprime un élément Entrée/Sortie/Faute

Supprime l'élément d'entrée/de sortie ou de faute Type de Port sélectionné.

Ajoute Nouveau message à l'élément Entrée/Sortie/Faute

Ajoute un nouveau message (par défaut), à l'élément d'entrée/de sortie ou de faute Type de Port actuellement sélectionné.

28.14.1.3 Type de port

Insérer Type de port

Ajoute un nouveau Type de Port à la colonne de la Fenêtre principale.

Supprimer Type de port

Supprime le Type de Port sélectionné depuis la colonne Types de Port de la Fenêtre principale.

28.14.1.4 Liaison

Insérer Liaison

Ajoute une nouvelle liaison à la colonne Liaisons de la Fenêtre principale.

Supprimer Liaison

Supprime la liaison sélectionnée de la colonne Liaisons de la Fenêtre principale.

Apposer Enfant

Permet l'ajout d"un nouvel élément d'extensibilité à un message d'entrée ou de sortie. Si l'item de menu n'est pas disponible, il n'est pas autorisé dans cette position. Voir les Specs W3C WSDL pour plus d'informations concernant les items d'Extensibilité.

Les items d'extensibilité suivants sont disponibles :

- soap:body
- soap:header
- soap:headerfault
- soap:fault
- mime:content
- mime:multipartrelated
- mime:part

- mime:mimeXml
- http:urlencoded
- http:urlreplacement

Supprimer extensibilité

Supprime l'item d'extensibilité sélectionné.

28.14.1.5 Service

Insérer Service

Ajoute un nouveau service dans la colonne Services de la Fenêtre principale.

Supprimer Service

Supprime le service sélectionné dans la colonne Services de la Fenêtre principale.

Insérer Port

Ajoute un nouveau port au service sélectionné dans la colonne Services de la Fenêtre principale.

Supprimer Port

Supprime le port sélectionné depuis le service actuellement sélectionné.

28.14.2 Composants WSDL 2.0

Passer avec la souris au-dessus de l'item de menu **Composants WSDL 2.0** permet de dérouler un sous-menu (*capture d'écran ci-dessous*) à partir duquel les différentes commandes pour l'édition des composants WSDL 2.0 peuvent être sélectionnés.

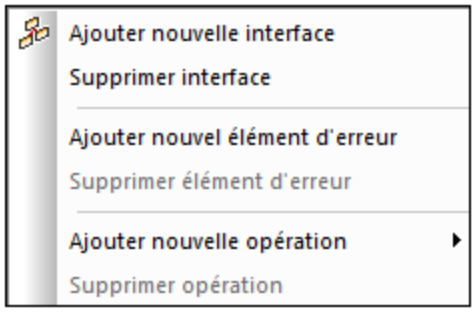


Chaque item du menu Composants WSDL 2.0 (*capture d'écran ci-dessus*) comporte un sous-menu individuel à partir duquel les commandes ayant trait à ce composant peuvent être sélectionnées. Les commandes dans chacune de ces sous-menus sont décrites dans les sous-sections de cette section.

© 2016-2022 Altova GmbH

28.14.2.1 Interface

Les commandes suivantes sont disponibles dans le menu **Interface** (*capture d'écran ci-dessous*).



Ajoute une nouvelle interface

Ajoute une nouvelle fenêtre d'interface à la colonne Interfaces dans la Fenêtre principale. Le nom par défaut de l'interface est marqué dans la fenêtre d'interface, vous permettant de saisir un nouveau nom directement.

Supprime une interface

Supprime l'interface sélectionnée.

Ajoute une nouvelle faute

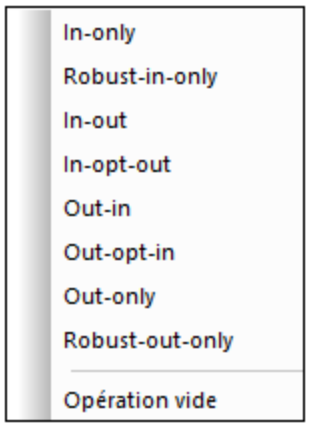
Ajoute un nouvel élément `fault` à l'interface sélectionnée. Le nom par défaut de la `fault` est marqué dans la fenêtre d'interface, vous permettant de saisir un nouveau nom directement.

Supprime une faute

Supprime la faute sélectionnée.

Ajoute une nouvelle opération

Ajoute un nouvel élément `operation` à l'interface sélectionnée. Le type de l'opération à ajouter est sélectionné depuis le menu pop-out (*capture d'écran ci-dessous*), et peut être un des types d'opération affiché dans la capture d'écran.



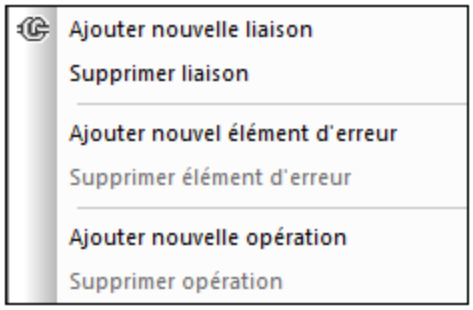
Le nom par défaut de l'`operation` est marqué dans la fenêtre d'interface, vous permettant de saisir un nouveau nom directement.

Supprimer l'opération

Supprime l'opération sélectionnée.

28.14.2.2 Liaison

Les commandes suivantes sont disponibles dans le menu **Liaison** (*capture d'écran ci-dessous*).



Ajoute une nouvelle liaison

Ajoute une nouvelle fenêtre de liaison à la colonne Liaison dans la Fenêtre principale. Le nom par défaut de la liaison est marqué dans la fenêtre de liaison, vous permettant de saisir un nouveau nom directement.

Supprime une liaison

Supprime la liaison sélectionnée.

© 2016-2022 Altova GmbH

Ajoute une nouvelle faute

Ajoute un nouvel élément `fault` à la liaison sélectionnée. Un élément `fault` dans une liaison contient un attribut `ref` qui référence une faute déclarée dans une interface. Dans la faute récemment créée dans la liaison, la faute d'interface qui doit être référencée peut être sélectionnée depuis la liste de choix de la faute récemment créée.

Supprime une faute

Supprime la faute sélectionnée.

Ajoute une nouvelle opération

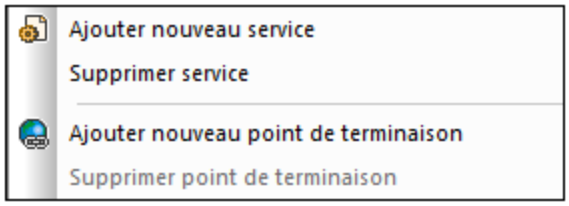
Ajoute un nouvel élément `operation` dans la liaison sélectionnée. Un élément `operation` dans une liaison contient un attribut `ref` qui référence une opération déclarée dans une interface. Dans l'opération récemment créée dans la liaison, l'opération d'interface qui doit être référencée peut être sélectionnée depuis la liste de choix de l'opération de liaison récemment créée.

Supprime une opération

Supprime l'opération sélectionnée.

28.14.2.3 Service

Les commandes suivantes sont disponibles dans le menu **Service** (*capture d'écran ci-dessous*).



Ajoute un nouveau service

Ajoute une nouvelle fenêtre à la colonne Services de la Fenêtre principale. Le nom par défaut du service est marqué dans la fenêtre de service, vous permettant de saisir un nouveau nom directement. La référence d'interface peut être sélectionnée depuis la liste de choix pour la propriété *Interface*.

Supprime service

Supprime le service sélectionné.

Ajoute un nouveau point d'arrêt

Ajoute un nouvel élément `endpoint` au service sélectionné. Le nom par défaut du point d'arrêt est marqué dans la fenêtre de service, vous permettant de saisir un nouveau nom directement. La référence de liaison peut être

sélectionnée depuis la liste de choix pour la propriété *Liaison*. L'adresse du point d'arrêt doit être saisie dans le champ pour la propriété *Adresse*

Supprime point d'arrêt

Supprime le point d'arrêt sélectionné.

28.14.3 Types, Enregistrer diagramme

L'item de menu **Types** a un sous-menu contenant les commandes suivantes. Celles-ci sont décrites ci-dessous.

- **Nouveau schéma**
- **Intégrer schéma**
- **Extraire schéma(s)**
- **Éditer schéma(s) dans le Mode Schéma**

La commande **Enregistrer le diagramme** enregistre le diagramme de design en tant que fichier PNG.

Types | Nouveau schéma

Cette option devient uniquement active si le fichier WSDL ne contient pas d'élément de schéma.
Veuillez noter que lorsque vous utilisez l'option de menu **Fichier | Nouveau**, un élément de schéma est inclus dans le fichier squelette WSDL. Dans ce cas, l'item de menu ne peut pas être sélectionné (voir ci-dessous).

```
<types>
              <xs:schema/>
</types>
```

Types | Incorporer schéma

La commande ouvre un dialogue Ouvrir fichier, dans lequel vous pouvez chercher le fichier de schéma que vous souhaitez incorporer. En cliquant sur **OK** dans le dialogue, le schéma est créé en tant qu'un schéma inline dans l'élément `types`. Si le schéma sélectionné a déjà été importé, vous serez invité à choisir si vous souhaitez incorporer le schéma déjà importé. Si vous décidez d'incorporer le schéma importé, il sera converti en un schéma inline dans le cadre de l'élément `types`.

Types | Extraire schéma(s)

En sélectionnant cette commande, chacun des schémas incorporés (défini inline dans l'élément `types`) est ouvert en tant que fichier temporaire dans le Mode Schéma et un dialogue Enregistrer sous s'ouvre pour chaque fichier. Si vous choisissez d'enregistrer un fichier de schéma, le schéma sera extrait depuis le fichier WSDL, enregistré à l'emplacement que vous spécifiez, puis importé dans le fichier WSDL. Il ne sera plus un schéma incorporé, mais un schéma externe, importé.

Types | Éditer schéma(s) dans le Mode Schéma

Ouvre un fichier de schéma squelette si le fichier WSDL ne contient pas fichier de référence à un schéma spécifique. C'est le cas si vous avez utilisé l'option de menu **Fichier | Nouveau**. Si une référence à un schéma spécifique existe, le schéma s'ouvre dans le Mode Schéma de l'éditeur graphique WSDL.

© 2016-2022 Altova GmbH

28.14.4 Générer documentation

La commande **WSDL | Générer documentation** génère une documentation détaillée du fichier WSDL actuel. Vous pouvez créer une documentation de format HTML, MS Word, RTF ou PDF. La documentation générée par cette commande peut être modifiée et utilisée selon votre gré ; une permission de la part d'Altova n'est pas nécessaire. La documentation est générée pour les composants que vous sélectionnez dans le dialogue Documentation WSDL (qui apparaît lorsque vous sélectionnez la commande Générer Documentation). Les composants liés sont reliés par hyperliens dans la sortie sur l'écran, vous permettant de naviguer d'un composant à l'autre. Veuillez noter que la documentation WSDL peut aussi être générée pour les **fichiers WSDL et de Schéma XML importés**. Les différentes options de génération de documentation sont décrites dans la section Documentation Options.

Note : Afin de générer une documentation dans un format MS Word, vous devez avoir installé MS Word (version 2000 ou plus).

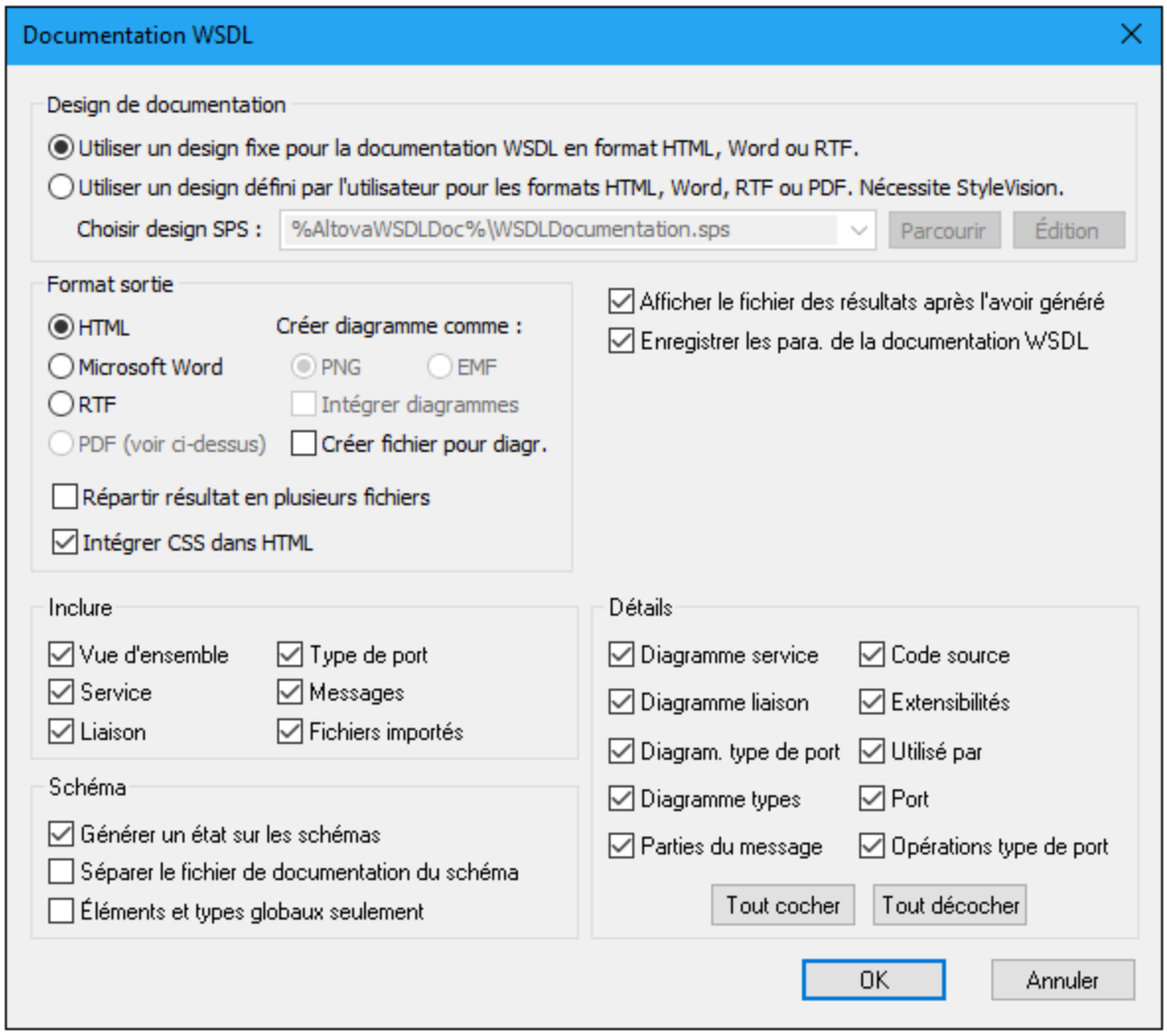
Vous pouvez utiliser soit le design fixe de XMLSpy pour le document généré, soit utiliser une SPS de StyleVision pour le design. L'utilisation d'une SPS de StyleVision vous permet de personnaliser le design de la documentation générée. Suivez les explications dans la section, Design défini par l'utilisateur.

Note : Afin d'utiliser une SPS pour générer une documentation WSDL, vous devez avoir installé StyleVision sur votre appareil.

28.14.4.1 Options de documentation

La commande **WSDL | Générer documentation** ouvre le dialogue Documentation WSDL (*capture d'écran ci-dessous*), dans lequel vous pouvez choisir des options pour la documentation.

Dans le volet Design de Documentation du dialogue, vous pouvez sélectionner si vous souhaitez utiliser le design fixe XMLSpy pour la documentation générée ou si vous souhaitez utiliser un design personnalisé créé dans une SPS StyleVision. Sélectionner l'option de votre choix. Veuillez noter que la sortie PDF est disponible uniquement pour la documentation générée avec une SPS StyleVision, et non pas pour la documentation générée avec un design fixe. Vous trouverez une description du maniement d'un design défini par l'utilisateur dans la section Design défini par l'utilisateur.



Les autres options dans le dialogue Documentation WSDL sont expliquées ci-dessous. Selon que le document actif est un WSDL 1.1 ou un WSDL 2.0, les items dans le volet Inclure et Détails du dialogue seront différents. La capture d'écran ci-dessus montre le dialogue de Documentation WSDL pour un document WSDL 1.1.

- Le format requis est spécifié dans le volet Format de sortie : soit HTML, Microsoft Word, RTF ou PDF. (Le format de sortie PDF est uniquement disponible si vous utilisez une SPS de StyleVision pour générer la documentation.) En cliquant sur **OK**, vous serez invité à choisir le nom du fichier et l'emplacement où il devra être enregistré.
- Les documents Microsoft Word sont créés avec l'extension de fichier `.doc` lorsqu'ils sont générés à l'aide d'un design fixe, et avec une extension de fichier `.docx` lorsqu'ils sont générés avec une SPS de StyleVision.
- La documentation peut être générée soit en tant que fichier unique soit entre plusieurs fichiers. Si plusieurs fichiers sont générés, chaque fichier correspond à un composant. Les composants inclus dans la sortie sont spécifiés à l'aide de cases à cocher dans le volet Inclure. Dans les designs fixes, les liens entre les documents multiples sont créés automatiquement.
- Pour la sortie HTML, les définitions de style CSS peuvent soit être enregistrées dans un fichier CSS séparé, soit intégrées dans le fichier HTML (dans l'élément `<head>`). Si un fichier CSS séparé est créé,

© 2016-2022 Altova GmbH

il recevra le même nom que le fichier HTML, mais aura une extension `.css`. Cocher ou décocher la case *Intégrer CSS dans HTML* pour définir l'option requise.

- L'option *Intégrer les diagrammes* est activée pour les options de sortie MS Word, RTF et PDF. Lorsque cette option est cochée, les diagrammes sont intégrés dans le fichier de résultats, soit dans un format PNG. Sinon, les diagrammes sont créés sous la forme de fichiers PNG qui sont affichés dans le fichier de résultats par le biais de liens d'objets.
- Si la sortie est un HTML, tous les diagrammes sont créés en tant que fichiers PNG externes. Si la case *Créer dossier pour les diagrammes* est cochée, un dossier sera créé dans le même dossier que le fichier HTML et les fichiers PNG seront enregistrés à l'intérieur. Ce dossier aura un nom de format *`HTMLFilename`*`_diagrams`. Si la case à cocher *Créer dossier pour les diagrammes* est décochée, les fichiers PNG seront enregistrés dans le même dossier que le fichier HTML.
- Dans le volet Inclure, vous choisissez les items que vous souhaitez inclure dans la documentation. L'option *Vue d'ensemble* liste tous les composants en haut du fichier, organisés par type de composant. Si l'option *Fichiers importés* (WSDL 1.1) ou *Fichiers importés/inclus* (WSDL 2.0) est cochée, les composants dans les fichiers importés (ainsi que les fichiers inclus dans le cas de WSDL 2.0) sont inclus dans la documentation de schéma.
- Dans le volet Schéma, vous pouvez sélectionner si des schémas dans le fichier sont rapportés ou pas. Si vous choisissez de rapporter les schémas, vous pouvez choisir si : (i) vous souhaitez rapporter la documentation de schéma dans un fichier séparé ou dans le fichier de documentation principal, et si (ii) le schéma entier doit être rapporté ou bien seulement les éléments globaux, les types simples et les types complexes.
- Le volet Détails liste les détails qui peuvent être inclus pour chaque composant. Sélectionner les détails que vous souhaitez inclure dans la documentation. Les touches **Tout cocher tout** et **Tout décocher** vous permettent de sélectionner ou désélectionner rapidement toutes les options dans le volet.
- L'option *Afficher le fichier de résultat* est activée pour toutes les options de sortie. Lorsque cette option est cochée, les fichiers de résultat sont affichés dans le Mode Navigateur (sortie HTML), MS Word (sortie MS Word), et les applications par défaut pour les fichiers `.rtf` (sortie RTF) et les fichiers `.pdf` (sortie PDF).

Valeurs de paramètre

Si la SPS de StyleVision contient une ou plusieurs définitions de paramètre, en cliquant sur **OK**, un dialogue s'ouvre et liste tous les paramètres définis dans la SPS. Vous pouvez saisir les valeurs de paramètres dans ce dialogue pour contourner les valeurs de paramètre par défaut qui ont été assignées dans la SPS.

28.14.4.2 Design défini par l'utilisateur

Au lieu d'utiliser le design standard fixe de XMLSpy, vous pouvez créer un design personnalisé pour la documentation WSDL. Le design personnalisé est créé dans une SPS StyleVision, qui est un modèle de design pour le document de sortie.

Créer la SPS

Une StyleVision Power Stylesheet (ou SPS) est créée à l'aide d'un produit StyleVision d'Altova. Les SPS pour générer une documentation WSDL doivent être basées sur un Schéma XML qui spécifie la structure de la documentation WSDL. Deux schémas, un pour WSDL 1.1 et le deuxième pour WSDL 2.0, sont fournis avec votre package XMLSpy. Il s'agit respectivement de `WSDLDocumentation.xsd` et `WSDL20Documentation.xsd`, situés respectivement dans les sous-dossiers de dossier (Mes documents) :

- C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Documentation\WSDL.
- C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Documentation\WSDL20.

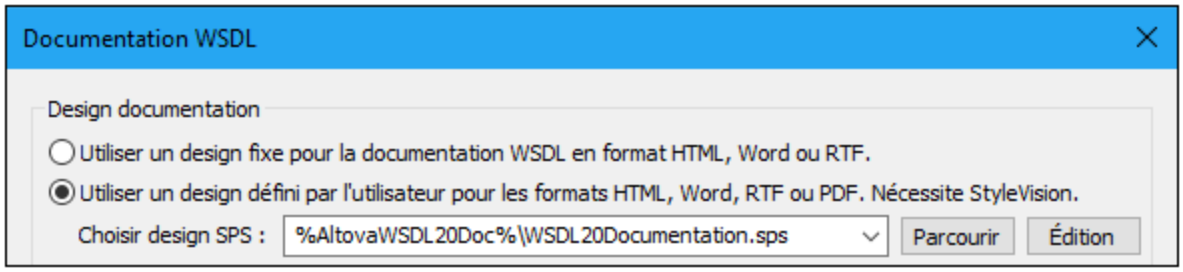
Lors de la création du design SPS dans StyleVision, les nœuds provenant du schéma sont placés dans le modèle de design, et des styles et propriétés leur sont attribués. Des composants supplémentaires, comme des liens, des tables et des images, peuvent aussi être ajoutés au design SPS. De cette manière, le document de sortie entier peut être conçu dans la SPS. Vous trouverez une description détaillée d'un design SPS de StyleVision dans le manuel d'utilisateur de StyleVision.

L'avantage d'utiliser une SPS pour générer la documentation WSDL est que vous avez le plein contrôle d'un design SPS. Veuillez aussi noter qu'une sortie PDF de la documentation WSDL est uniquement disponible si une SPS définie par l'utilisateur est utilisée ; la sortie PDF n'est pas disponible si le design fixe de XMLSpy est utilisé.

Spécifier la SPS à utiliser pour la documentation WSDL

Une fois qu'une SPS a été créée, elle peut être utilisée pour générer une documentation WSDL. La SPS que vous souhaitez utiliser pour générer la documentation WSDL est sélectionnée dans le dialogue de Documentation WSDL (accédé par le biais de la commande **WSDL | Générer documentation**). Dans le volet Design de documentation de ce dialogue (*capture d'écran ci-dessous*), sélectionner le bouton radio *Utiliser un design défini par l'utilisateur*. Vous pouvez cliquer sur la touche **Parcourir** et chercher la SPS que vous souhaitez. Cliquer sur la touche **OK** du dialogue et, dans le dialogue enregistrer qui s'ouvre, sélectionner le dossier du fichier de sortie et saisir son nom.

Note : Le fichier SPS doit localiser correctement le schéma sur lequel il est basé : `WSDLDocumentation.xsd` ou `WSDL20Documentation.xsd` (*voir ci-dessus*).



Deux designs SPS éditables, un pour WSDL 1.1 et pour WSDL 2.0, sont fournis avec XMLSpy. Ils se trouvent, respectivement, dans les sous-dossiers `WSDL` et `WSDL20` du dossier (Mes documents) : `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2011\Documentation\`. Ils se nomment :

- `WSDL\WSDLDocumentation.sps`
- `WSDL20\WSDL20Documentation.sps`

Ces fichiers, avec les autres fichiers SPS que vous avez récemment parcourus, seront disponibles dans la liste de choix de l'option *Utiliser un design défini par l'utilisateur* (*voir capture d'écran ci-dessous*).

Cliquer sur la touche **Éditer** dans le volet Design de Documentation pour lancer StyleVision et ouvrir la SPS sélectionnée dans une fenêtre StyleVision. Afin de consulter le document de résultat dans StyleVision, vous devrez obtenir un fichier de Travail XML. Des fichiers XML de modèle sont disponibles à cette fin :

© 2016-2022 Altova GmbH

`TimeService.xml` et `TimeService20.xml`, et sont fournis avec votre application, ils sont situés sous dossier (Mes documents) :

```
C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Documentation\WSDL(20)
\SampleData
```

Note : Afin d'utiliser une SPS pour générer la documentation WSDL, vous devez avoir installé StyleVision sur votre appareil.

28.14.5 Reparser le Document WSDL

Reparse the document. Cela est exigé dans certaines situations, par exemple, lorsqu'un schéma associé avec le document WSDL a été modifié. Reparser le document WSDL permettra de le mettre à jour avec des informations provenant du document de schéma modifié.

28.14.6 Convertir en WSDL 2.0

La commande **Convertir en WSDL 2.0** est activée uniquement lorsqu'un WSDL 1.1 est actif dans le Mode WSDL. Il génère un document WSDL 2.0 depuis le document WSDL 1.1 actif. Cliquer sur cette commande pour ouvrir un dialogue Enregistrer fichier dans lequel vous pouvez spécifier l'emplacement et le nom du fichier WSDL 2.0 qui sera généré par XMLSpy.

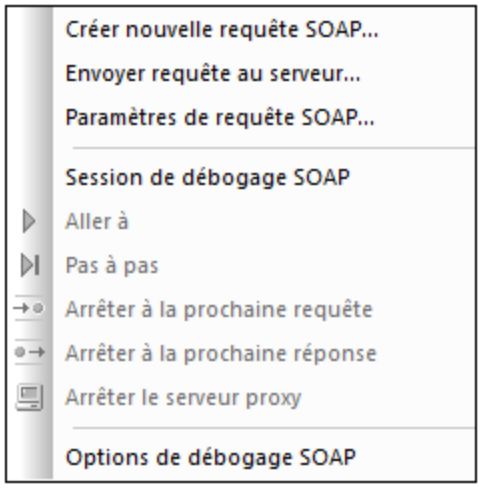
En cliquant sur **OK** dans le dialogue Enregistrer fichier, un document WSDL 2.0 est généré et enregistré dans l'emplacement spécifié et s'ouvre dans le Mode WSDL dans un nouvel onglet. Le fichier peut ensuite être édité selon vos besoins, comme tout autre document WSDL 2.0.

28.14.7 Générer le code de programme WSDL avec MapForce

La commande **Générer le code de programme WSDL avec MapForce** lance l'application MapForce d'Altova si celle-ci est installée. MapForce vous permet de mapper un schéma vers un autre DTD, Schéma XML, ou base de données pour générer XML et pour générer un code de programme depuis le fichier WSDL.

28.15 Menu SOAP

XMLSpy prend en charge SOAP versions 1.1 et 1.2, et WSDL versions 1.1 et 2.0.



La section Tutoriel SOAP qui suit les descriptions de menu vous montre comment utiliser le débogueur SOAP à l'aide du serveur **nanonull.com timeservice** fourni par Altova. Veuillez utiliser ce service pour tester le débogueur SOAP. Le service web AirportWeather, décrit dans les pages suivantes n'est pas forcément toujours disponible.

Utiliser les fonctions SOAP :

- Pour tester vos services web sans devoir mettre en place des applications client
- pour un test rapide de services web tiers

Informations supplémentaires

Pour plus d'infos concernant ces spécifications veuillez consulter :
SOAP http://www.w3.org/TR/SOAP/
WSDL http://www.w3.org/TR/wsdl

28.15.1 Créer une nouvelle requête SOAP

Cette commande crée un nouveau document de requête SOAP. Elle comporte les étapes suivantes :

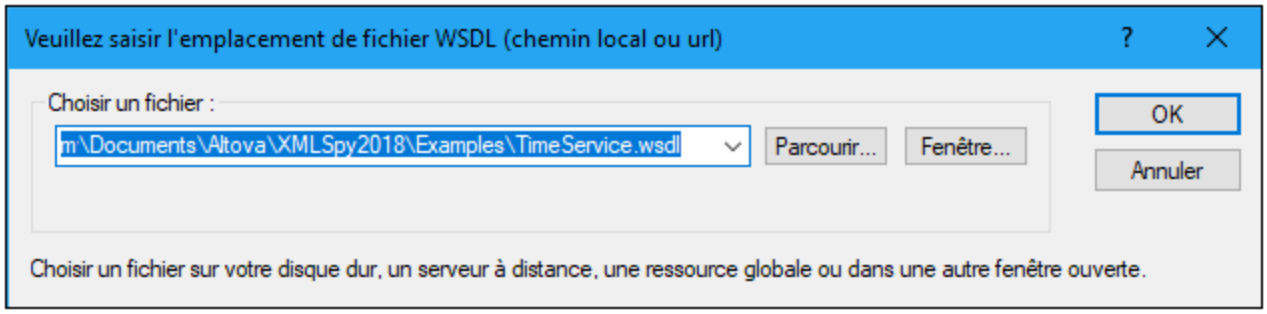
1. Saisir l'emplacement de fichier WSDL et se connecter au serveur SOAP.
2. Le serveur répond avec une liste d'opérations. Sélectionner l'opération SOAP que vous souhaitez.
3. Le serveur répond avec un formulaire de requête SOAP de format XML. Définir le formulaire de requête SOAP.

Nous démontrons le processus ci-dessous en créant une requête SOAP pour le US National Digital Forecast Database (NDFD) SOAP Service (http://www.nws.noaa.gov/xml/).

© 2016-2022 Altova GmbH

Se connecter au serveur SOAP

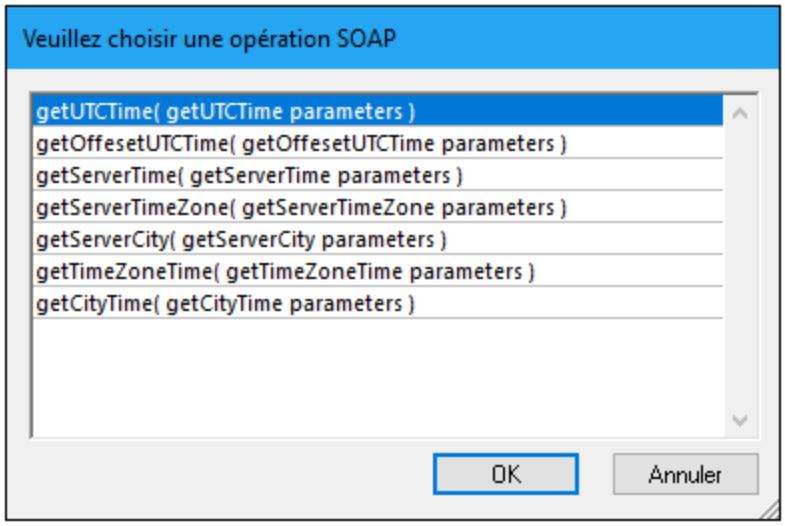
La connexion est établie par le biais d'un fichier WSDL. Dans notre exemple, l'URI du fichier WSDL est : http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl. Pour établir la connexion, cliquer sur la commande **Créer nouvelle requête SOAP**, et saisir l'URI de fichier dans le dialogue qui apparaît (*capture d'écran ci-dessous*).



Cliquer sur **OK** pour confirmer la sélection.

Sélectionner l'opération SOAP requise

Le serveur répond avec une liste d'opérations qui sont affichées dans un dialogue (*capture d'écran ci-dessous*).



Choisir une opération et cliquer sur **OK**. Nous avons choisi l'opération `NDFDgenByDay`.

Définir la requête SOAP

Le serveur répond en envoyant un fichier XML, qui est affiché dans le Mode Texte de XMLSpy. En ce qui concerne l'opération que nous avons sélectionnée, nous recevons le fichier XML suivant :

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:m0="http://www.weather.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
   <SOAP-ENV:Body>
      <m:NDFDgenByDay
xmlns:m="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl"
       SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <latitude xsi:type="xsd:decimal">0.0</latitude>
         <longitude xsi:type="xsd:decimal">0.0</longitude>
         <startDate xsi:type="xsd:date">1967-08-13</startDate>
         <numDays xsi:type="xsd:integer">0</numDays>
         <format xsi:type="m0:formatType">String</format>
      </m:NDFDgenByDay>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Nous remplissons les paramètres tels que requis par le XML (*en gras et rouge ci-dessous ; remplir une date de démarrage qui est la date actuelle ou une date dans le courant de la semaine prochaine*) :

```
<SOAP-ENV:Body>
   <m:NDFDgenByDay
xmlns:m="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <latitude xsi:type="xsd:decimal">45</latitude>
         <longitude xsi:type="xsd:decimal">-90</longitude>
         <startDate xsi:type="xsd:date">2019-12-10</startDate>
         <numDays xsi:type="xsd:integer">1</numDays>
         <format xsi:type="m0:formatType">24 hourly</format>
   </m:NDFDgenByDay>
</SOAP-ENV:Body>
```

Cela complète la **définition** de cette requête SOAP. Dans l'étape suivante, nous allons envoyer la requête.

28.15.2 Envoyer la requête au serveur

La définition d'une requête SOAP est décrite dans la rubrique précédente, Créer nouvelle requête SOAP. Une fois que la requête SOAP (un document XML) a été créée, vous devez en faire le document actif. Ensuite, sélectionner cette commende pour envoyer la requête SOAP au serveur SOAP.

Une fois que la requête SOAP a été envoyée, une réponse est reçue de la part du serveur SOAP. Cette réponse est un document XML, qui est affiché dans le Mode Texte de XMLSpy. Par exemple, ci-dessous vous trouverez une capture d'écran du document XML qui a été retournée en réponse à la requête SOAP que nous avons définie dans la section Créer nouvelle requête SOAP.

© 2016-2022 Altova GmbH



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SO
http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:
http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/enc
   <SOAP-ENV:Body>
      <ns1:NDFDgenByDayResponse xmlns:ns1="http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/n
         <dwmlByDayOut xsi:type="xsd:string"><?xml version="1.0"?>
<dwml version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
xsi:noNamespaceSchemaLocation="http://www.nws.noaa.gov/forecasts/xml/DWMLgen/schema/DWML.xsd">
  <head>
  <data>
    <location>
      <location-key>point1</location-key>
      <point latitude="45.00" longitude="-90.00"/>
    </location>
    <moreWeatherInformation applicable-location="point1">
http://forecast.weather.gov/MapClick.php?textField1=45.00&amp;amp;textField2=-90.00
    </moreWeatherInformation>
    <time-layout time-coordinate="local" summarization="24hourly">
    <time-layout time-coordinate="local" summarization="12hourly">
    <parameters applicable-location="point1">
      <temperature type="maximum" units="Fahrenheit" time-layout="k-p24h-n1-1">
        <name>Daily Maximum Temperature</name>
        <value>45</value>
      </temperature>
      <probability-of-precipitation type="12 hour" units="percent" time-layout="k-p12h-n2-2">
        <name>12 Hourly Probability of Precipitation</name>
        <value>14</value>
        <value xsi:nil="true"/>
      </probability-of-precipitation>
    </parameters>
  </data>
</dwml>
</dwmlByDayOut>
      </ns1:NDFDgenByDayResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Enregistrer et réutiliser une requête SOAP

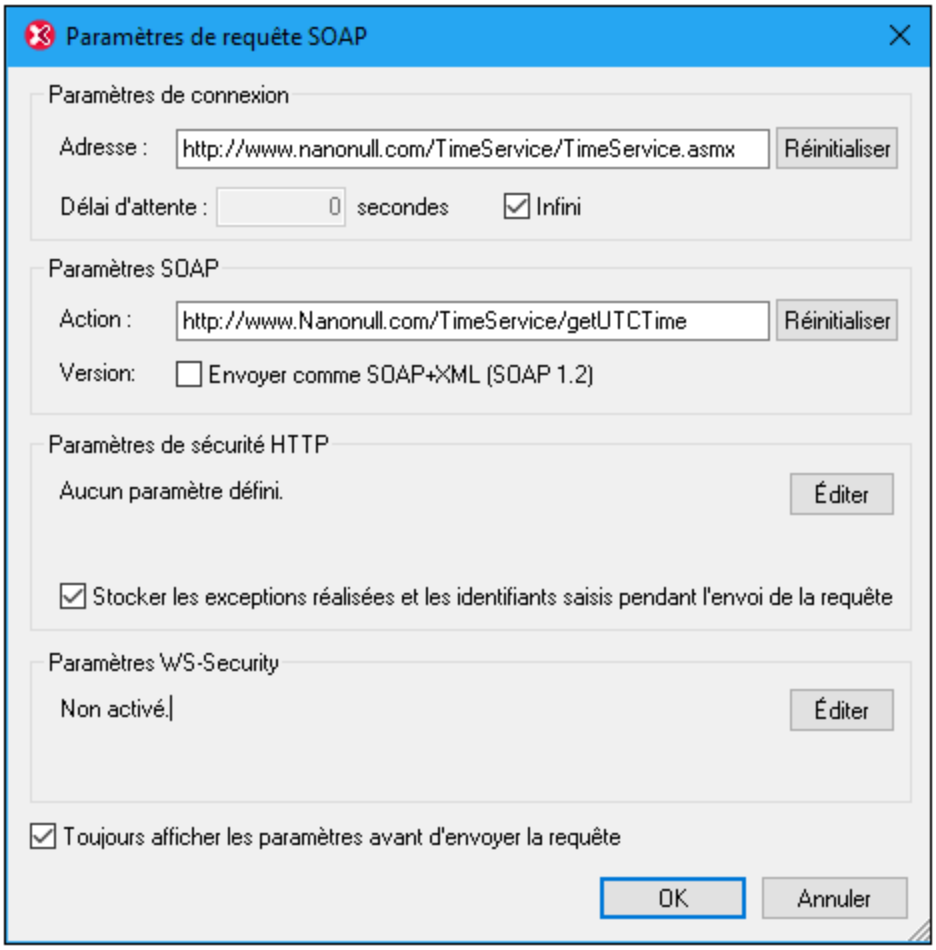
XMLSpy vous permet d'enregistrer une requête SOAP et de la renvoyer ultérieurement. Procéder comme suit :

1. Enregistrer le document XML de requête SOAP (**Fichier | Enregistrer sous**).
2. Fermer le fichier de requête SOAP.
3. Rouvrir le document XML de requête SOAP et sélectionner l'option de menu **SOAP | Envoyer requête au serveur**. (Tout fichier XML peut être utilisé en tant que document de requête SOAP.)

28.15.3 Paramètres de requête SOAP

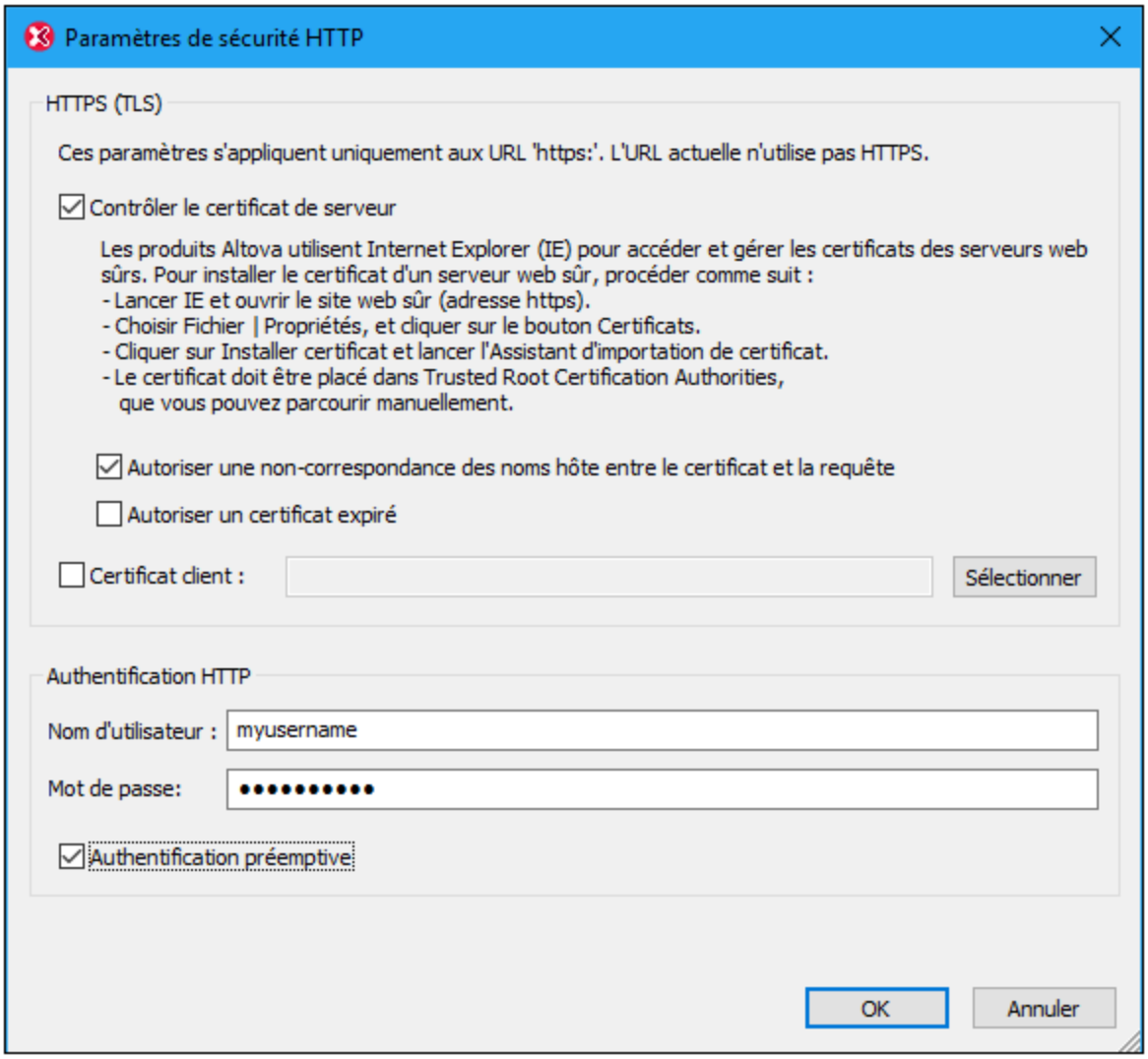
Cette commande affiche le dialogue de Paramètres de requête SOAP (*capture d'écran ci-dessous*), dans lequel vous pouvez spécifier des paramètres différents de la requête SOAP. Ces paramètres sont décrits ci-dessous.

1. Rendre le document de requête SOAP actif.
2. Sélectionner l'option de menu **Soap | Paramètres requête SOAP**. Cela ouvre le dialogue Paramètres de requête SOAP (*capture d'écran ci-dessous*).



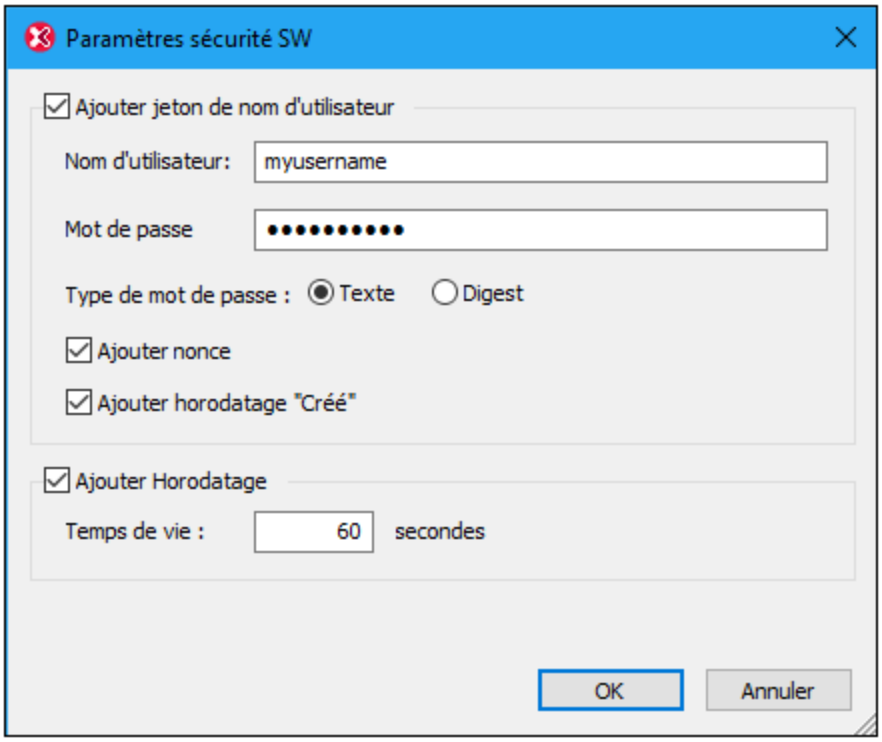
3. Dans le champ *Adresse*, saisir le point d'arrêt de connexion désiré. Si la requête SOAP a été créée depuis un fichier WSDL dans XMLSpy, la valeur du champ *Adresse* sera l'emplacement du point d'arrêt sélectionné dans le WSDL. Cliquer sur **Réinitialiser** pour obtenir ce point d'arrêt. Une valeur de délai d'attente de connexion peut être spécifiée en secondes. Pour ne définir aucune valeur de délai d'attente, cocher la case *Infini*.
4. Dans le champ *Action*, saisir l'action SOAP à effectuer. Pour envoyer la requête en tant que SOAP 1.2, cocher la case *SOAP+XML (SOAP 1.2)*. Si la requête SOAP a été créée depuis un fichier WSDL dans XMLSpy, alors l'action SOAP est reçue depuis l'élément extensible sous l'opération de liaison SOAP correspondante dans le fichier WSDL. Dans ce cas, la version SOAP est aussi présélectionnée depuis le WSDL. (La version SOAP touche la valeur de l'en-tête HTTP Content-Type: `text/xml` ou `application/soap+xml`.) Cliquer **Réinitialiser** pour obtenir l'action SOAP depuis le fichier WSDL.
5. Le volet Paramètres de sécurité HTTP fournit un sommaire des listes de paramètres de sécurité. Si l'option *Stocker les exceptions pendant l'envoi de la requête*, tous les paramètres sont enregistrés lorsque la requête est envoyée et peut être réutilisée pour la prochaine requête. Cliquer sur la touche **Éditer** pour afficher le dialogue Paramètres de sécurité HTTP (*capture d'écran ci-dessous*). Vous trouverez une description pour installer les certificats de serveur ci-dessous.

© 2016-2022 Altova GmbH



Si vous souhaitez permettre une ***non-correspondance du nom d'hôte*** (entre le nom d'hôte dans le certificat de serveur et l'adresse que vous utilisez réellement) ou un ***certificat de serveur expiré***, alors cocher ces options dans le dialogue. Si le serveur nécessite un ***certificat client***, vous pouvez spécifier l'emplacement du certificat de client. Si l'authentification est exigée par le serveur, spécifier le ***nom d'utilisateur*** et le ***mot de passe*** pour l'authentification standard. Lorsque la requête de client initiale vers le serveur contient les informations d'authentification requises, ce processus est qualifié d'***authentification préemptive***. Si exigé par le serveur, sélectionner l'option *Authentification préemptive*. Sinon, ne pas sélectionner l'option *Authentification préemptive*.

6. Outre l'ajout de sécurité au niveau du transport (paramètres de sécurité HTTP), vous pouvez également spécifier les paramètres de sécurité de service si ceux-ci sont exigés par le service web. Cliquer sur la touche **Éditer** dans le volet de Paramètres de sécurité SW pour afficher le dialogue Paramètres de sécurité SW (*capture d'écran ci-dessous*). Les informations de sécurité comprennent le nom d'utilisateur, le mot de passe, le type de mot de passe, un string de code nonce généré automatiquement, et un horodatage. Vous pouvez aussi spécifier la période de validité de l'information de sécurité (*Ajouter horodatage*). Le dialogue crée un fragment XML qui contient les informations de sécurité et intègre ce fragment dans la requête SOAP. *Voir la liste ci-dessous*.



7. Une fois terminé, cliquer sur **OK**.

À propos des certificats autorisés

Les produits Altova utilisent Internet Explorer (IE) pour accéder et gérer les certificats autorisés des serveurs web sûrs. L'installation du certificat d'un serveur web dans IE permet à celui-ci d'accéder au serveur web sans émettre un avertissement et sans abandonner le processus. Pour installer le certificat d'un serveur web sûr, procéder comme suit :

- Dans Internet Explorer 8, ouvrir le site web sûr.
- Choisir **Fichier | Propriétés**, et cliquer sur la touche Certificats.
- Cliquer sur **Installer le certificat** et lancer l'assistant Importer certificat. (Cet assistant peut aussi être accédé depuis **Outils | Options Internet | Contenu | Certificats | Importer**.)
- Le certificat doit être placé dans le stock Autorités de certification approuvées que vous pouvez parcourir manuellement.
- Terminer les étapes de l'assistant, fermer les dialogues Certificats et Propriétés respectivement en cliquant sur **OK**. Vous devrez éventuellement redémarrer Internet Explorer.

Note : Ne procédez à des modifications des paramètres d'action SOAP que si vous pouvez accéder à toutes les méthodes SOAP et à leurs actions SOAP correspondantes.

Information de sécurité de service Web

Certains services web exigent une authentification d'utilisateurs. (Le niveau de sécurité du service web se trouve en plus du niveau de sécurité HTTP mis en place par le serveur.) L'information d'authentification du service web est stocké dans la requête SOAP en tant qu'un fragment XML présentant une structure comme celle listée ci-dessous. Ce fragment XML est généré automatiquement dans la requête SOAP à partir de l'information d'authentification du service web que vous saisissez dans le dialogue Paramètres de sécurité SW.

© 2016-2022 Altova GmbH

```
<wsse:Security xmlns:wsse="…" xmlns:wsu="…" SOAP-ENV:mustUnderstand="true">
   <wsse:UsernameToken>
      <wsse:Username>usr</wsse:Username>
      <wsse:Password Type="...#PasswordText">pwd</wsse:Password>
      <wsse:Nonce EncodingType="…#Base64Binary">UqrtD963797WBRgWiJPu2w==</wsse:Nonce>
      <wsu:Created>2014-11-17T16:08:07.016Z</wsu:Created>
   </wsse:UsernameToken>
   <wsu:Timestamp>
      <wsu:Created>2014-11-17T16:08:07.016Z</wsu:Created>
      <wsu:Expires>2014-11-17T16:09:07.016Z</wsu:Expires>
   </wsu:Timestamp>
</wsse:Security>
```

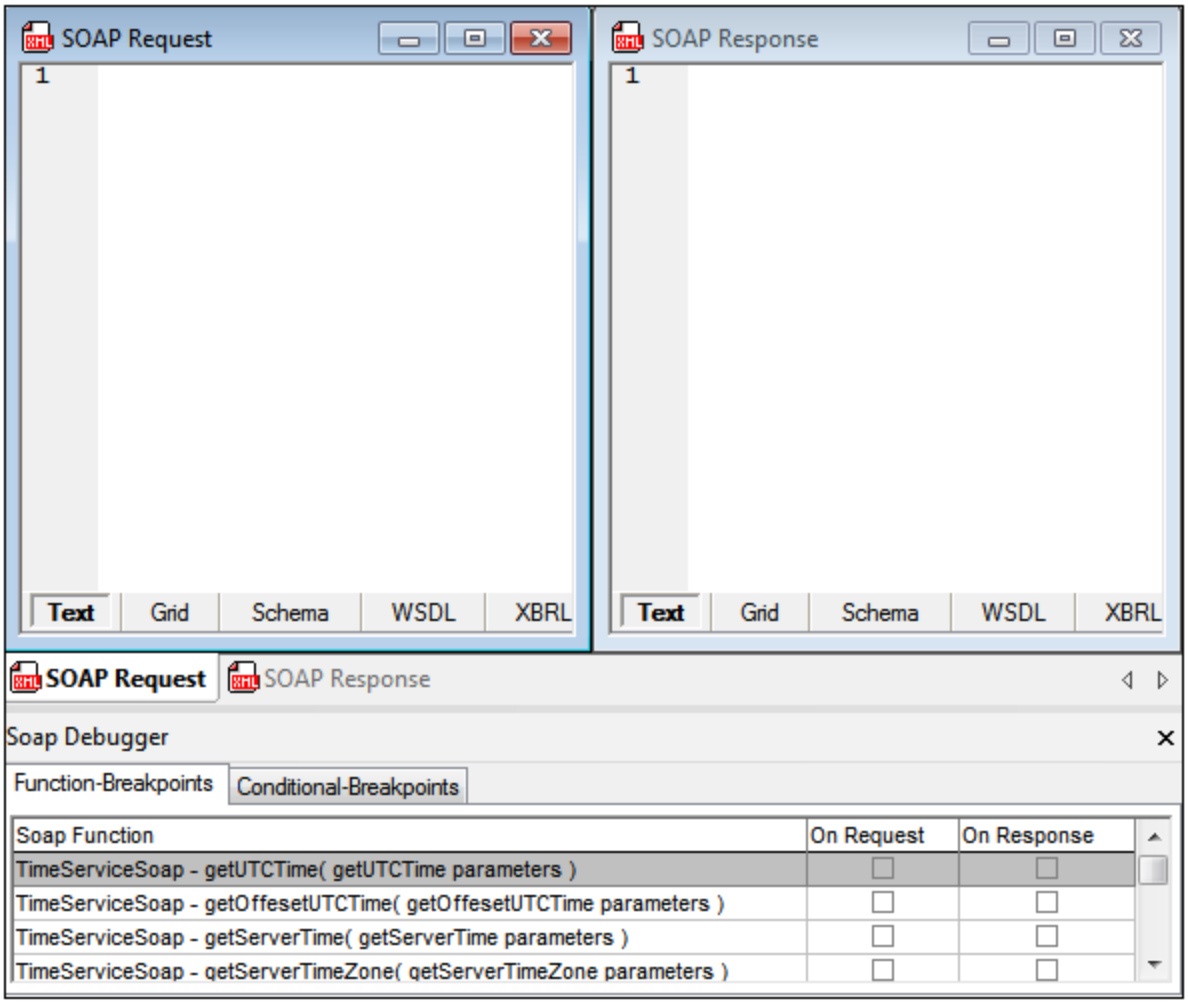
28.15.4 Session de débogage Soap

Cette commande lance la session SOAP de débogage.

- Une boîte de dialogue est immédiatement ouverte après avoir sélectionné cette commande. Puis, vous devez sélectionner un emplacement de fichier WDSL, généralement une URL. Vous pouvez aussi sélectionner un fichier par le biais d'une ressource globale (cliquez sur l'icône Ressource globale et naviguez) ou un fichier dans une des fenêtres ouvertes dans XMLSpy.
- Sélectionnez les ports source et cible dont vous avez besoin pour le serveur proxy de débogage et le service web dans la boîte de dialogue suivante.

  Ceci ouvre le serveur proxy de débogage SOAP dans son état inactif. Cliquer sur l'une des icônes de la barre d'outils SOAP lance le débogage SOAP et attend les requêtes client.

  Veuillez voir la section Soap - How to pour une description plus détaillée.



28.15.5 Aller à



Cette commande active le serveur proxy SOAP et traite le fichier WSDL jusqu'à ce qu'il rencontre un point d'arrêt. Le document SOAP restrictif apparaît ensuite dans une des fenêtres de document SOAP.

© 2016-2022 Altova GmbH

28.15.6 Pas à pas



Cette commande vous permet de passer pas à pas dans les requêtes et les réponses SOAP entrantes et sortantes. Le débogueur SOAP s'arrête à chaque requête et réponse. Le serveur proxy est aussi démarré s'il était inactif.

28.15.7 Arrêter à la prochaine requête



Cette commande entraîne l'arrêt de la requête SOAP suivante et affiche les données dans la fenêtre de requête de document SOAP. Vous pouvez éditer les données directement dans cette fenêtre avant de l'envoyer vers le service web.

28.15.8 Arrêter à la prochaine réponse



Cette commande entraîne l'arrêt du débogueur à la réponse SOAP suivante et affiche les données dans la fenêtre de requête de document SOAP. Vous pouvez éditer les données directement dans cette fenêtre avant de l'envoyer au client.

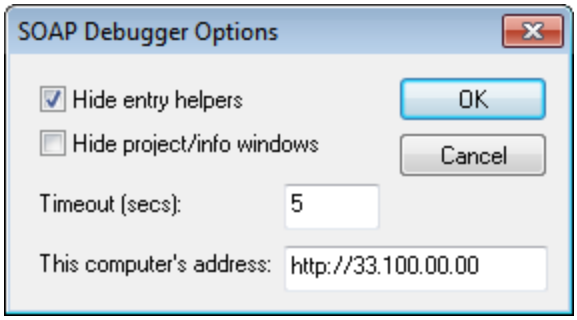
28.15.9 Arrêter le serveur proxy



Cette commande arrête le serveur proxy de débogage.

28.15.10 Options de débogage SOAP

Le dialogue Débogage SOAP (*capture d'écran ci-dessous*) vous permet de spécifier l'adresse IP de l'ordinateur et les autres options de débogage qui sont listées ci-dessous. Accéder à ce dialogue avec la commande de menu **SOAP | Options de débogage SOAP**.



- *Adresse de cet ordinateur :* l'adresse du serveur proxy depuis laquelle le débogage est effectué. Le débogueur sur le serveur proxy prend les requêtes depuis les machines sur le réseau et les envoie au service web. Puisque le débogueur est exécuté dans XMLSpy, la machine sur laquelle XMLSpy est installé sert aussi en tant que serveur proxy. L'adresse IP de la machine est détectée automatiquement et saisie dans ce champ. Seule l'adresse IP ne peut pas être détectée automatiquement, si vous nécessitez de saisir l'adresse IP (en tant qu'adresse `http`) dans ce champ. Pour connaître l'adresse IP de votre ordinateur, ouvrez une invite de commande, saisir la commande `ipconfig /all`, et appuyer sur **Entrée**.
- *Délai d'expiration :* Cette valeur est le temps pendant lequel le déboguer SOAP demeure dans un point d'arrêt. Le défaut est de 5 secondes.
- *Masquer assistants à la saisie ; Masquer fenêtres projet/info :* ces options sont utiles pour offrir un espace d'écran plus important à la fenêtre de débogage SOAP.

© 2016-2022 Altova GmbH

28.16 Menu XBRL

Le menu **XBRL** (*voir la capture d'écran ci-dessous*) contient des commandes qui sont activées lorsqu'une taxonomie est éditée dans le Mode XBRL. Ces commandes sont recensées ci-dessous et sont décrites en détail dans les sous-sections de cette section :

- Rôles d'arc : définit des rôles d'arc
- Rôles de lien : définit des rôles de lien
- Préfixes d'espace de nom : gère les espaces de noms de la taxonomie
- Définir espace de nom cible : définit et déclare l'espace de nom cible de la taxonomie
- Valeurs de paramètre : affiche les formules et les paramètres de table dans un dialogue, où ils peuvent être édités
- Importer/Référence : importe une taxonomie XBRL ou référence un base de lien
- Touver des composants de formule par ID : trouve des composants de formule sur la base de l'ID fournie par l'utilisateur
- Générer une documentation : génère une documentation de la taxonomie XBRL actuelle
- Paramètres du Mode : définit un défaut pour le Mode XBRL
- Générer XBRL depuis BD, Excel, CSV avec MapForce : lance Altova MapForce pour générer un fichier d'instance XBRL
- Présenter XBRL en tant que HTML/PDF/Word avec StyleVision : lance Altova StyleVision pour concevoir un rapport XBRL
- Exécuter Formule : exécute des formules et/ou des assertions depuis le DTS associé avec le document d'instance XBRL actif
- Générer Table : génère des tables XBRL depuis une instance XBRL
- Détecter Duplicatas : détecte les faits dupliqués dans les instances XBRL
- Exécuter XULE : exécute un document XULE dans un document d'instance XBRL
- Transformer Inline XBRL : génère la partie inline XBRL d'un document XHTML en tant que XBRL
- Valider EDGAR sur Serveur : valide des instances XBRL en utilisant EDGAR sur RaptorXML+XBRL Server
- Processing Options : active les paramètres des options pour le traitement des instances XBRL

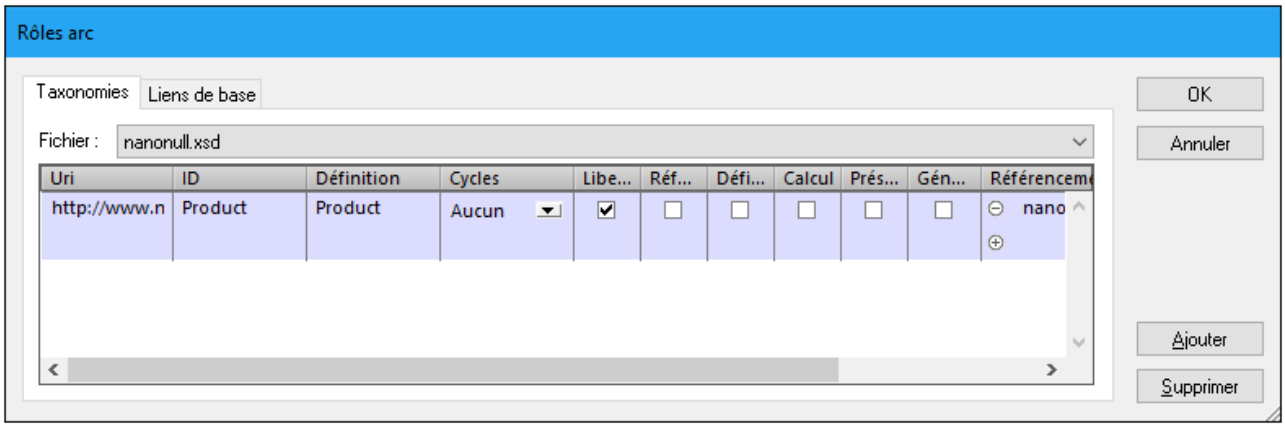
Pour plus d'informations concernant XBRL, voir les sections XBRL et Modes d'édition | Mode XBRL.

Pour la conversion en formats OIM xBRL, voir Menu Conversion.

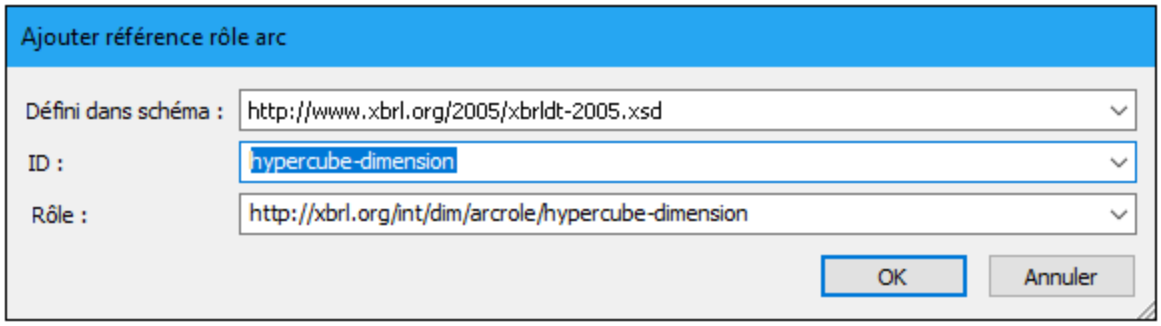
28.16.1 Rôles d'arc

La commande **Rôles d'arc** permet d'ouvrir le dialogue Rôles d'arc (*capture d'écran ci-dessous*) dans lequel des rôles d'arc peuvent être créés pour une taxonomie. Les rôles d'arc sont stockés dans le fichier de définitions de concept, dans l'élément `appinfo`. Ils spécifient le rôle d'un arc.

Dans l'onglet **Taxonomies** du dialogue Rôles d'arc (*capture d'écran ci-dessous*), seules des taxonomies qui sont éditables ou qui contiennent un rôle d'arc ou un rôle de lien sont listées dans la liste de choix. Vous pouvez ajouter un rôle d'arc à une taxonomie en cliquant sur la touche **Ajouter**. Ensuite, vous pourrez définir l'URI, l'D, la définition et les cycles du rôle d'arc. Pour spécifier dans quels types de relations le rôle d'arc doit être disponible, cochez les cases des types de relation requis. Les bases de lien qui référencent un rôle d'arc peuvent être ajoutés à la colonne Référencement des fichiers de base de lien.



L'onglet **Bases de lien** propose un autre mode d'affichage des rôles d'arc de la taxonomie. Dans ce mode, vous ajoutez et consultez des rôles d'arc conformément aux bases de lien individuelles (par exemple, le calcul ou la présentation de bases de lien). Sélectionner une base de lien dans la liste de choix et ajoutez ou supprimez un rôle d'arc selon vos besoins. Lorsque vous cliquez sur la touche **Ajouter**, le dialogue Ajouter référence de rôle d'arc (*capture d'écran ci-dessous*) s'ouvre.

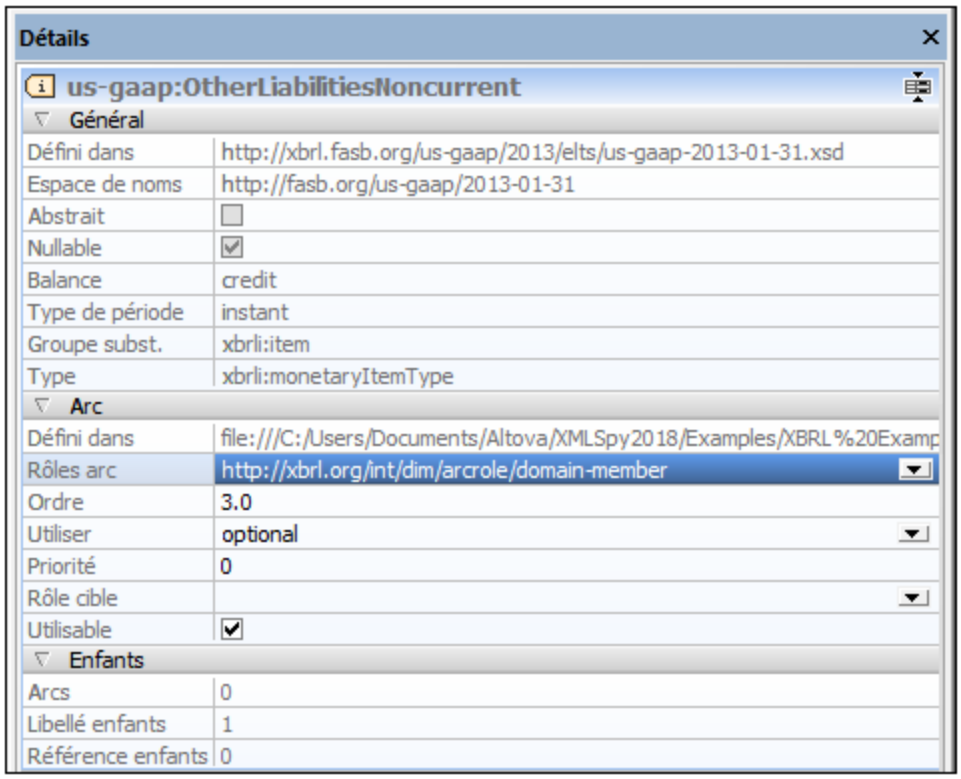


Chacune des entrées dans ce dialogue est une liste de choix qui vous permet d'effectuer des choix depuis des options disponibles. Le champ *Défini dans le Schéma* vous permet de sélectionner la taxonomie dans laquelle le rôle d'arc est défini. Les listes de choix *ID* et *Rôle* fournissent les rôles d'arc disponibles. Une fois avoir sélectionné un rôle d'arc et avoir cliqué sur **OK**, la référence est ajoutée à la base de lien. Dans l'onglet Taxonomies, le rôle d'arc que vous avez référencé affichera la base de lien de référencement dans la colonne Référencement des fichiers de base de lien. Si vous souhaitez rendre ce rôle d'arc disponible pour un type de relation particulier, vous devrez néanmoins cocher la case de type de relation approprié.

Une fois qu'un rôle d'arc a été créé dans la taxonomie, il peut être utilisé pour spécifier le rôle d'un arc dans un type de relation pour lequel l'arc est disponible conformément à sa définition. Dans la capture d'écran ci-dessus, par exemple, le rôle d'arc a été rendu disponible pour les arcs dans des relations de libellé.

Le rôle d'arc d'un arc est sélectionné dans l'assistant à la saisie Détails (*capture d'écran ci-dessous ; rôle d'arc marqué*).

 © 2016-2022 Altova GmbH



L'élément a été sélectionné à la fin d'un arc, dans l'Assistant à la saisie Détails, sélectionner l'item requis depuis la liste déroulante de l'entrée Arc role.

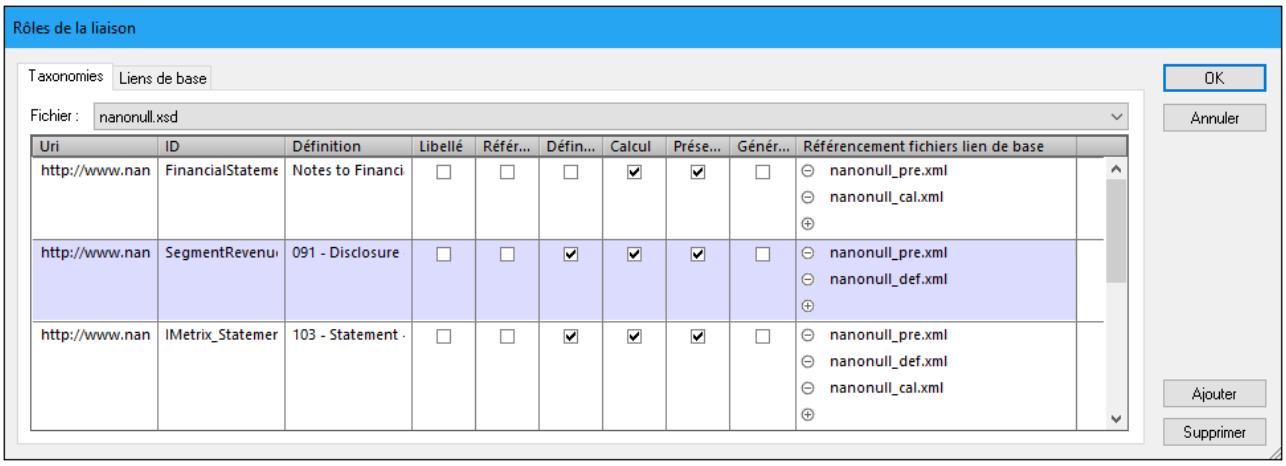
28.16.2 Rôles de lien

La commande **Rôles de lien** permet d'ouvrir le dialogue Rôles de lien (*capture d'écran ci-dessous*) dans lequel des rôles de lien peuvent être créés pour une taxonomie. Les rôles de lien sont stockés dans le fichier de définitions de concept, dans l'élément `appinfo` (*voir liste ci-dessous*). Les rôles de lien sont utilisés non seulement dans les éléments `definitionLink` mais aussi dans les éléments contenants d'autres types de relation (par exemple, dans les éléments `calculationLink` et `presentationLink`).

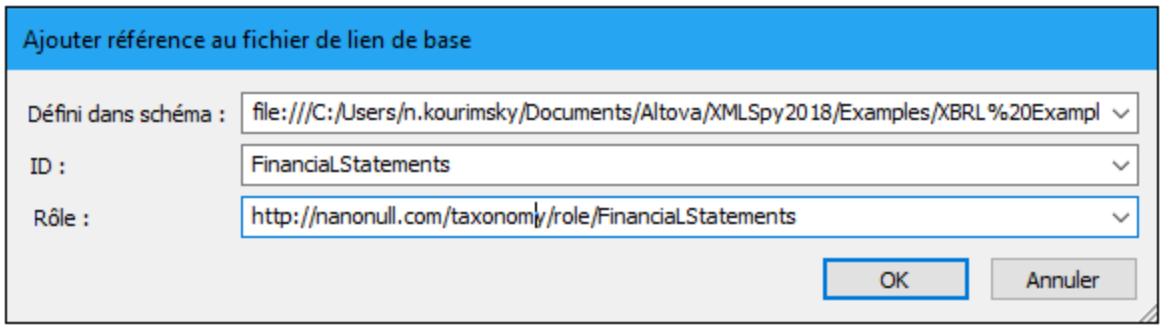
```
<xs:appinfo>
   <link:roleType id="SegmentRevenueAndOperatingIncome"
   roleURI="http://www.nanonull.com/taxonomy/role/SegmentRevenueAndOperatingIncome">
      <link:definition>006091 - Disclosure - Segment Revenue and Operating
Income</link:definition>
      <link:usedOn>link:calculationLink</link:usedOn>
      <link:usedOn>link:definitionLink</link:usedOn>
      <link:usedOn>link:presentationLink</link:usedOn>
   </link:roleType>
</xs:appinfo>
```

Dans la liste ci-dessus, veuillez noter qu'il y a des éléments `usedOn` qui spécifient dans quel type de relation ce rôle de lien peut être utilisé.

Dans l'onglet **Taxonomies** du dialogue Rôles de lien (*capture d'écran ci-dessous*), vous pouvez ajouter un rôle de lien à une taxonomie en cliquant sur la touche **Ajouter**. Ensuite, vous pourrez définir l'URI, l'ID du rôle de lien (*se référer à la liste ci-dessus*). Pour spécifier dans quels types de relations un rôle de lien doit être disponible, cochez les cases des types de relation requis. Dans la colonne Référencement des fichiers de base de lien, pour chaque rôle de lien, vous pouvez ajouter ou supprimer les fichiers de base de lien qui référencent le rôle de lien.



L'onglet **Bases de lien** propose un autre mode d'affichage des rôles de lien de la taxonomie. Dans ce mode, vous ajoutez et consultez des rôles de lien conformément aux bases de lien individuelles (par exemple, bases de lien de calcul ou de présentation). Sélectionner une base de lien dans la liste de choix et ajoutez ou supprimez un rôle de lien selon vos besoins. Par exemple, vous pouvez ajouter un rôle de lien dans la base de lien du calcul. Lorsque vous cliquez sur la touche **Ajouter**, le dialogue Ajouter référence de rôle de lien (*capture d'écran ci-dessous*) s'ouvre.



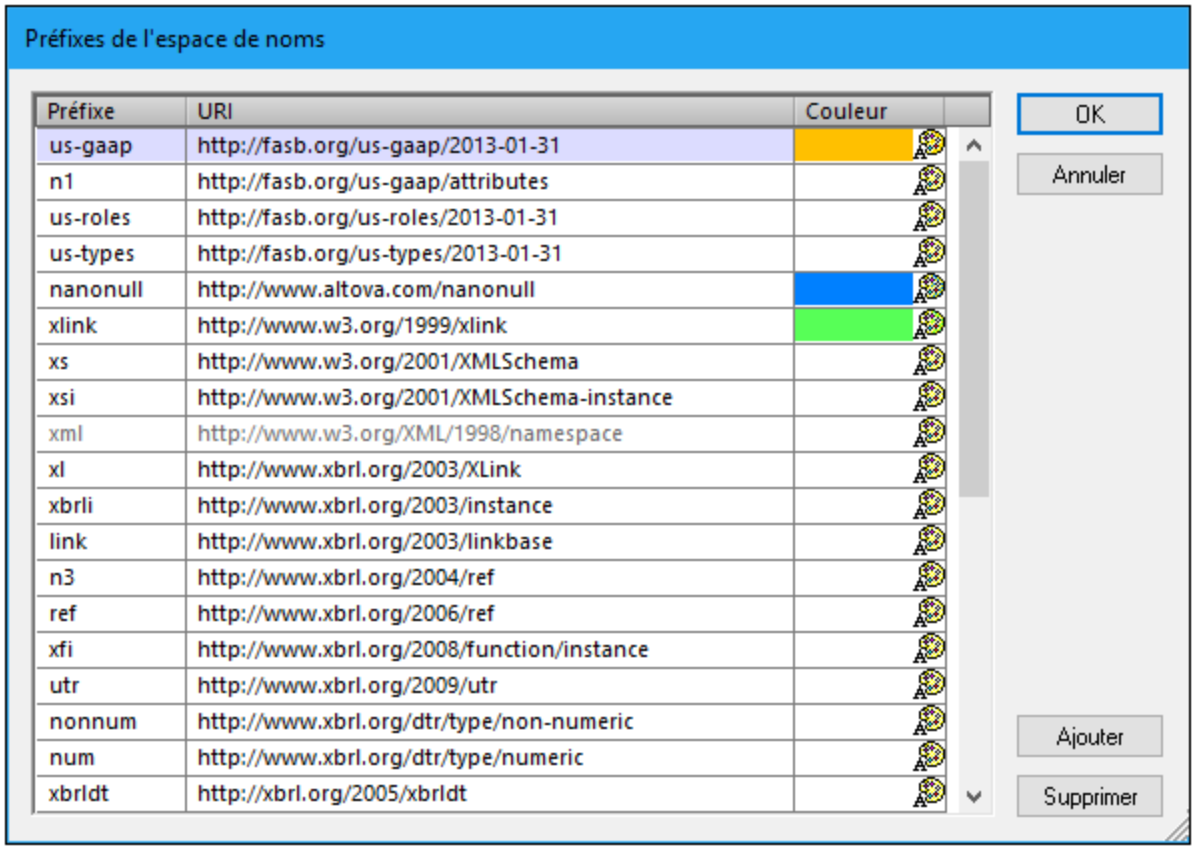
Chacune des entrées dans ce dialogue est une liste de choix qui vous permet d'effectuer des choix depuis des options disponibles. Le champ *Défini dans le Schéma* vous permet de sélectionner la taxonomie dans laquelle le rôle de lien est défini. Les listes de choix *ID* et *Rôle* fournissent les rôles de lien disponibles. Une fois avoir sélectionné un rôle de lien et avoir cliqué sur **OK**, la référence est ajoutée à la base de lien. Dans l'onglet Taxonomies, le rôle de lien que vous avez référencé affichera la base de lien de référencement dans la colonne Référencement des fichiers de base de lien. Si vous souhaitez rendre ce rôle de lien disponible pour un type de relation particulier, vous devrez néanmoins cocher la case de type de relation appropriée.

Une fois qu'un rôle de lien a été créé dans la taxonomie, il peut être utilisé pour créer des relations.

© 2016-2022 Altova GmbH

28.16.3 Préfixes d'espace de noms

La commande **Préfixes d'espace de noms** permet d'ouvrir le dialogue Préfixes d'espace de noms (*capture d'écran ci-dessous)*, qui affiche tous les espaces de noms dans la taxonomie, y compris ceux des taxonomies importées. Dans le dialogue Préfixes d'espace de noms, vous pouvez éditer des espaces de noms et des préfixes et définir des couleurs d'arrière-plan pour des espaces de noms individuels. Lorsqu'une couleur d'arrière-plan est définie pour un espace de noms, les éléments dans cet espace de noms dans la Fenêtre principale et les assistants à la saisie apparaissent avec cette couleur d'arrière-plan. Veuillez noter qu'un réglage de couleur pour un espace de noms donné s'applique pour cet espace de noms à travers tous les documents de taxonomie ouverts dans le Mode XBRL.

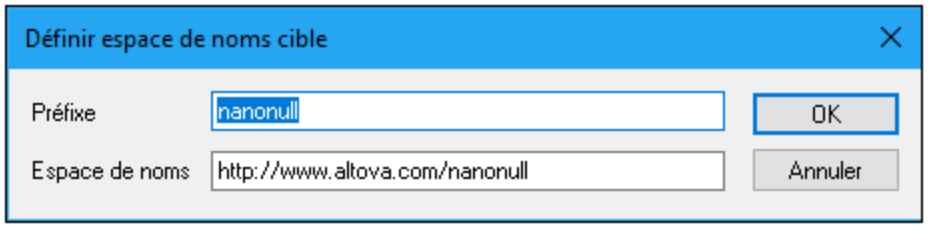


Pour ajouter ou supprimer un espace de noms, utiliser les touches **Ajouter** ou **Supprimer**, respectivement. Une couleur est attribuée à un espace de noms via la palette de couleurs pour cet espace de noms. Une fois que vous avez terminé l'édition dans le dialogue Espace de noms, cliquer sur **OK** pour terminer.

L'**espace de noms cible** de la taxonomie est aussi listé dans l'Espace de noms. L'espace de noms cible, néanmoins, ne doit pas être modifié dans ce dialogue, mais avec la commande **Définir Espace de noms cible**. Pour plus d'informations concernant les espaces de noms cibles, voir la section XBRL de la documentation.

28.16.4 Définir l'espace de noms cible

La commande **Définir espace de noms cible** permet de définir un espace de noms cible pour un document de taxonomie. Cliquer sur la commande pour ouvrir le dialogue Définir espace de noms cible (*capture d'écran ci-dessous*). Vous pourrez y saisir l'espace de noms cible souhaité et le compléter d'un préfixe. Cliquer sur **OK** pour terminer.



L'espace de noms cible sera défini et sera également déclaré :

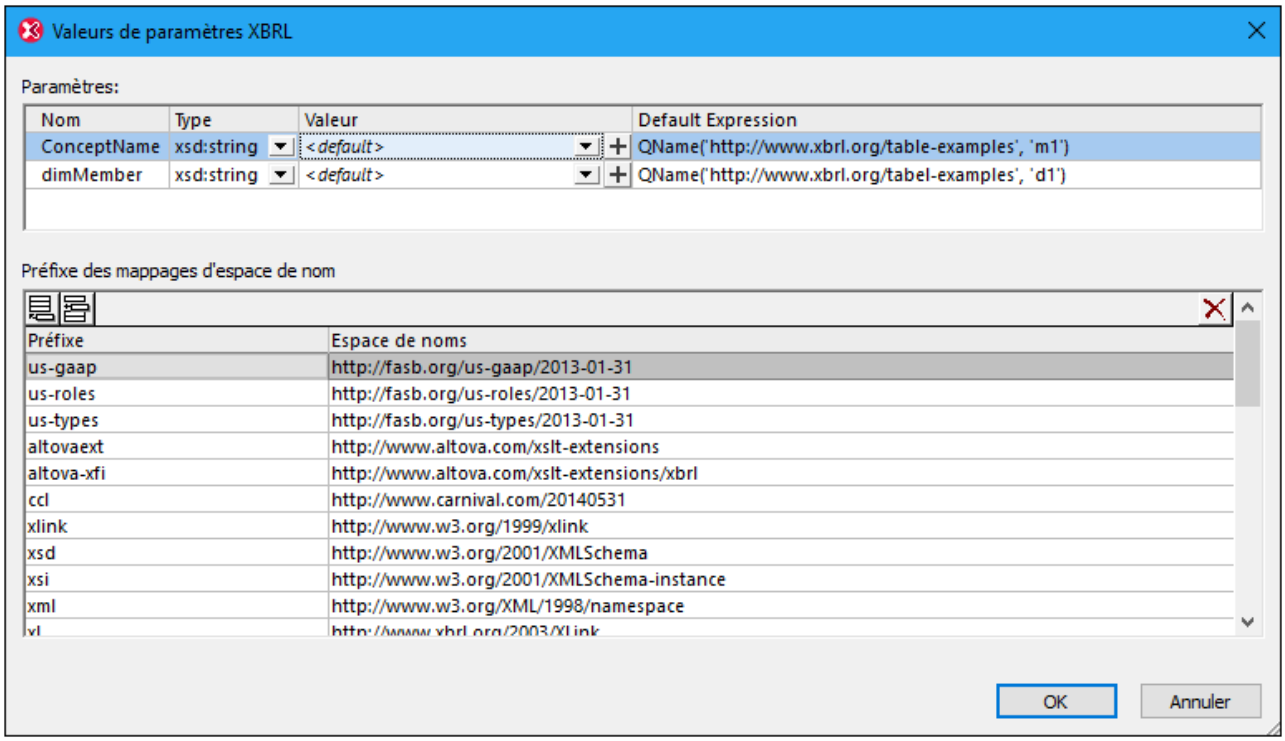
```
<xs:schema targetNamespace="http://www.altova.com/XBRL/Taxonomies"
      xmlns:ns1="http://www.altova.com/XBRL/Taxonomies" >
       ...
</xs:schema>
```

Dans la liste ci-dessus, l'espace de noms cible est défini avec l'attribut `targetNamespace` et il est ensuite déclaré avec un préfixe de `ns1`.

28.16.5 Valeurs de paramètres

La commande **Valeurs de paramètres** affiche le dialogue Valeurs de paramètres XBRL (*capture d'écran ci-dessous*). Il affiche des paramètres définis dans l'onglet Formule (paramètres de formule) et l'onglet Table (paramètres de table). Dans le dialogue Valeurs de paramètre XBRL, vous pouvez éditer le type de données du paramètre et offrir une valeur de paramètre qui contourne la valeur par défaut. La valeur de paramètres que vous saisissez contournera la valeur par défaut que vous avez saisie par le biais du diagramme. Puisque les paramètres de table peuvent prendre plusieurs valeurs, vous pouvez ajouter des valeurs de paramètres supplémentaires pour un paramètre en cliquant sur l'icône **+** dans la colonne *Valeur*.

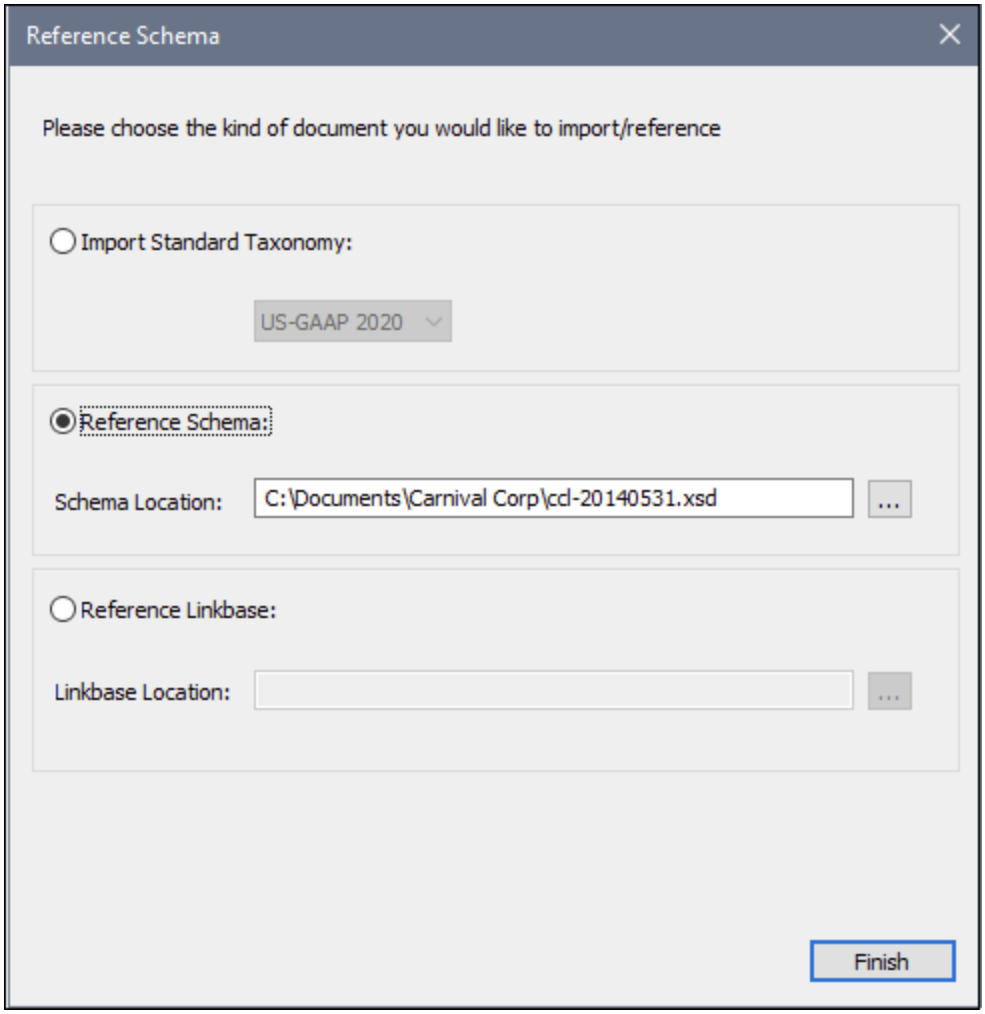
© 2016-2022 Altova GmbH



Les valeurs des paramètres globaux telles qu'attribuées dans ce dialogue sont évaluées uniquement pour les paramètres de table. Les paramètres de Formule, bien qu'affichés, ne sont pas éditables dans ce dialogue.

28.16.6 Importer/Référence

La commande **Importer/Référence** ouvre un dialogue (*capture d'écran ci-dessous*) dans lequel vous pouvez spécifier le schéma à importer ou la base de lien à référencer.



Le dialogue fournit les options de bouton radio suivantes :

- *Importer une taxonomie standard :* cette option vous permet de corriger rapidement et correctement une taxonomie US-GAAP ou une taxonomie IFRS. Sélectionner la taxonomie standard requise depuis le menu déroulant de la liste de choix et cliquer sur Suivant. Le dialogue suivant apparaît qui vous permettra de compléter votre spécification de la taxonomie requise. Le processus est décrit dans la section, Créer une nouvelle taxonomie.
- *Importer toute taxonomie (Schéma de référence):* cette option vous permet d'importer n'importe quelle taxonomie en spécifiant l'emplacement du fichier de taxonomie (fichier `.xsd`).
- *Référencer une base de lien :* une base de lien peut être spécifiée pour être incluse dans la taxonomie. Pour ce faire, spécifier l'emplacement du fichier de base de lien et cliquer sur **Terminer**. Une référence au fichier de base de lien est créée dans la taxonomie. Le type de relation de la base de lien récemment référencée peut ensuite être spécifié en cliquant avec la touche de droite sur le nom du fichier et en choisissant la commande **Définir genre de base de lien**.

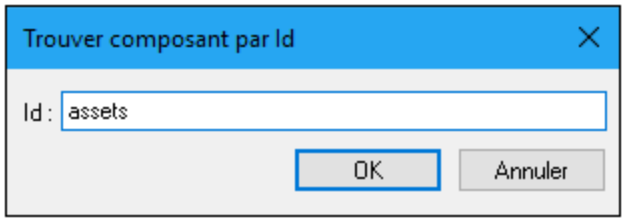
```
<xsl:template match="*">
   <xsl:copy>
      <xsl:copy-of select="@*[not(.='')]"/>
      <xsl:apply-templates/>
```

 © 2016-2022 Altova GmbH

```
    </xsl:copy>
</xsl:template>
```

28.16.7 Trouver le composant par ID

Dans des taxonomies avec des formules volumineuses ou des bases de lien de table contenant plusieurs composants du même type, il peut être utile de chercher un composant par le biais de son ID. La commande de menu **XBRL | Trouver composant par Id** permet une recherche par ID. En cliquant sur la commande, un dialogue s'affiche vous invitant à saisir l'ID (*capture d'écran ci-dessous*).



Cliquer sur **OK** pour lancer la recherche.

28.16.8 Générer documentation

La commande **XBRL | Générer Documentation** génère une documentation détaillée de la taxonomie XBRL actuelle. Vous pouvez obtenir une sortie de la documentation sous la forme d'un fichier HTML, MS Word, RTF ou PDF. La documentation générée par cette commande peut être modifiée et utilisée librement ; il n'est pas nécessaire d'obtenir la permission de la part d'Altova. La documentation est générée pour les composants que vous sélectionnez dans le dialogue Documentation Taxonomie XBRL (qui apparaît lorsque vous sélectionnez la commande Générer Documentation). Les composants liés présentent un hyperlien dans la sortie de l'écran, vous permettant de naviguer d'un composant à un autre. Les différentes options de génération de documentation sont décrites dans la section, Options de Documentation .

Note : Afin de générer une documentation dans un format MS Word, vous devez installer MS Word (version 2000 ou plus).

Vous pouvez soit utiliser le design standard fixe de XMLSpy pour le document généré, soit utiliser une SPS StyleVision pour le design. Le fait d'utiliser une SPS StyleVision vous permet de personnaliser le design de la documentation générée ainsi que de générer un PDF en tant qu'un format de sortie supplémentaire. Voir la section Design défini par l'utilisateur pour découvrir comment travailler avec une SPS.

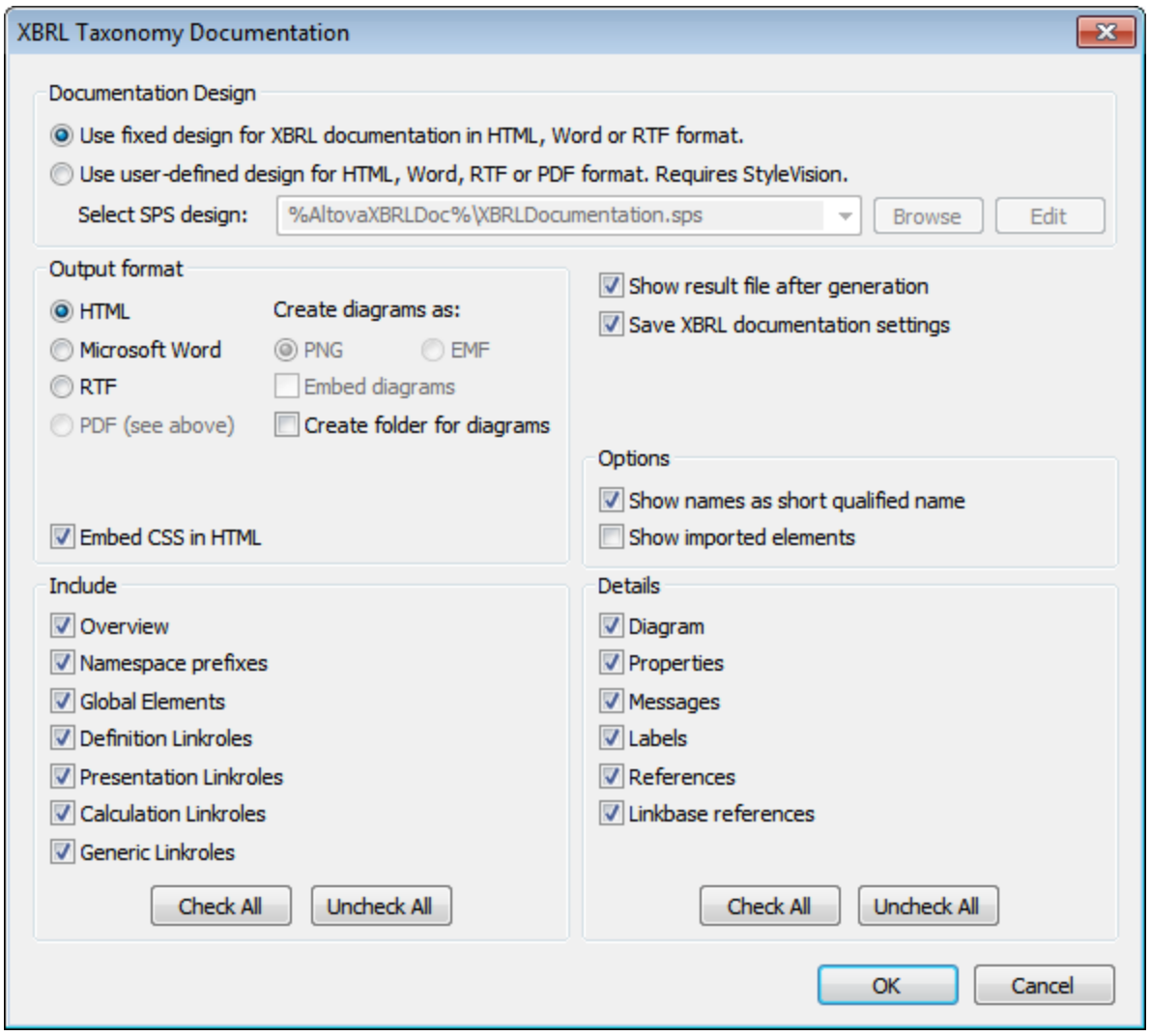
Note : Afin d'utiliser une SPS pour générer une documentation de schéma, vous devez avoir installé StyleVision sur votre appareil.

28.16.8.1 Options de documentation

La commande **XBRL | Générer Documentation** ouvre le dialogue Documentation de taxonomie XBRL (*capture d'écran ci-dessous*), dans lequel vous pouvez choisir des options pour la documentation.

Dans le panneau Design de Documentation du dialogue, vous pouvez choisir si vous souhaitez utiliser un design XMLSpy standard fixe pour la documentation générée ou si vous souhaitez utiliser un design personnalisé créé dans une SPS StyleVision. Choisir l'option que vous souhaitez. Veuillez noter que la sortie PDF est disponible uniquement pour une documentation générée avec une SPS StyleVision, et non pas pour une documentation générée à l'aide d'un design fixe. Voir la section Design défini par l'utilisateur pour découvrir comment travailler avec une SPS.

Note : Afin d'utiliser une SPS pour générer une documentation de schéma, vous devez avoir installé StyleVision sur votre appareil.



Cliquer sur la commande **Générer Documentation** pour ouvrir le dialogue Documentation Taxonomy XBRL :

- Le format requis est spécifié dans le panneau de format de sortie : soit HTML, Microsoft Word, RTF ou PDF. (Le format de sortie PDF est uniquement disponible si vous utilisez une SPS StyleVision pour générer la documentation.) En cliquant sur **OK**, vous serez invité à saisir le nom du fichier de sortie et l'emplacement où il a dû être enregistré.

© 2016-2022 Altova GmbH

- Les documents Microsoft Word sont créés avec l'extension de fichier `.doc` lorsqu'ils sont générés avec un design fixe et avec une extension de fichier `.docx` lorsqu'ils sont générés avec une SPS StyleVision. Afin de générer une documentation en format MS Word, vous devez avoir installé MS Word (version 2000 ou plus).
- En ce qui concerne la sortie HTML, les définitions de style CSS peuvent être sauvegardées soit dans un fichier CSS séparé soit intégrées dans le fichier HTML (dans l'élément `<head>`). Si un fichier CSS séparé est créé, il recevra le même nom que le fichier HTML, mais aura une extension `.css`. Cocher ou décocher la case *Intégrer CSS dans HTML* pour définir l'option requise.
- L'option *Intégrer Diagrammes* est activée pour les options de sortie MS Word et RTF. Lorsque cette option est cochée, les diagrammes sont intégrés dans le fichier de résultat, soit dans un format PNG ou EMF. Sinon, les diagrammes sont créés en tant que fichiers PNG ou EMF, qui sont affichés dans le fichier de résultat par des liens d'objet.
- Lorsque la sortie est HTML, tous les diagrammes sont créés en tant que fichier PNG externes. Si la case à cocher *Créer dossier pour les diagrammes* est cochée, un dossier sera créé dans le même dossier que le fichier HTML, et les fichiers PNG seront enregistrés à l'intérieur. Ce dossier aura un nom du format *`HTMLFilename`*`_diagrams`. Si la case *Créer dossier pour les diagrammes* n'est pas cochée, les fichiers PNG seront enregistrées dans le même dossier que le fichier HTML.
- Dans le volet Inclure, vos sélectionnez les items que vous souhaitez inclure dans la documentation. L'option *Aperçu* liste tous les composants, organisés par type de composant en haut du fichier. Les boutons **Cocher tout** et **Décocher tout** vous permettent de sélectionner ou désélectionner rapidement toutes les options dans le volet.
- Le volet Détails liste les détails qui peuvent être inclus pour chaque composant. Sélectionner les détails que vous souhaitez inclure dans la documentation. Les boutons **Cocher tout** et **Décocher tout** vous permettent de sélectionner ou désélectionner rapidement toutes les options dans le volet. La case à cocher *Messages* est uniquement activée si *Rôles de lien génériques* est coché dans le volet Inclure. Toutes les autres cases à cocher sont activée si la case *Éléments globaux* ou *Rôles de lien génériques* est coché dans le volet Inclure.
- L'option *Afficher fichier de résultat* est activée pour toutes les options de sortie. Lorsque cette option est cochée, les fichiers de résultat sont affichés dans le Mode Navigateur (sortie HTML), MS Word (sortie MS Word), et les applications par défaut pour les fichiers `.rtf` (sortie RTF) et les fichiers `.pdf` (sortie PDF).
- Dans le panneau Options, vous pouvez choisir (i) si le nom d'élément doit être affiché avec juste un préfixe (nom qualifié bref) ou dans sa forme agrandie (avec l'espace de noms complet) ; et (ii) si les éléments importés doivent aussi être affichés.

Valeurs de paramètre

Si la SPS StyleVision contient une ou plusieurs définitions de paramètres, en cliquant sur **OK**, un dialogue s'ouvre listant tous les paramètres définis dans la SPS. Vous pouvez saisir des valeurs de paramètre dans ce dialogue pour contourner les valeurs de paramètre par défaut qui ont été assignées dans la SPS.

28.16.8.2 Design défini par l'utilisateur

Au lieu du design XMLSpy standard fixe, vous pouvez créer un design personnalisé pour la documentation de taxonomie XBRL. Le design personnalisé est créé dans une SPS StyleVision qui est un modèle de design pour le document de sortie.

Créer la SPS

Une StyleVision Power Stylesheet (ou SPS) est créée en utilisant le produit Altova StyleVision. La SPS de génération de la documentation de taxonomie XBRL doit être basée sur un schéma XML qui spécifie la structure de la documentation de taxonomie XBRL. Ce schéma est appelé `XBRLDocumentation.xsd`, et il est fourni avec votre package XMLSpy. Il est stocké dans le dossier : `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2023\Documentation\XBRL`.

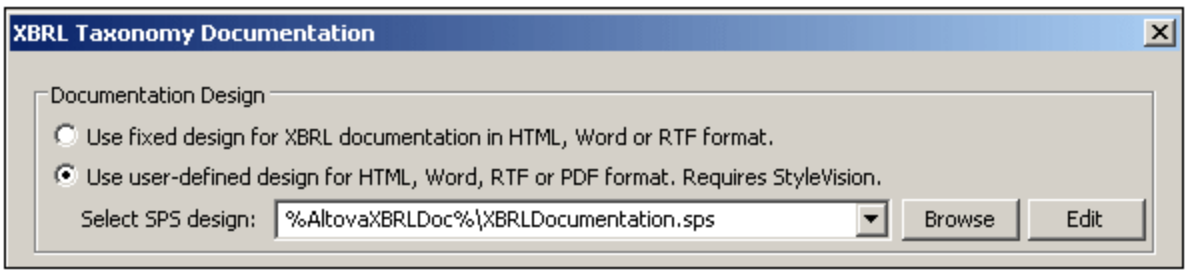
Lors de la création du design SPS dans StyleVision, des nœuds provenant du schéma `XBRLDocumentation.xsd` sont placés dans le modèle de design et des styles et des propriétés leurs sont assignés. Des composants supplémentaires, comme des liens, des tables et des images peuvent aussi être ajoutés au design SPS. De cette manière, le document de sortie complet peut être conçu dans la SPS. Voir le manuel d'utilisation StyleVision pour consulter des explications détaillées de la création d'un design de SPS dans StyleVision.

L'avantage d'utiliser une SPS pour générer la documentation de taxonomie XBRL est que vous détenez le contrôle absolu du design SPS. Veuillez aussi noter que la sortie PDF de la documentation de taxonomie XBRL est uniquement disponible si une SPS définie par l'utilisateur est utilisée ; la sortie PDF n'est pas disponible si le design XMLSpy fixe est utilisé.

Spécifier la SPS à utiliser pour la documentation de taxonomie XBRL

Une fois qu'une SPS a été créée, elle peut être utilisée pour générer une documentation de taxonomie XBRL. La SPS que vous souhaitez utiliser pour générer la documentation de taxonomie XBRL est sélectionnée dans le dialogue Documentation de taxonomie XBRL (accédé par le biais de la commande **XBRL | Générer Documentation**). Dans le volet Design de Documentation de ce dialogue (*voir capture d'écran ci-dessous*), sélectionner le bouton radio *Utiliser design défini par l'utilisateur*. Vous pouvez ensuite cliquer sur la touche **Parcourir** et chercher la SPS que vous souhaitez. Cliquer sur la touche **OK** du dialogue et, dans le dialogue Enregistré qui s'ouvre, sélectionner le dossier pour et saisir le nom du fichier de sortie.

Note : le fichier SPS doit situer correctement le schéma sur lequel il est basé : `XBRLDocumentation.xsd` (*voir ci-dessus*).



Un design de SPS éditable pour la génération de la documentation de taxonomie XBRL est fourni avec XMLSpy. Il s'appelle `XBRLDocumentation.sps` et se trouve dans le dossier : `C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2011\Documentation\XBRL\`. Ce fichier SPS, avec les autres fichiers SPS que vous avez récemment parcourus sera disponible dans la liste de choix de l'option *Utiliser défini par l'utilisateur* (*voir capture d'écran ci-dessous*).

Cliquer sur la touche **Éditer** dans le volet de Design de Documentation pour lancer StyleVision et ouvrir la SPS choisie dans une fenêtre StyleVision. Afin de préconsulter de document de résultat dans StyleVision,

© 2016-2022 Altova GmbH

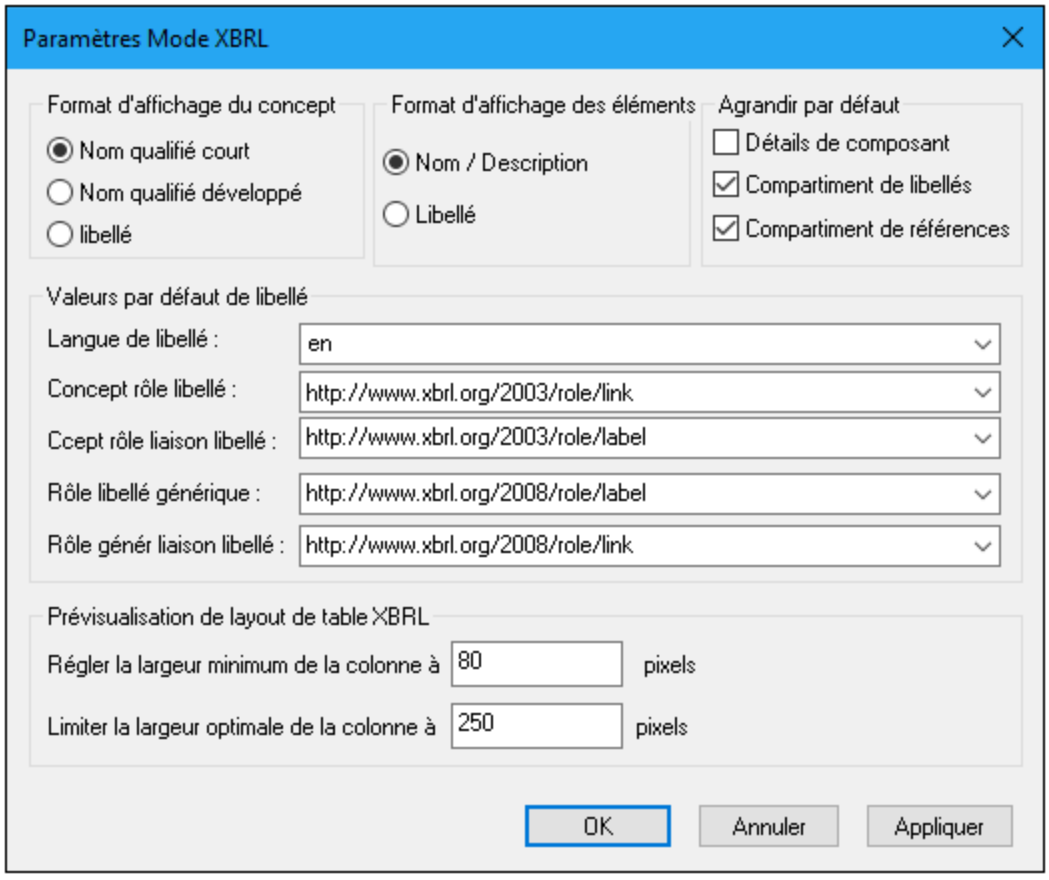
vous nécessiterez un fichier de travail XML. Un fichier d'échantillon XML est fourni à cet effet avec votre application, et il s'appelle `nanonull.xml`. Il se trouve dans le dossier :

```
C:\Documents and Settings\<username>\My
Documents\Altova\XMLSpy2023\Documentation\XBRL\SampleData
```

Note: Afin de pouvoir utiliser une SPS pour générer une documentation de taxonomie XBRL, vous devez avoir installé StyleVision sur votre appareil.

28.16.9 Paramètres de mode

La commande **Paramètres de mode** permet d'ouvrir le dialogue Paramètres de mode XBRL (*capture d'écran ci-dessous*), dans lequel vous pourrez spécifier les paramètres par défaut pour le Mode XBRL.



Les paramètres suivants peuvent être effectués :

- *Affichage des noms de concept* peut être réglé sur le nom qualifié long ou court ou sur des libellés. Ces paramètres s'appliquent à la Fenêtre principale et aux assistants à la saisie Détails mais pas à l'assistant à la saisie Éléments globaux. L'affichage des items dans l'assistant à la saisie Éléments globaux est défini dans la barre de menu de l'assistant à la saisie.

- *Format d'affichage de ressource :* dans les onglets Formule et Table, les ressources peuvent être affichées soit par leurs noms, soit par des libellés. (Si aucun nom n'a été attribué, la description de la ressource est utilisée).
- *Agrandir par défaut :* dans la Fenêtre principale, les détails d'élément, la fenêtre de libellés et la fenêtre de références peuvent être définies par défaut à l'état étendu. Veuillez noter que, si les fenêtres de libellés ou de références sont définies pour montrer un agrandissement par défaut, les fenêtres agrandies seront uniquement visibles lorsque les détails d'Élément seront étendus (soit par défaut, soit manuellement). À chaque fois que le mode est réinitialisé (par exemple, lorsque le mode passe de Mode Texte à Mode XBRL), le Mode XBRL retourne aux paramètres par défaut.
- *Défauts de libellé* spécifie la langue par défaut et les rôles de libellé par défaut à utiliser si les libellés ne sont pas définis. La liste de choix pour chaque propriété affiche une liste des valeurs disponibles.
- *Prévisualisation de layout de table XBRL :* les largeurs minimum et maximum des colonnes peuvent être définies en pixels.

28.16.10 Générer XBRL depuis BD, Excel, CSV avec MapForce

Cette commande lance la génération d'un fichier d'instance XBRL basé sur la taxonomie active actuellement. Les données pour le fichier d'instance sont obtenues depuis une feuille de calcul MS Excel, une base de données ou un fichier CSV. Le fichier d'instance XBRL est généré par programme MapForce d'Altova que vous devez avoir préalablement installé sur votre appareil. La commande lance Altova MapForce et y charge la taxonomie. Vous pouvez ensuite spécifier le fichier de données de source et élaborer graphiquement la sortie de fichier d'instance requise.

28.16.11 Présenter XBRL en tant que HTML/PDF/Word avec StyleVision

Cette commande charge la taxonomie active actuellement dans Altova StyleVision, qui vous permet de générer un design pour les rapports sur la base de la taxonomie. Afin de pouvoir utiliser cette commande, vous devez avoir préalablement installé Altova StyleVision sur votre appareil.

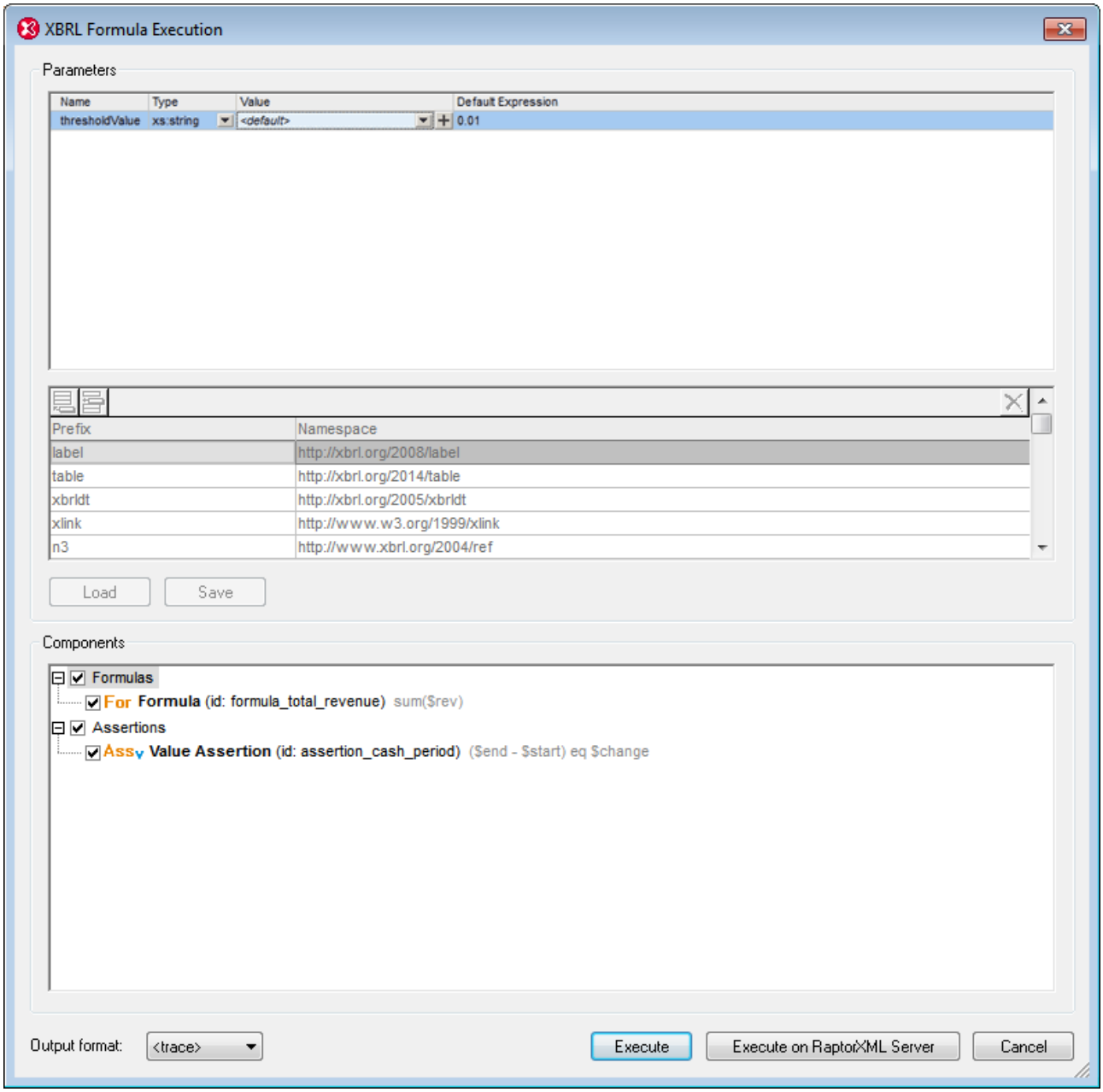
28.16.12 Exécuter Formule (sur Serveur)

Les commandes **Exécuter Formule** et **Exécuter les Formules sur Serveur (haute performance)** sont activées lorsqu'un document d'instance XBRL est le document actif dans le Mode Texte ou le Mode Grille. Ces commandes exécutent des formules et/ou des assertions définies dans le DTS associées avec le fichier d'instance XBRL. (Un DTS (Discoverable Taxonomy Set) est une collection de taxonomies.) Les formules sont évaluées avec des données dans le fichier d'instance XBRL, et les résultats sont des sorties dans un fichier d'instance XBRL. Les assertions sont évaluées séparément et les résultats sont la sortie dans un fichier JSON ou XML.

La commande **Exécuter les Formules sur Serveur (haute performance)** utilise un serveur RaptorXML+XBRL associé pour exécuter les formulaires. Utiliser la commande **Outils | Gérer Serveurs Raptor** pour configurer un Serveur RaptorXML+XBRL.

© 2016-2022 Altova GmbH

Si aucune formule ou assertion n'est définie dans le DTS, un message s'affichera. Si un formulaire valide ou une assertion est définie dans le DTS, le dialogue d'Exécution de la formule XBRL (*capture d'écran ci-dessous*) s'ouvre.



Paramètres

Si des paramètres sont définis dans le DTS, chaque paramètre sera affiché dans le volet Paramètre et une valeur peut être saisie. Les noms de paramètre sont disponibles en lecture seule uniquement. Les paramètres obligatoires sont affichés par un point d'exclamation rouge et le bouton **OK** est désactivé jusqu'à ce qu'une valeur est assignée au paramètre. Les paramètres optionnels ont une valeur par défaut. Si un type requis est spécifié, le type sera affiché. Les paramètres qui nécessitent des valeurs multiples sont indiqués par une icône **+**, qui peut être cliquée pour ajouter une nouvelle valeur. Veuillez noter que les paramètres optionnels sans

valeur ne seront pas transmis au moteur pour l'exécution. Les valeurs par défaut sont disponibles en lecture seule uniquement et seront exécutées si l'utilisateur ne saisit pas de valeur.

Mappages d'espace de noms
Cette table définit les préfixes qui sont utilisés dans les QNames de paramètres et de types. Des espaces de noms supplémentaires à utiliser dans l'évaluation des paramètres peuvent être définis ici.

Enregistrer et charger les paramètres
La configuration des paramètres, y compris les mappages d'espace de noms, peut être enregistrée sous format JSON ou XML en cliquant sur la touche **Enregistrer**. Le format de fichier est déterminé par l'extension de fichier donnée au fichier. Veuillez noter que les paramètres optionnels ne contenant pas de valeur ne seront pas enregistrés. Une fois enregistré, un fichier de paramètre peut être chargé dans le dialogue par le biais du bouton **Charger**.

Options de traitement XBRL
La touche **Options** ouvre le dialogue Options de traitement XBRL dans lequel vous pouvez passer sur la déduplication (pour ignorer automatiquement les faits dupliqués).

Composants

Ce volet contient un affichage arborescent qui permet d'exécuter la sélection de composants de formule et/ou d'assertion. Chaque item montre une icône et sa description, ainsi que l'ID et l'expression, si disponibles. Pour sélectionner un composant à exécuter, cocher sa case. Les sorties des exécutions d'assertion peuvent être soit JSON ou XML ; sélectionner le format de sortie dans la liste de choix *Format de sortie*. Le format d'une exécution de formule XBRL est toujours XML.

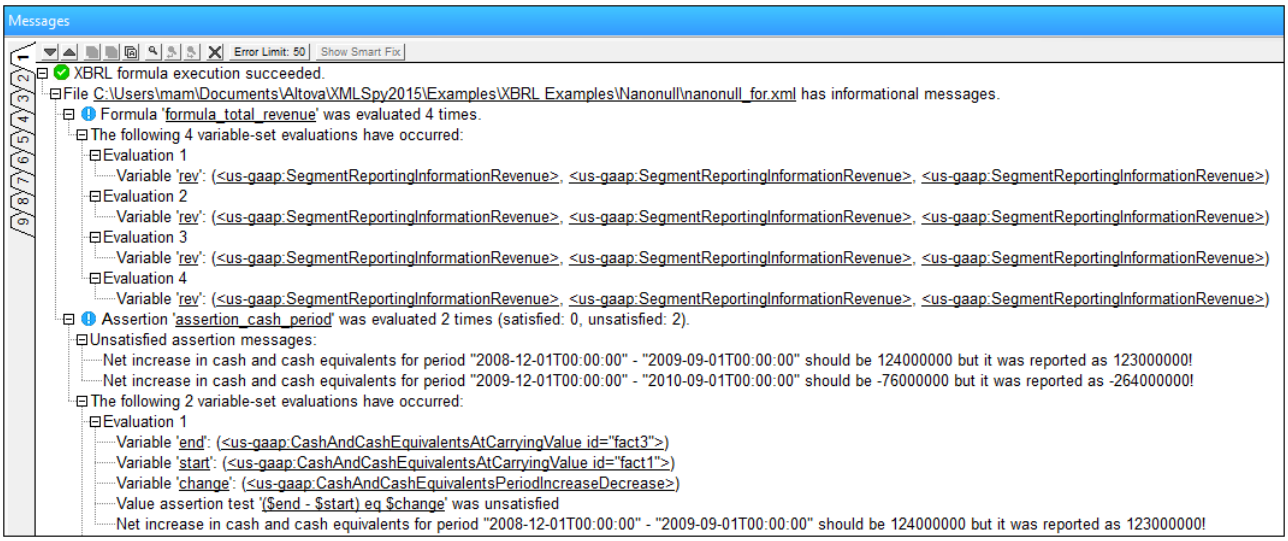
Exécution

Vous pouvez sélectionner si l'exécution doit être effectuée avec le moteur interne de XMLSpy ou avec le Serveur RaptorXML d'Altova. En cas d'une erreur d'exécution, un message d'erreur est affiché dans la fenêtre de sortie. Sinon, un message de réussite est affiché. Les fichiers de sortie, `assertions-ouput-file.xml/json` et/ou `formula-output-file.xml`, sont ouverts dans de nouvelles fenêtres de document, ils ne sont pas sauvegardés sur le disque dur. Vous devrez enregistrer explicitement le fichier à l'emplacement que vous souhaitez sur le disque.

Trace

Si vous choisissez **<trace>** dans la liste de choix *Format de sortie* (située en bas à gauche du dialogue), Des informations de débogage supplémentaires seront collectées pour toutes les “évaluations de set de variable” au cours de l'exécution de formule et puis affichés dans la fenêtre Messages (*voir capture d'écran ci-dessous*). La trace liste les évaluations de set de variable individuelles pour chaque formule/assertion aux endroits où l'assignation réelle des variables dans cette évaluation est affichée. Si les variables référencent des faits d'instance, cliquer sur les valeurs pour vous amener à l'élément de fait correspondant dans l'instance. Cliquer sur la formule/assertion ou sur le nom de variable pour vous emmener à la définition correspondante dans les fichiers de base de lien de formule. Dans le cas d'assertions de validation, les messages d'assertion qui ont été générés dans l'étape d'évaluation sont aussi affichés.

© 2016-2022 Altova GmbH



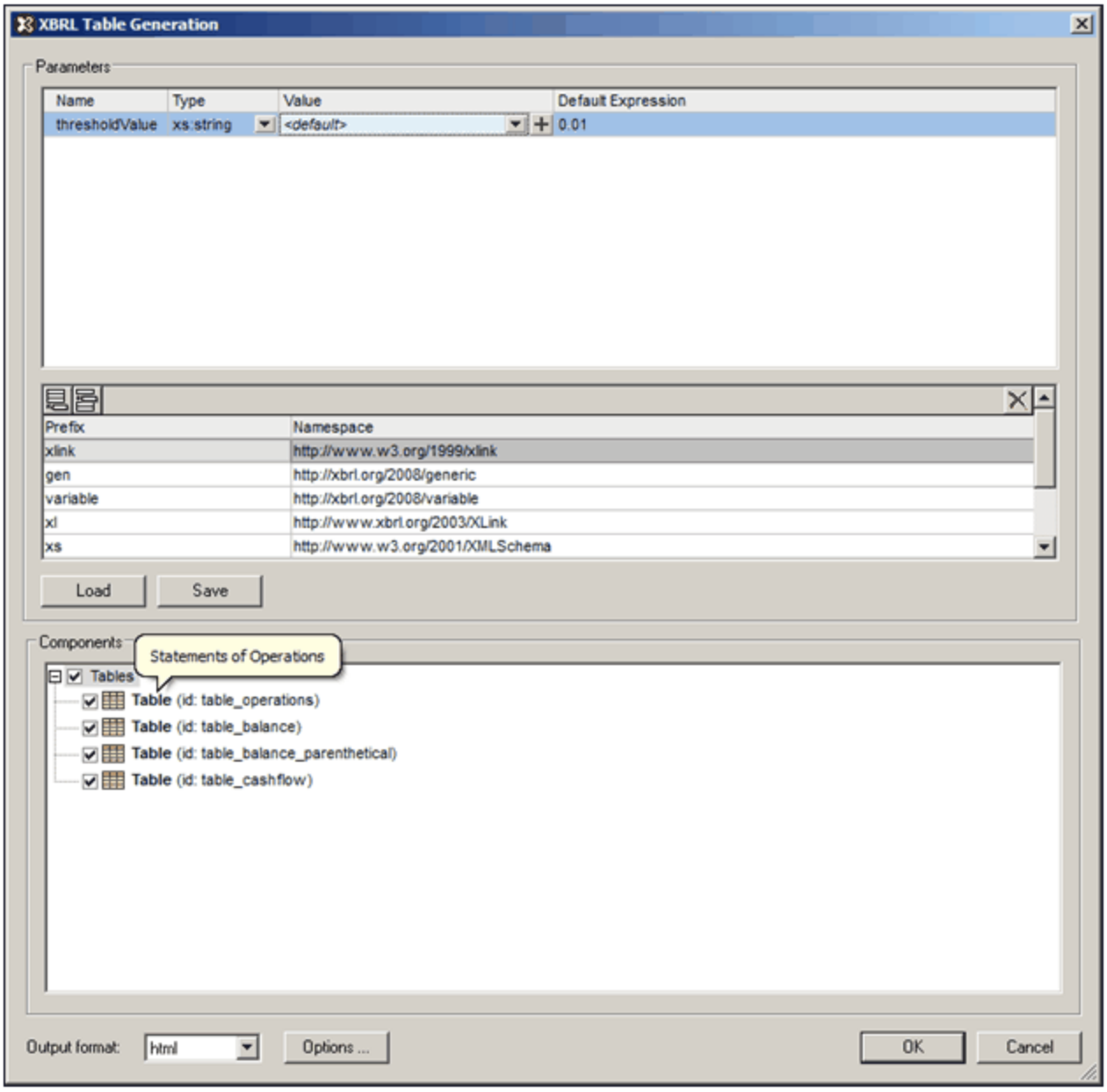
Note : L'exécution d'une trace peut entraîner des dépenses considérables en termes de mémoire et de vitesses de calcul. Lorsque vous utilisez des instances XBRL volumineuses, les assertions peuvent être évaluées des millions de fois et chaque évaluation peut nécessiter de stocker les valeurs de jusqu'à 40 variables. Ainsi, cette fonction doit uniquement être utilisée pour le débogage avec des échantillons petits/réduits, sinon l'exécution sera lente et XMLSpy pourrait manquer de mémoire. C'est pour cela que chaque trace a une limite codée en dur de 1000 évaluations.

28.16.13 Générer Table (sur Serveur)

Les commandes **Générer Table** et **Générer Table sur Serveur (haute performance)** sont activées lorsqu'(i) un document d'instance XBRL est le document actif dans le Mode Texte ou le Mode Grille ou (ii) une taxonomie XBRL est le document actif dans le Mode XBRL, dans le Mode Texte, le Mode Grille, ou la Mode Schéma. Ces commandes génèrent un document XML ou HTML contenant des tables XBRL définies dans le DTS associé avec le document actif. (un DTS, (Discoverable Taxonomy Set) est une collection de taxonomies.) Dans le cas des fichiers d'instance XBRL, les tables sont générées avec des données dans le fichier d'instance XBRL.

La commande **Générer Table sur Serveur (haute performance)** utilise un serveur RaptorXML+XBRL associé pour générer les tables. Utiliser la commande **Outils | Gérer Serveurs Raptor** pour configurer un Serveur RaptorXML+XBRL.

Si aucune table n'est définie dans le DTS, un message s'affichera. Si une définition de table valide est définie dans le DTS, le dialogue de Génération de table XBRL (*capture d'écran ci-dessous*) s'ouvre.



Paramètres

Si des paramètres sont définis dans le DTS, chaque paramètre sera affiché dans le volet Paramètre et une valeur peut être saisie. Les noms de paramètre sont disponibles en lecture seule uniquement. Les paramètres obligatoires sont affichés par un point d'exclamation rouge et le bouton **OK** est désactivé jusqu'à ce qu'une valeur est assignée au paramètre. Les paramètres optionnels ont une valeur par défaut. Si un type requis est spécifié, le type sera affiché. Les paramètres qui nécessitent des valeurs multiples sont indiqués par une icône **+**, qui peut être cliquée pour ajouter une nouvelle valeur. Veuillez noter que les paramètres optionnels sans valeur ne seront pas transmis au moteur pour l'exécution. Les valeurs par défaut sont disponibles en lecture seule uniquement et seront exécutées si l'utilisateur ne saisit pas de valeur.

© 2016-2022 Altova GmbH

Mappages d'espace de noms
Cette table définit les préfixes qui sont utilisés dans les QNames de paramètres et de types. Des espaces de noms supplémentaires à utiliser dans l'évaluation des paramètres peuvent être définis ici.

Enregistrer et charger les paramètres
La configuration des paramètres, y compris les mappages d'espace de noms, peut être enregistrée sous format JSON ou XML en cliquant sur la touche **Enregistrer**. Le format de fichier est déterminé par l'extension de fichier donnée au fichier. Veuillez noter que les paramètres optionnels ne contenant pas de valeur ne seront pas enregistrés. Une fois enregistré, un fichier de paramètre peut être chargé dans le dialogue par le biais du bouton **Charger**.

Options de traitement XBRL
La touche **Options** ouvre le dialogue Options de traitement XBRL dans lequel vous pouvez passer sur la déduplication (pour ignorer automatiquement les faits dupliqués).

Composants

Ce volet contient un affichage arborescent qui permet d'exécuter la sélection de composants de table. Chaque item montre une icône et sa description, ainsi que l'ID, si disponible. Pour sélectionner un composant de table à exécuter, cocher sa case. Les sorties peuvent être en format XML ou HTML ; sélectionner le format de sortie dans la liste de choix *Format de sortie*. Cliquer sur la touche **Options** pour afficher le dialogue des Options de traitement XBRL dans lequel vous pouvez spécifier les options de génération de table XBRL et éventuellement ignorer les duplicats. Si les options supplémentaires pour les libellés et le traitement des lignes vides sont disponibles par le biais de la touche **Options** (*capture d'écran ci-dessous*). Si les options de libellés préférés sont disponibles, elles seront utilisées ; sinon, les défauts spécifiés dans le dialogue des Options de traitement XBRL seront utilisés.

Exécution

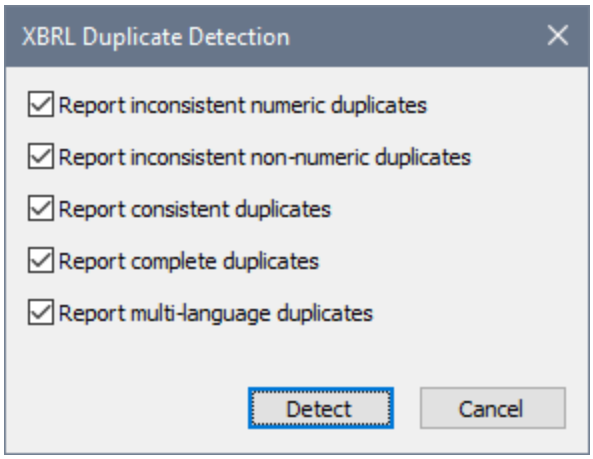
Dans le cas d'une erreur d'exécution, un message d'erreur est affiché dans la fenêtre de sortie. Sinon, un message de réussite sera affiché. Le fichier de sortie, `table-ouput-file.xml/html` est ouvert dans la nouvelle fenêtre de document, il ne sera pas enregistré sur le disque. Vous devrez enregistrer le fichier explicitement à l'endroit précis que vous souhaitez sur le disque.

28.16.14 Détecter les duplicatas (sur Serveur)

Les commandes **Détecter duplicata** et **Détecter duplicata sur Serveur (haute performance)** sont activées lorsqu'un document d'instance XBRL est le document actif dans le Mode Texte ou Grille. Ces commandes vérifient le document d'instance pour les faits doubles et rapportent tous les duplicatas trouvés dans la fenêtre Messages. La commande **Détecter duplicata sur Serveur (haute performance)** utilise un RaptorXML+XBRL Server associé pour vérifier la présence de duplicatas. Utiliser la commande **Outils | Gérer Raptor Servers** pour configurer un RaptorXML+XBRL Server. En cliquant sur une des commandes, le dialogue Détection de duplicata XBRL (*capture ci-dessous*) apparaît. Dans ce dialogue, vous pouvez sélectionner le type de duplicatas que vous souhaitez détecter.

Les différents types de duplicatas sont expliqués en détails dans la spécification Gérer les faits dupliqués dans XBRL et Inline XBRL 1.0. Ils sont révisés brièvement ici :

- *Duplicatas complets* sont des duplicatas qui sont identiques en terme du nom, du contexte, de la valeur ;
- *Duplicatas cohérents* sont des faits numériques dupliqués qui ont la même valeur jusqu'à la place décimale spécifiée pour l'arrondissement ;
- *Duplicatas multi-langage* répètent le même fait dans plusieurs langues ;
- *Duplicatas incohérents* sont des duplicatas qui remplissent les conditions pour des faits dupliqués définis dans la spécification XBRL 2.1, mais qui ne sont pas des duplicatas complets, des duplicatas cohérents ou des duplicatas multi-langage (par exemple, un duplicatas numérique avec une valeur numérique différente).



28.16.15 Execute XULE

La commande **Exécuter XULE** exécute des règles XULE dans un document d'instance XBRL. Les règles XULE peuvent être contenues dans un seul fichier `.xule` ou dans une archive zip (`.zip`) contenant des documents XULE. *Voir la section XBRL | XULE pour plus d'informations.*

La commande est activée dans les cas suivants :

- Lorsqu'un document XULE est le document actif. Généralement, un document XULE possède l'extension de fichier `.xule`. Dans ce cas, vous serez invité à sélectionner l'instance XBRL dans laquelle le document XULE doit être traité.
- Lorsqu'un document d'instance XBRL (possédant généralement une extension de fichier `.xbrl` ou `.xml`) est le document actif. Dans ce cas, vous serez invité à choisir le document XULE à utiliser, ou l'archive zip des fichiers XULE (le ruleset XULE).

Note : Si le document XULE et le document d'instance XBRL font tous les deux partie d'un projet XMLSpy, alors vous pouvez spécifier le fichier d'instance XBRL cible dans les propriétés du projet XMLSpy. Si, ensuite, vous cliquez avec la touche de droite sur le fichier XULE et que vous sélectionnez la commande **Exécuter XULE**, alors d'exécution sera effectuée dans le document XBRL qui est spécifié en tant que la cible pour l'exécution XULE.

© 2016-2022 Altova GmbH

Options d'exécution

Sortie XULE

La sortie de l'exécution XULE est envoyée soit : (i) à la fenêtre Messages, soit (ii) à un nouveau document qui est affiché dans une nouvelle fenêtre XMLSpy et stockée temporairement dans la mémoire ; ce document peut être stocké dans le fichier avec la commande **Fichier | Enregistrer sous**. Pour spécifier si la sortie va dans une fenêtre ou un nouveau document, choisir l'option que vous souhaitez dans l'onglet d'options XBRL XULE (**Outils | Options | XBRL | XULE**); *voir capture d'écran ci-dessous*.



Faits dupliqués

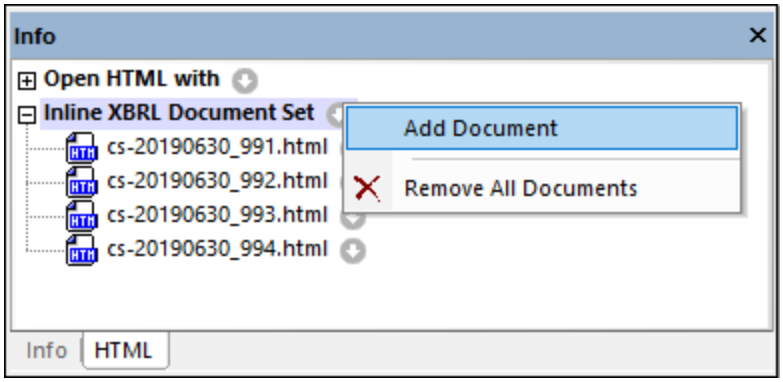
Les faits dupliqués se réfèrent à plusieurs références du même fait. Vous pouvez éviter la présence de faits dupliqués dans le résultat en cochant l'option *Ignorer faits dupliqués* dans l'onglet d'options XBRL XULE (**Outils | Options | XBRL | XULE**); *voir capture d'écran ci-dessus*. Dans ce cas, les faits dupliqués seront rapportés une seule fois.

28.16.16 Transformer Inline XBRL

La commande **Transformer Inline XBRL** est activée lorsqu'un document XHTML contenant un inline XBRL est le document actif dans le Mode Texte, le Mode Grille ou le Mode Navigateur. La commande extrait les données Inline XBRL depuis le document XHTML actif et génère un document XBRL contenant les données extraites. Le document XBRL généré est ouvert dans les nouvelles fenêtres et peut être enregistré sous un fichier. Pour que la commande puisse fonctionner correctement, toutes les ressources référencées par le document Inline XBRL doivent être disponibles pour le traitement.

Traiter plusieurs documents Inline XBRL

Vous pouvez traiter plusieurs documents Inline XBRL en ajoutant les documents Inline XBRL supplémentaires dans le Inline XBRL Document Set de l'onglet HTML de la fenêtre Info *(voir capture d'écran ci-dessous)*. Veuillez noter que cet onglet apparaît dans la fenêtre Info seulement lorsqu'un document HTML est actif dans la fenêtre principale.



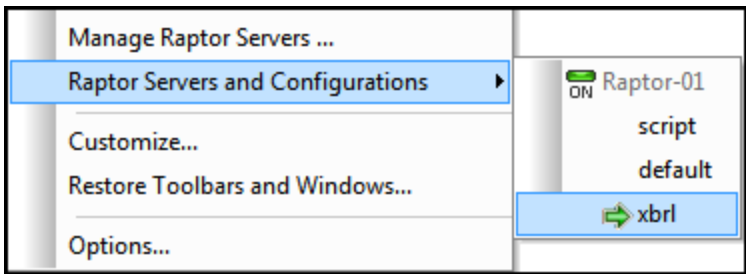
Cliquer sur la touche de menu de *Inline XBRL Document Set*, puis cliquer sur **Ajouter document** *(voir capture d'écran)* et chercher les fichiers Inline XBRL que vous souhaitez ajouter. Lorsque vous exécutez la commande **Transformer Inline XBRL**, le fichier actif et les fichiers du set de documents sera traité. Les données Inline XBRL extraites de tous ces fichiers seront combinés dans un seul document XBRL qui est ouvert dans une nouvelle fenêtre.

28.16.17 Valider EDGAR sur serveur

La commande **Valider EDGAR sur serveur (haute performance)** valide le document d'instance XBRL actif en utilisant le RaptorXML+XBRL Server actif actuellement et sa configuration active. Lorsque vous validez via EDGAR, Raptor valide le document d'instance XBRL en utilisant un script EDGAR interne. La commande effectue immédiatement la validation et affiche les résultats dans la fenêtre Messages.

Note : La performance actuelle dépend du nombre de cœurs de processeurs PC utilisés par RaptorXML+XBRL Server pour la validation : Plus le nombre de cœurs est élevé, plus le traitement sera rapide.

Si vous avez défini plusieurs configurations sur plusieurs serveurs, vous pouvez sélectionner un serveur et une de ses configurations en tant que la configuration active. Celle-ci sera utilisée pour les validations suivantes. En plaçant le curseur sur la commande **Outils | Serveurs Raptor et Configurations** (*capture d'écran ci-dessous*), un sous-menu apparaît qui contient tous les serveurs ajoutés, avec la configuration de chacun d'entre eux. Sélectionner la configuration de serveur que vous souhaitez pour rendre la configuration active. Dans la capture d'écran ci-dessous, la configuration `xbrl` du serveur nommé `Raptor-01` a été sélectionnée en tant que la configuration active (indiquée par la flèche verte).



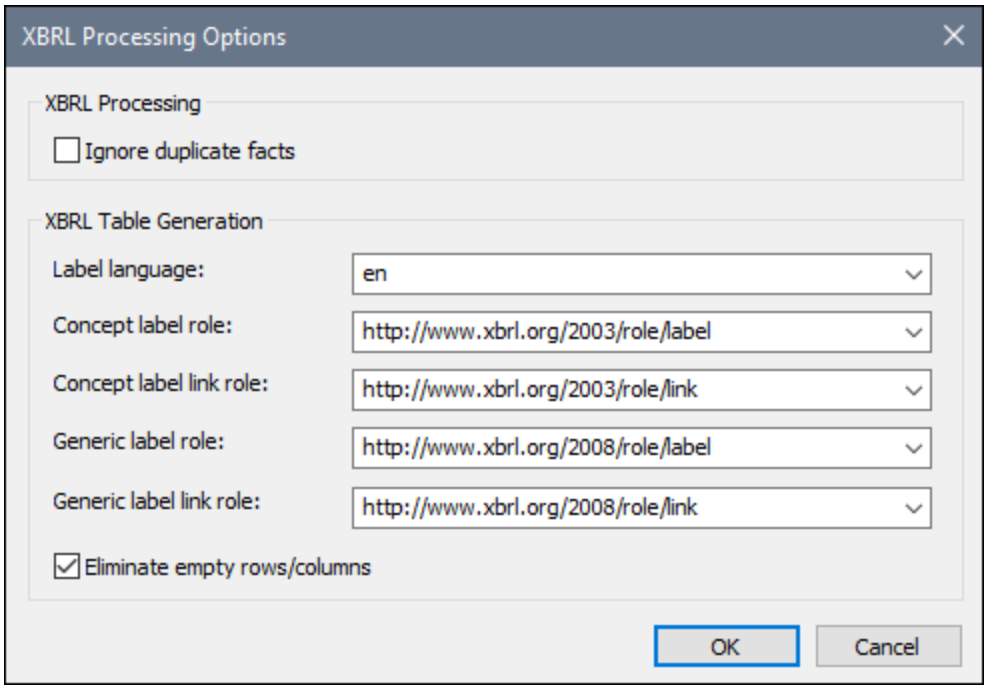
 © 2016-2022 Altova GmbH

Pour plus d'informations, voir la section qui décrit comment utiliser RaptorXML+XBRL Server.

28.16.18 Options de traitement

La commande **Options de traitement XBRL** affiche le dialogue Options de traitement XBRL (*capture d'écran ci-dessous*). Vous pouvez y spécifier :

- que les faits dupliqués sont ignorés pour : (i) l'exécution de formule XBRL, (ii) la génération de table XBRL, (iii) transformations XBRL Inline
- pour des exécutions de formule XBRL, (i) la limite de trace pour des assertions, et (ii) si uniquement des assertions non satisfaites sont tracées ou si les deux assertions satisfaites ou non satisfaites sont tracées
- des paramètres de libellés pour la génération de table XBRL
- que les lignes et les colonnes vides sont éliminées pendant la génération de table XBRL



Note : le dialogue Options de traitement XBRL peut aussi être accédé depuis le dialogue Exécuter Formule XBRL et le dialogue Générer Table XBRL.

Note concernant la dé-duplication

Seuls les types suivants de fait dupliqué sont ignoré : (i) duplicatas complets, et (ii) duplicatas cohérents. Le fait (parmi les duplicatas) qui est sélectionné (pas ignoré) est le plus précis de tous. Par exemple, parmi les duplicatas cohérents d'un fait numérique, le fait présentant la précision numérique la plus haute est sélectionnée.

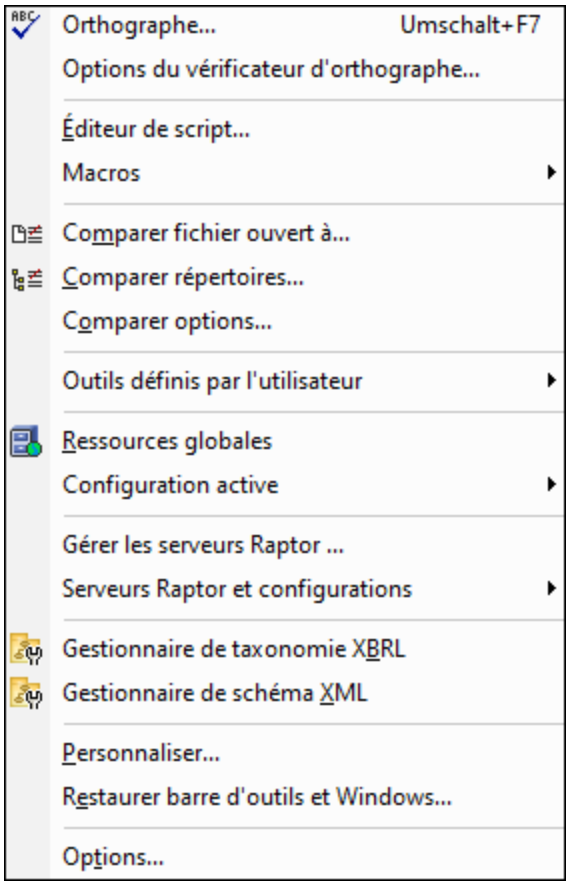
Pour plus d'informations concernant les duplicatas, voir la spécification Gérer les faits dupliqués dans XBRL et Inline XBRL 1.0.

© 2016-2022 Altova GmbH

28.17 Menu Outils

Le menu Outils vous permet de :

- Contrôler l'orthographe de vos documents XML
- Accéder à l'environnement des scripts de XMLSpy. Vous pouvez créer, gérer et stocker vos propres formulaires, macros et gestionnaires d'événements
- Consulter les macros attribués actuellement
- Comparer deux fichiers quelconques pour contrôler les différences
- Comparer deux dossiers quelconques pour contrôler les différences
- Accéder aux commandes personnalisées qui utilisent des applications externes. Ces commandes peuvent être créées dans l'onglet Outils du dialogue Personnaliser.
- Définir les ressources globales
- Changer la configuration active pour les ressources globales dans XMLSpy
- Ajouter des serveurs RaptorXML pour la validation XML et XBRL, et pour configurer les options de validation RaptorXML
- Sélectionner une configuration Raptor Server en tant que la configuration active
- Gérer vos packages de taxonomie XBRL via une application propre, Gestionnaire de taxonomie
- Personnaliser votre version de XMLSpy : définir vos propres barres d'outils, raccourcis de clavier, menus et macros
- Définir des paramètres globaux de XMLSpy



28.17.1 Orthographe

Le vérificateur orthographique de XMLSpy contenant des dictionnaires de langue intégré (*voir la note ci-dessous*) est activé dans le Mode Texte, le Mode Grille et Authentic View. Si vous souhaitez contrôler l'orthographe d'un document que vous avez édité dans un autre mode, vous pouvez passer au Mode Texte ou Grille et effectuez une vérification orthographique. Par exemple, si vous avez édité un document de Schéma XML dans le Mode Schéma, passer au Mode Texte ou Grille pour la vérification orthographique.

Note : La sélection des dictionnaires intégrés fournis avec le logiciel Altova ne constitue pas des préférences linguistiques de la part d'Altova, mais est basée largement sur la disponibilité des dictionnaires qui permettant la redistribution avec des logiciels commerciaux, comme par exemple les licences MPL, LGPL ou BSD. Il existe nombre d'autres dictionnaires open-source, mais ceux-ci sont distribués sous des licences plus restrictives, comme la licence GPL. Nombre de ces dictionnaires sont disponibles dans un installateur séparé situé sous http://www.altova.com/dictionaries. À vous de choisir si vous souhaitez accepter les termes de la licence applicables au dictionnaire et si le dictionnaire est approprié pour votre utilisation avec le logiciel sur votre ordinateur.

Cette section décrit comment utiliser le vérificateur orthographique. Elle est organisée dans les sous-sections suivantes :

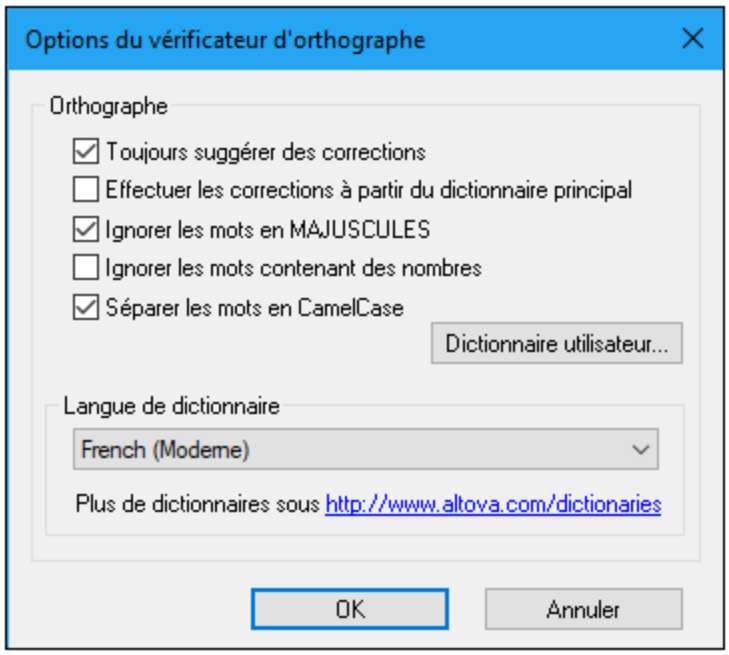
- Choisir la langue du vérificateur automatique
- Définir l'étendue du contrôle
- Exécuter la vérification orthographique

Choisir la langue du vérificateur orthographique

La langue du vérificateur orthographique peut être définie comme suit :

1. Cliquez sur la commande de menu **Outils | Options du vérificateur orthographique**.
2. Dans le dialogue Options du vérificateur orthographique qui s'ouvre maintenant (*capture d'écran ci-dessous*), choisissez un des dictionnaires installés à partir de la liste déroulante de la liste de choix de la Langue du dictionnaire.

© 2016-2022 Altova GmbH

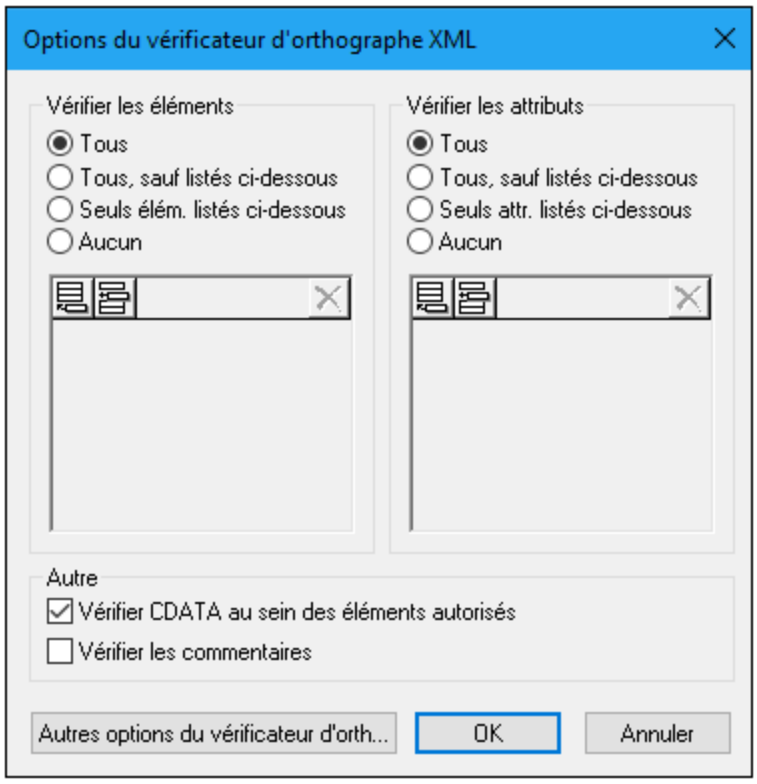


3. Cliquer sur **OK** pour finir.

La Langue du dictionnaire que vous avez choisie sera utilisée par la vérification orthographique pour les vérifications. Si la langue que vous souhaitez n'est pas encore installée, vous pouvez télécharger des dictionnaires de langues supplémentaires. Vous trouverez une description dans la section Ajouter des dictionnaires pour la vérification orthographique.

Définir l'étendue du contrôle

Lorsque la vérification orthographique est exécutée dans le Mode Texte ou dans le Mode Grille, l'étendue de la vérification peut être définie immédiatement avec de commencer la vérification. Pour ce faire, sélectionnez la commande **Outils | Options du vérification orthographique** et en définissant l'étendue requise dans le dialogue Options du vérificateur orthographique XML qui s'ouvre (*voir la capture d'écran ci-dessous*).

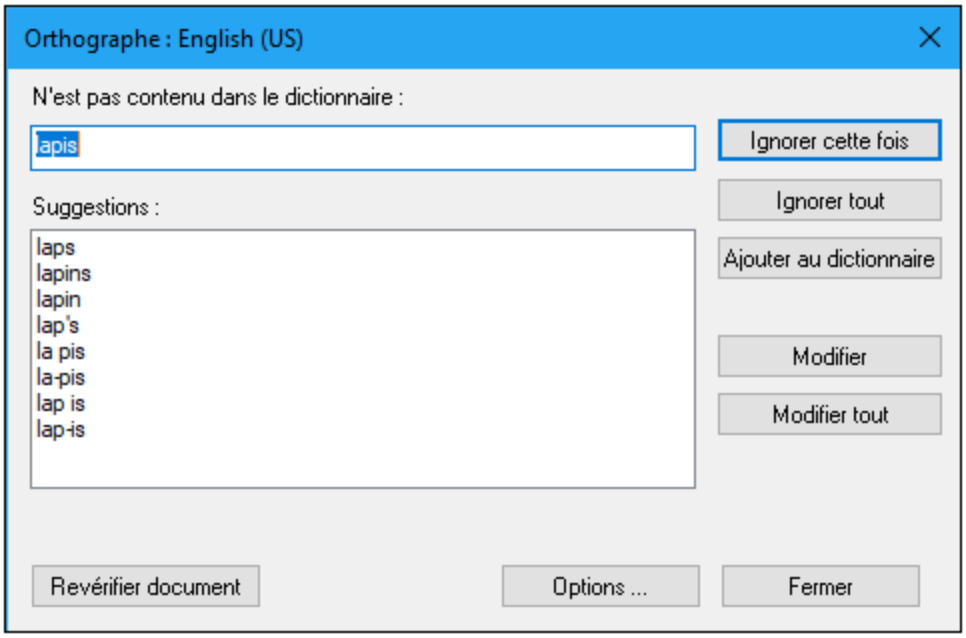


Vous pouvez sélectionner les éléments et les attributs qui doivent être vérifiés et si les sections CDATA et les commentaires doivent être cochés. Voir également la description de la commande Options du vérificateur orthographique pour les informations liées.

Exécuter la vérification orthographique

La commande **Outils | Orthographe (Shift+F7)** commence automatiquement à vérifier le document XML actuellement actif conformément à l'étendue définie. Si un mot inconnu est rencontré, le dialogue *Orthographe: N'est pas contenu dans le dictionnaire* s'ouvre (voir la *capture d'écran ci-dessous*). Sinon, la vérification orthographique se poursuit jusqu'au bout du document.

© 2016-2022 Altova GmbH



Les différentes parties du dialogue *: N'est pas contenu dans le dictionnaire* et les options disponibles sont décrites ci-dessous :

N'est pas contenu dans le dictionnaire
Cette fenêtre contient le mot qui ne peut pas être trouvé dans le dictionnaire de langue ou le dictionnaire utilisateur sélectionné. Les options suivantes sont disponibles :

- Vous pouvez éditer le mot dans la fenêtre manuellement ou sélectionner une suggestion depuis le volet *Suggestions*. Ensuite, cliquez sur **Modifier** pour remplacer le mot dans le document XML avec le mot éditer. (Double-cliquez sur une suggestion pour l'insérer directement dans le document XML.) Lorsqu'un mot est affiché dans la fenêtre *N'est pas contenu dans le dictionnaire*, il est également marqué dans le document XML, vous pouvez donc éditer le mot directement dans le document si vous le souhaitez. Cliquer sur **Modifier tout** pour remplacer toutes les occurrences du mot dans le document XML avec le mot édité.
- Vous pouvez choisir de ne pas effectuer de modifications et d'ignorer les avertissements du vérificateur orthographique, soit uniquement pour l'occurrence actuelle du mot soit pour toutes les occurrences.
- Vous pouvez ajouter le mot au dictionnaire d'utilisateur, le mot sera donc considéré comme étant correct pour toutes les vérifications depuis la vérification actuelle jusqu'à la fin du document.

Suggestions
Cette liste affiche des mots qui ressemblent au mot inconnu (fourni par les dictionnaires de langue et d'utilisateur). Double-cliquer sur un mot dans cette liste pour l'insérer automatiquement dans le document et continuer le processus de vérification orthographique.

Ignorer cette fois
Cette commande vous permet de continuer la vérification du document tout en ignorant la première occurrence du mot inconnu. Le même mot sera marqué à nouveau s'il réapparaît dans le document.

Ignorer tout
Cette commande ignore toutes les instances du mot inconnu dans le document entier.

Ajouter au dictionnaire
Cette commande ajoute le mot inconnu au **dictionnaire utilisateur**. Vous pouvez accéder au dictionnaire utilisateur (afin de l'éditer) par le biais du dialogue Options du vérificateur orthographique.

Modifier
Cette commande remplace le mot actuellement marqué dans le document XML avec le mot (édité) dans la fenêtre *N'est pas contenu dans le dictionnaire*.

Modifier tout
Cette commande remplace toutes les occurrences du mot actuellement marqué dans le document XML avec le mot (édité) dans la fenêtre *N'est pas contenu dans le dictionnaire*.

Revérifier le document
La touche **Revérifier document** redémarre la vérification à partir du début du document.

Options
Cliquer sur la touche **Options** pour ouvrir un dialogue dépendant du mode actuel.

- Si le mode actuel est Authentic View, le dialogue Options du vérificateur orthographique est ouvert.
- Si le mode actuel est Mode Texte ou Grille, alors le dialogue Options du vérificateur orthographique XML est ouvert.

Pour plus d'informations concernant ces dialogues, voir la section Options du vérificateur orthographique.

Fermer
Cette commande ferme la fenêtre du dialogue Orthographe.

28.17.2 Options du vérificateur orthographique

La commande **Outils | Options du vérificateur orthographique** ouvre les **Options du vérificateur orthographique**. Selon le mode actif, la commande **Outils | Options du vérificateur orthographique** ouvre soit directement le dialogue **Options du vérificateur orthographique** (Mode Schéma, Mode WSDL, Mode XBRL, Mode Authentic, Mode Browser), ou le dialogue **Options du vérificateur orthographique** (Mode Text, Mode Grille). Le dialogue Options du vérificateur orthographique XML a une touche **Options du vérificateur orthographique** pour accéder au dialogue Options du vérificateur orthographique.

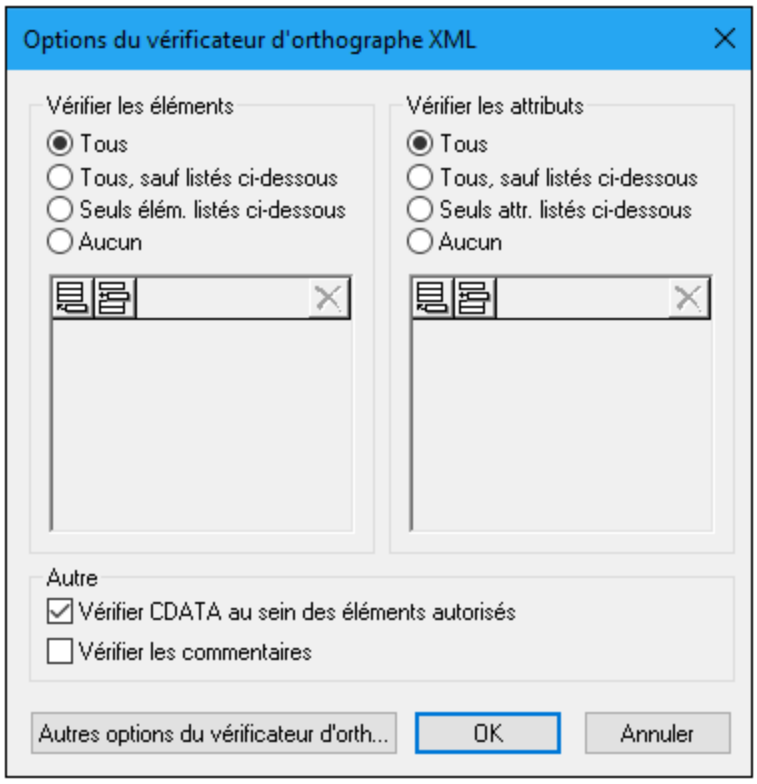
Les paramètres variés dans ces deux dialogues sont décrits dans les sous-sections de cette section :

- Contexte de la vérification orthographique
- Options du vérificateur orthographique
- Ajouter des dictionnaires au vérificateur orthographique
- Travailler avec le dictionnaire d'utilisateur

Dialogue Options du vérificateur orthographique XML

Cliquer sur la commande **Outils | Options du vérificateur orthographique** dans le Mode Texte ou le Mode Grille pour ouvrir le dialogue Options du vérificateur orthographique XML (*capture d'écran ci-dessous*), dans lequel vous pouvez définir l'étendue de la vérification orthographique. Vous pouvez sélectionner quels éléments et attributs doivent être cochés et si les sections et les commentaires doivent être cochés.

© 2016-2022 Altova GmbH

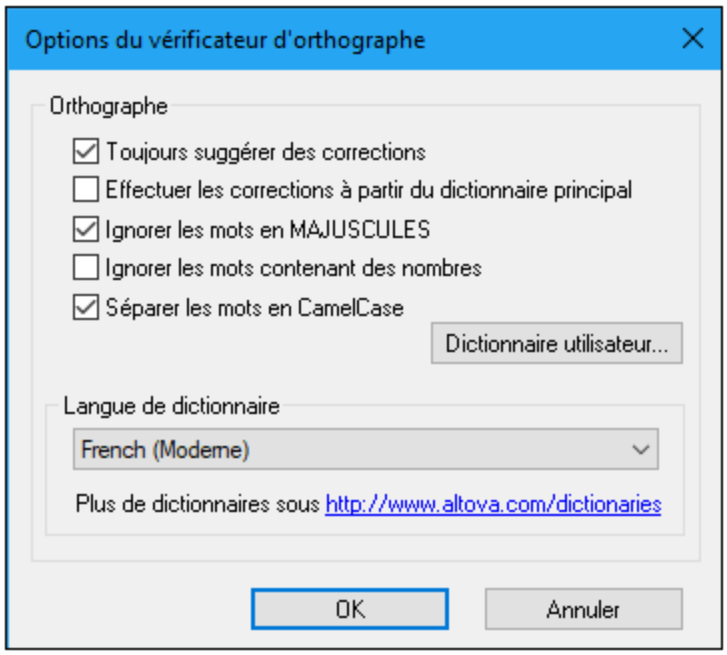


Vous pouvez compiler une liste des éléments et/ou attributs que vous souhaitez vérifier ou que vous souhaitez exclure de la vérification. En alternative, vous pouvez choisir d'exécuter la vérification orthographique sur tous les éléments et/ou attributs ou sur aucun élément ou attribut.

Cliquer sur la touche **Autres options du vérificateur orthographique** tout en bas du dialogue pour ouvrir le dialogue Options du vérificateur orthographique.

Options du vérificateur orthographique

Le dialogue Options du vérificateur orthographique est utilisé pour définir des options de vérification orthographique globales.



Toujours suggérer des corrections :
En activant cette option, des suggestions (provenant du dictionnaire de langue et du dictionnaire d'utilisateur) s'affichent dans la liste de choix Suggestions. En désactivant cette option, aucune suggestion ne sera affichée.

Effectuer les corrections à partir du dictionnaire principal :
En activant cette option, seul le dictionnaire de langue (dictionnaire principal) sera utilisé. Le dictionnaire d'utilisateur n'est pas utilisé pour proposer des suggestions. Il désactive aussi le bouton **Dictionnaire utilisateur**, empêchant toute édition par le dictionnaire d'utilisateur.

Ignorer les mots en MAJUSCULES :
En activant cette option, les mots en majuscule seront ignorés.

Ignorer les mots contenant des numéros :
En activant cette option, les mots contenant des nombres seront ignorés.

Séparer les mots en CamelCase
Les mots en CamelCase sont des mots qui contiennent une majuscule à l'intérieur du mot. Par exemple, dans le mot "CamelCase", le "C" de "Case" est en majuscule et peut donc être considéré comme étant une CamelCase. Étant donné que les mots en CamelCase sont rarement trouvés dans les dictionnaires, le correcteur orthographique les marquerait comme des erreurs. Pour éviter ce problème, l'option *Séparer les mots en CamelCase* sépare les mots en CamelCase dans leurs composants à majuscules et contrôle chaque composant individuellement. Cette option est contrôlée par défaut.

Langue de dictionnaire
Utiliser cette liste de choix pour sélectionner la langue de dictionnaire que vous souhaitez utiliser pour la vérification orthographique. La sélection par défaut est US English. Les autres dictionnaires en d'autres langues peuvent être téléchargés gratuitement depuis le site web Altova.

© 2016-2022 Altova GmbH

Ajouter des dictionnaires au vérificateur orthographique

Pour chaque langue de dictionnaire, il existe deux fichiers de dictionnaire Hunspell qui fonctionnent ensemble : un fichier `.aff` et un fichier `.dic`. Tous les dictionnaires de langue sont installés dans un dossier `Lexicons` sous : `C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons`.

Dans le dossier `Lexicons`, plusieurs dictionnaires de langue différents sont stockés dans un dossier différent : `<language name>\<dictionary files>`. Par exemple, les fichiers pour les deux dictionnaires de langue anglaise (`English (British)` et `English (US)`) seront stockés comme dans l'exemple ci-dessous :

```
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.dic
```

Dans le dialogue Options du vérificateur d'orthographe, la liste déroulante de la liste de choix *Langue de dictionnaire* affiche les dictionnaires de langue. Ces dictionnaires sont ceux disponibles dans le dossier `Lexicons` et ont les mêmes noms que les sous-dossiers de langue dans le dossier `Lexicons`. Par exemple, dans le cas des dictionnaires de langue anglaise affichés ci-dessus, les dictionnaires apparaîtraient dans la liste de choix Langue de dictionnaire en tant que : *English (British)* et *English (US)*.

Tous les dictionnaires installés sont partagés par les différents utilisateurs de l'appareil et les versions principales différentes des produits Altova (32-bit ou 64-bit).

Vous pouvez ajouter des dictionnaires pour le vérificateur orthographique de deux manières, aucune des deux ne nécessite que les fichiers soient enregistrés avec le système :

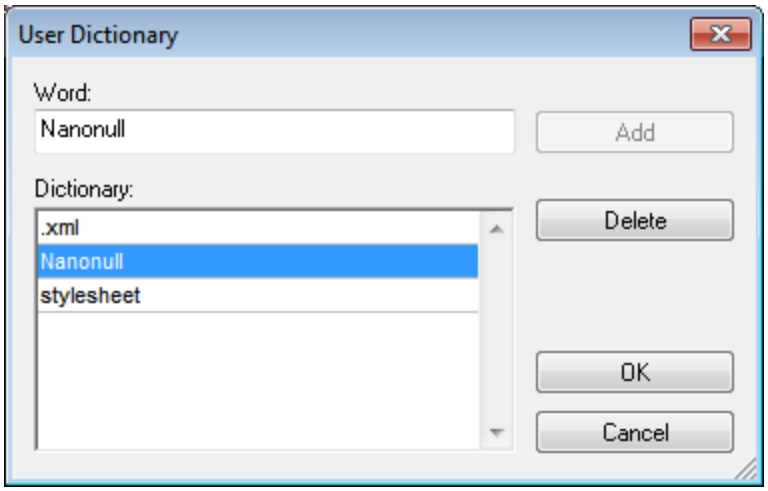
- En ajoutant les dictionnaires Hunspell dans un nouveau sous-dossier du dossier `Lexicons`. Les dictionnaires Hunspell peuvent être téléchargés, par exemple depuis http://wiki.services.openoffice.org/wiki/Dictionaries ou http://extensions.services.openoffice.org/en/dictionaries. (Veuillez noter qu'OpenOffice utilise le format zippé `OXT`. Il faut donc changer l'extension en `.zip` et dézipper le fichier `.aff` et `.dic` dans les dossiers de langue dans le dossier `Lexicons`. Veuillez aussi noter que les dictionnaires Hunspell sont basés sur les dictionnaires Myspell. Les dictionnaires Myspell peuvent donc aussi être utilisés.)
- En utilisant l'installateur de dictionnaire Altova, qui installe un pack de dictionnaires en plusieurs langues par défaut dans l'emplacement correct de votre appareil. L'installateur peut être téléchargé par le biais du lien dans le volet Langue de dictionnaire du dialogue Options du vérificateur orthographique (*voir capture d'écran ci-dessous*). L'installation des dictionnaires doit être effectuée avec des droits d'administrateur, sinon l'installation échouera avec une erreur.



Note : À vous de choisir si vous acceptez les termes de la licence applicables au dictionnaire et si le dictionnaire est approprié pour votre utilisation avec le logiciel sur votre ordinateur.

Travailler avec le dictionnaire d'utilisateur

Chaque utilisateur a un dictionnaire d'utilisateur dans lequel les mots autorisés par l'utilisateur peuvent être stockés. Pendant une vérification orthographique, l'orthographe est vérifiée par rapport à une liste de mots contenant les mots dans le dictionnaire de langue et le dictionnaire d'utilisateur. Vous pouvez ajouter et supprimer des mots du dictionnaire d'utilisateur par le biais du Dialogue Dictionnaire utilisateur (*capture d'écran ci-dessous*). Ce dialogue est accédé en cliquant sur la touche Dictionnaire utilisateur dans le dialogue Options du vérificateur d'orthographe (*voir seconde capture d'écran dans cette section*).



Pour ajouter un mot au dictionnaire utilisateur, saisir le mot dans le champ de saisie Mot et cliquer sur **Ajouter**. Le mot sera ajouté à la liste alphabétique dans le volet Dictionnaire. Pour supprimer un mot du dictionnaire, choisir le mot dans le volet Dictionnaire et cliquer sur **Supprimer**. Le mot sera supprimé du volet Dictionnaire. Lorsque vous avez terminé d'éditer le dialogue Dictionnaire utilisateur, cliquer sur **OK** pour que les changements soient enregistrés dans le dictionnaire utilisateur.

Les mots peuvent être ajoutés au Dictionnaire utilisateur pendant une vérification orthographique. Si un mot inconnu est rencontré pendant une vérification orthographique, le dialogue Orthographe s'ouvre et vous invite à choisir une action. Si vous cliquez sur la touche **Ajouter au dictionnaire**, le mot inconnu sera ajouté au dictionnaire utilisateur.

Le Dictionnaire utilisateur se trouve sous : `C:\Users\<user>\Documents\Altova\SpellChecker\Lexicons\user.dic`

28.17.3 Éditeur de script

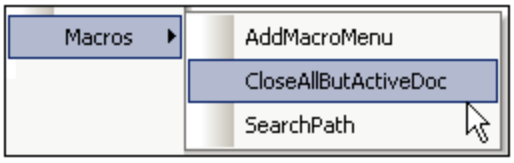
La commande **Éditeur de script** ouvre la fenêtre Éditeur de script. Le fonctionnement de l'Éditeur de script est décrit dans la section Script de cette documentation.

Note : La version de .NET Framework 2.0 ou plus doit être installée sur votre appareil pour que l'Éditeur de script puisse être exécuté.

© 2016-2022 Altova GmbH

28.17.4 Macros

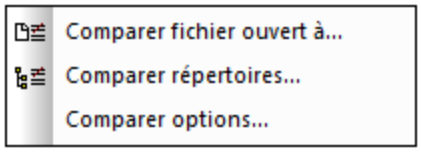
Passer avec la souris sur la commande **Macros** pour ouvrir un sous-menu contenant les macros définis dans le projet de script actif actuellement dans XMLSpy (*capture d'écran ci-dessous*).



Cliquer sur un macro dans le sous-menu (*voir capture d'écran ci-dessus*) pour exécuter le macro.

28.17.5 Comparaisons

XMLSpy offre une fonction de comparaison (ou différenciation) avec laquelle vous pouvez comparer des fichiers XML et Texte, ainsi que des dossiers, afin de contrôler la présence éventuelle de différences.



Ci-dessous, vous trouverez la liste des points du menu **Outils** permettant d'effectuer des tâches de comparaison sur les fichiers et les dossiers :

- Comparer fichier ouvert à
- Comparer répertoires
- Options de comparaison

Ces commandes sont décrites en détails dans les sous-sections suivantes.

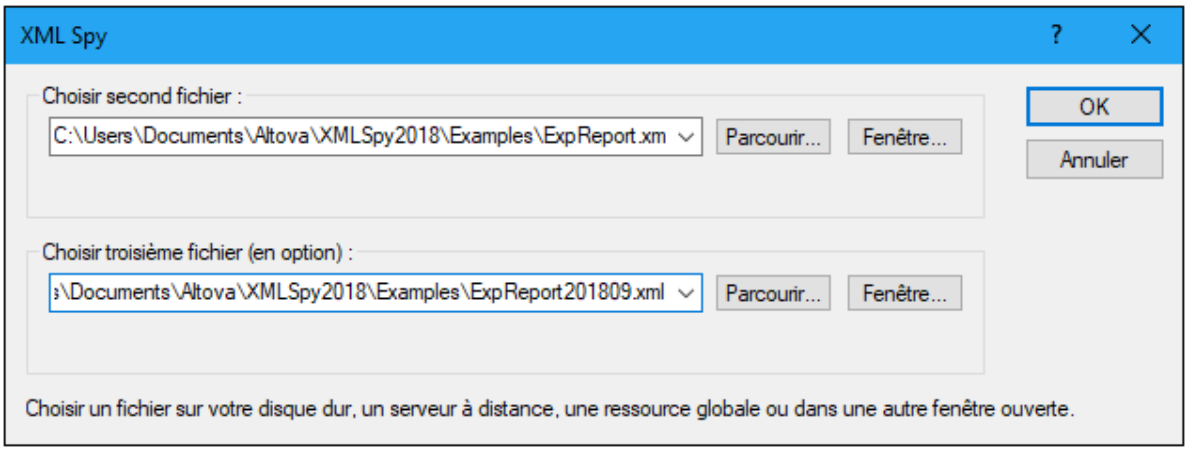
28.17.5.1 Comparer fichier ouvert avec

Cette commande vous permet de comparer le fichier ouvert avec un autre fichier. La comparaison montre les fichiers à comparer sous forme de mosaïque verticale dans la fenêtre principale et les différences sont marquées dans chaque fichier. Si deux fichiers présentent une différence, celle-ci sera marquée en vert. Si le contenu diffère dans les trois fichiers (*uniquement disponible dans Enterprise Edition*), cela entraîne un conflit, et celui-ci est marqué en rose. Vous pouvez comparer les fichiers en tant que documents XML (lorsque la structure et la sémantique des balises est significative) ou en tant que documents de texte.

Note : les comparaisons de fichier à trois voies sont disponibles dans l'**Enterprise Edition uniquement.**

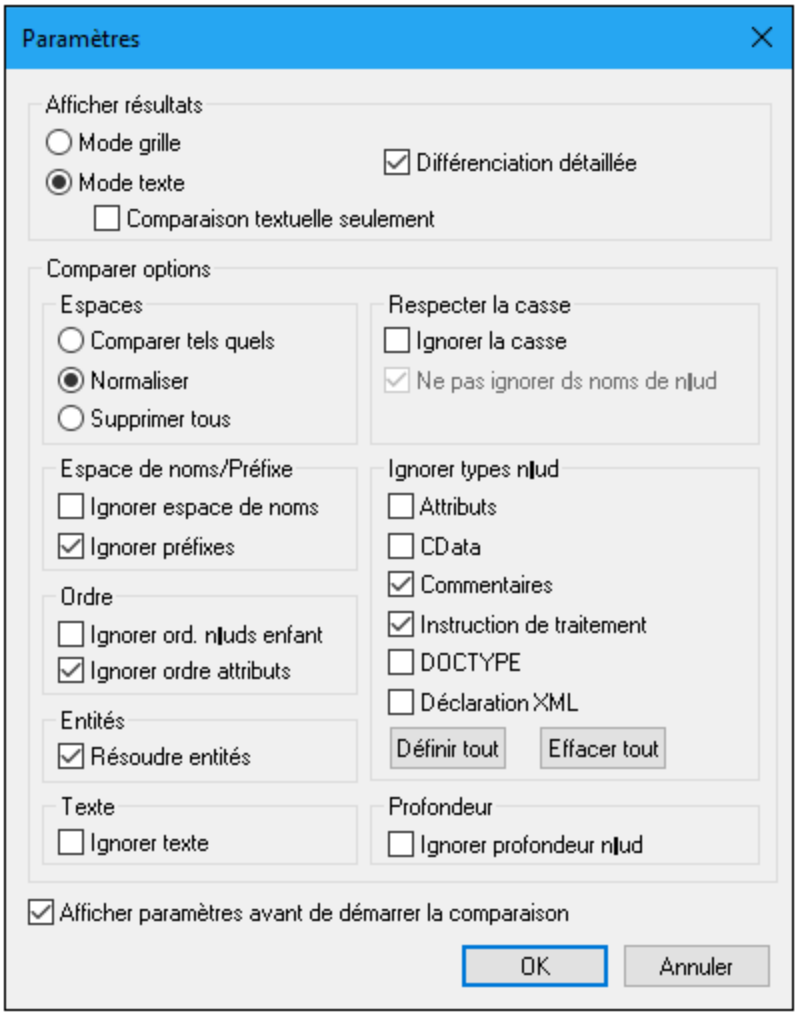
Pour comparer le fichier actif (dans la GUI) avec un autre fichier, procéder comme suit :

1. Le fichier à comparer doit être activé dans la Fenêtre principale (uniquement un fichier doit être actif à un moment donné), cliquer sur **Outils | Comparer fichier ouvert avec**. Un dialogue Parcourir apparaît (*capture d'écran ci-dessous*), dans lequel vous pouvez chercher un ou deux fichiers à comparer avec le fichier actif.



2. Cliquer sur **Parcourir** pour choisir un fichier par le biais de l'explorateur de fichiers, d'une ressource globale ou d'une URL. Cliquer sur **Fenêtre** pour choisir un fichier qui est ouvert dans une des fenêtres de XMLSpy.
3. Cliquer sur **OK**. Le dialogue Paramètres apparaît (*capture d'écran ci-dessous*). Ces paramètres sont décrits dans la section Options de comparaison. Si vous ne souhaitez pas voir apparaître ce dialogue à chaque fois que vous lancez une session de comparaison, décocher *Afficher paramètres avant de démarrer la comparaison* (situé en bas du dialogue).

© 2016-2022 Altova GmbH



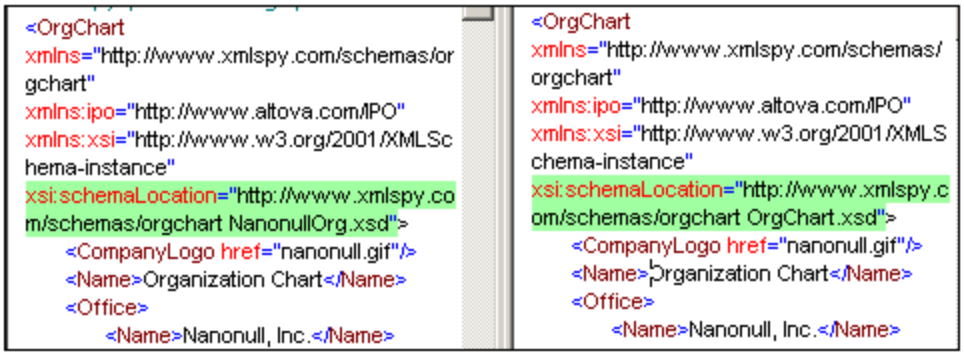
4. Choisir les paramètres requis, puis cliquer sur **OK**. Deux choses se produiront : (i) les fichiers à comparer sont affichés côte -à-côte dans des panneaux séparés ; (ii) la fenêtre de contrôle Comparer les fichiers apparaît. (*Captures d'écran dans la section suivante, Comparaison de fichiers.*)
5. Vous pouvez parcourir et fusionner les différences en utilisant les touches dans la fenêtre Comparer les fichiers. (*pour plus d'informations d'utilisation, voir la section suivante, Comparaison de fichiers.*)
6. Une fois avoir terminé de revoir et/ou de fusionner les différences, cliquez sur **Terminé**.

Note : La comparaison des paramètres peut être modifiée au cours d'une session de comparaison (en choisissant **Outils | Options de comparaison**), mais elle ne prendra effet qu'à partir de la session de Comparaison suivante ; les nouveaux paramètres n'influeront pas sur la session actuelle.

Comparaison de fichiers et fusion des différences

Il existe deux types de comparaison :

- *Comparaison à deux voies :* Deux fichiers sont comparés. Le fichier qui a été activé (lorsque la comparaison est requise ; appelé actif en abrégé) est affiché dans le volet de gauche et est comparé par rapport à un deuxième fichier *(voir capture d'écran ci-dessous)*.



- *Comparaison à trois voies (Enterprise Edition uniquement) :* Trois fichiers sont comparés. Le fichier actif est affiché dans le volet de gauche. Le premier des deux fichiers sélectionné pour la comparaison par rapport au fichier actif est affiché dans le volet central. Ce fichier central est nommé le fichier de base. Vous pouvez fusionner une différence entre le fichier de gauche et le fichier de base, ou une différence entre le fichier de droite et le fichier de base.

Note : Ne modifiez pas l'ordre des volets dans la fenêtre, sinon les directions Copier Vers des touches dans la fenêtre de contrôle Comparer les fichiers pourraient être incorrectes.

Différences et conflits

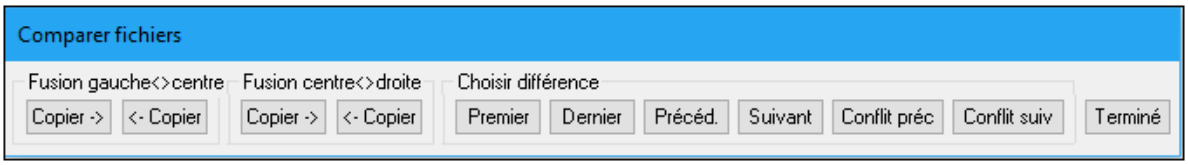
Les différences et les conflits sont marqués dans des couleurs différentes (*voir Marquer les couleurs ci-dessous*) et peuvent être recherchés séparément.

- *Différences :* lorsque le contenu dans des emplacements correspondants dans deux fichiers sont différents. Marquées en vert.
- *Conflits :* lorsque le contenu dans des emplacements correspondants dans trois fichiers sont différents. Marqués en rose.

Note : Les comparaisons à deux voies montrent uniquement les différences, pas les conflits.

Naviguer entre les différences et les conflits

Vous pouvez naviguer dans le document en utilisant les touches dans le volet Choisir la différence dans la fenêtre de contrôle Comparer les fichiers (*affiché ci-dessous : comparaison à deux voies, suivie d'une comparaison à trois voies*). Le panneau contient quatre touches pour une comparaison à deux voies, et six touches pour une comparaison à trois voies (*voir capture d'écran*).



La fenêtre de contrôler Comparer les fichiers contient les touches suivantes :

- *Premier :* se rend au premier des différences et des conflits dans le document.
- *Dernier :* se rend au dernier des différences et des conflits dans le document.
- *Précédent :* se rend à la différence précédente depuis la différence ou le conflit actif.

 © 2016-2022 Altova GmbH

- *Suivant :* se rend à la différence suivante depuis la différence ou le conflit actif.
- *Conflit précédent :* se rend au conflit précédent depuis la différence ou le conflit actif.
- *Conflit suivant :* se rend au conflit suivant depuis la différence ou le conflit actif.

Fusionner des différences et les conflits
Vous pouvez utiliser les touches dans le panneau *Fusionner* pour copier le contenu marqué d'un volet à l'autre. Utiliser la touche de *direction* de Copie appropriée d'une paire de panneau. Afin de permettre la fusion, il est nécessaire de définir les options de comparaison suivantes :

- *La différenciation détaillée* doit être cochée et
- *Ignorer profondeur nœud* doit être décochée.

Une fois que vous avez terminé de revoir et/ou de fusionner les différences, cliquer sur **Terminé**.

Note : Si vous souhaitez annuler une fusion, interrompre la session de Comparaison, sélectionner le fichier dans lequel le changement doit être annulé et cliquer sur **Édition | Annuler** ou appuyer sur **Ctrl+Z**.

Note : Lorsque la session de Comparaison est active, aucune édition ou modification des affichages n'est autorisée. Toute tentative d'édition ou de modification de l'affichage d'un des fichiers entraînera l'affichage d'un message vous informant que la session de Comparaison va se terminer.

Couleurs de marquage

Si des lignes correspondantes diffèrent dans deux fichiers, elles seront marquées en vert dans les deux fichiers (différences). Si des lignes diffèrent dans trois fichiers, elles seront marquées en rose dans les trois fichiers. Si une différence est sélectionnée, elle sera indiquée dans les deux fichiers dans un ton de vert plus foncé. Si un conflit est sélectionné, il sera indiqué dans les deux fichiers dans un ton de rose plus foncé.

Les couleurs marquées utilisées dans les sessions Comparer sont indiquées dans le tableau ci-dessous.

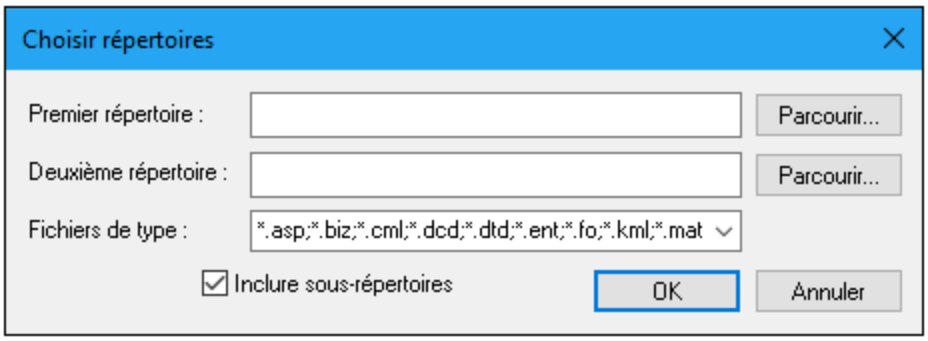
	Différence
	Différence actuelle
	Différence fusionnée
	Différence fusionnée actuelle
	Conflit
	Conflit actuel

28.17.5.2 Comparer répertoires

La commande **Comparer les répertoires** vous permet de comparer deux répertoires, avec ou sans leurs sous-répertoires. Les répertoires sont comparés pour indiquer des fichiers manquant, et pour vérifier que des fichiers portant le même nom sont différents ou pas.

Pour comparer deux répertoires :

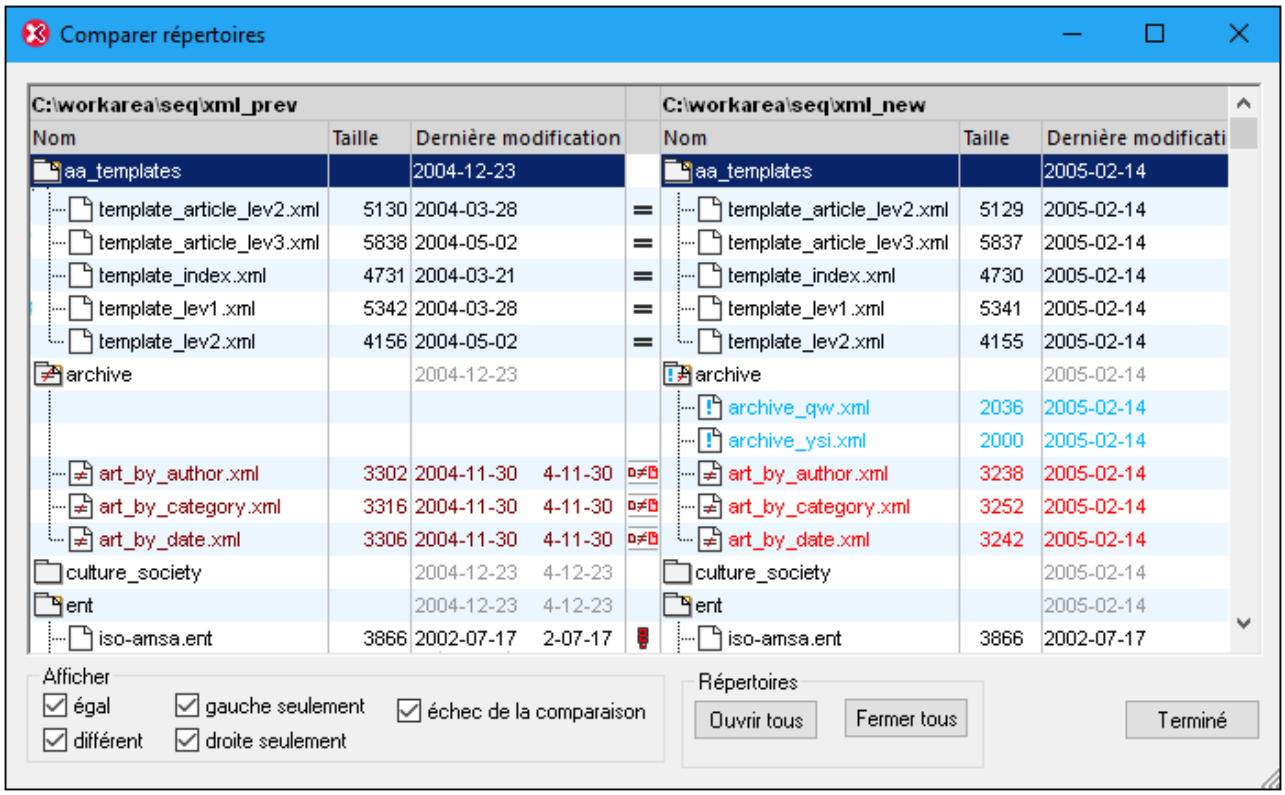
1. Cliquer sur **Outils | Comparer répertoires**. Le dialogue suivant apparaît.



2. Chercher les répertoires que vous souhaitez comparer et cocher la case **Inclure sous-répertoires** si vous souhaitez inclure les sous-répertoires à la comparaison.
3. Choisir les types de fichier que vous souhaitez comparer dans le champ **Fichiers de type**. Vous trouverez trois options dans la liste déroulante : (i) les fichiers de type XML ; (ii) les types de fichier définis dans XMLSpy; et (iii) tous les types de fichiers. Les archives Zip sont traitées en tant que répertoire, et les répertoires et les fichiers dans l'archive Zip est affichée dans la fenêtre de résultats. Afin de vous assurer que les archives Zip sont lues en tant que répertoires, veuillez vous assurer que l'extension nécessaire est inclue dans l'entrée Fichier de Type.
4. Cliquer sur **OK**. Le dialogue Paramètres (décrit dans les Options de comparaison) apparaît.
5. Choisir les paramètres requis pour comparer les fichiers.
6. Cliquer sur **OK**. Un dialogue s'affichera indiquant la progression de la comparaison.

Le résultat apparaîtra dans une fenêtre, et ressemblera à l'image ci-dessous :

© 2016-2022 Altova GmbH



Symboles de répertoire

Tous les noms des répertoires sont indiqués en noir.

Le répertoire est réduit et son contenu n'est pas affiché

Le répertoire est agrandi, ce qui est indiqué par le coin plié. Le contenu est affiché.

Le répertoire contient des fichiers ; ceux-ci soit ne peuvent pas être comparés soit ne sont pas différents du fichier correspondant dans le répertoire comparé.

Le répertoire contient un ou plusieurs fichiers qui n'existent pas dans le répertoire comparé.

Le répertoire contient un ou plusieurs fichiers qui sont différents du fichier correspondant dans le répertoire comparé.

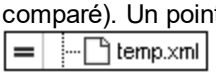
Le répertoire contient un ou plusieurs fichiers qui n'existent pas dans le répertoire comparé et un ou plusieurs fichiers qui sont différents du fichier correspondant dans le répertoire comparé.

Symboles de fichier

Les couleurs dans lesquelles les noms de fichier apparaissent dépendent de leur statut de comparaison. Les couleurs sont indiquées ci-dessous.



Ce fichier ne peut pas être comparé (avec le fichier correspondant dans le répertoire comparé). Un point d'interrogation apparaît dans la colonne centrale. Le nom du fichier apparaît en noir.



Ce fichier n'est pas différent du fichier correspondant dans le répertoire comparé. Un signe = (égal à) apparaît dans la colonne centrale. Le nom du fichier apparaît en noir.



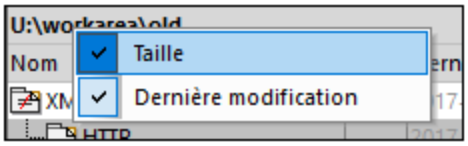
Ce fichier n'existe pas dans le répertoire comparé. La colonne centrale est vide. Le nom du fichier apparaît en bleu.



Ce fichier est différent du fichier correspondant dans le répertoire comparé, et la dernière modification de ce fichier est plus récent que la dernière modification du fichier correspondant. Le fichier le plus récent apparaît dans un rouge plus vif, et l'icône montre le fichier en rouge vif sur le côté avec le fichier le plus récent.

Options d'affichage

- Choisir les fichiers que vous souhaitez afficher en cochant ou en décochant les options dans le volet Afficher situé en bas de la fenêtre Résultat.
- Ouvrir ou fermer tous les sous-répertoires en cliquant sur la touche appropriée dans le volet Répertoires.
- Agrandir ou réduire les sous-répertoires en double-cliquant l'icône Dossier.
- Les options Taille et Dernière modification peuvent être activées et désactivées en cliquant avec la touche de droite sur la barre de titre du fichier et en cliquant sur Taille et Dernière modification.



- Modifier la largeur des colonnes en les glissant.
- La fenêtre Résultats peut être maximisée, minimisée et redimensionnée.

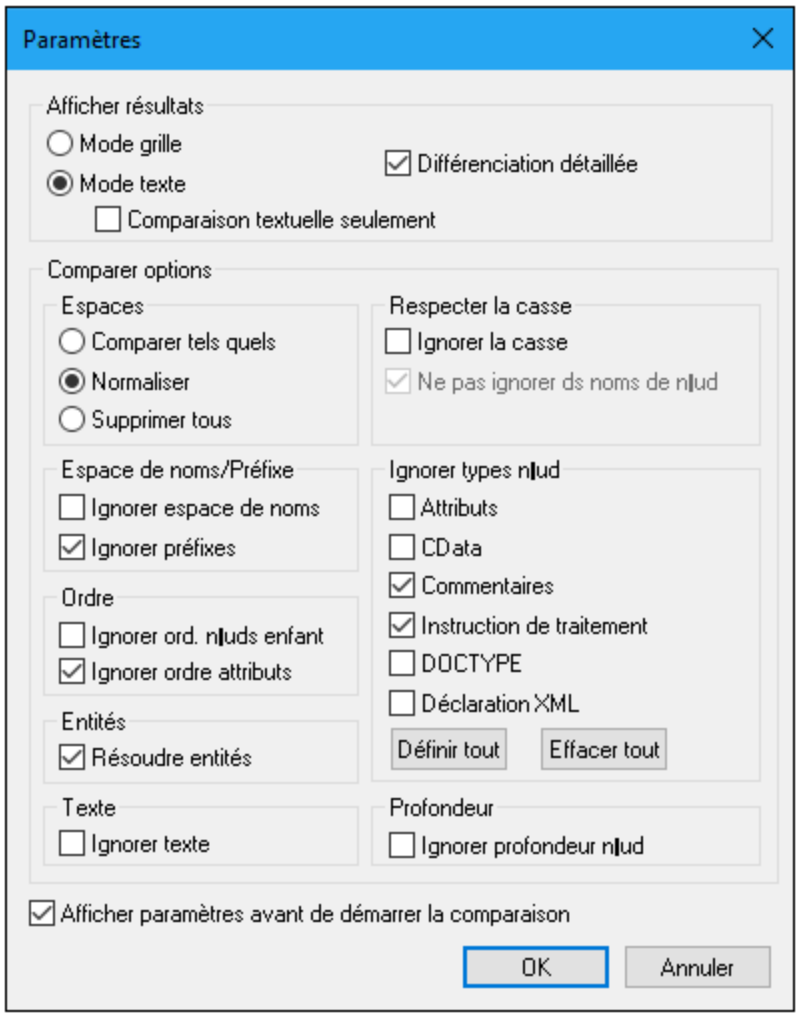
Comparer et fusionner les fichiers

Double-cliquer sur une ligne pour ouvrir les deux fichiers sur cette ligne dans la Fenêtre principale et démarrer directement une Comparaison de fichier pour deux fichiers. Vous pouvez ensuite continuer comme dans une session de Comparaison normale (voir Comparer fichier ouvert avec...).

28.17.5.3 Options de comparaison

Cliquer sur **Outils** | **Options de comparaison** pour ouvrir le dialogue Paramètres (*voir capture d'écran*). Vous pourrez y définir les paramètres pour vos sessions de Comparaison. Les paramètres en vigueur lors du démarrage d'une session de Comparaison sont les paramètres qui sont appliqués à cette session de Comparaison.

© 2016-2022 Altova GmbH



Voir les résultats

Choisir le mode dans lequel les résultats sont affichés. Vous pouvez les choisir à partir des options suivantes :

- Grille (comparaison XML)
- Mode Texte avec Comparaison textuelle seulement décochée (comparaison XML)
- Mode Texte Comparaison textuelle seulement cochée (comparaison Texte)

Si un mode qui propose une comparaison XML a été sélectionné, les documents sont traités en tant que documents XML et les Options de Comparaison XML sont activées. Si la comparaison de Texte a été sélectionnée, seule les Options Comparer valides pour la Comparaison de Texte (Espaces blancs et Sensibilité à la casse) sont activées ; toutes les autres Options de Comparaison sont désactivées.

Note : Vous pouvez fusionner des différences dans les Modes Grille et Texte et dans les modes de comparaison XML et Texte. Si vous souhaitez annuler une fusion, arrêter la session de Comparaison, sélectionner le fichier dans lequel la modification est effectuée et sélectionner **Édition | Annuler** ou appuyer sur **Ctrl + Z**.

Différenciation détaillée

Si cette option est décochée, les différences apparaissant dans les éléments frères immédiats sont représentées en tant qu'une seule différence, et l'option de fusion est désactivée. Une fois sélectionnées, les différences dans les frères immédiats sont représentées en tant que différences séparées et la fusion est activée.

Note : la case **Différenciation détaillée** doit être cochée pour permettre la fusion.

Espaces blancs

Les caractères d'espace blanc sont : espace, tabulateur, retour chariot et saut de ligne. Lorsque l'espace blanc est normalisé, les caractères d'espace blanc consécutifs sont réduits à un caractère d'espace blanc ; néanmoins, veuillez noter que, conformément à la spécification XML, les espaces blancs de début et de fin dans les valeurs d'attribut sont entièrement supprimés lorsque l'espace blanc est normalisé. Les options ici comparent les fichiers avec : (i) espace blanc inchangé ; (ii) espace blanc normalisé ; et (iii) tous les espaces blancs retirés. L'option Espaces blancs est disponible aussi bien pour les comparaisons XML que Texte.

Sensibilité à la casse

Si la case **Ignorer la casse** est cochée, vous avez la possibilité d'ignorer ou pas la casse dans les noms de nœud (uniquement pour les comparaisons XML). L'option Sensibilité à la casse est disponible aussi bien pour les comparaisons XML que Texte.

Espace de noms / Préfixe

Il s'agit des options pour ignorer les espaces de noms et les préfixes lorsque vous recherchez des différences.

Ordre

Si la case **Ignore ordre des nœuds enfants** est cochée, la position des nœuds enfants entre eux n'a pas d'importance. La comparaison est effectuée pour l'ensembles des nœuds enfant, et si la seule différence entre un nœud enfant dans un document et un nœud enfant dans le document de comparaison est la position relative dans l'ensemble de nœud, alors cette différence sera ignorée. Chaque nœud d'élément enfant est identifiée par son nom, ses attributs, et sa position. Si les noms de plus d'un nœud dans l'ensemble de frère sont les mêmes, alors la position de ces nœuds est utilisée pour identifier les nœud même si l'option "Ignorer l'ordre des nœuds enfants" a été cochée. Cette option s'applique à chaque niveau séparément. Si la case **Ignore ordre des nœuds enfants** est décochée, les différences de l'ordre sont représentées en tant que différences.

L'option d'ignorer l'**ordre des attributs** est aussi disponible et s'applique à l'ordre des attributs dans un seul élément.

Entités

Si "Résoudre entités" est sélectionné, toutes les entités dans le document sont résolues. Sinon les fichiers sont comparés avec les entités telles qu'elles sont.

Texte

Si "Ignorer texte" est sélectionné, les différences dans les nœuds de texte correspondant ne sont pas rapportées.

© 2016-2022 Altova GmbH

Ignorer types de nœuds

Cocher les types de nœud qui ne seront pas comparés dans la session de Comparaison. Les types de nœud qui peuvent être ignorés sont les Attributs, CDATA, les Commentaires, les instructions de traitement, les instructions DOCTYPE et la déclaration XML.

Profondeur

Si **Ignorer profondeur de nœud** est cochée, la profondeur supplémentaire de tout élément (c.à.d. plus de niveaux des descendants) relative à la profondeur de l'élément correspondant dans le fichier comparé est ignorée. Cette option doit être décochée pour permettre la fusion.

Afficher les paramètres avant de démarrer la comparaison

Cocher cette option pour faire apparaître le dialogue Paramètres (ce dialogue) avant que chaque comparaison de fichier ou de répertoire soit effectuée (par le biais des commandes **Comparer fichier ouvert avec** et **Comparer répertoires**). Le fait de faire apparaître le dialogue Paramètres avant chaque comparaison vous permet de cocher et de vérifier les paramètres pour chaque comparaison.

Si cette commande est décochée, la session de Comparaison démarrera immédiatement lorsqu'une comparaison est invoquée.

28.17.6 Outils définis par l'utilisateur

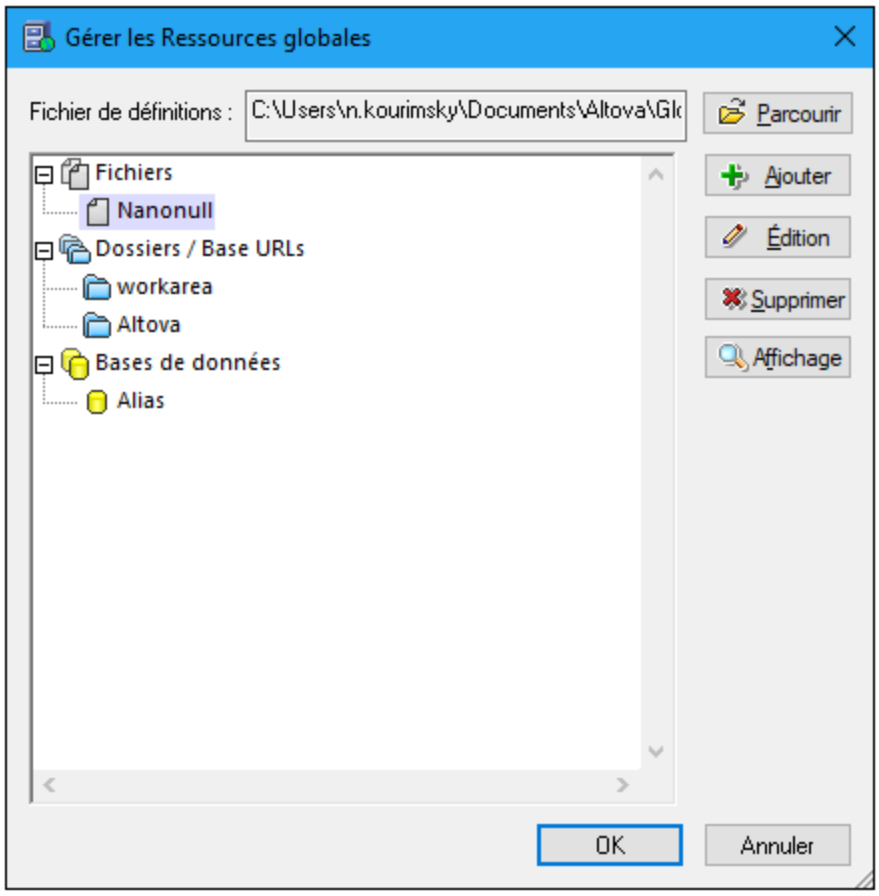
Placer le curseur sur la commande **Outils définis par l'utilisateur** pour ouvrir un sous-menu contenant des commandes personnalisées qui utilisent des applications externes. Vous pouvez créer ces commandes dans l'onglet Outils du dialogue Personnaliser. Cliquer sur une de ces commandes personnalisées pour exécuter l'action associée avec cette commande.

La commande **Outils définis par l'utilisateur | Personnaliser** ouvre l'onglet Outils du dialogue Personnaliser (dans lequel vous pouvez créer les commandes personnalisées qui apparaissent dans le menu de la commande **Outils définis par l'utilisateur**.)

28.17.7 Ressources globales

La commande **Ressources globales** ouvre le dialogue Ressources Globales (*capture d'écran ci-dessous*), dans lequel vous pouvez :

- Spécifier le Fichier XML de Ressources globales à utiliser pour les ressources globales.
- Ajouter un fichier, dossier et ressources globales de base de données (ou alias)
- Spécifier des configurations variées pour chaque ressource globale (alias). Chaque configuration mappe vers une ressource spécifique.

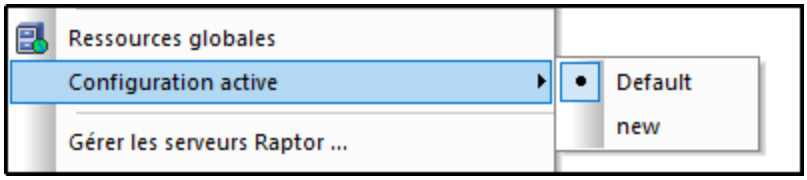


Vous trouverez une description détaillée de la définition d'une ressource globale dans la section Définir des Ressources globales.

Note: le dialogue Ressources globales Altova peut aussi être accédé par le biais de la barre outils Ressources globales (**Outils | Personnaliser | Barres outils | Ressources globales**).

28.17.8 Configuration Active

Passer avec la souris sur l'item de menu **Configuration Active** pour dérouler un sous-menu contenant toutes les configurations définies dans le fichier XML Ressources Globales actuellement actif (*capture d'écran ci-dessous*).



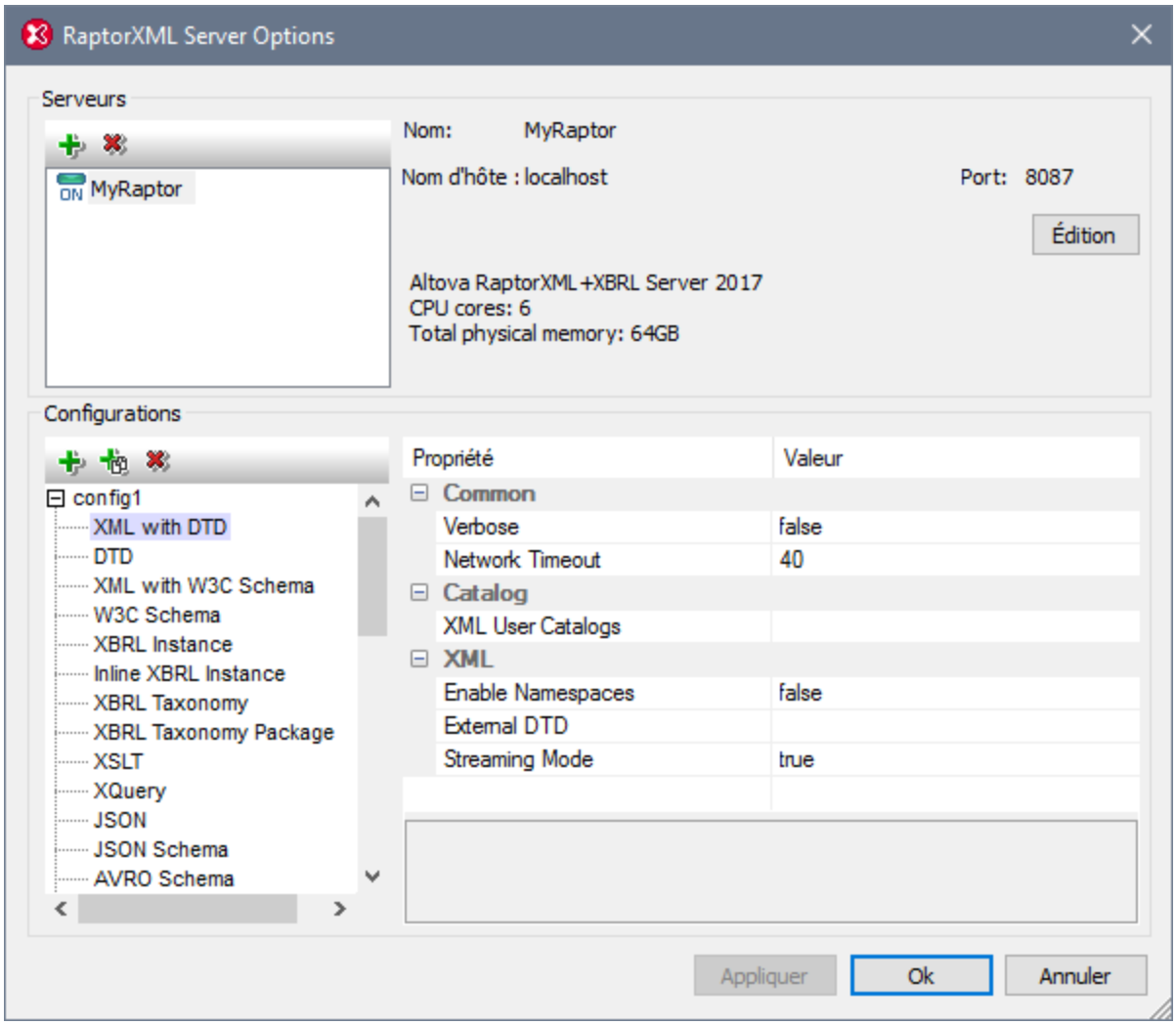
© 2016-2022 Altova GmbH

La configuration actuellement active est indiquée par une puce. Dans la capture d'écran ci-dessus, la configuration actuellement active est `Default`. Pour la modifier, choisir la configuration que vous souhaitez rendre active.

Note: La configuration active peut aussi être sélectionnée par le biais de la barre d'outils Ressources globales (**Outils | Personnaliser | Barres d'outils | Ressources globales**).

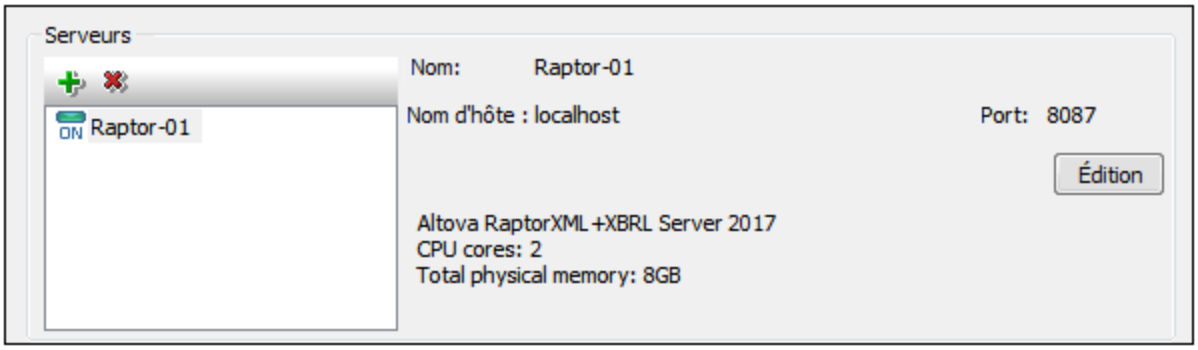
28.17.9 Gérer les serveurs Raptor

La commande **Gérer les serveurs Raptor** vous permet d'ajouter plusieurs serveurs Raptor dans le pool de serveurs Raptor disponibles puis de définir plusieurs configurations pour chaque serveur. Pour un aperçu de l'utilisation du serveur RaptorXML pour la validation de document XML et XBRL, voir le sujet Valider avec RaptorXML Server.



Ajouter un serveur Raptor

Dans le volet *Serveurs* du panneau (*capture d'écran ci-dessous*), cliquer sur l'icône **Ajouter serveur**, puis saisir le nom avec lequel vous souhaitez identifier le serveur Raptor, le nom du réseau de la machine sur lequel Raptor est installé (nom d'hôte), et le port du serveur Raptor. Cliquer sur **OK** pour enregistrer les paramètres.



- *Nom :* une chaîne de votre choix. Elle est utilisée dans XMLSpy pour identifier un RaptorXML Server particulier.
- *Nom de l'hôte :* le nom ou l'adresse IP de la machine de réseau sur laquelle le Raptor server est installé. Le traitement sera plus rapide si vous utilisez une adresse IP au lieu d'un nom d'hôte. L'adresse IP correspondant à `localhost` (la machine locale) est `127.0.0.1`.
- *Port :* le port permettant d'accéder au serveur Raptor. Ce port est spécifié dans le fichier de configuration de Raptor (appelé `server_config.xml`). Le port doit être fixé et connu de manière à ce que les requêtes puissent être adressées correctement au service. Pour plus d'informations concernant le fichier de configuration Raptor, voir les manuels d'utilisateur : *RaptorXML Server* et *RaptorXML+XBRL Server*.

Une fois avoir saisi les informations de serveur, cliquer sur **OK**. Le nom de serveur que vous avez saisi apparaît dans la liste de serveur (à gauche du volet). Une icône verte apparaît à côté du nom du serveur, indiquant que le serveur Raptor a été lancé et est en marche. Les détails du serveur sont affichés dans le volet (*capture d'écran ci-dessus*). Une icône rouge indique que le serveur est hors ligne. Si le serveur ne peut pas être trouvé, un message d'erreur s'affiche.

Note : le serveur Raptor doit être en marche lorsque le serveur est ajouté. Cela est nécessaire pour que XMLSpy puisse obtenir des informations concernant le serveur et les stocker. Si, une fois que le serveur a été ajouté, le serveur est hors ligne ou ne peut pas être trouvé, ces situations sont indiquées, respectivement, par une icône rouge ou un message d'erreur.

Pour éditer un nom de serveur, un nom d'hôte ou un port, sélectionner le serveur dans le volet de gauche, cliquer sur la touche **Édition**, et, dans le dialogue qui apparaît, éditer les informations que vous souhaitez modifier. Pour supprimer un serveur du pool, sélectionner le serveur et cliquer sur l'icône **Supprimer le serveur sélectionné**.

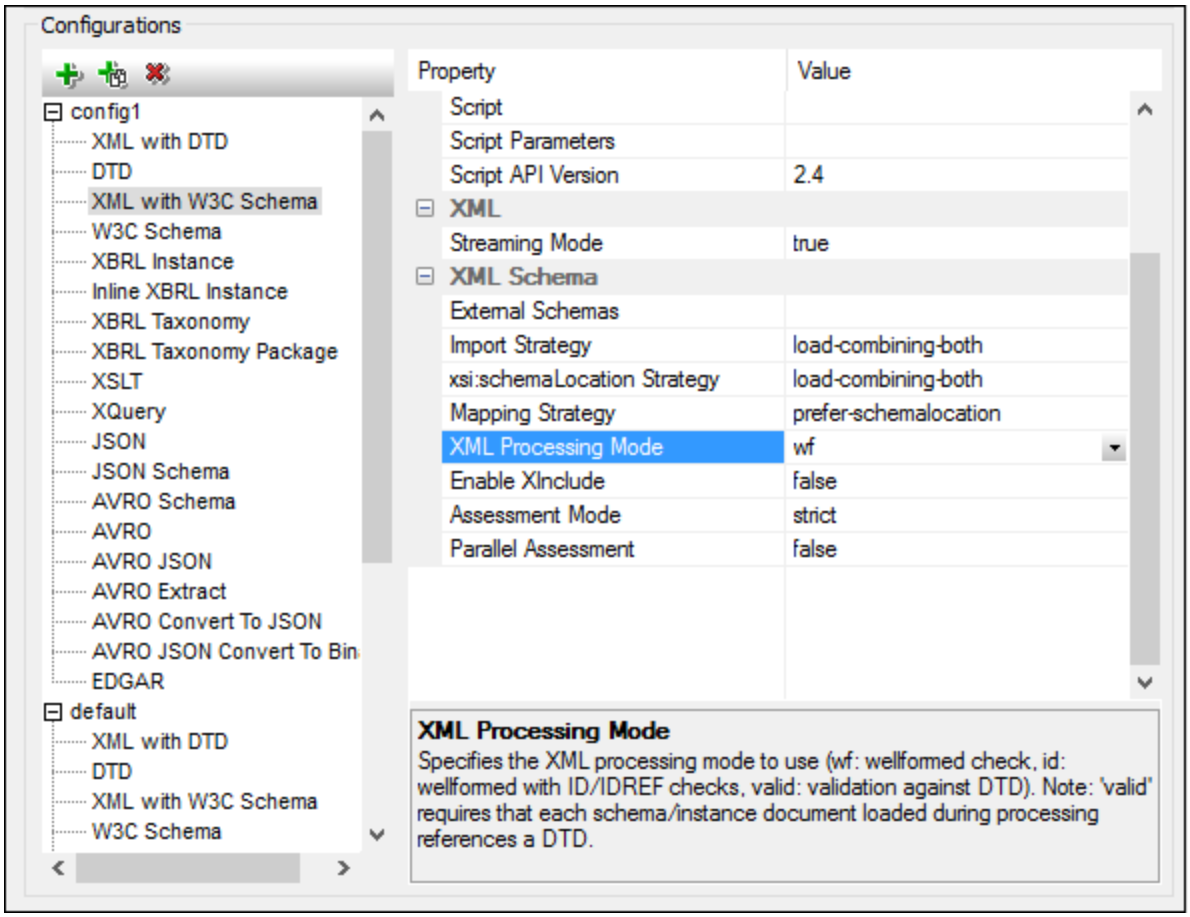
Server Configurations

Une configuration est un ensemble d'options de validation RaptorXML. Lorsqu'un serveur est ajouté, il sera doté d'une configuration nommée `default`. Il s'agit d'un ensemble d'options RaptorXML configurées sur leurs valeurs par défaut. Vous pouvez ajouter de nouvelles configurations qui contiennent d'autres valeurs d'options. Une fois avoir défini plusieurs configurations de serveur, vous pouvez sélectionner une configuration qui sera la

© 2016-2022 Altova GmbH

configuration active. Il s'agit de la configuration qui sera utilisée lorsque la commande **Valider sur le serveur** sera exécutée.

Le volet *Configurations* présente deux parties : (i) un volet de gauche, qui montre les configurations et les types, chacune contenant une liste des document-type pouvant être validés ; (ii) un panneau de droite qui affiche les options de validation pour le document-type sélectionné dans le panneau de gauche ; en bas du volet de droite, vous trouverez une description de l'option sélectionnée (*capture d'écran ci-dessous*).



Ajouter une configuration
Dans le panneau *Configurations* du dialogue Options de RaptorXML Server (*capture d'écran ci-dessus*), cliquer sur **Ajouter une configuration**. Une nouvelle configuration est ajoutée avec des valeurs d'option par défaut. Vous pouvez aussi créer une nouvelle configuration en cliquant sur **Copier la configuration sélectionnée**. Cela permet de créer une nouvelle configuration avec des valeurs d'option qui sont les mêmes que celle de la configuration copiée. Les nouvelles configurations sont créées avec les noms par défaut du type `config<X>`; vous pouvez éditer le nom d'une configuration en double-cliquant dessus et en saisissant le nouveau nom. Vous pouvez ensuite éditer une des valeurs d'option de la configuration.

Éditer les valeurs d'options d'une configuration
Tout d'abord, sélectionner le document-type dans le volet de gauche. Cela permet d'afficher les options de validation du document-type sélectionné dans le volet de droite. Pour éditer la valeur d'une option, choisir une des possibilités suivantes (dépendant du type de la valeur d'option) :

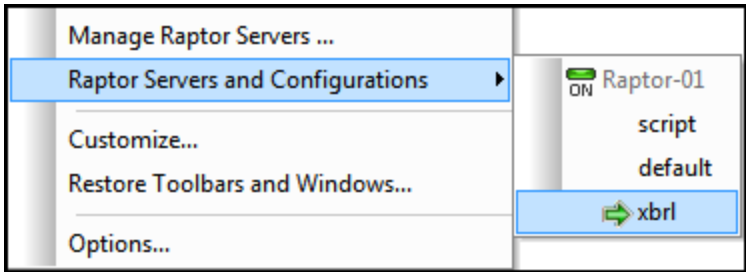
- Si la valeur peut être une dans un ensemble de valeurs prédéfinies, sélectionner la valeur que vous souhaitez depuis la liste de choix de la colonne de valeur de cette option.
- Si la valeur n'est pas limitée, cliquer dans la valeur d'option enregistrée et saisir la valeur que vous souhaitez.
- Si la valeur est un chemin de fichier, outre la possibilité de saisir la valeur, vous pouvez aussi parcourir le fichier que vous souhaitez en utilisant la touche **Parcourir** dans la colonne de valeur de l'option.

Si vous sélectionnez une option, sa description est affichée dans une fenêtre en bas du panneau de droite. Pour des descriptions plus détaillées de chaque option, voir les chapitres de l'interface de ligne de commande des manuels RaptorXML Server et RaptorXML(+XBRL) Server.

Supprimer une configuration
Dans le volet de gauche, choisir la configuration à supprimer et cliquer sur **Supprimer la configuration sélectionnée**.

28.17.10 Serveurs Raptor et Configurations

Si vous avez défini plusieurs configurations sur plusieurs serveurs, vous pouvez sélectionner un serveur et une de ses configurations en tant que la configuration active. Celle-ci sera utilisée pour les validations suivantes. En plaçant le curseur sur la commande **Outils | Serveurs Raptor et Configurations** (*capture d'écran ci-dessous*), un sous-menu apparaît qui contient tous les serveurs ajoutés, avec la configuration de chacun d'entre eux. Sélectionner la configuration de serveur que vous souhaitez pour rendre la configuration active. Dans la capture d'écran ci-dessous, la configuration `xbrl` du serveur nommé `Raptor-01` a été sélectionnée en tant que la configuration active (indiquée par la flèche verte).



Voir la section RaptorXML(+XBRL) Server pour découvrir comment utiliser Raptor depuis XMLSpy.

28.17.11 Gestionnaire de taxonomie

Cette commande ouvre le dialogue Gestionnaire de taxonomie qui décrit comment gérer vos packages de taxonomie XBRL.

Pour installer une taxonomie XBRL, cochez la case située à côté de la taxonomie que vous souhaitez installer et cliquer sur **Appliquer**. Vous pouvez désinstaller des taxonomies, mettre à niveau des taxonomies, recherchez de nouvelles taxonomies et gérer généralement toutes vos taxonomie dans un emplacement central.

© 2016-2022 Altova GmbH

Pour plus d'informations, voir Gestionnaire de taxonomie.

28.17.12 XML Gestionnaire de schéma

Cette commande ouvre le dialogue Gestionnaire de schéma qui vous permet de gérer vos packages de schéma XML.

Pour installer un schéma XML, cochez la case située à côté du schéma que vous souhaitez installer et cliquez sur **Appliquer**. Vous pouvez désinstaller des taxonomies, mettre à niveau des Schémas, recherchez de nouveaux schémas et gérer généralement toutes vos schémas dans un emplacement central.

Pour plus d'informations, voir Gestionnaire de schéma.

28.17.13 Personnaliser

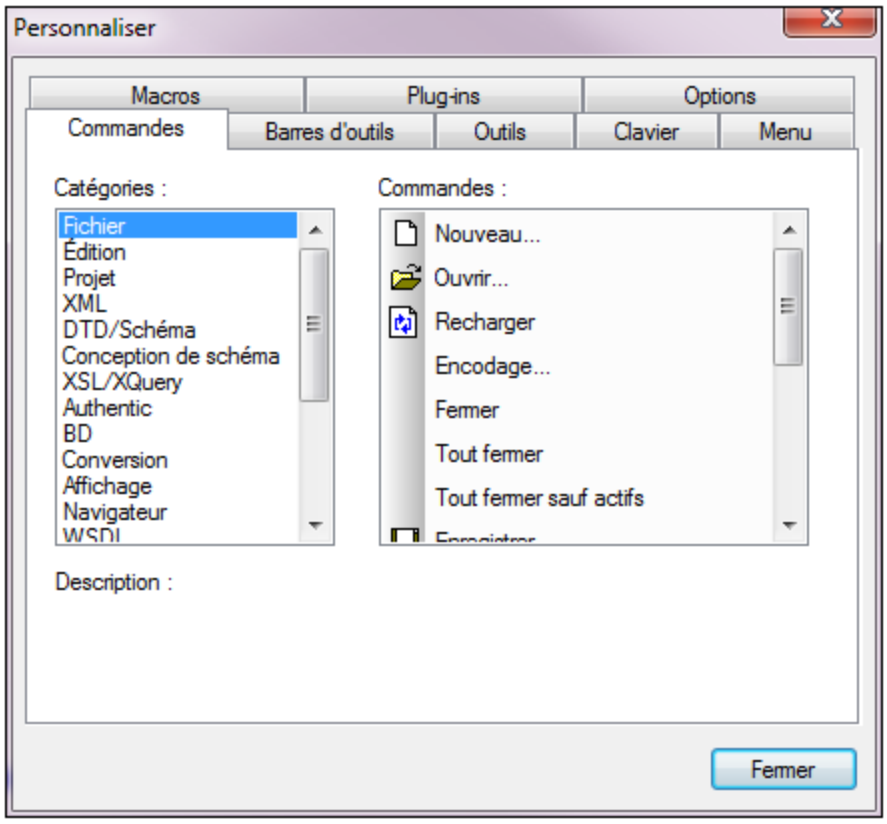
La commande **Personnaliser** vous permet de personnaliser les menus d'application et les barres d'outils selon vos besoins. Cliquer sur la commande pour ouvrir le dialogue Personnaliser qui comporte les onglets suivants :

- Commandes : toutes les applications et les commandes macro peuvent être glissées depuis cet onglet dans les barres de menu, les menus et les barres d'outils.
- Barres d'outils : les barres d'outils peuvent être activées, désactivées et réinitialisées individuellement.
- Outils : les commandes qui ouvrent les programmes externes depuis l'interface peuvent être ajoutés à l'interface.
- Clavier : les raccourcis de clavier peuvent être créés pour les applications individuelles et les commandes macro.
- Menu : les barres de menu et les menus contextuels à personnaliser sont sélectionnés et rendus actifs dans cet onglet. Travaille en collaboration avec l'onglet Commandes.
- Macros : les macros peuvent être associés à de nouvelles commandes.
- Plug-ins : les plug-ins peuvent être activés et intégrés dans l'interface.
- Options : l'affichage des options pour les barres d'outils est défini dans cet onglet.

Cette section décrit aussi le menu contextuel qui apparaît lorsque le dialogue Personnaliser est ouvert et que vous cliquez avec la touche de droite sur la barre de menu, le menu ou les items de barre d'outils.

28.17.13.1 Commandes

L'onglet **Commandes** vous permet de personnaliser vos menus et barres d'outils. Vous pouvez ajouter des commandes d'application aux menus et barres d'outils selon vos préférences. Veuillez noter, néanmoins que vous ne pouvez pas vous-même créer de nouvelles commandes d'application ou de menus.



Pour ajouter une commande à une barre d'outils ou un menu :

1. Sélectionner l'item de menu **Outils | Personnaliser**. Le dialogue Personnaliser apparaît.
2. Sélectionner la catégorie **Commandes** dans la fenêtre de liste *Catégories*. Les commandes disponibles apparaissent dans la liste *Commandes*.
3. Cliquer sur une commande dans la liste *Commandes* et la glisser dans un menu existant ou dans une barre d'outils. Une barre **I** apparaît lorsque vous placez le curseur sur une position valide pour déposer la commande.
4. Relâcher le bouton de la souris à l'endroit où vous souhaitez insérer la commande.

Veuillez noter les points suivants.

- Lorsque vous glissez une commande, un petit bouton apparaît à la pointe du pointeur de la souris : cela indique que la commande est en train d'être glissée.
- Un "**x**" se trouvant en-dessous du pointeur indique que la commande ne peut pas être déposée à l'endroit actuel du curseur.
- Si le curseur est déplacé dans une position à laquelle la commande peut être déposée (une barre d'outils ou un menu), le "**x**" disparaît et une barre **I** indique la position valide.
- Les commandes peuvent être placées dans des menus ou dans des barres d'outils. Si vous avez créé votre propre barre d'outils, vous pouvez utiliser ce mécanisme de personnalisation pour la remplir.
- Déplacer le curseur au-dessus d'un menu fermé pour ouvrir ce menu, vous permettant d'insérer la commande n'importe où dans ce menu.

© 2016-2022 Altova GmbH

Ajouter les commandes aux menus contextuels

Vous pouvez aussi ajouter les commandes aux menus contextuels en glissant les commandes depuis la liste *Commandes* dans le menu contextuel. Suivre la procédure suivante :

1. Dans le dialogue Personnaliser, cliquer sur l'**onglet Menu**.
2. Dans le volet Menu Contextuel, sélectionner un menu contextuel depuis la liste de choix. Le menu contextuel s'ouvre.
3. Dans le dialogue Personnaliser, retourner à l'onglet Commandes.
4. Glisser la commande que vous souhaitez créer depuis la liste *Commandes* et la déposer dans l'emplacement souhaité dans le menu contextuel.

Supprimer une commande ou un menu

Pour supprimer une commande d'un menu, d'un menu contextuel (voir ci-dessus pour plus de détails concernant l'accès aux menus contextuels), ou d'une barre d'outils ou pour supprimer un menu entier, procéder comme suit :

1. Ouvrir le dialogue Personnaliser (**Outils | Personnaliser**). Le dialogue Personnaliser apparaît.
2. Une fois le dialogue Personnaliser ouvert (et un onglet sélectionné), cliquer avec la touche de droite sur un menu ou une commande de menu, et puis sélectionner **Supprimer** depuis le menu contextuel qui s'ouvre. En alternative, glisser le menu ou la commande de menu jusqu'à ce qu'une icône "**x**" apparaisse en-dessous du pointeur de la souris, et puis déposer le menu ou la commande de menu. Le menu ou la commande de menu sera supprimée.

Pour réinstaller les commandes de menu, utiliser les mécanismes décrits dans cette section. Pour réinstaller un menu supprimé, aller sur **Outils | Personnaliser | Menu**, et cliquer sur la touche **Réinitialiser** dans le volet *Menus cadre de l'application*. En alternative, se rendre sur **Outils | Personnaliser | Barres d'outils**, choisir la Barre Menu et cliquer sur la touche **Réinitialiser**.

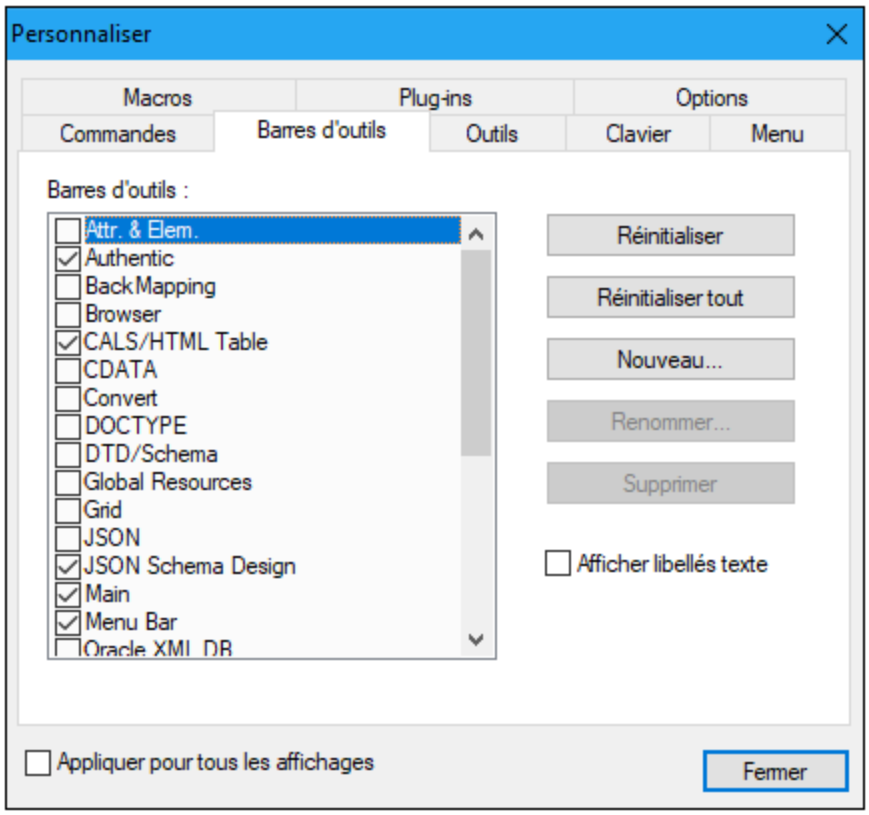
28.17.13.2 Barres d'outils

L'onglet **Barres d'outils** vous permet : (i) d'activer ou de désactiver des barres d'outils spécifiques (donc de décider quelles barres afficher dans l'interface); (ii) de définir quelles icônes sont affichées dans chaque barre d'outils et (iii) de créer vos propres barres d'outils spécialisées.

Les barres d'outils contiennent des icônes pour les commandes de menu les plus fréquemment utilisées. L'information concernant chaque icône est affichée dans une infobulle et dans la Barre de statut lorsque le curseur est placé sur l'icône. Vous pouvez glisser une barre d'outils sur n'importe quel endroit dans l'écran où il apparaît comme une fenêtre flottante.

Note : pour ajouter une commande à une barre d'outils, glisser la commande que vous souhaitez depuis la liste *Commandes* dans l'onglet **Commandes** vers la barre d'outils. Pour supprimer une commande depuis une barre d'outils, ouvrir le dialogue Personnaliser, et avec chaque onglet sélectionné, glisser la commande hors de la barre d'outils (voir **Commandes** pour plus de détails).

Note : les paramètres de barre d'outils définis dans un mode particulier sont, par défaut, valides pour ce mode uniquement. Pour appliquer les paramètres à tous les modes, cliquer sur la case à cocher en bas du dialogue.



Les fonctions suivantes sont disponibles :

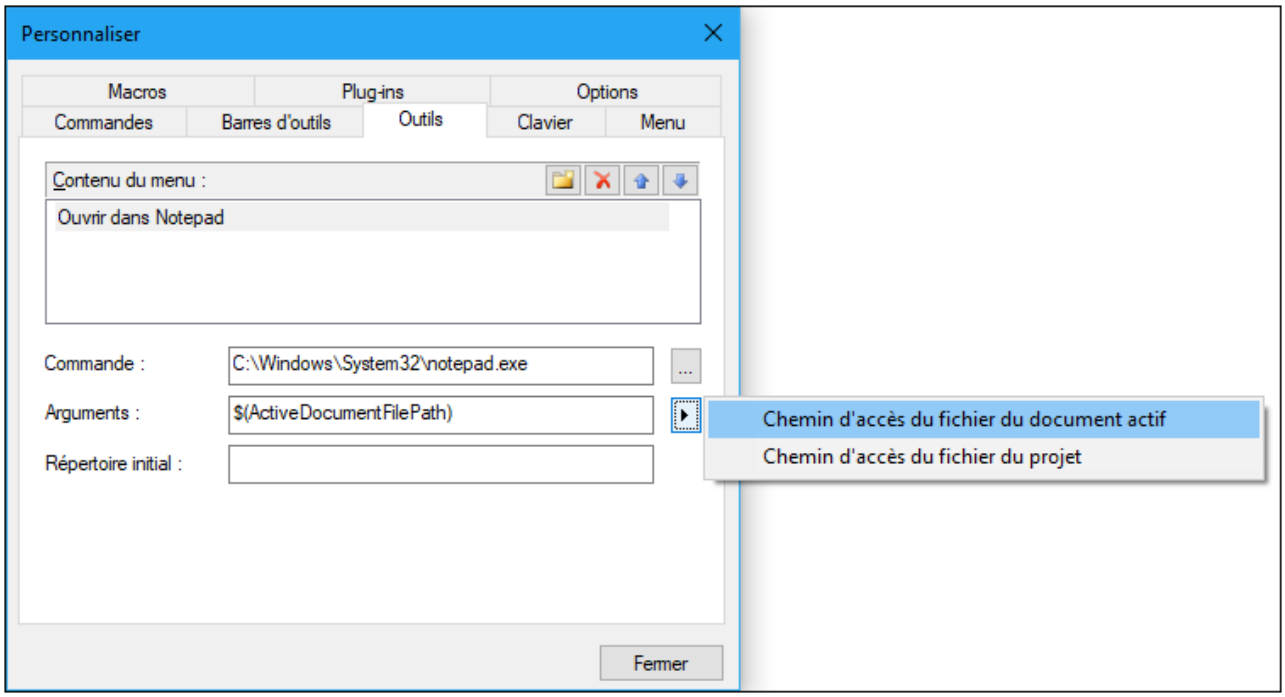
- *Activer ou désactiver une barre d'outils :* cliquer la case à cocher dans la liste *Barres d'outils*.
- *Appliquer les changements à tous les modes :* cliquer la case à cocher en bas du dialogue. Sinon, les changements seront appliqués uniquement au mode actif. Veuillez noter que seuls les changements effectués **après** avoir cliqué sur la case *Tous les modes* s'appliqueront à tous les modes.
- *Ajouter une nouvelle barre d'outils* : cliquer sur la touche **Nouveau** et donner à la barre d'outils un nom dans le dialogue Nom de Barre d'outils qui s'ouvre. À partir de l'onglet **Commandes** glisser les commandes dans la nouvelle barre d'outils.
- *Changer le nom d'une barre d'outils ajoutée :* sélectionner la barre d'outils ajoutée dans le volet Barres d'outils, cliquer sur la touche **Renommer** et éditer le nom dans le dialogue Nom de la barre outils qui apparaît.
- *Réinitialiser la barre de Menu* : sélectionner *Barre de menu* dans le volet Barres d'outils, puis cliquer sur **Réinitialiser**. Cela réinitialise la barre Menu à l'état au moment de l'installation de l'application.
- *Réinitialiser toutes les barres d'outils et les commandes de menu :* cliquer sur la touche **Tout Réinitialiser**. Cela réinitialise toutes les barres d'outils et les menus à l'état au moment de l'installation de l'application.
- *Supprimer une barre d'outils :* sélectionner la barre d'outils que vous souhaitez supprimer dans le volet Barres d'outils et cliquer sur **Supprimer**.

© 2016-2022 Altova GmbH

- *Afficher les libellés de texte des commandes dans une barre d'outils particulière* : sélectionner cette barre d'outils et cliquer la case *Afficher libellés texte*. Veuillez noter que les libellés de texte doivent être activés pour chaque barre d'outils séparément.

28.17.13.3 Outils

L'onglet **Outils** vous permet de définir les commandes à utiliser pour les applications externes depuis l'intérieur de XMLSpy. Ces commandes seront ajoutées au menu **Outils | Outils définis par l'utilisateur**. Par exemple, le fichier actif dans la fenêtre principale de XMLSpy peut être ouverte dans une application externe, comme Notepad, en cliquant sur une commande dans le menu **Outils | Outils définis par l'utilisateur** que vous avez créé.



Pour définir une commande à utiliser dans une application externe, procéder comme suit :

1. Dans le volet *Contenus du menu* (*voir capture d'écran ci-dessus*), cliquer sur l'icône **Nouveau** dans la barre de titre du volet et, dans la ligne d'item qui est créée, saisir le nom de la commande de menu que vous souhaitez. Dans la capture d'écran ci-dessus, nous avons saisi une seule commande de menu, **Ouvrir dans le Notepad**. Nous souhaitons utiliser cette commande pour ouvrir le document actif dans l'application de bloc-notes externe. D'autres commandes peuvent être ajoutées à la liste de commandes en cliquant sur l'icône **Nouveau**. Une commande peut être déplacée vers le haut ou vers le bas dans la liste par rapport aux autres commandes en utilisant les icônes **Déplacer l'élément vers le haut** et **Déplacer l'élément vers le bas**. Pour supprimer une commande, la sélectionner et cliquer sur l'icône **Supprimer**.
2. Afin d'associer une application externe avec une commande, choisir la commande dans le volet *Contenus de menu*. Ensuite, dans le champ *Commande*, saisir le chemin d'accès vers ou chercher le

fichier exécutable de l'application externe. Dans la capture d'écran ci-dessus, le chemin vers l'application Notepad a été saisie dans le champ *Commande*.

3. Les actions disponibles pour être performées avec l'application externe sont affichées lorsque vous cliquez sur la touche fléchée du champ *Arguments* (*voir capture d'écran ci-dessus*). Ces actions sont décrites dans la liste ci-dessous. Lorsque vous sélectionnez une action, un string de code pour l'action est saisi dans le champ *Arguments*.
4. Si vous souhaitez spécifier un répertoire de travail actuel, il faut le saisir dans le champ *Répertoire initial*.
5. Cliquer sur **Fermer** pour terminer.

La commande que vous avez créée apparaîtra dans le menu **Outils | Outils définis par l'utilisateur**, et dans le menu contextuel des fichiers et des dossiers de la fenêtre Projet, dans le sous-menu **Outils définis par l'utilisateur**.

Lorsque vous cliquez sur la commande (dans le menu **Outils | Outils définis par l'utilisateur**) que vous avez créé, l'action que vous avez associée avec la commande sera exécutée. L'exemple de commande affiché dans la capture d'écran ci-dessus effectue les étapes suivantes : il ouvre, dans le bloc-notes, le document qui est actif dans la Fenêtre principale de XMLSpy. La commande d'application externe est aussi disponible dans le menu contextuel des fichiers dans la fenêtre Projet (cliquer avec la touche de droite sur un fichier dans la fenêtre Projet pour afficher le menu contextuel de ce fichier). Par le biais de la fenêtre Projet, vous pouvez aussi ouvrir plusieurs fichiers (lorsque les applications le permettent) en effectuant une sélection multiple et en sélectionnant la commande depuis le menu contextuel.

Arguments

Le champ *Arguments* spécifie l'action à exécuter par la commande d'application externe. Les arguments suivants sont disponibles :

- *Chemin d'accès de fichier du document actif :* la commande dans le menu **Outils | Outils définis par l'utilisateur** ouvre le document qui est actif dans XMLSpy dans l'application externe. La commande dans le menu contextuel d'un fichier dans la fenêtre Projet ouvre le fichier sélectionné dans l'application externe.
- *Chemin d'accès de fichier du projet :* ouvre le fichier de projet XMLSpy (le fichier `.spp`) dans l'application externe.

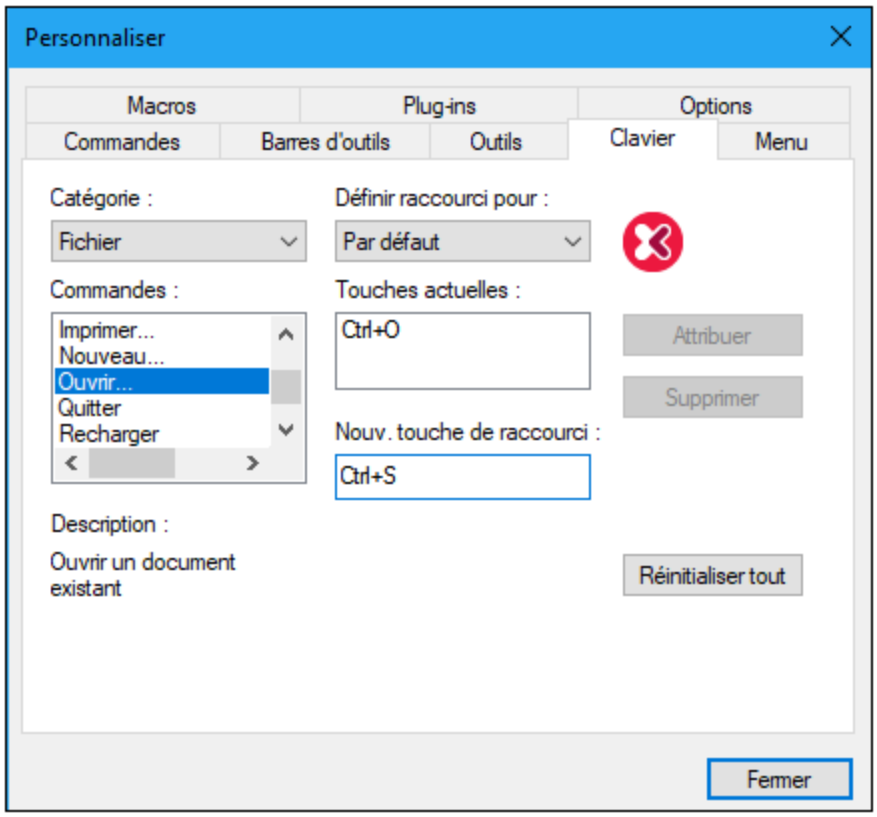
Répertoire initial

L'entrée du *Répertoire initial* est optionnelle et est un chemin qui sera utilisé en tant que le répertoire actuel.

28.17.13.4 Clavier

L'onglet **Clavier** vous permet de créer de nouveaux raccourcis de clavier ou de changer des raccourcis existants, pour toute commande d'application.

© 2016-2022 Altova GmbH



Pour attribuer un nouveau raccourci à une commande, ou pour changer un raccourci existant, procéder comme suit.

1. Choisir la catégorie *Toutes les commandes* dans la liste de choix *Catégorie*. Veuillez noter que si un macro a été sélectionné en tant qu'une Commande Associée, les macros sont aussi disponibles pour une sélection dans la liste de choix *Catégorie* et un raccourci pour le macro peut être défini.
2. Dans la liste de choix *Commandes*, sélectionner la commande à laquelle vous souhaitez attribuer un nouveau raccourci ou choisir la commande du raccourci que vous souhaitez modifier.
3. Cliquer dans la fenêtre de saisie *Nouvelle touche de raccourci* et appuyer sur le raccourci que vous souhaitez attribuer à cette commande. Le raccourci apparaît dans la fenêtre *Nouvelle touche de raccourci*. Si le raccourci n'a encore été attribué à aucune commande, la touche **Attribuer** est activée. Si le raccourci a déjà été attribué à une commande, alors celle-ci sera affichée sous la fenêtre et la touche **Attribuer** est désactivée. (Pour supprimer la fenêtre de saisie *Nouvelle touche de raccourci*, appuyer sur une des clés de commande, **Ctrl**, **Alt** ou **Shift**).
4. Cliquer sur la touche **Attribuer** pour attribuer le raccourci. Celui-ci apparaît maintenant dans la liste *Touches actuelles*. Vous pouvez attribuer plusieurs raccourcis à une seule commande.
5. Cliquer sur la touche **Fermer** pour confirmer.

Supprimer un raccourci

Un raccourci ne peut pas être attribué à plusieurs commandes. Si vous souhaitez supprimer un raccourci, cliquer dans une des fenêtres Touches actuelles puis cliquer sur la touche **Supprimer**.

Définir l'accélérateur pour

Actuellement, les accélérateurs peuvent uniquement être définis en tant que défaut. Aucun autre mode n'est disponible.

Raccourcis de clavier par défaut

Les raccourcis par défaut des commandes utilisées communément sont listés ci-dessous. Un aperçu de toutes les commandes du menu de l'application est disponible dans le modèle de clavier (**Aide | Modèle de clavier**).

- *Function-key shortcuts (y compris pour la validation et la transformation)*

F1	Menu d'aide
F1 + Alt	Ouvrir le dernier fichier
F3	Trouver suivant
F4 + CTRL	Fermer la fenêtre active
F4 + Alt	Fermer XMLSpy
F5	Réinitialiser
F6 + CTRL	Faire défiler les fenêtres ouvertes
F7	Vérifier la bonne formation
F8	Valider
F10	Transformation XSL
F10 + CTRL	Transformation XSL:FO

- *Commandes de fichier et d'applications*

Alt + F1	Ouvrir le dernier fichier
Ctrl + O	Ouvrir le fichier
Ctrl + N	Nouveau fichier
Ctrl + P	Imprimer fichier
Ctrl + S	Enregistrer fichier
CTRL + F4	Fermer la fenêtre active
CTRL + F6	Faire défiler les fenêtres ouvertes
CTRL + TAB	Sauter entre des documents ouverts
Alt + F4	Fermer XMLSpy

- *Touches diverses*

Touches fléchées haut/bas	Déplacer le curseur ou la barre de sélection
Échap	Abandonner les éditions ou fermer le dialogue
Retour	Confirmer la sélection
Suppr	Supprimer le caractère ou la sélection

© 2016-2022 Altova GmbH

Shift + Del	Couper

Commandes d'édition

CTRL + A	Sélectionner tout
Ctrl + F	Recherche
Ctrl + G	Aller à la ligne/caractère
Ctrl + H	Remplacer
Ctrl + V	Coller
Ctrl + X	Couper
CTRL + Y	Rétablir
Ctrl + Z	Annuler

Commandes de Mode Texte

CTRL + E	Passer entre les balises de début/fin
CTRL + Shift + E	Sélectionner l'élément qui contient le curseur
CTRL + Alt + E	Aller à l'élément parent
CTRL + "+"	Zoom avant
CTRL + "-"	Zoom arrière
CTRL + 0	Réinitialiser Zoom
CTRL + roulette de souris avant	Zoom avant
CTRL + roulette de souris arrière	Zoom arrière

Commandes Mode Grille

CTRL + D	Ajouter CDATA
CTRL + E	Ajouter Élément
CTRL + I	Ajouter Attribut
CTRL + M	Ajouter Commentaire
CTRL + T	Ajouter Texte
CTRL + Shift + D	Insérer CDATA
CTRL + Shift + E	Insérer Élément
CTRL + Shift + I	Insérer Attribut
CTRL + Shift + M	Insérer Commentaire
CTRL + Shift + T	Insérer Texte

CTRL + Alt + D	Ajouter CDATA enfant
CTRL + Alt + E	Ajouter Élément enfant
CTRL + Alt + I	Ajouter Attribut enfant
CTRL + Alt + M	Ajouter Commentaire enfant
CTRL + Alt + T	Ajouter Texte enfant

Commandes Mode Schéma

CTRL + Dbl-clic Élément	Afficher la définition élément

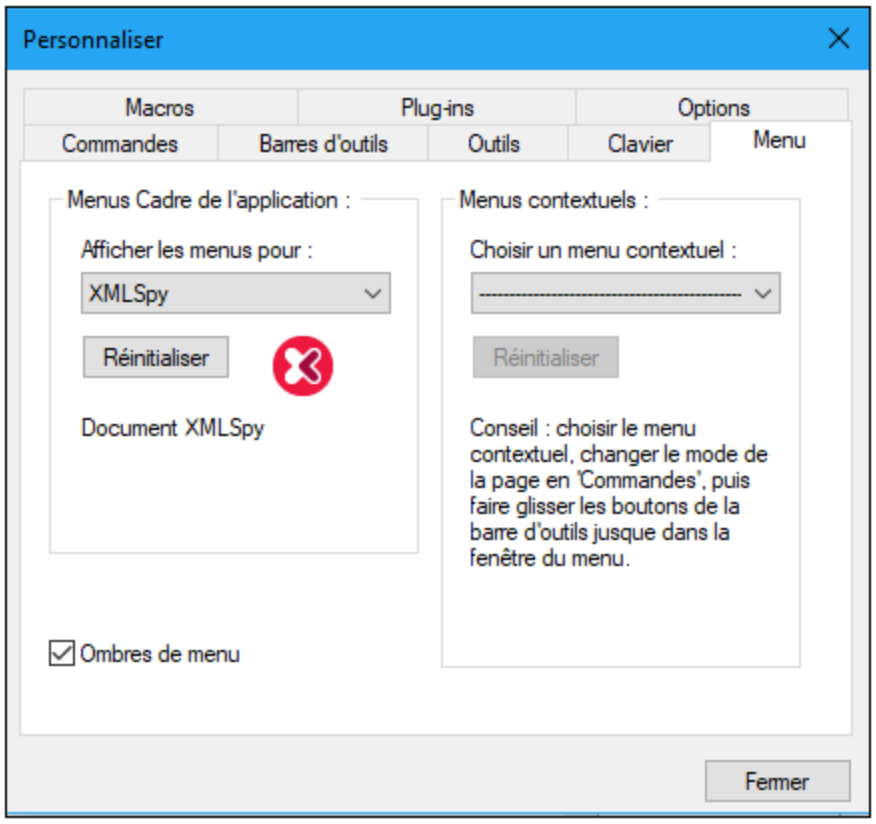
Commandes de débogueur

F9	Insérer/Supprimer point d'arrêt
F9 + Shift	Insérer/Supprimer point de trace
F9 + CTRL	Activer/Désactiver point d'arrêt
F9 + Shift + CTRL	Activer/Désactiver point de trace
F11	Pas à pas détaillé
F11 + Shift	Pas à pas principal
F11 + CTRL	Pas à pas sortant
F11 + Alt	Démarrer débogueur/OK

28.17.13.5 Menu

L'onglet **Menu** vous permet de personnaliser les deux barres de menu principal (barres de menu par défaut et d'application) ainsi que les menus contextuels de l'application.

© 2016-2022 Altova GmbH



Personnaliser la barre de menu par défaut et la barre de menu d'application

La barre de menu par défaut est la barre de menu qui s'affiche lorsqu'aucun autre document n'est ouvert dans la fenêtre principale. La barre de menu d'application est la barre de menu qui s'affiche lorsqu'un ou plusieurs documents sont ouverts dans la fenêtre principale. Chaque barre de menu peut être personnalisée séparément et les changements de personnalisation effectués dans l'une d'entre elle ne touchent pas l'autre.

Pour personnaliser une barre de menu, la sélectionner dans la liste de choix *Afficher les menus pour* (*voir capture d'écran ci-dessous*). Ensuite passer à l'onglet Commandes du dialogue Personnalisation et glisser les commentaires depuis la liste Commandes dans la barre de menu ou dans n'importe lequel des menus.

Supprimer les commandes depuis les menus et réinitialiser les barres de menu

Pour **supprimer** un menu complet ou une commande à l'intérieur d'un menu, procéder comme suit :

1. Dans le volet Application Frame Menus, sélectionner soit *Défaut* (qui montre les menus disponibles lorsqu'aucun document n'est ouvert) ou *XMLSpy* (qui montre les menus disponibles lorsqu'un ou plusieurs documents sont ouverts).
2. Ouvrir le dialogue Personnaliser, sélectionner (i) le menu que vous souhaitez supprimer depuis la barre de menu de l'application, ou (ii) la commande que vous souhaitez supprimer depuis un de ces menus.
3. Soit (i) glisser le menu depuis la barre du menu ou la commande de menu depuis le menu, soit (ii) cliquer avec la touche de droite sur le menu ou la commande de menu et sélectionner **Supprimer**.

Vous pouvez **réinitialiser** chacune de ces deux barres de menu (barres de menu par défaut et d'application) à son état d'installation original en sélectionnant le menu dans la liste de choix *Afficher les menus pour* et puis en cliquant sur la touche **Réinitialiser** en-dessous de la liste de choix.

Personnaliser les menus contextuels de l'application

Les menus contextuels sont les menus qui apparaissent lorsque vous cliquez avec la touche de droite certains objets dans l'interface de l'application. Chacun de ces menus contextuels peut être personnalisé en suivant les étapes suivantes :

1. Sélectionner le menu contextuel que vous souhaitez dans la liste de choix *Choisir un menu contextuel*. Le menu contextuel s'ouvre.
2. Passer à l'onglet Commandes du dialogue Personnalisation.
3. Glisser une commande depuis la liste *Commandes* dans le menu contextuel.
4. Si vous souhaitez supprimer une commande depuis le menu contextuel, cliquer avec la touche de droite sur la commande dans le menu contextuel et cliquer sur **Supprimer**. En alternative, vous pouvez glisser la commande que vous souhaitez supprimer du menu contextuel.

Vous pouvez réinitialiser tout menu contextuel à son état d'installation d'origine en le sélectionnant dans la liste de choix *Choisir un menu contextuel* et puis en cliquant la touche **Réinitialiser** située sous la liste de choix.

Ombres de menu

Cliquer sur la case *Ombres de menu* pour ajouter une ombre à tous les menus.

28.17.13.6 Macros

L'onglet **Macros** vous permet de créer des commandes d'application pour les macros qui ont été créés à l'aide de d'éditeur de script de XMLSpy. Ces commandes d'application (qui exécutent les macros associés à eux) peuvent ensuite être rendues disponibles dans les menus et barres d'outils, soit directement depuis l'onglet Macros ou en utilisant les mécanismes disponibles dans l'onglet Commandes du dialogue Personnaliser. En tant que commandes d'application, ils peuvent aussi être attribués à des raccourcis dans l'onglet Clavier du dialogue Personnaliser.

Comment les macros fonctionnent dans XMLSpy

Dans XMLSpy, les macros fonctionnent comme suit :

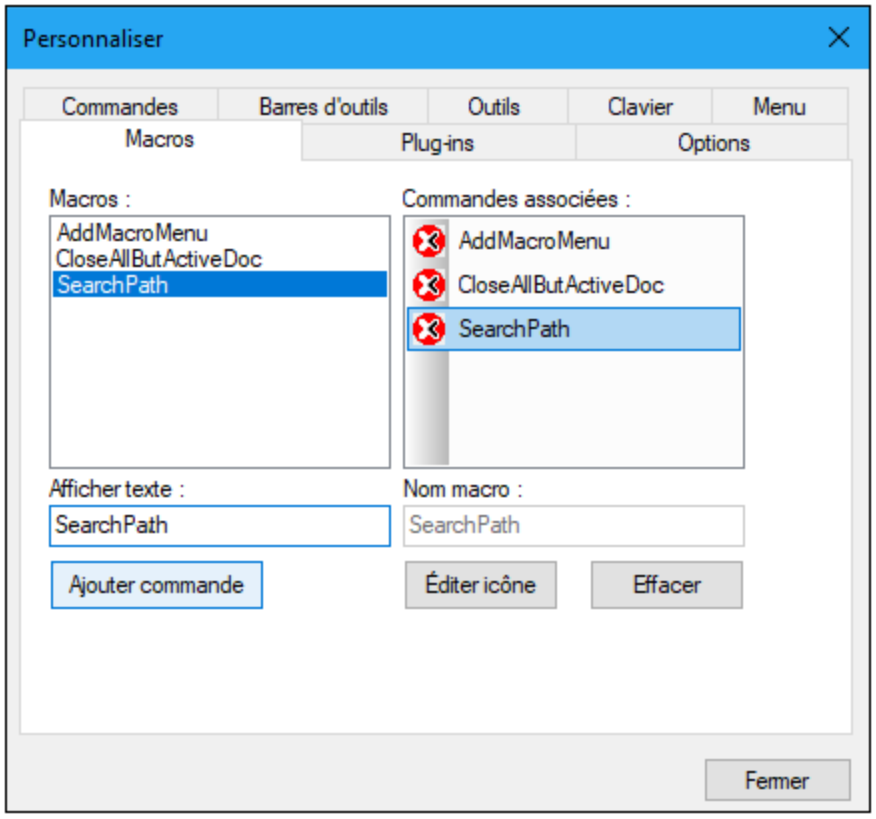
- Des projets de script Altova (fichier `.asprj`) sont créés dans l'Éditeur de script de XMLSpy. Ce sont ces projets de script qui peuvent contenir les macros utilisés dans XMLSpy.
- Deux projets de script peuvent être actifs dans le même temps dans XMLSpy : (i) un projet de script d'application, qui est spécifié dans la section Script du dialogue Options, et (ii) le projet de script du projet XMLSpy actif, qui est spécifié dans le dialogue Paramètres de Script (**Projet | Paramètres du script**).
- Les macros dans ces deux projets de script sont disponibles dans l'application : dans le menu **Projet | Macros** (à partir duquel les macros peuvent être exécutés), et dans l'onglet Macros du dialogue Personnaliser (*capture d'écran ci-dessous*), dans lequel ils peuvent être définis en tant que commandes d'application. Une fois qu'un macro a été défini en tant que commande d'application, la commande peut être placée dans un menu et/ou une barre d'outils.

© 2016-2022 Altova GmbH

Créer une commande d'application pour un macro

Dans l'Éditeur de script (**Outils | Éditeur de script**), créer le macro que vous souhaitez et l'enregistrer dans un projet de script. Spécifier ce fichier pour qu'il soit le projet de script de l'application (par le biais de la section Script du dialogue Options) ou bien le projet de script du projet d'application actif (par le biais du dialogue Paramètres de Script (**Projet | Paramètres du script**)) du projet d'application. Les macros dans le projet de script apparaîtront maintenant dans le volet *Macros* de l'onglet Macros (*voir capture d'écran ci-dessous*).

Pour créer une commande d'application pour un macro, sélectionner le macro dans le volet *Macros*, définir le texte de la commande dans la fenêtre *Afficher texte* et cliquer sur **Ajouter Commande** (*voir capture d'écran ci-dessous*). Une commande associée avec le macro sélectionné sera ajouté à la liste *Commandes associées*.



Pour éditer l'icône d'une commande associée, sélectionner la commande et cliquer sur **Éditer icône**. Pour supprimer une commande associée, cliquer sur **Supprimer**.

Placer une commande à association macro dans un menu ou une barre d'outils

Il existe deux moyens de placer une commande à association macro dans un menu ou une barre d'outils :

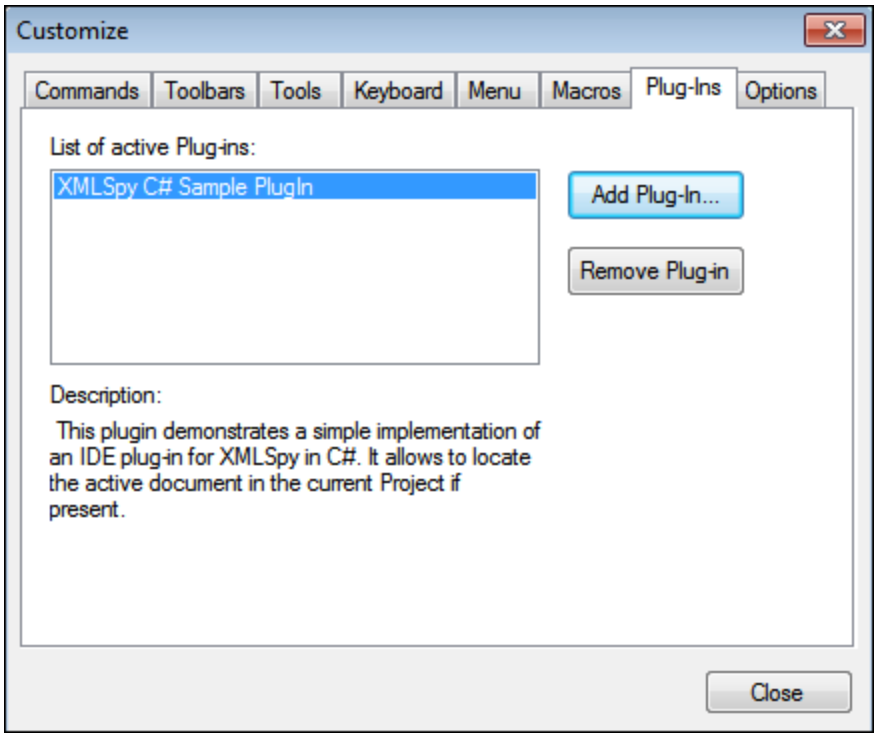
- Glisser la commande depuis la liste Commandes associées à l'emplacement désiré dans le menu ou la barre d'outils.
- Utiliser les mécanismes disponibles dans l'onglet Commandes du dialogue Personnaliser.

Dans les deux cas, la commande sera créée à l'emplacement désiré. Cliquer sur la commande dans le menu ou la barre d'outils permettra d'exécuter le macro.

Note : Si un macro a été défini en tant que commande associée, vous pouvez définir un raccourci de clavier pour ce macro. Dans l'onglet Clavier du dialogue Personnaliser, choisir *Macros* dans la liste de choix *Catégorie*, puis sélectionner le macro requis et définir le raccourci. Vous devez définir un macro en tant que commande associée pour qu'elle soit disponible en tant que raccourci de clavier.

28.17.13.7 Plug-Ins

L'onglet **Plug-Ins** vous permet d'intégrer les plug-ins et de placer des commandes, lorsque celles-ci ont été programmées de cette manière, dans un menu d'application et/ou barre d'outils. Dans l'onglet Plug-In (*capture d'écran ci-dessous*), cliquer sur **Ajouter Plug-In**, et chercher le fichier DLL du plug-in (*voir 'Créer plug-ins' ci-dessous*). Cliquer sur **OK** pour ajouter le plug-in. Plusieurs plug-ins peuvent être ajoutés.



Une fois qu'un plug-in a bien été ajouté, une description du plug-in apparaît dans le dialogue et la touche **Supprimer le Plug-In** est activée. Si le code de plug-in crée des barres d'outils et des menus, ceux-ci seront immédiatement visibles dans l'interface d'application. Pour supprimer un plug-in, le sélectionner et cliquer sur **Supprimer le Plug-In**.

Créer des plug-ins

Le code source pour les plug-ins modèles a été fourni dans le dossier (Mes) Documents de l'application : dossier `Examples\IDEPlugin`. Pour générer un plug-in à partir du code de source, procéder comme suit :

1. Ouvrir la solution que vous souhaitez générer en tant que plug-in dans Visual Studio.

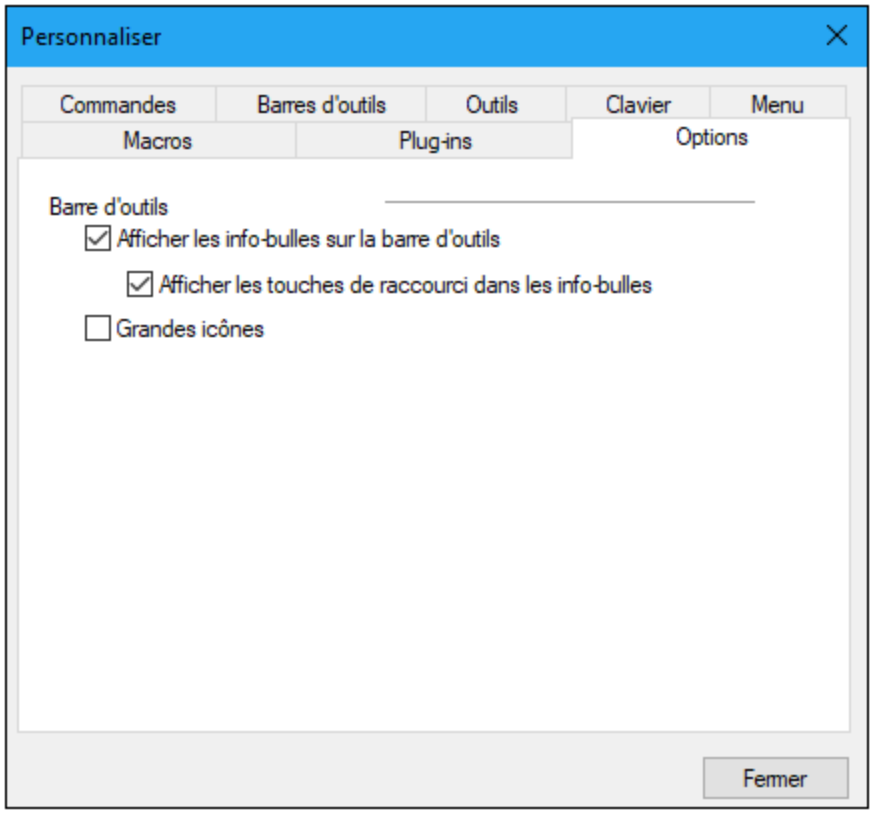
© 2016-2022 Altova GmbH

2. Générer le plug-in avec la commande dans le menu Générer.
3. Le fichier DLL du plug-in sera créé dans le dossier `Bin` ou `Debug`. Ce fichier DLL est le fichier qui doit être ajouté en tant que plug-in (*voir ci-dessus*).

Pour plus d'information sur les plug-in, voir la section IDE Plugins.

28.17.13.8 Options

L'onglet **Options** vous permet de définir des paramètres d'environnement généraux.

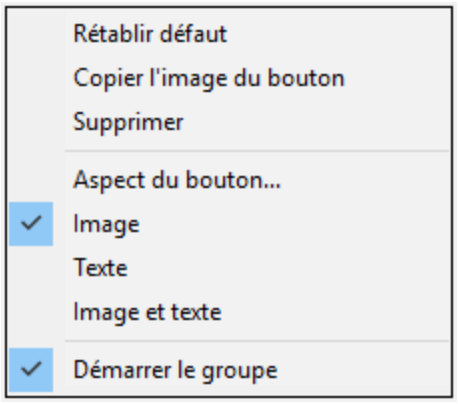


Cliquer sur les cases à cocher pour passer aux options suivantes :

- *Afficher les infobulles sur la barre d'outils :* affiche une fenêtre popup lorsque le pointeur de la souris est placé sur une icône dans une barre d'outils. La fenêtre popup contient une brève description du fonctionnement de l'icône et du raccourci de clavier associé, si un raccourci a été attribué et si l'option *Afficher raccourci* a été cochée.
- *Afficher les touches de raccourci dans les infobulles :* définit si les informations de raccourci seront affichées dans les infobulles.
- *Grandes icônes :* fait passer la taille des icônes de la barre d'outils de standard à grand.

28.17.13.9 Personnaliser le menu Contextuel

Le menu **Personnaliser le menu contextuel** (*capture d'écran ci-dessous*) est le menu qui apparaît lorsque le dialogue Personnaliser est ouvert et que vous cliquez avec la touche de droite dans un menu d'application, une commande de menu ou une icône de barre d'outils dans la fenêtre principale.



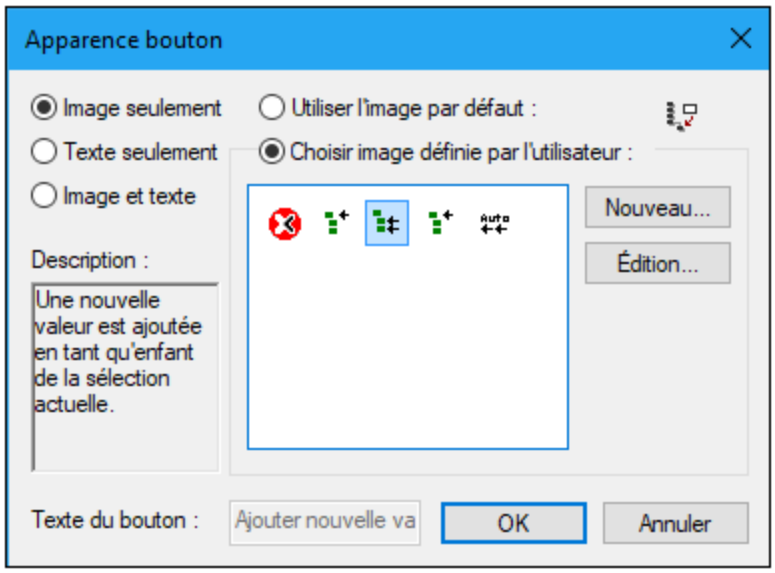
Les fonctions suivantes sont disponibles :

- *Rétablir défaut :* actuellement aucune fonction.
- *Copier l'image du bouton :* copie l'icône que vous avez cliqué avec la touche de droite dans le presse-papiers.
- *Supprimer :* supprime le menu sélectionné, la commande de menu ou l'icône de la barre d'outils. Pour plus d'informations concernant la manière de restaurer des items supprimés, voir ci-dessous.
- *Aspect du bouton :* ouvre le dialogue Apparence du bouton (*voir capture d'écran ci-dessous*), dans lequel vous pouvez déterminer des propriétés qui définissent l'aspect de l'icône de barre d'outils sélectionnée. Voir la description pour plus de détails.
- *Image, Texte, Image et Texte* : options exclusives mutuellement qui déterminent si l'icône de la barre d'outils sélectionnée sera uniquement une icône, uniquement du texte ou les deux. Vous pouvez sélectionner une de ces options pour effectuer le changement. En alternative, vous pouvez effectuer ce changement dans le dialogue Apparence du bouton.
- *Démarrer le groupe :* insère un diviseur de groupe vertical à gauche de l'icône de la barre d'outils sélectionnée. Cela permet de faire de l'icône de la barre d'outils sélectionnée, la première d'un groupe d'icônes.

Le dialogue Apparence du bouton

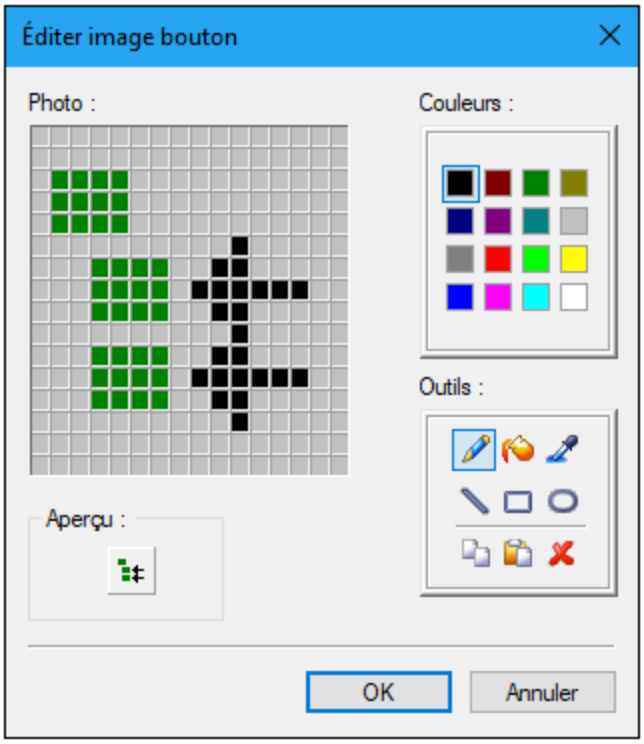
Cliquer avec la touche de droite sur une icône de barre d'outils et cliquer sur **Apparence du bouton** pour ouvrir le dialogue Apparence du bouton (*capture d'écran ci-dessous*). Ce dialogue vous permet d'éditer l'image d'icône de la barre outils, ainsi que son texte. Actuellement, seules les icônes de barre d'outils pour les macros et pour les plug-ins peuvent être éditées à l'aide de ce dialogue.

© 2016-2022 Altova GmbH



Les fonctions d'édition suivantes sont disponibles pour l'icône de la barre d'outils sélectionnée (celle qui a été cliquée avec la touche de droite de la souris pour aboutir au menu contextuel Personnaliser) :

- *Image seulement, texte seulement, image et texte* : choisir le bouton radio désiré pour spécifier l'aspect que prendra l'icône de la barre d'outils.
- *Édition de l'image :* lorsque *Image seulement* ou *Image et texte* sont sélectionnées, les options d'édition de l'image sont activées. Cliquer sur **Nouveau** pour créer une nouvelle image qui sera ajoutée aux images définies par l'utilisateur dans le volet d'images. Sélectionner une image et cliquer sur **Édition** pour l'éditer.



- *Sélection d'image :* choisir une image depuis le volet Images et cliquer sur OK pour utiliser l'image sélectionnée en tant que l'icône de la barre d'outils.
- *Édition et sélection du texte :* si vous sélectionnez *Texte seulement* ou *Image et texte*, le champ de saisie *Texte du bouton* est activé. Saisir ou éditer le texte et cliquer sur **OK** pour en faire le texte de l'icône de la barre d'outils.

Note : le dialogue Apparence du bouton peut aussi être utilisé pour éditer le texte des commandes de menu. Cliquer avec la touche de droite sur la commande de menu (lorsque le dialogue Personnaliser est ouvert), cliquer sur **Apparence du bouton**, puis éditer le texte de la commande de menu dans le champ de saisie *Texte du bouton*.

Restaurer les menus supprimés, les commandes de menu et les icônes de barres d'outils

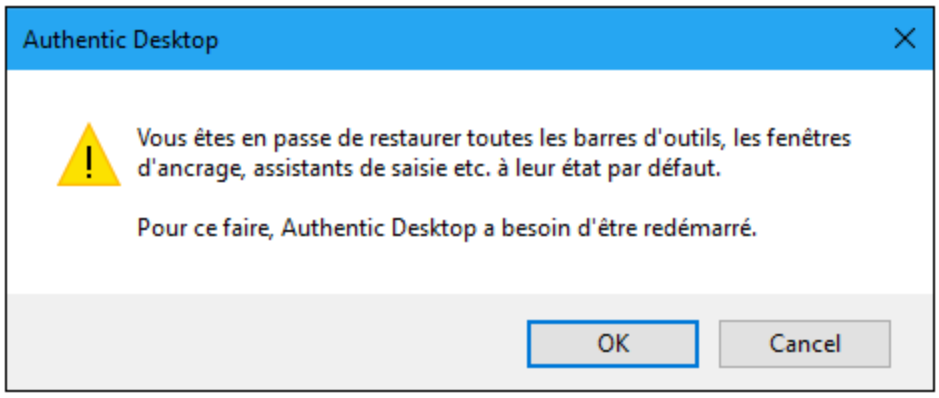
Si un menu, une commande de menu, ou une icône de barre d'outils a été supprimée en utilisant la commande **Supprimer** dans le menu contextuel Personnaliser, ils peuvent être restaurés comme suit :

- *Menus :* Aller sur **Outils | Personnaliser | Menu**, et cliquer sur la touche **Réinitialiser** dans le volet *Menus Cadre de l'application*. En alternative, aller sur **Outils | Personnaliser | Barres outils**, choisir Barre de menus, et cliquer sur la touche **Réinitialiser**.
- *Commandes de menu :* Aller sur **Outils | Personnaliser | Commandes**, et glisser la commande depuis la fenêtre Commandes dans le menu.
- *Icônes de barre d'outils :* Aller sur **Outils | Personnaliser | Commandes**, et glisser la commande depuis la fenêtre Commandes dans le menu.

© 2016-2022 Altova GmbH

28.17.14 Restaurer barres d'outils et fenêtres

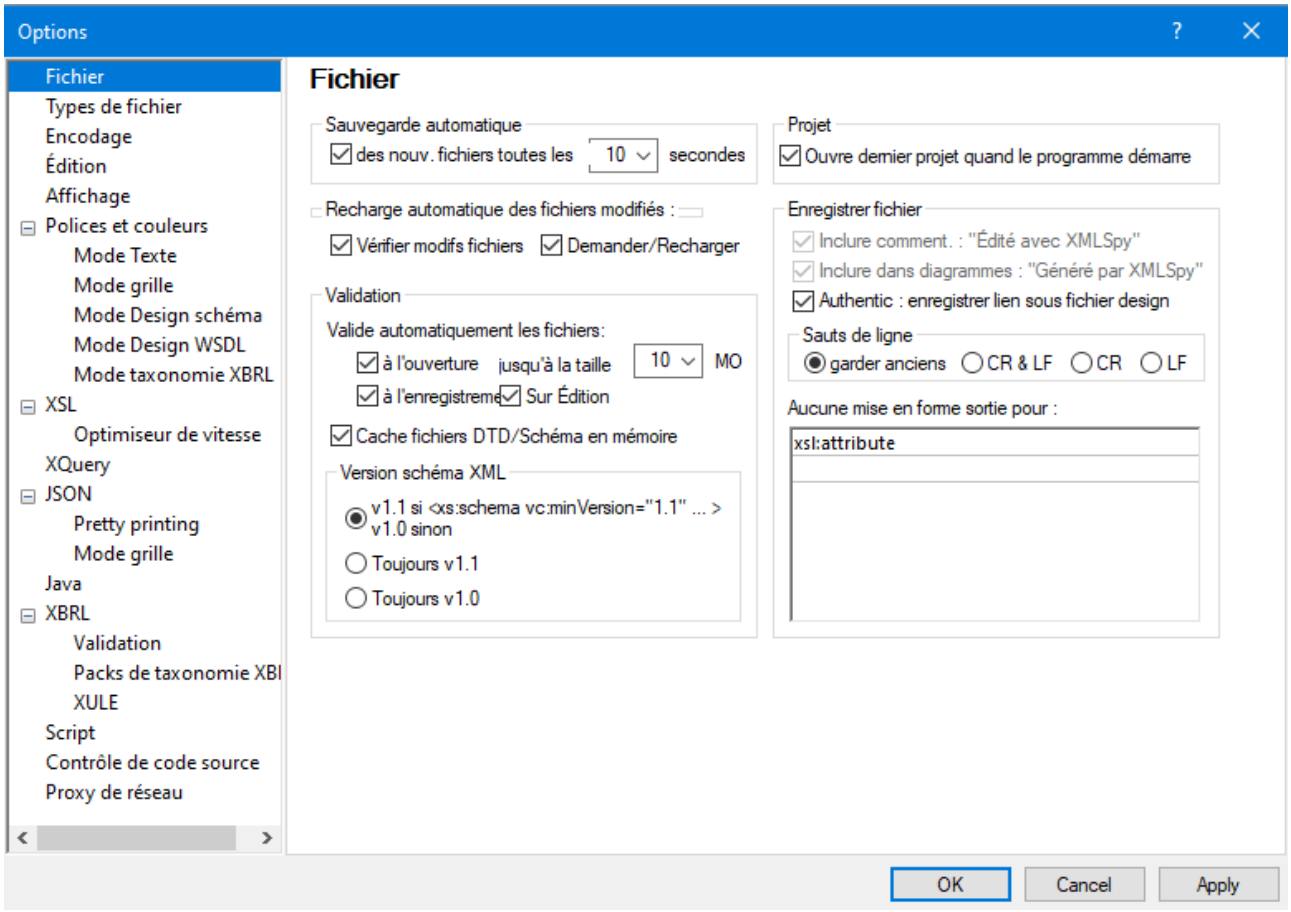
La commande **Restaure barres d'outils et fenêtres** ferme XMLSpy et la redémarre avec les paramètres par défaut. Avant de fermer le programme un dialogue s'ouvre et vous demande de confirmer si vous souhaitez fermer XMLSpy (*capture d'écran ci-dessous*).



Cette commande est utile si vous avez redimensionné, déplacé ou dissimulé des barres d'outil ou des fenêtres et que vous souhaitez à présent avoir toutes les barres d'outils et les fenêtres à leur place originale.

28.17.15 Options

La commande **Outils | Options** vous permet de définir les paramètres d'application globaux. Ces paramètres sont organisés en sections (*voir le panneau gauche dans la capture d'écran ci-dessous*). Par exemple, la section Fichier (*affichée dans la capture d'écran ci-dessous*) contient des options qui spécifient comment vous voulez XMLSpy pour ouvrir et enregistrer les fichiers. Pour spécifier les options d'une section particulière, sélectionnez cette section dans le panneau gauche et spécifiez les valeurs de propriété que vous souhaitez avoir. Le bouton **OK** enregistre les modifications dans le registre et ferme le dialogue. La touche **Appliquer** entraîne l'affichage des modifications dans les documents ouverts actuellement.

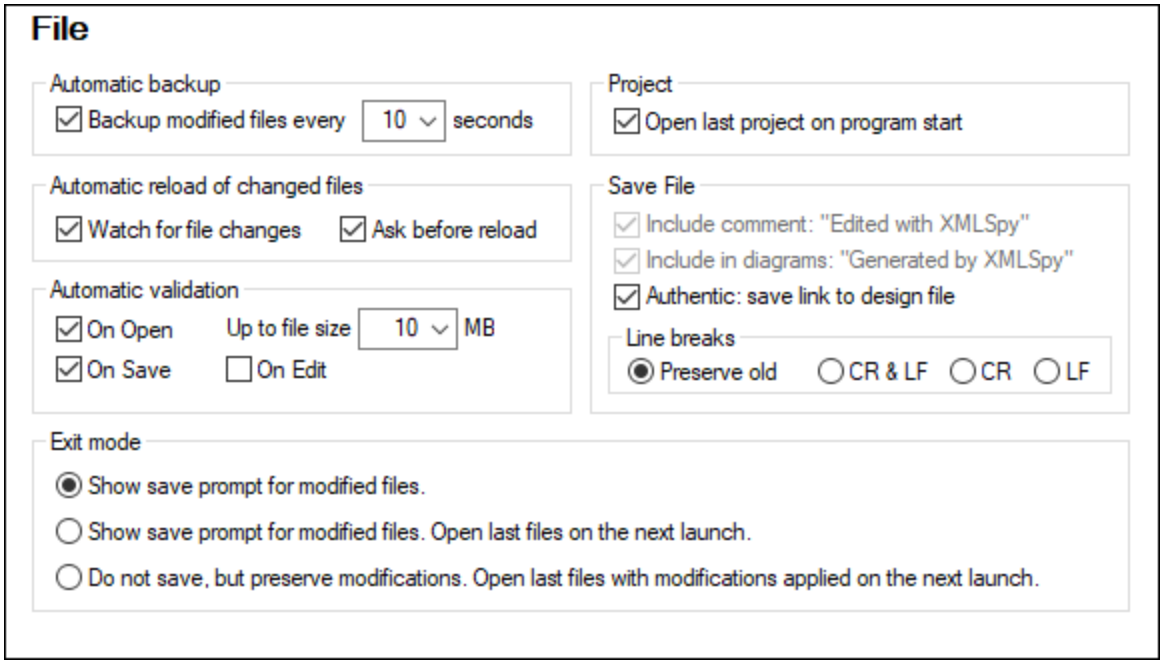


Chaque section du dialogue des Options est décrite en détail dans sa sous-section de cette section.

28.17.15.1 Fichier

La section **Fichier** définit la manière dont XMLSpy ouvre et enregistre des documents. Vous trouverez les paramètres liés dans la section Encodage.

© 2016-2022 Altova GmbH



Sauvegarde automatique

Les fichiers que vous éditez actuellement seront sauvegardés automatiquement si cette option est activée. Vous pouvez choisir une fréquence de sauvegarde allant de 5 à 60 secondes dans la liste de choix ou saisir une valeur individuelle de jusqu'à 300 secondes. Pour plus d'informations, voir la section *Sauvegarde automatique de fichiers*.

Le rechargement automatique des fichiers modifiés

Si vous utilisez un environnement à utilisateurs multiples, ou si vous travaillez sur des fichiers générés dynamiquement sur un serveur, vous pouvez détecter les modifications des fichiers actuellement ouverts dans l'interface. Chaque fois que XMLSpy détecte un changement dans un document ouvert, il vous demandera si vous souhaitez recharger le fichier modifié.

Validation automatique

Si vous utilisez des schémas DTD ou XML pour définir la structure de vos documents XML, vous pouvez valider automatiquement vos documents d'instance dans les situations suivantes :

- En ouvrant le fichier si le fichier a une taille inférieure à celle que vous spécifiez dans MB
- Sur Enregistrement du fichier
- En éditant le fichier. Si l'option est sélectionnée, la validation sera effectuée lorsque vous tapez Text View ou JSON Grid View. Pour plus d'information, voir également Validation XML dans le Mode Texte.

Si le document n'est pas valide, un message d'erreur sera affiché. S'il est valide, un message aucun message ne sera affiché et l'opération sera traitée sans aucune notification.

Projet

Lorsque vous lancez XMLSpy, vous pouvez ouvrir le dernier projet utilisé en dernier automatiquement.

Enregistrer fichier

Lorsque vous enregistrez un document XML, XMLSpy inclut un commentaire succinct `<!-- Edited with XMLSpy http://www.altova.com -->` en haut du fichier. Cette option ne peut être désactivée par des utilisateurs de licence et prend effet lors de l'édition ou en enregistrant les fichiers dans la Grille améliorée ou le Mode Design de schéma.

Lors de l'enregistrement d'un diagramme modèle de contenu (utilisant l'option de menu **Design Schéma | Générer Documentation**), XMLSpy inclut le logo XMLSpy. Cette option peut uniquement être désactivée par des utilisateurs de licence.

Si StyleVision Power Stylesheet est associée à un fichier XML, l'option 'Authentic: save link to design file' fera que le lien vers StyleVision Power Stylesheet sera enregistré avec un fichier XML.

Sauts de ligne

Lorsque vous ouvrez un fichier, le codage de caractère pour les sauts de ligne dans celui-ci sera préservé si **Preserve old** est sélectionné. En alternative, vous pouvez choisir de coder les sauts de ligne dans un des trois codages : **CR&LF** (pour PC), **CR** (pour MacOS), ou **LF** (pour Unix).

Mode Quitter

Ces options déterminent comment gérer des fichiers qui sont ouverts lorsque XMLSpy a quitté. Les options suivantes sont disponibles :

- *Show save prompt for modified files*: Si un fichier ouvert contient des modifications non enregistrées, une invite apparaîtra demandant si vous souhaitez enregistrer les modifications de fichier. Dépendant de votre réponse, le fichier est enregistré ou pas enregistré et le programme est quitté par la suite.
- *Afficher invite enregistrer pour les fichiers modifiés. Open last files on the next launch*: Le dialogue Enregistrer apparaît pour les fichiers ouverts qui contiennent des modifications non enregistrées. L'utilisateur peut enregistrer un ou plusieurs fichiers ou pas. Lorsque le programme est relancé après avoir quitté, tous les fichiers qui ont été ouverts sur quitter seront ouverts sur redémarrage. (Si les modifications n'avaient pas été enregistrées, elles seraient perdues.)
- *Ne pas enregistrer mais préserver les modifications. Open last files with modifications applied on the next launch*: Le programme quitte directement sans enregistrer les modifications non enregistrées. En relançant le programme, tous les fichiers qui ont été ouverts en quittant seront ouverts en redémarrant, et ils contiendront les modifications non enregistrées. Ce serait une situation comme si vous continuez là où vous avez arrêté.

Si vous quittez le programme pour la première fois, les options en Mode Quitter sont présentées pour que vous puissiez choisir le comportement Quitter de votre choix. Ensuite, les options sont disponibles dans la section Fichier du dialogue Options.

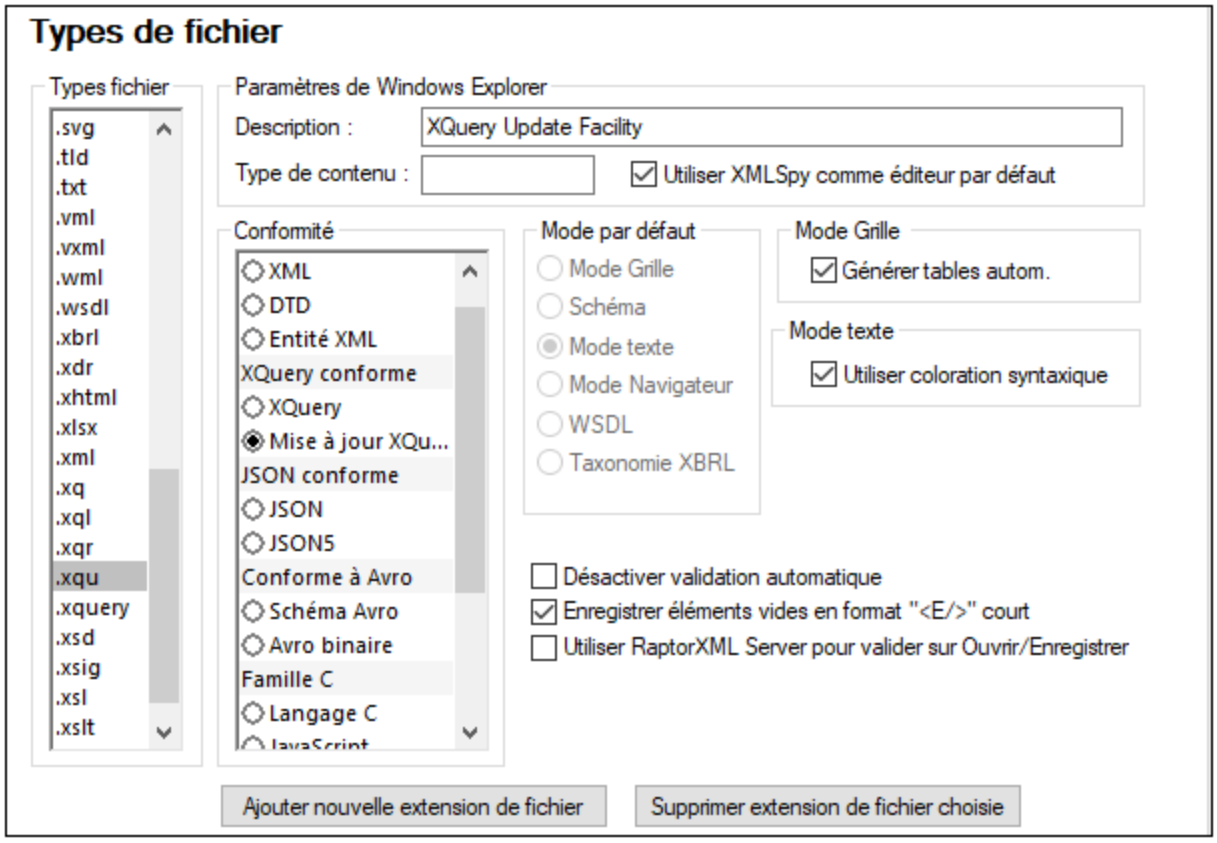
Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

© 2016-2022 Altova GmbH

28.17.15.2 Types de fichier

La section **Types de fichier** *(capture d'écran ci-dessous)* vous permet de personnaliser le comportement de XMLSpy sur une base "par type de fichier".



Choisir un type de fichier depuis la fenêtre Types de fichiers, puis personnaliser les fonctions pour ce type de fichier particulier comme décrit ci-dessous.
Veuillez noter qu'il y a deux saisies spéciales dans la liste des types de fichier :

- <*default*> peut être utilisé pour spécifier le traitement de fichiers dotés d'une extension qui ne se trouve pas dans la liste du file-type.
- <*none*> peut être utilisé pour spécifier le traitement de fichiers qui n'ont pas d'extension du tout.

Paramètres Windows Explorer

Vous pouvez définir la description de type de fichier et le type de contenu conforme à MIME utilisé par Windows Explorer et si XMLSpy doit être l'éditeur par défaut pour les documents de ce type de fichier.

Conformité

XMLSpy fournit des fonctions d'édition intelligentes spécifiques, ainsi que d'autres fonctions, pour des types de fichier différents. XMLSpy définit les fonctions pour un type de fichier particulier sur la base de la conformité que vous avez définie dans cette option. Par exemple, dans la capture d'écran ci-dessus, les fichiers présentant l'extension de fichier `.xqu` sont définis pour être conforme à XQuery Update. C'est pourquoi XMLSpy ouvrira les

fichiers `.xqu` avec la prise en charge de l'édition de XQuery Update. XMLSpy vous permet de définir les options de conformité suivantes : XML, XQuery, ZIP, JSON, Avro, autres formats. De plus, une différenciation de conformité XML est également effectuée entre les types de fichier d'Entité XML, DTD et XML. Les fichiers de conformité JSON sont différenciés selon qu'ils sont des fichiers JSON ou de Schéma Avro. L'option *Conformité Avro* invoque la prise en charge pour les fichiers binaires Avro. Un grand nombre de types de fichiers est défini avec une conformité par défaut qui est appropriée pour le type de fichier. Nous vous recommandons de ne pas modifier ces paramètres à moins que vous souhaitiez ajouter un nouveau type de fichier ou que vous souhaitez volontairement définir une type de fichier sous une autre conformité.

Mode par défaut

Ce groupe vous permet de définir le mode par défaut à utiliser pour chaque type de fichier. Si une conformité particulière peut être consultée dans un seul mode, ce mode sera sélectionné par défaut et la sélection des modes sera désactivée. Par exemple, des documents XQuery Update peuvent uniquement être consultés dans le Mode Texte, ce mode est donc sélectionné par défaut et la sélection des modes est désactivée ; de même, les documents conformes Avro (binaires Avro) peuvent être consultés uniquement dans le Mode Grille.

Mode Grille

Cette case à cocher vous permet de définir si le Mode Grille doit automatiquement construire des tables.

Mode Texte

Cette case à cocher vous permet de configurer la coloration syntaxique pour les types de fichier particuliers.

Désactiver la validation automatique

Cette option vous permet de désactiver la validation automatique par type de fichier. La validation automatique a généralement lieu lorsqu'un fichier est ouvert ou enregistré ou lorsqu'un mode est modifié.

Utiliser RaptorXML Server pour valider lors de Ouvrir/Enregistrer

Spécifie si le RaptorXML Server doit être utilisé pour valider les fichiers du type de fichier sélectionné lorsque le fichier est ouvert et enregistré. Pour que cela fonctionne un RaptorXML Server doit être configuré.

Ajoute une nouvelle extension de fichier

Ajoute un nouveau type de fichier à la liste des Types de fichier. Vous devez ensuite définir les paramètres pour ce nouveau type de fichier en utilisant les autres options dans cet onglet.

Supprimer l'extension de fichier choisie

Supprime le type de fichier actuellement sélectionné et tous ses paramètres associés.

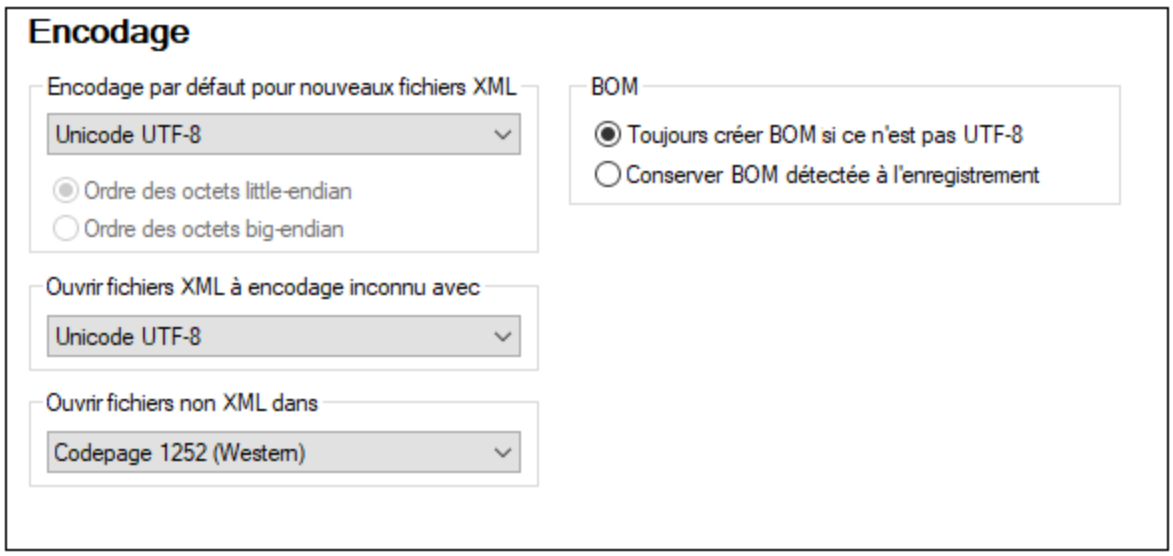
Enregistrer et quitter

Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer.

© 2016-2022 Altova GmbH

28.17.15.3 Encodage

La section **Encodage** spécifie les options pour les encodages de fichier.



Encodage par défaut pour les nouveaux fichiers XML

L'encodage par défaut pour les nouveaux fichiers XML peut être défini en sélectionnant une option depuis la liste déroulante. Un nouveau document est créé avec une déclaration XML contenant la valeur d'encodage que vous spécifiez ici. Si un encodage de deux ou quatre octets est sélectionné en tant que l'encodage par défaut (par ex. UTF-16, UCS-2 ou UCS-4) vous pouvez aussi choisir entre un ordre des octets little-endian et big-endian.

L'encodage des fichiers XML existants sera retenu et peut uniquement changer avec la commande **Fichier | Encodage**.

Ouvrir les fichiers XML à encodage inconnu comme

Si l'encodage d'un fichier XML ne peut pas être déterminé ou si le document XML n'a pas de spécification d'encodage, le fichier sera ouvert avec l'encodage que vous sélectionnez dans cette liste de choix.

Ouvrir fichiers non XML dans

Les fichiers existants et les nouveaux fichiers non-XML sont ouverts avec l'encodage que vous avez sélectionné dans cette liste de choix. Vous pouvez changer l'encodage du document en utilisant la commande **Fichier | Encodage**.

BOM (Byte Order Mark)

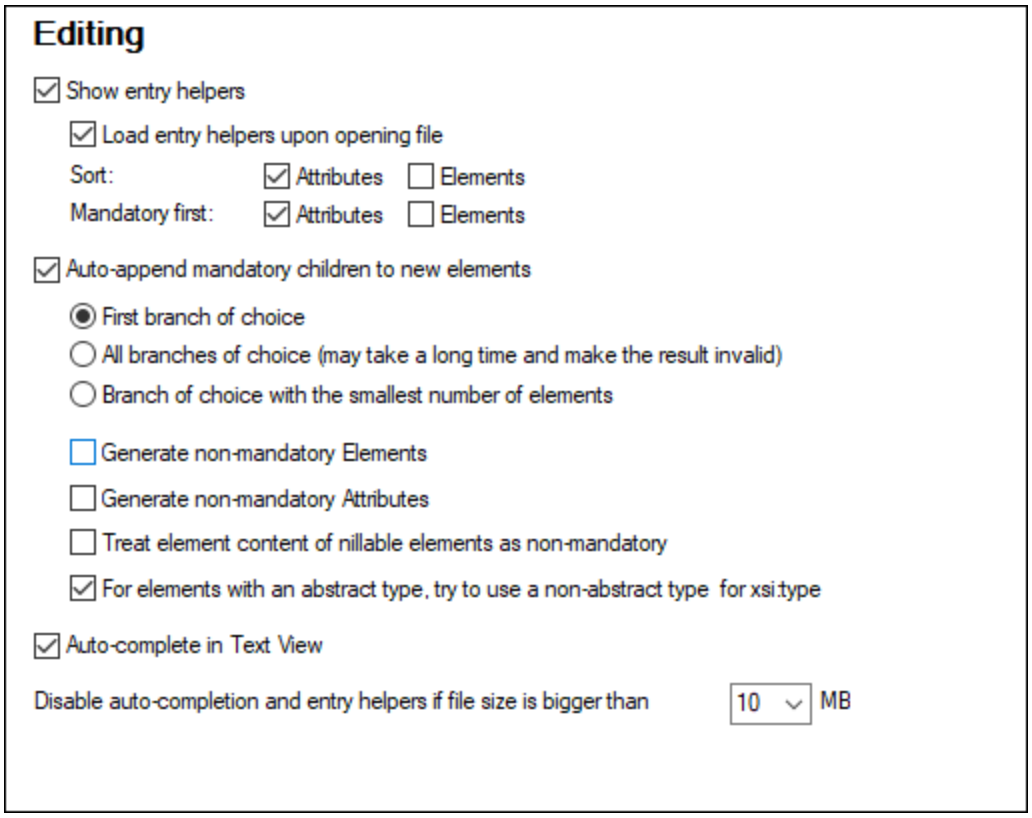
Lorsqu'un document à encodage de caractère deux octets ou quatre octets est enregistré, le document peut être enregistré soit avec (i) un ordre des octets little-endian et un BOM little-endian (*Toujours créer BOM si différent de UTF-8*); soit (ii) avec l'ordre des octets détecté et le BOM détecté (*Préserver BOM détecté à l'enregistrement*).

Enregistrer et quitter

Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer.

28.17.15.4 Édition

La section **Édition** vous permet de spécifier le comportement d'édition dans XMLSpy.



Assistants de saisie

Lors de l'édition de documents, XMLSpy fournit une édition intelligente sur la base de ces paramètres. Vous pouvez aussi personnaliser les différents aspects du comportement de l'Assistant à la saisie dans ce volet. Les paramètres de personnalisation effectués ici seront appliqués s'ils sont pertinents pour le type de fichier édité. Par exemple, l'option de charger les assistants à la saisie pour ouvrir le fichier et trier les attributs ne sera pas applicable aux documents DTD ou XQuery.

Les éléments enfant obligatoires des groupes `choice` sont auto-ajoutés sur la base des paramètres effectués dans ce volet. Vous pouvez sélectionner si (i) la première branche (élément) du groupe `choice`, (ii) toutes les branches, ou (iii) la branche avec le plus petit nombre d'éléments descendants est générée. Veuillez noter que la sélection *Toutes les branches* peut générer un document invalide puisque seule une branche provenant d'un groupe `choice` est autorisée.

© 2016-2022 Altova GmbH

Créer une structure XML depuis le Schéma XML

Lorsque vous créez un nouveau document XML qui est basé sur le Schéma XML, le document sera généré avec une structure qui est dérivée depuis les définitions dans le schéma. Les paramètres décrits ci-dessous déterminent certains aspects ambigus liés à la création de cette structure.

Auto-apposer des enfants obligatoires
Les éléments d'enfants obligatoires de groupes `choice` dans le schéma sont auto-apposés sur la base du paramètre défini dans ce panneau. Vous pouvoir choisir si (i) la première branche (élément) du groupe `choice`, (ii) toutes les branches, ou (iii) la branche avec le petit nombre d'éléments descendants sont générées. Veuillez noter que la sélection *Toutes les branches* pourrait générer un document invalide étant donné qu'une seule branche du groupe `choice` est autorisée.

Nœuds non obligatoires et éléments de type abstrait
Pour ajouter des éléments ou attributs non obligatoires, sélectionnez l'option respective. Si ces options ne sont pas sélectionnées, seuls des nœuds obligatoires seront ajoutés. Vous pouvez également (i) définir du contenu d'élément des éléments qui peuvent être annulés pour être définis comme non obligatoires, et (ii) tenter d'utiliser un type non abstrait comme élément `xsi:type` d'un type abstrait.

Mode Texte

L'option *Saisie automatique* ajoute automatiquement des composants structurels non ambigus. Par exemple, lors de la saisie du crochet de fermeture de la balise de démarrage d'un élément, alors la balise de fin de cet élément est ajouté automatiquement si l'option est activée.

Dans le Mode Texte, la saisie automatique et les assistant à la saisie peuvent être désactivés si un fichier est plus important que la taille spécifiée dans la liste de choix *Désactiver saisie automatique*. Cela est utile si vous souhaitez accélérer l'édition de fichiers volumineux et que vous pouvez vous débrouiller sans la fonction de saisie automatique et des assistants à la saisie. Si la taille du fichier est plus grande que celle spécifiée pour cette option, le menu contextuel Mode Texte contient une commande à bascule pour allumer et éteindre ces fonctions. Vous pouvez donc toujours allumer et éteindre ces aides à l'édition à tout moment pendant l'édition (Dans le cas de fichiers dont la taille est plus grande que la taille spécifiée pour cette option). Si la valeur spécifiée pour cette option est plus petite que la taille du fichier ouvert, les emplacements indiqués dans les messages d'erreur ne correspondront pas exactement à l'emplacement dans le Mode Texte.

Enregistrer et quitter

Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer.

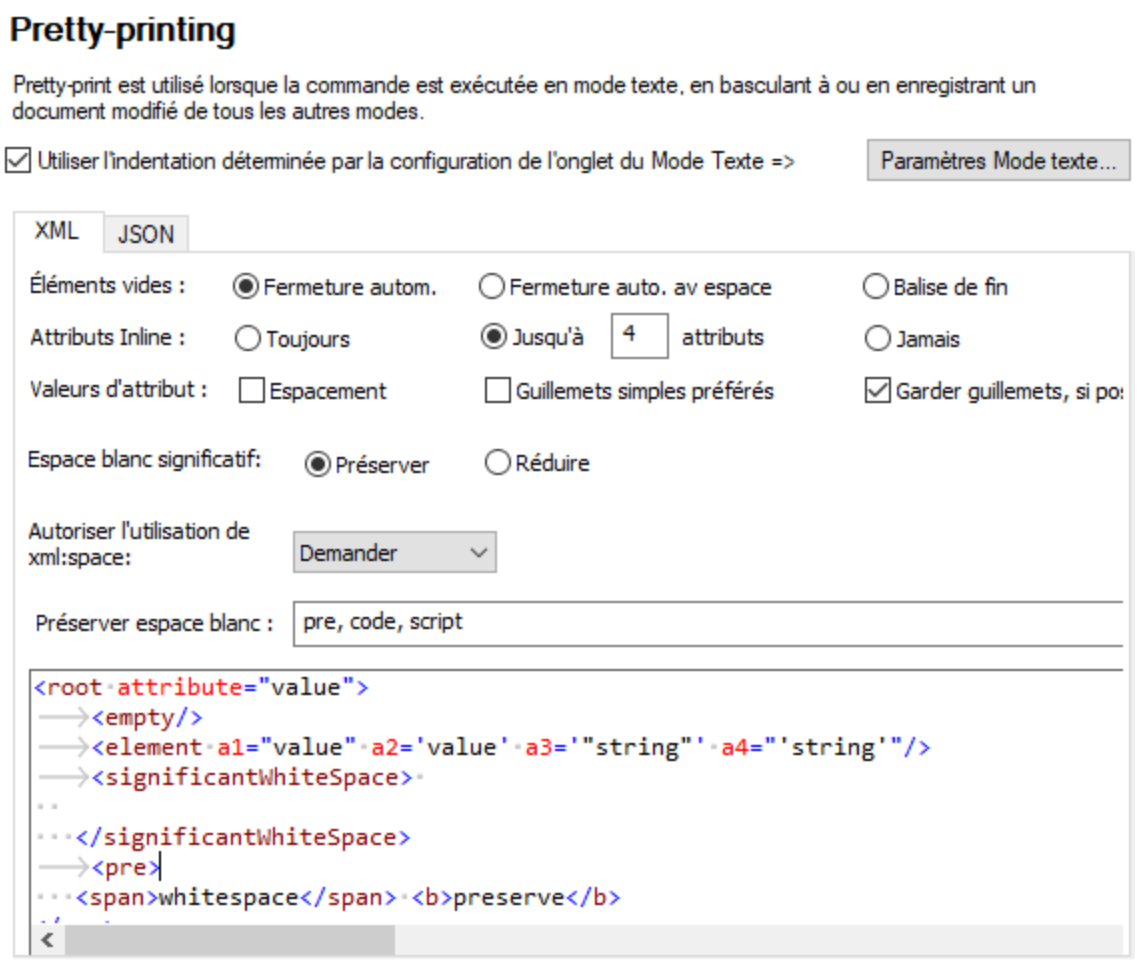
28.17.15.5 Pretty Printing

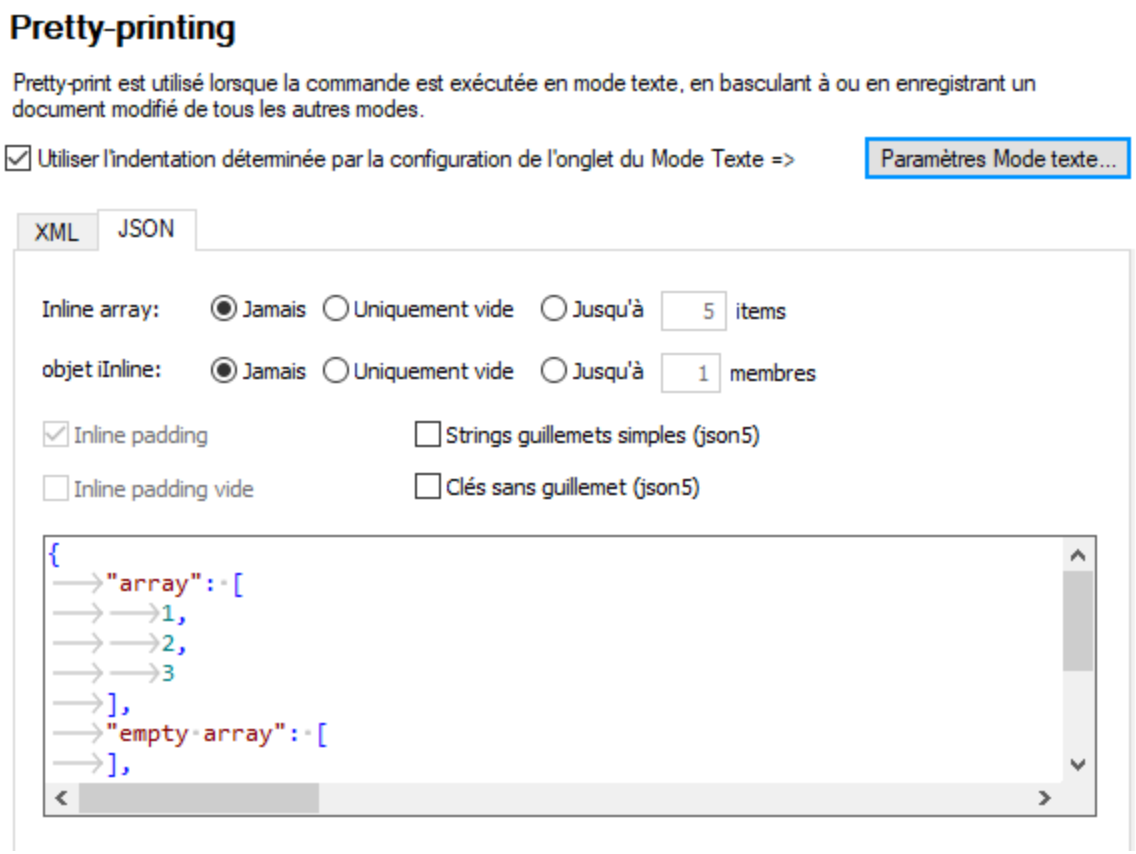
La section **Pretty Printing** *(voir la capture d'écran ci-dessous)* vous permet de spécifier comment le texte est affiché dans le Mode Texte. Les définitions dans cette section sont groupées dans les catégories suivantes :

- Paramètres XML (sélectionnez le tab XML pour accéder ces paramètres ; *voir la capture d'écran ci-dessous gauche*)
- Paramètres JSON (sélectionnez le tab JSON pour accéder à ces paramètres ; *voir la capture d'écran ci-dessous droite*)
- Paramètres du Mode Texte (cliquez sur la touche pour accéder au dialogue des paramètres)

La case à cocher *Utiliser indentation* active/désactive pretty-printing. La case à cocher *pretty-print automatiquement dans le Mode Texte* peut être sélectionnée pour appliquer automatiquement l'option pretty-printing lorsqu'un document est chargé.

© 2016-2022 Altova GmbH


Pretty-printing
Pretty-print est utilisé lorsque la commande est exécutée en mode texte, en basculant à ou en enregistrant un document modifié de tous les autres modes.
Utiliser l'indentation déterminée par la configuration de l'onglet du Mode Texte =>
Paramètres Mode texte...
XML
JSON
Éléments vides :
Fermeture autom.
Fermeture auto. av espace
Balise de fin
Attributs Inline :
Toujours
Jusqu'à
4
attributs
Jamais
Valeurs d'attribut :
Espacement
Guillemets simples préférés
Garder guillemets, si po
Espace blanc significatif:
Préserver
Réduire
Autoriser l'utilisation de xml:space:
Demander
Préserver espace blanc :
pre, code, script
<root attribute="value">
<empty/>
<element a1="value" a2='value' a3='"string"' a4="'string'"/>
<significantWhiteSpace>
</significantWhiteSpace>
<pre>
whitespace preserve



Paramètres XML

Les paramètres suivants sont disponibles :

- Comment les éléments vides sont écrits et affichés dans le document : avec un tag (*Self-closing*) ou deux tags (*End tag*).
- Si les attributs sont affichés inline (sur la même ligne que son élément parent) ou non. Les attributs sont affichés inline si *Toujours* est sélectionné, ou si *Jusqu'à X attributs* est sélectionné et le nombre d'attributs n'excède pas X. Les attributs sont affichés sur de nouvelles lignes si *Jamais* est sélectionné ou si *Jusqu'à X attributs* est sélectionné et le nombre d'attributs n'excède pas X.
- Comment les valeurs d'attributs sont écrites : (i) avec des espaces de chaque côté du signe égal ou non ; (ii) si des valeurs sont incluses entre guillemets simples ou guillemets doubles ; (iii) si les guillemets dans le texte de source sont préservés tels que vous les avez saisis, ou s'ils sont écrasés par d'autres options (telles que *Guillemets simples préférés*) ; veuillez noter que, si sélectionné, *Garder les guillemets* est appliqué uniquement s'il est possible de le faire sans invalider le document.
- Préserver ou comprimer les espaces blancs. Les caractères d'espace blanc sont : les espaces, le tabulateur, le retour de chariot et le saut de ligne. Voir la section *Espace blanc* pour plus de détails.
- Si les éléments d'un document contiennent la paire attribut-valeur **`xml:space="preseve"`**, alors vous pouvez spécifier, avec le paramètre *Permettre l'utilisation de xml:space*, si cette paire attribut-valeur devrait être traitée lors du pretty-printing. L'option *Toujours* spécifie que l'intention de l'attribut doit être suivie lors du pretty-printng de tout document : les espaces blancs dans les éléments respectifs seront préservés et ces éléments ne seront pas pretty-printed. L'option *Jamais* fait que l'attribut *xml:space* est ignoré et que les éléments respectifs sont en pretty-print. L'option *Demander* incite XMLSpy à

© 2016-2022 Altova GmbH

demander ce qui doit être fait chaque fois qu'un document contenant **`xml:space="preseve"`** est en pretty-printed.
- Définir quels éléments préserveront des espaces blancs
- Si l'indentation spécifiée dans le dialogue Paramètres du Mode Texte est utilisée ou si aucune indentation n'est utilisée (spécifiée via la case à cocher au-dessus du volet)

Lorsque vous sélectionnez une option, ses effets sont affichés dans le volet de visualisation en bas du dialogue, ce qui vous permet de voir l'effet avant de confirmer avec **OK**.

Paramètres JSON

Pour chaque option que vous choisissez, l'effet est indiqué immédiatement dans le volet de visualisation en bas du dialogue.

- *Inline Array :* Affiche les éléments d'un array sur une seule ligne (ou inline). Autrement, les éléments pourraient être sur des lignes séparées. Vous pouvez choisir de ne jamais appliquer de formatage inline aux arrays, ou d'appliquer le formatage inline aux arrays vides ou aux arrays jusqu'à une taille spécifique.
- *Objet Inline :* Affiche les propriétés d'un objet sur une seule ligne (ou inline). Autrement, les propriétés pourraient être sur des lignes séparées. Vous pouvez choisir de ne jamais appliquer de formatage inline aux objets, ou d'appliquer le formatage inline aux objets vides ou aux objets jusqu'à un nombre spécifié de propriétés.
- *Remplissage Inline :* Si elle est sélectionnée, la fonction ajoute des espaces entre les éléments d'arrays inline non vides et des objets inline non vides. Cette option est activée uniquement si l'array inline non vide ou l'option d'objet inline non vide a été sélectionné.
- *Remplissage Inline Vide :* Si elle est sélectionnée, la fonction ajoute des espaces à l'intérieur des délimiteurs d'arrays inline vides et d'objets inline vides. L'option est activée si soit un array ou un objet est défini pour être inline (vide ou non vide).
- *Strings à guillemet unique (JSON5) :* Si elle est sélectionnée, la fonction convertit tous les guillemets dans les documents JSON5 à guillemets uniques.
- *Strings sans guillemets (JSON5) :* Si elle est sélectionnée, elle supprime, dans les documents JSON5, des guillemets de toutes les clés (des paires `key:value`).

Vous pouvez spécifier via la case à cocher au-dessus du volet si vous voulez utiliser l'indentation spécifiée dans le dialogue Paramètres du Mode Texte ou si aucune indentation ne doit être utilisée.

Paramètres du Mode Texte

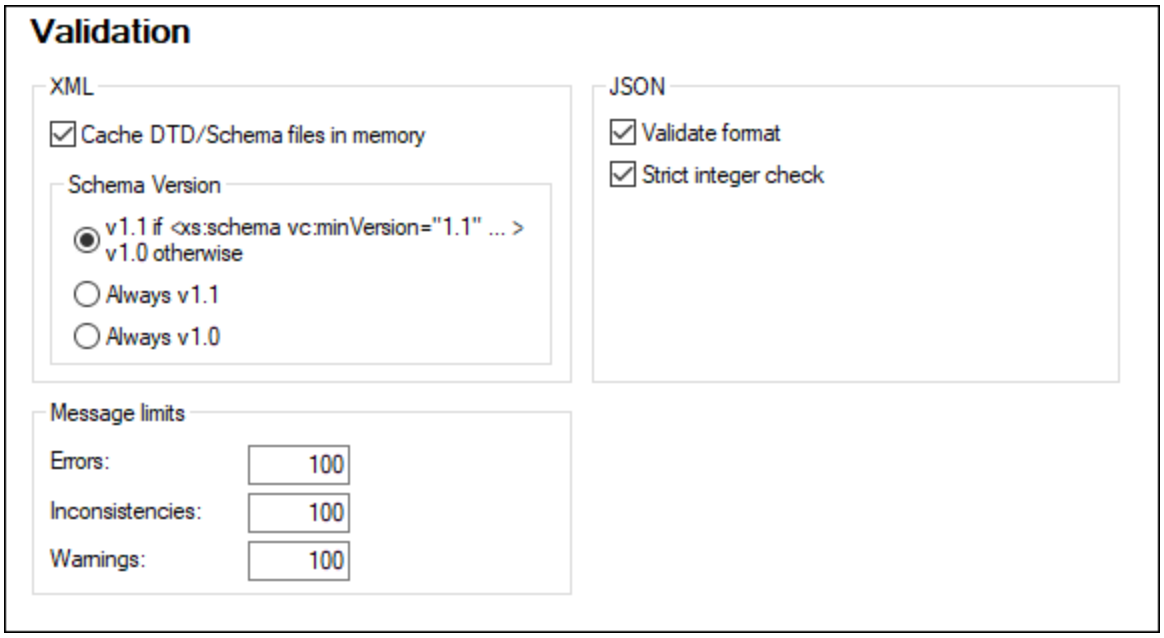
Cliquez sur **Paramètres du Mode Texte** pour ouvrir le dialogue des Paramètres du Mode Texte, où vous pouvez activer les propriétés du Mode Texte telles que l'indentation, les marges de signets et la mise en surbrillance automatique. Le dialogue des Paramètres du Mode Texte peut aussi être accédé par la commande de menu **Mode | Paramètres du Mode Texte**. Le dialogue est décrit ici.

Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.6 Validation

La section **Validation** vous permet de spécifier des options pour valider les documents XML et JSON

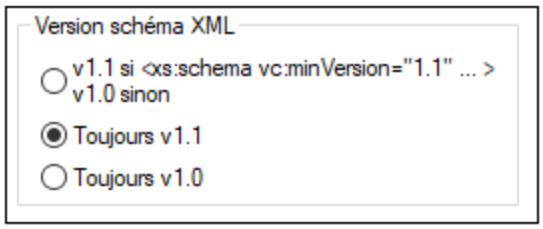


XML

XMLSpy peut mettre en cache des fichiers de schéma DTD et XML en mémoire pour éviter un rechargement non nécessaire (par exemple, quand le schéma n'est pas local mais est accédé via un URL). Veuillez noter, toutefois, que si vous utilisez des versions en cache de schémas, les modifications apportées à votre schéma ne seront pas immédiatement reflétées quand vous les validez ; dans ce cas, vous devrez recharger le fichier XML ou redémarrer XMLSpy.

Version Schéma

Le mode XSD qui est activé dans le mode Schéma dépend de (i) la présence/l'absence - et, si présent, la valeur - de l'attribut `/xs:schema/@vc:minVersion` du document XSD, et (ii) l'option de la version de Schéma XML sélectionnée dans la section Fichier du dialogue des Options (**Outils | Options**, *voir la capture d'écran ci-dessous*).



Les situations suivantes sont possibles. La *Version de schéma XML* dans la table ci-dessous fait réf´rence à la sélection dans le volet de la version de schéma XML ci-dessus. Les valeurs `vc:minVersion` dans la table se

 © 2016-2022 Altova GmbH

réfèrent à la valeur de l'attribut `xs:schema/@vc:minVersion` dans le document de schéma XML. Pour plus de détails, voir la section Éditer les Modes | Mode Schéma | ModeXSD.

Version Schéma XML	attribut `vc:minVersion`	Mode XSD
Toujours v1.0	est absent, ou est présent avec toute valeur	1.0
Toujours v1.1	est absent, ou est présent avec toute valeur	1.1
Valeur de @vc:minVersion	Attribut a une valeur de `1.1`	1.1
Valeur de @vc:minVersion	Attribut est absent, ou attribut est présent avec une valeur différente à `1.1`	1.0

Limites de message

Ces options vous permettent de définir les limites séparées pour le nombre d'erreurs, inconsistances XBRL et avertissements qui sont affichés. Le nombre par défaut pour chaque catégorie est de 100. Modifiez-le au nombre souhaité.

JSON

Les options de validation suivantes pour la validation de document JSON sont disponibles :

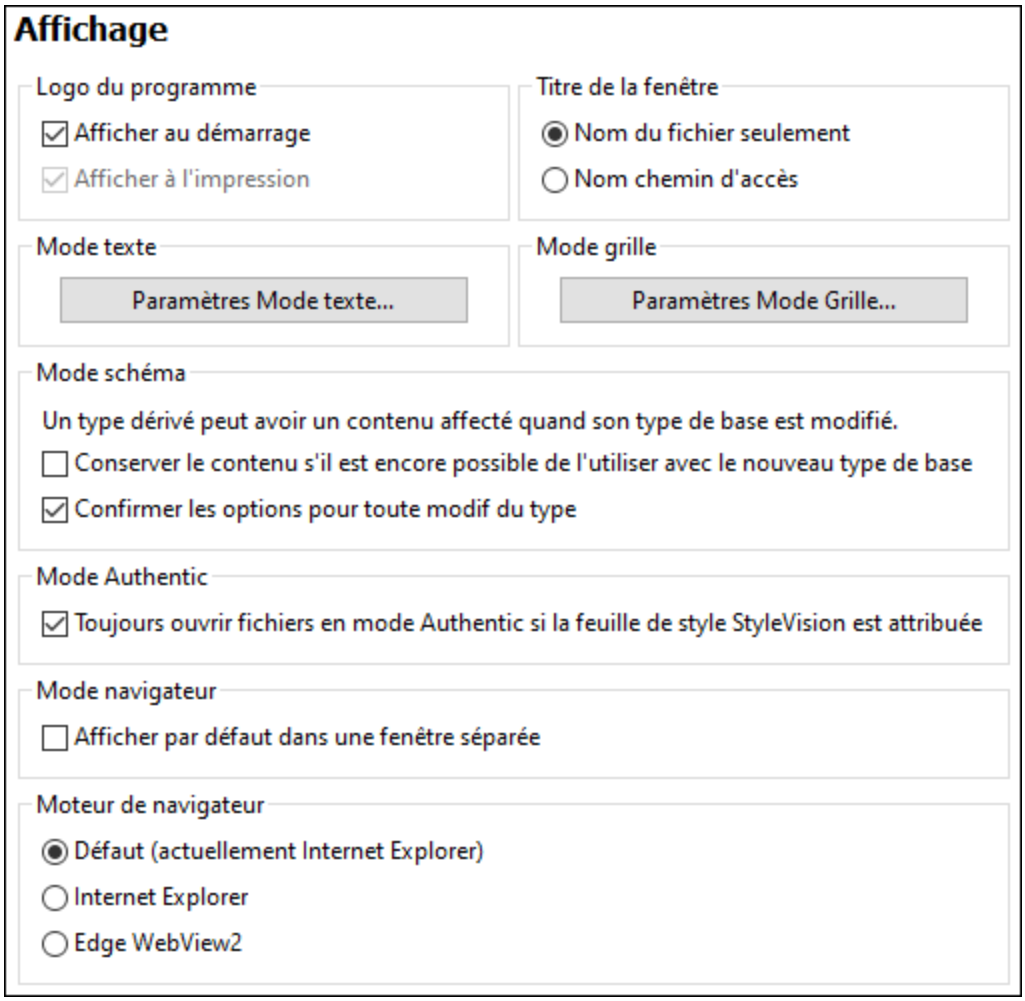
- *Valider le format :* le format de types de string dans les documents d'instance JSON est validé.
- *Vérification d'entier stricte :* il existe deux types numériques JSON : `number` et `integer`. Cette option vérifie que les entiers sont de type entier (et pas, par exemple, des nombres à virgule flottante (par exemple, `7.0`), ou signé (par exemple, `+7`), ou des strings (par exemple, `"7"`).

Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.7 Affichage

La section **Affichage** vous permet de personnaliser la présentation des documents XML dans XMLSpy.



Logo du programme

Vous pouvez désactiver l'écran de démarrage sur démarrage du programme pour rendre l'application plus rapide. De même, si vous avez une licence achetée (comparé à, par exemple, une licence d'évaluation), vous aurez l'option de désactiver le logo de programme, la notice de copyright et les détails d'inscription lorsque vous imprimez un depuis XMLSpy.

Titre de fenêtre

Le titre de fenêtre pour chaque fenêtre de document peut contenir soit uniquement le nom de fichier ou le nom entier du chemin.

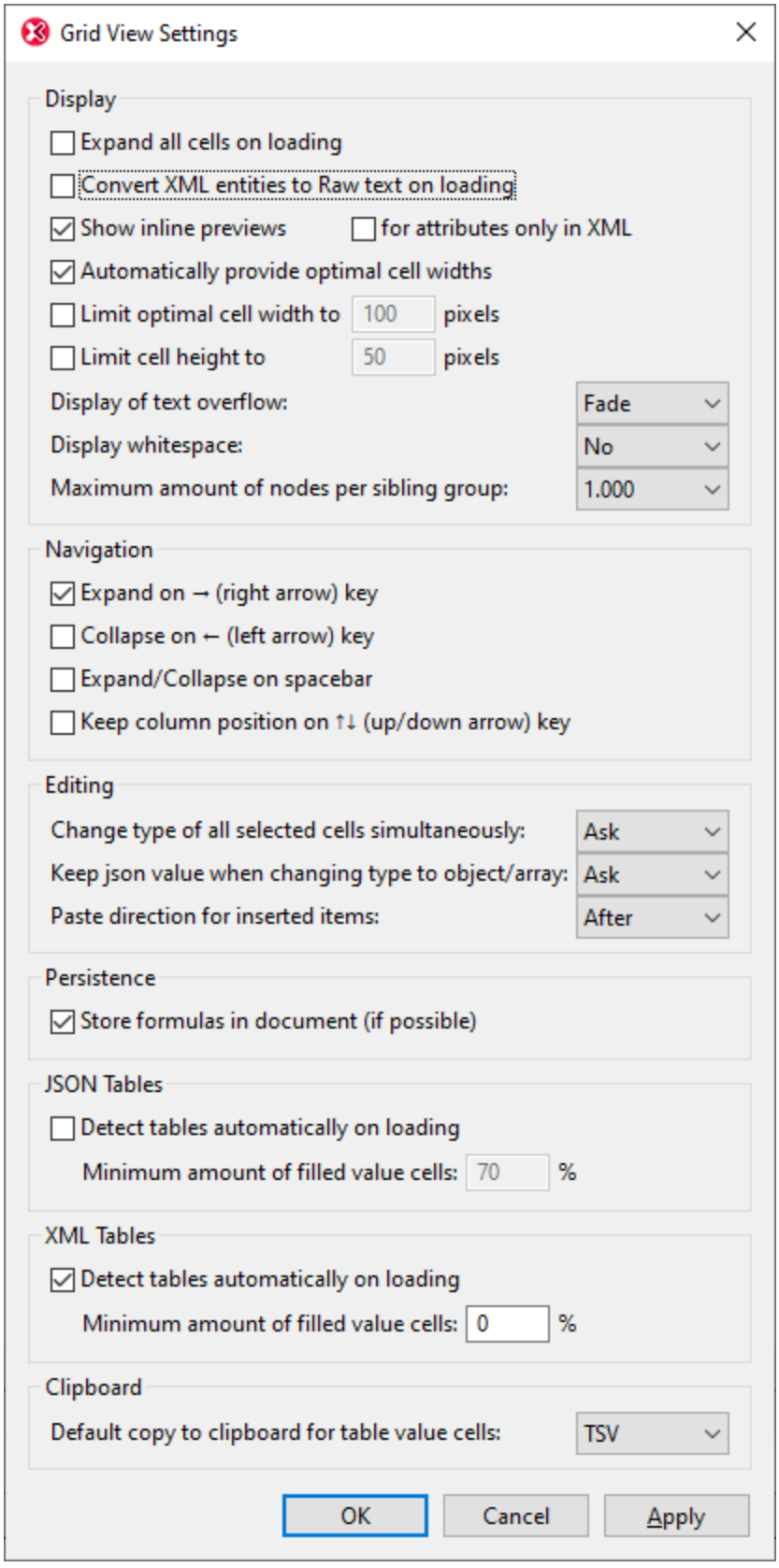
© 2016-2022 Altova GmbH

Paramètres du Mode Texte

Cliquez sur les **Paramètres du Mode Texte** pour ouvrir le dialogue des Paramètres du Mode Texte, où vous pouvez activer les propriétés du Mode Texte telles que l'indentation, les marges de signets et l'auto-mise en surbrillance. Le dialogue des Paramètres du Mode Texte peut aussi être accédé par la commande de menu **Mode | Paramètres du Mode Texte**. Le dialogue est décrit ici.

Paramètres du Mode Grille

Cliquez sur les **Paramètres du Mode Grille** pour ouvrir le dialogue des Paramètres du Mode Grille. Les options suivantes pour le Mode Grid peuvent être définies :

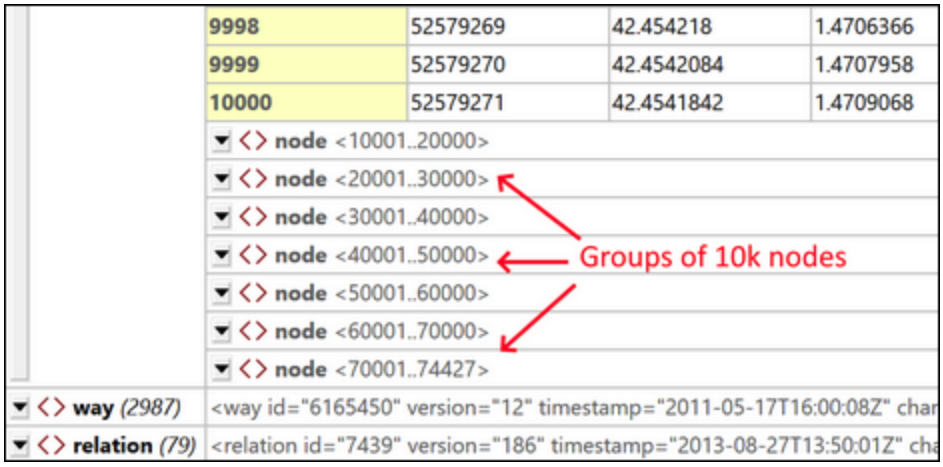


© 2016-2022 Altova GmbH

Afficher

Les cases à cocher dans la section *Affichage* sont assez explicites. Veuillez trouver ci-dessous quelques clarifications.

- Si les cellules ne sont pas agrandies lors du chargement, le nœud racine et tous ses descendants sont compressés. Vous devrez étendre chaque nœud au fur et à mesure que vous plongez au cœur du document.
- Les largeurs optimales peuvent être changées automatiquement (i) lorsque le document est chargé dans le Mode Grille, et (ii) lorsqu'un ensemble d'éléments est permuté entre l'affichage de table et le Mode Grille normal.
- Lorsque les largeurs optimales sont activées, toute la largeur des cellules est affichée pour pouvoir placer le texte dans une seule ligne. Pour y arriver, les points suivants pourraient se produire : (i) la table pourrait s'élargir et dépasser la fenêtre principale, résultant en une barre de défilement qui apparait en bas de la fenêtre ; (ii) le texte dans quelques cellules s'élargira au-delà de la limite droite de la cellule (ou limite gauche, dépendant de la direction de l'écriture) ; le dépassement de texte sera indiqué par la disparition en fondu du texte ou par une ellipse (*voir le prochain point*).
- Si le texte n'est pas encapsulé et la largeur du texte est supérieure à la largeur de la cellule, l'option *dépassement de texte* vous laisse choisir si vous voulez indiquer le dépassement par une disparition en fondu du texte près de la limite ou par une ellipse.
- Les nœuds siblings peuvent être affichés en groupes frères de 100 nœuds, 1k ou 10k nœuds *(voir la capture d'écran ci-dessous)*. Ceci est utile si vous voulez garder de l'espace d'affichage dans les fichiers qui contiennent un grand nombre de nœuds frères.



Édition

Les cases à cocher dans la section *Éditer* sont assez explicites. Veuillez trouver ci-dessous quelques clarifications.

- Lorsque vous modifiez le type de cellules multiples sélectionnées, vous avez les options suivantes qui définissent si vous pouvez poursuivre l'action du paramètre : *Toujours, Jamais,* ou *Demander* (pour la décision de l'utilisateur).
- Lorsque vous changez un type JSON de type atomique à un type objet ou array, vous pouvez choisir entre les options suivantes qui vous permettent de continuer ou non avec l'action du paramètre : (i) *Demander* (si la valeur du type atomique devrait être maintenue comme valeur d'une paire d'enfants sans nom **`key:value`** ou ignorée), (ii) *Toujours* (maintenir la valeur d'une paire d'enfants sans nom **`key:value`**), (iii) *Jamais* (maintenir la valeur, à savoir ignorer la valeur).
- L'option de la *direction Coller* détermine si une sélection dans le presse-papiers est collée au-dessus ou en dessous de la cellule sélectionnée.

Persistance
Les filtres et formules peuvent être enregistrés dans les métadonnées des documents XML et JSON. Les formules sont stockées comme instructions de traitement dans les documents XML et comme commentaires dans certains types de documents JSON. Ces paramètres permettent de spécifier comment procéder avec le stockage des filtres et des formules. Dans le cas de formules, la terminologie, *si possible*, fait référence au fait que les formules peuvent être enregistrées en tant que commentaires uniquement dans des documents JSON5 et JSONC—et pas tous les documents JSON.

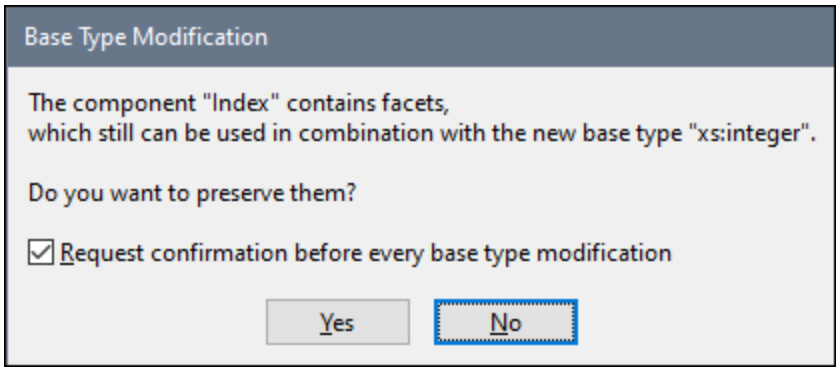
Tables
Les paramètres suivants sont disponibles :

- Si vous sélectionnez le paramètre qui détecte les tables automatiquement lors du chargement, vous pouvez choisir le pourcentage minimum des cellules de la table remplies qui qualifient les tables comme tables à être détectées. Si le nombre de cellules de table remplies ne dépasse pas ce niveau, la structure est affichée comme grille normale avec les éléments répétitifs recensés l'un en dessous de l'autre.
- Vous pouvez aussi choisir si les contenus du presse-papiers doivent être stockés comme valeurs séparées par tabulation (TSV), ou comme XML/JSON (dépendant du type de document). Ceci est une fonction très utile : Si vous souhaitez coller une table depuis le presse-papiers dans un autre document, le paramètre vous permet de choisir si la table copiée est stockée comme TSV ou avec un marquage. (Pour voir la différence, collez une table dans un éditeur de texte après avoir copié la table vers le presse-papiers dans chacun des formats.)

Mode Schéma

Un type de données Schéma XML peut être dérivé depuis un autre type de données. Par exemple, un type de données pour les éléments d'e-mail peut être dérivé depuis un type de données de base de `xs:string` (par exemple, en limitant le type de données `xs:string` à un ensemble spécifique de caractères). Si le type de données est changé ultérieurement, vous pouvez définir les options suivantes :

- *Préserver le contenu :* Si les définitions utilisées pour définir le type dérivé peuvent être utilisée avec un nouveau type de base, vérifier cette option préservera automatiquement les définitions.
- *Confirmer pour chaque modification* : Après avoir modifier le type de base, un dialogue (*voir la capture d'écran ci-dessous*) s'ouvrira demandant si les anciennes définitions doivent être préservées et utilisées avec le nouveau type de base.



 © 2016-2022 Altova GmbH

Authentic View

Les fichiers XML sur **StyleVision Power Stylesheet** sont automatiquement ouverts dans Authentic View lorsque cette option est active.

Mode Navigateur

Vous pouvez choisir de visualiser le mode navigateur dans une fenêtre séparée, permettant de placer les modes éditer et navigateur côte-à-côte.

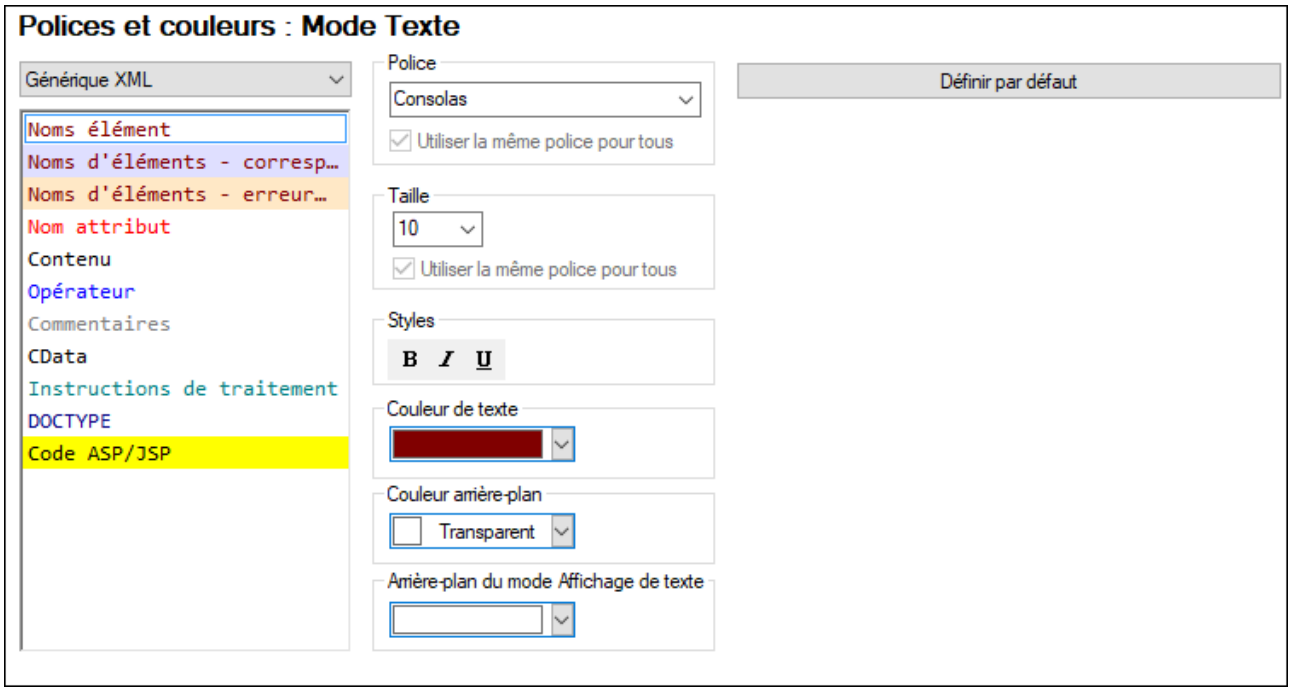
Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.8 Polices et couleurs

La section **Polices et couleurs** vous permet de personnaliser les options pour l'apparence des items de texte dans les différents modes de XMLSpy.

Note : Les options de police et couleurs s'appliquent au thème actif actuel. Pour modifier les polices et couleurs dans un autre thème, faites-en le thème actif avant de changer les options.



Les options ci-dessous sont disponibles pour les modes suivants :

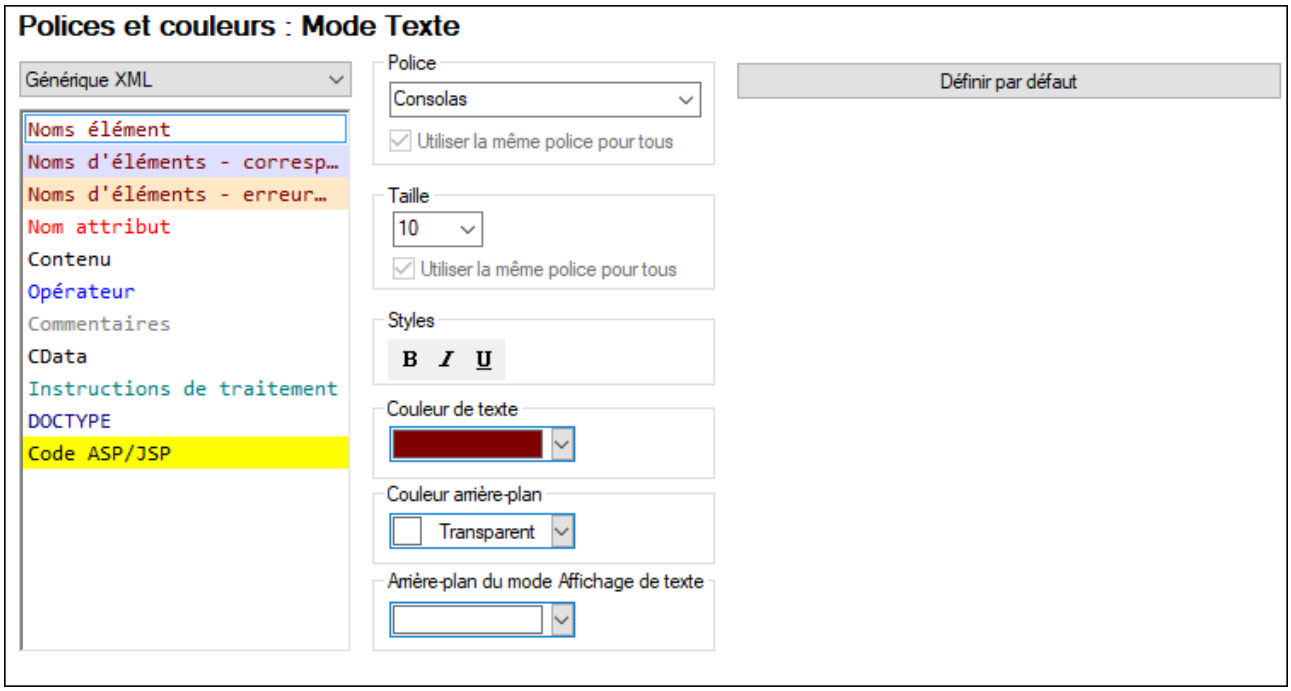
- Mode Texte
- Mode Grille
- Mode Design schéma

- Mode Design WSDL
- Mode taxonomie XBRL

Dans le panneau de gauche du dialogue, choisir le mode que vous souhaitez personnaliser. Les types d'item de texte qui peuvent être formatés sont affichés dans le volet de droite (*capture d'écran ci-dessus*). Choisir le type d'item de texte que vous souhaitez formater, puis l'attribuer aux valeurs de propriété de formatage désirées.

28.17.15.8.1 Mode Texte

La section **Mode Texte** vous permet de personnaliser, pour le thème actif actuel, l'apparence des types text-item individuels dans les différents types de documents (*voir capture d'écran ci-dessous et la liste des documents dans la section Types de document ci-dessous*). Par exemple, vous pouvez spécifier des formatages différents pour l'affichage des noms d'élément et d'attributs dans les documents XML, ou pour l'affichage de mots-clés et de variables dans les documents XQuery.



Note : Le formatage sera défini pour le thème actif actuel. Pour définir le formatage d'un autre thème, faites de l'autre thème le thème actif.

Types de document

Vous pouvez sélectionner un type de document spécifique dans la zone de liste déroulante dans la partie supérieure gauche du dialogue, puis définir le formatage des items de texte du type de document.

Les items de texte des types de documents suivants peuvent être formatés :

- Générique XML
- XQuery

© 2016-2022 Altova GmbH

- CSS
- json
- Famille C
- Python
- XULE
- Divers
- Sortie

Comment personnaliser

Pour personnaliser des types d'item de texte individuels d'un type de document particulier, procéder comme suit :

1. Dans la liste de choix située en haut à gauche, choisir le type de document pour lequel vous souhaitez personnaliser du texte. Ainsi, les types d'item de texte pour ce type de document apparaissent dans la fenêtre située en-dessous de la liste de choix. (*Dans la capture d'écran ci-dessus,* `XML generic` *a été sélectionné en tant que le type de document.*)
2. Choisir le type d'item de texte que vous souhaitez personnaliser en cliquant dessus. (*Dans la capture d'écran ci-dessus,* `Element names` *a été sélectionné.*)
3. Définir les propriétés de police du type d'item de texte sélectionné en utilisant les options dans les volets du côté droit. L'option *Arrière-plan du mode Affichage de texte* permet la sélection d'une couleur d'arrière-plan pour tout le Mode Texte.

Veuillez prendre note des points suivants :

- La même police, taille et style sont utilisés pour **tous les types d'item de texte** d'un type de document particulier (comme le type de document de XML générique). Dans le cadre d'un type de document, seule la couleur du texte et la couleur d'arrière-plan peuvent être modifiés. Ceci active la fonction de coloration de la syntaxe.
- Dans la catégorie *XML générique*, le type de texte *Noms d'éléments* consiste en trois sous-types : (i) *Noms d'éléments* s'applique aux noms d'élément qui ne sont pas sélectionnés ; (ii) *Noms d'éléments - correspondance* s'applique aux noms d'élément qui sont sélectionnés (les noms dans lequel le curseur est placé) et lorsque le nom de balise de démarrage correspond au nom de balise de fin ; (iii) *Noms d'éléments - erreur de correspondance* s'applique aux noms d'élément qui sont sélectionnés, mais dont le nom de balise de démarrage ne correspond pas au nom de balise de fin. Les noms d'élément qui sont édités seront donc marqués avec des couleurs d'arrière-plan différentes selon le fait que le nom de balise de démarrage corresponde au nom de balise de fin ou pas. Ces couleurs de marquage peuvent être modifiées en changeant les couleurs d'arrière-plan respectives. Le marquage est activé par défaut et peut être éteint en décochant l'option *Marquer les éléments* dans le dialogue Paramètres Mode Texte.
- Les paramètres contenus dans la catégorie XULE s'appliquent : (i) au document XULE dans la fenêtre principale, et (ii) aux règles XULE qui sont saisies dans la fenêtre XULE.
- Dans la catégorie *Divers* : (i) *Sélection* se réfère au contenu de texte sélectionné actuellement ; *sélection inactive* se réfère à d'autres occurrences dans le document du même contenu de texte ; (ii) *Trouver le marqueur actif* se réfère à l'occurrence sélectionnée actuellement d'un résultat de recherche, alors que *Trouver le marqueur* se réfère à d'autres occurrences (inactives) du résultat de recherche ; (iii) *Déboguer/Appeler marqueur* fait appel à l'étape sélectionnée actuellement dans une session de débogage ; (iv) *Espace blanc visible* fait référence aux marqueurs d'espace blanc dans un document. L'espace blanc dans un document peut être rendu visible en passant aux marqueurs d'espace blanc. (v) *Back-mapping actif* se réfère au contenu sélectionné actuellement dans le trio de document de back-mapping (**`XML—XSLT/XQuery—Result`**); *Back-mapping inactif* se réfère au contenu back-mappé dans les deux autres documents.

Définir par défaut

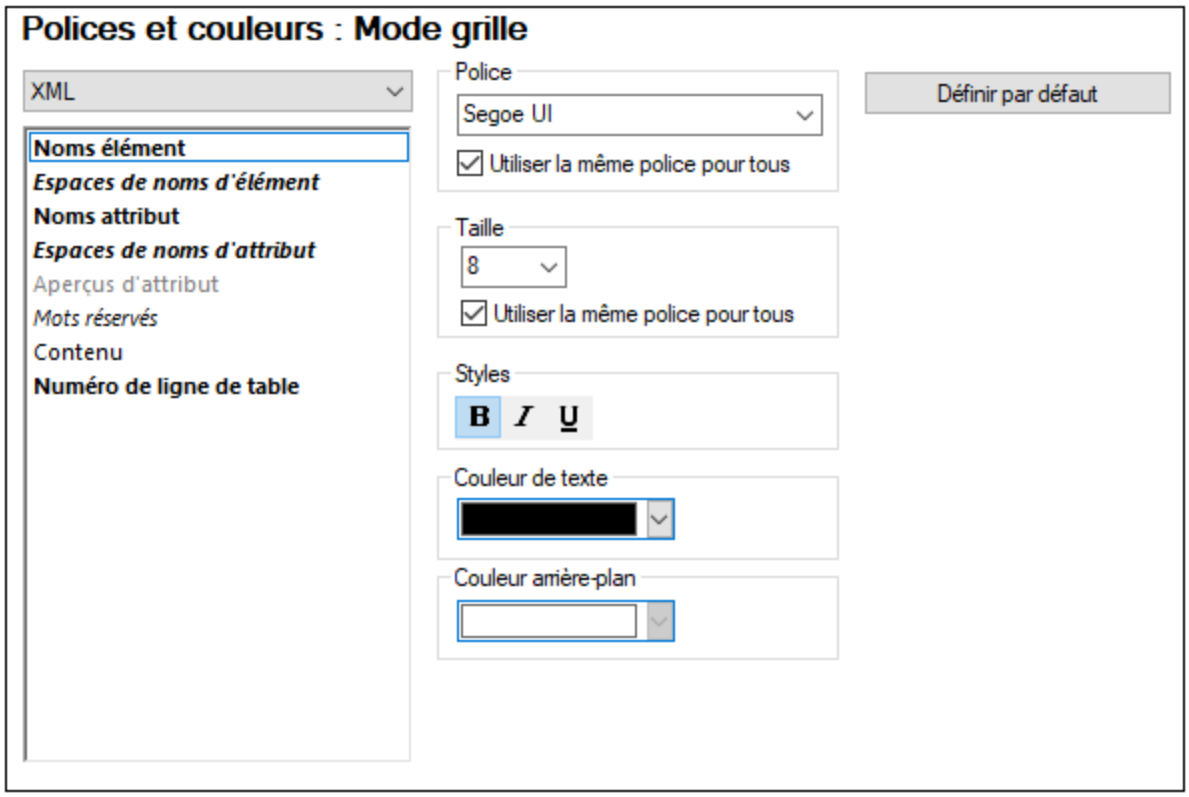
La touche **Définir par défaut** réinitialise les polices aux paramètres d'installation d'origine.

Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.8.2 Mode Grille

La section **Mode Grille** (*capture d'écran ci-dessous*) vous permet de personnaliser l'apparence du texte dans XML Grid View, JSON Grid View et Avro Grid View. L'option *Couleurs de cellule (voir la capture d'écran)* vous permet de définir les couleurs de cellules quand les composants grille sont affichés comme tables. Dans la liste de choix, sélectionnez l'affichage de document pour lequel vous souhaitez configurer le Mode Grille. Ensuite, sélectionnez le type d'item de texte que vous souhaitez formater et attribuez-le aux propriétés de formatage désirées (*listé ci-dessous*).



Note : Le formatage sera défini pour le thème actif actuel. Pour définir le formatage d'un autre thème, faites de l'autre thème le thème actif.

XML Grid, JSON Grid, Avro Grid

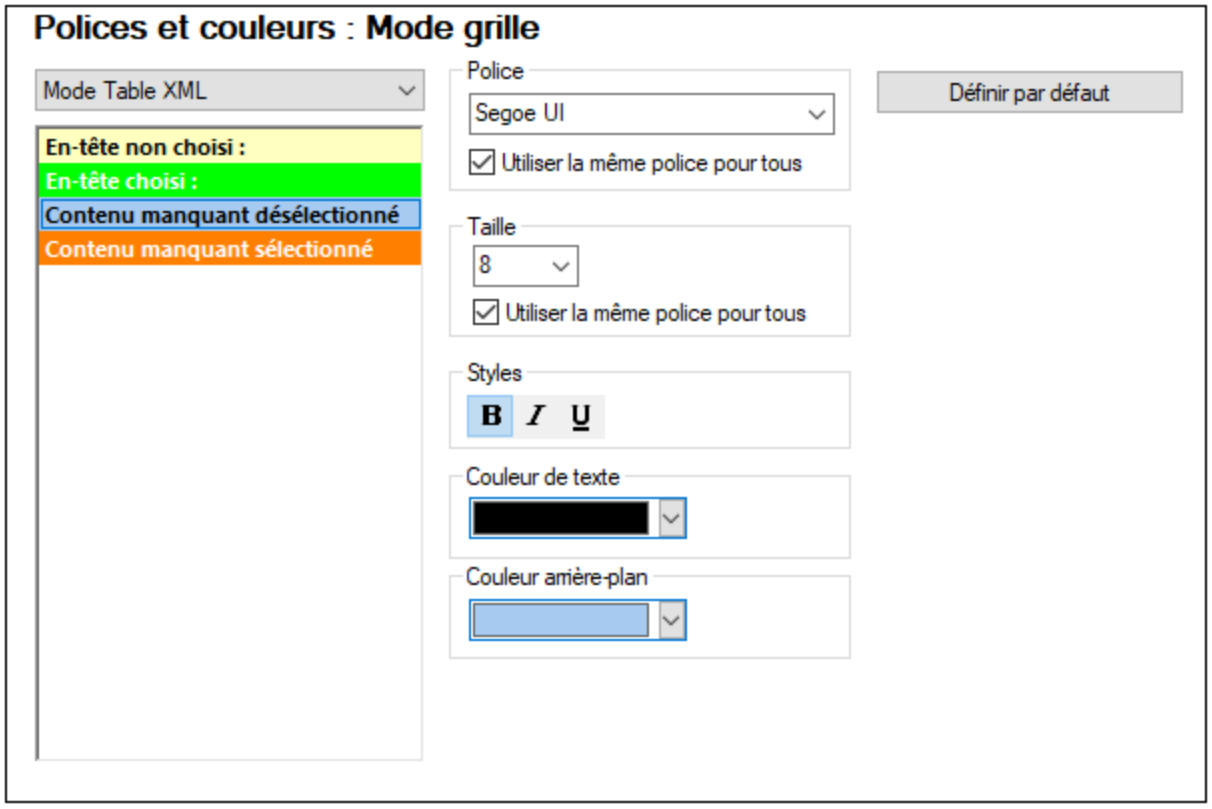
Vous pouvez définir les propriétés suivantes pour le type d'item de texte sélectionné :

© 2016-2022 Altova GmbH

- *Police et taille :* La deuxième police sera aussi utilisée en imprimés de l'Affichage Grille. Si vous souhaitez utiliser la même police et la même taille de police pour tous les types d'item de texte, cochez la case respective *Utiliser la même pour tous*.
- *Style de texte, couleur de texte, arrière-plan de texte :* Définit le style, la couleur et l'arrière-plan des types d'items de texte individuels. Les paramètres actuels sont immédiatement reflétés dans la liste du panneau de gauche, de manière à ce que vous puissiez visualiser l'apparence de l'item de texte

Couleurs de cellule (des Affichages de table dans toutes les Grilles)

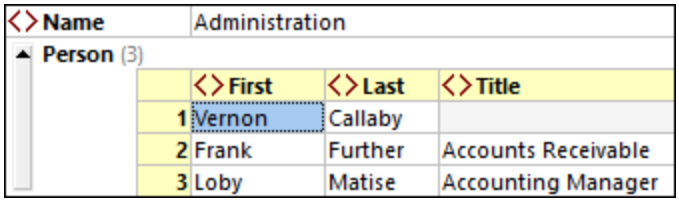
Les affichages de Table dans toutes les grilles (XML, JSON, Avro) peuvent aussi être configurées non seulement sur la base de leurs sémantiques liées aux langages (par exemple, un élément XML peut être formaté différemment qu'un commentaire XML). Vous pouvez aussi formater le Mode Grille sur la base d'une fonction de cellule : par exemple, pour différencier l'état sélectionné/non-sélectionné des différents composants de table. Une telle différentiation est obtenue en attribuant des couleurs d'arrière-plan variées à chaque item (*voir capture d'écran ci-dessous*), mais vous pouvez aussi utiliser d'autres propriétés de formatage et additionnelles pour différents types de cellules.



Les paramètres pour les composants suivants sont disponibles :

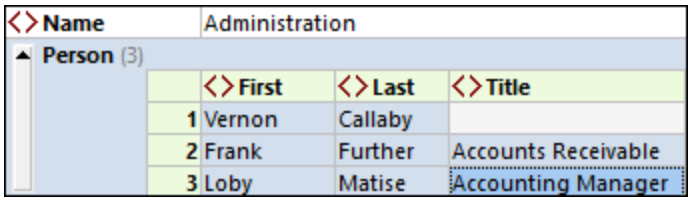
- *En-tête de table (non choisie)* et *Sélection d'en-tête de table* : Se réfèrent aux en-têtes de colonne et de ligne. La capture d'écran ci-dessous affiche les en-têtes non sélectionnées ; sa couleur d'arrière-fond est définie comme dans le dialogue ci-dessus. La couleur *En-tête sélectionné* est activée lorsque tous les en-têtes sont sélectionnés, pas lorsqu'un en-tête individuel est sélectionné. Tous les en-têtes

peuvent être sélectionnés en cliquant sur la cellule située au croisement de l'en-tête de ligne et de colonne, ou en sélectionnant l'élément créé en tant que la table ou un de ses ancêtres.



Name	Administration		
Person (3)			
	First	Last	Title
1	Vernon	Callaby	
2	Frank	Further	Accounts Receivable
3	Loby	Matise	Accounting Manager

- *Entrée de table manquante :* Se réfère aux éléments ou attributs non-existants dans le document (*voir capture d'écran ci-dessous*).
- *Sélection* et *Focus:* Se réfère aux cellules qui sont sélectionnées et aux cellules qui sont mises en focus. Par exemple, dans la capture d'écran ci-dessous, la table entière `Person` est sélectionnée et la cellule dans la partie inférieure droite est mise en focus.



Name	Administration		
Person (3)			
	First	Last	Title
1	Vernon	Callaby	
2	Frank	Further	Accounts Receivable
3	Loby	Matise	Accounting Manager

- *Sélection inactive, focus inactif :* Si un bloc de cellules a été sélectionné et qu'une ou plus de cellules dans ce bloc ont été mises en focus, alors la sélection et le focus sont actifs aussi longtemps que le Mode Grille est actif. Si, toutefois, un autre dialogue ou fenêtre est rendu actif sans avoir supprimé la sélection ou le focus dans le Mode Grille, alors la sélection et le focus dans le Mode Grille sont inactifs. Si le Mode Grille est de nouveau rendu actif, alors la sélection et le focus deviennent de nouveau actifs.

Note : Outre les couleurs que vous définissez ici, XMLSpy utilise la sélection habituelle et les préférences de couleurs de menu définies dans les Paramètres d'affichage dans le Panneau de contrôle de votre installation Windows.

Définir par défaut

La touche **Définir par défaut** réinitialise les polices aux paramètres d'installation d'origine.

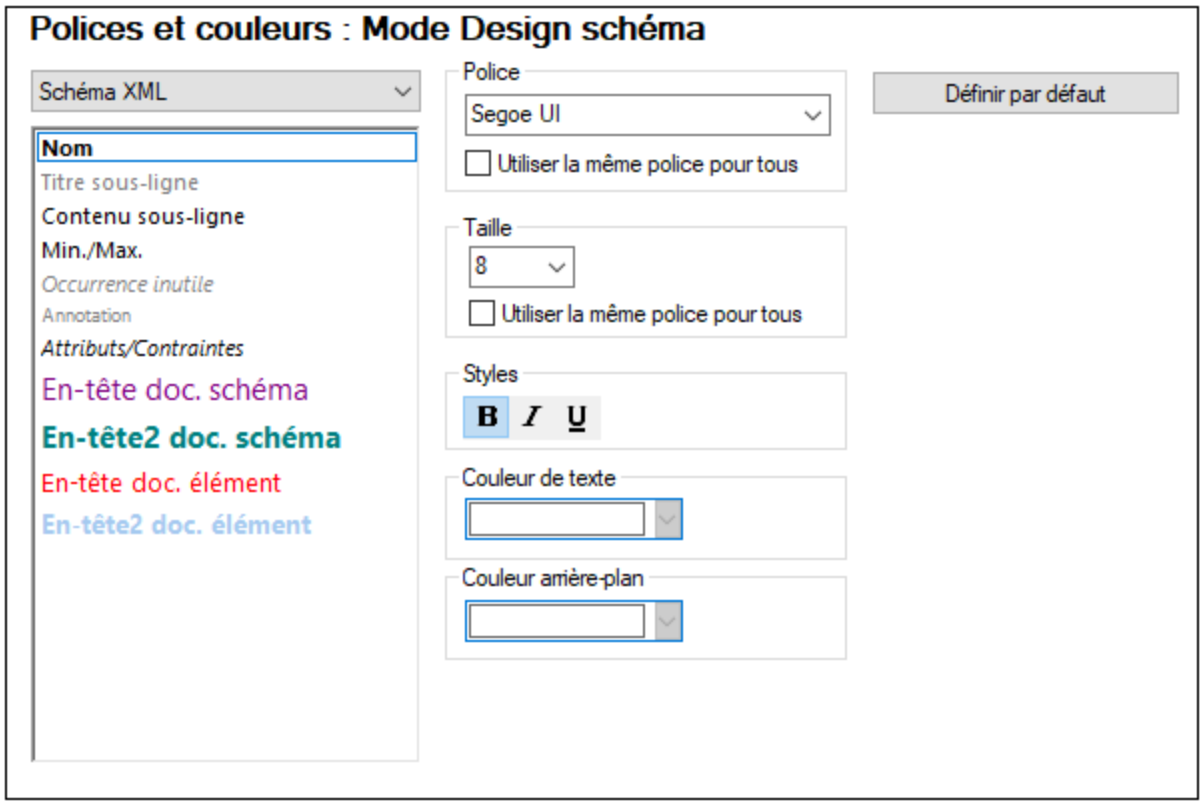
Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.8.3 Mode Conception de Schéma

La section **Mode Schema Design** vous permet de personnaliser l'apparence de l'affichage Mode Schéma des documents XML Schéma et JSON Schéma. Le formatage sera défini pour le thème actif actuel. Pour définir le formatage d'un autre thème, faites de l'autre thème le thème actif.

© 2016-2022 Altova GmbH



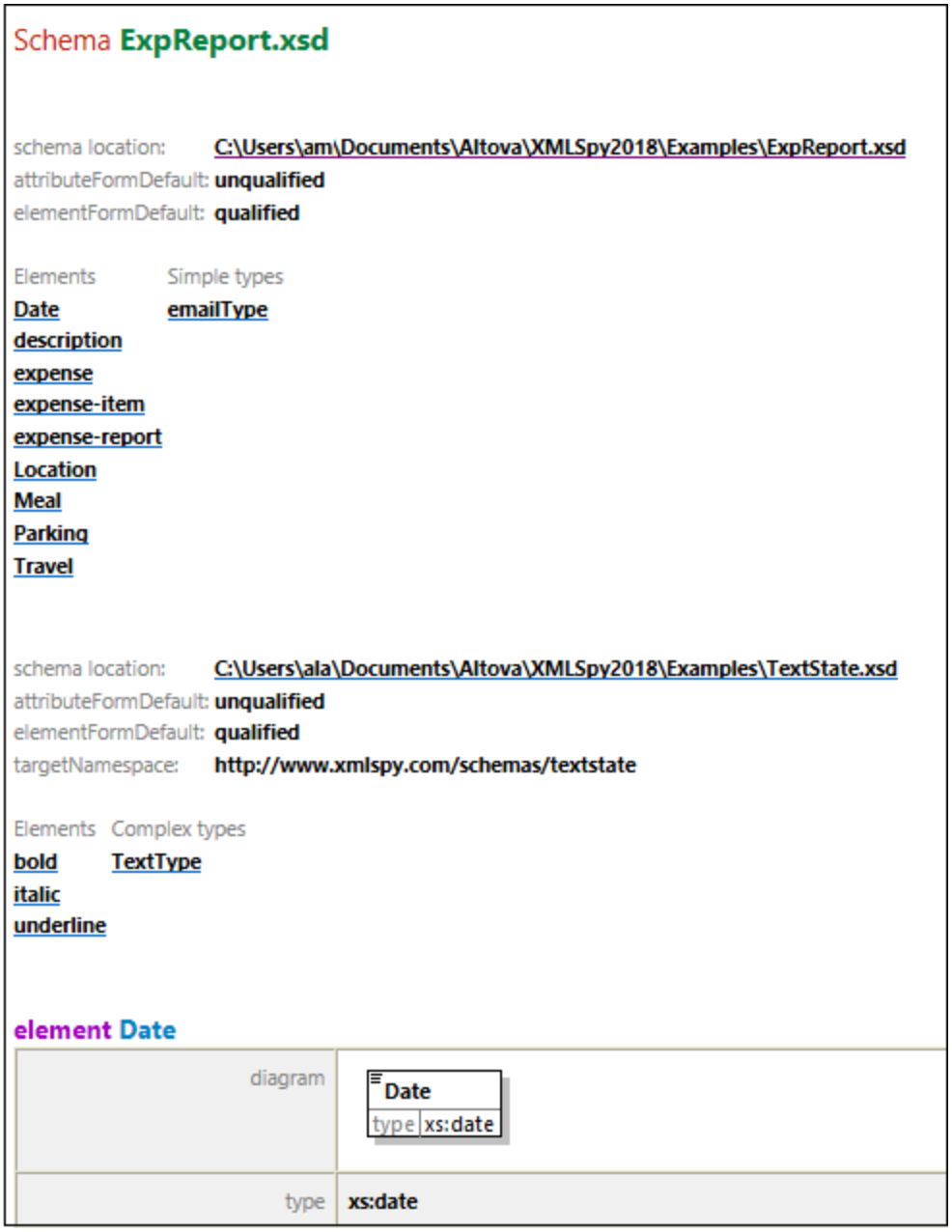
Comment personnaliser

Pour personnaliser des types d'item de texte individuels du type de document sélectionné, procédez comme suit :

1. Dans la liste de choix en haut à gauche, choisir le Schéma XML ou le Schéma JSON. Les types d'item de texte du type de document sélectionné apparaissent dans la fenêtre en-dessous de la liste de choix.
2. Choisir le type d'item de texte que vous souhaitez formater en cliquant dessus.
3. Définir les propriétés de police du type d'item de texte sélectionné en utilisant les options dans les volets du côté droit.

Note :

Les types d'item de texte *Doc.Schema Header(2)* et *Doc.Element Header(2)* se réfèrent, respectivement, à l'en-tête de schéma et aux en-tête d'élément dans la documentation générée du schéma. Comparer les couleurs de ces propriétés dans le dialogue ci-dessus avec les couleurs de l'en-tête de schéma et l'en-tête d'élément de la documentation générée dans la capture d'écran ci-dessous.



Définir par défaut

La touche **Définir par défaut** réinitialise les polices aux paramètres d'installation d'origine.

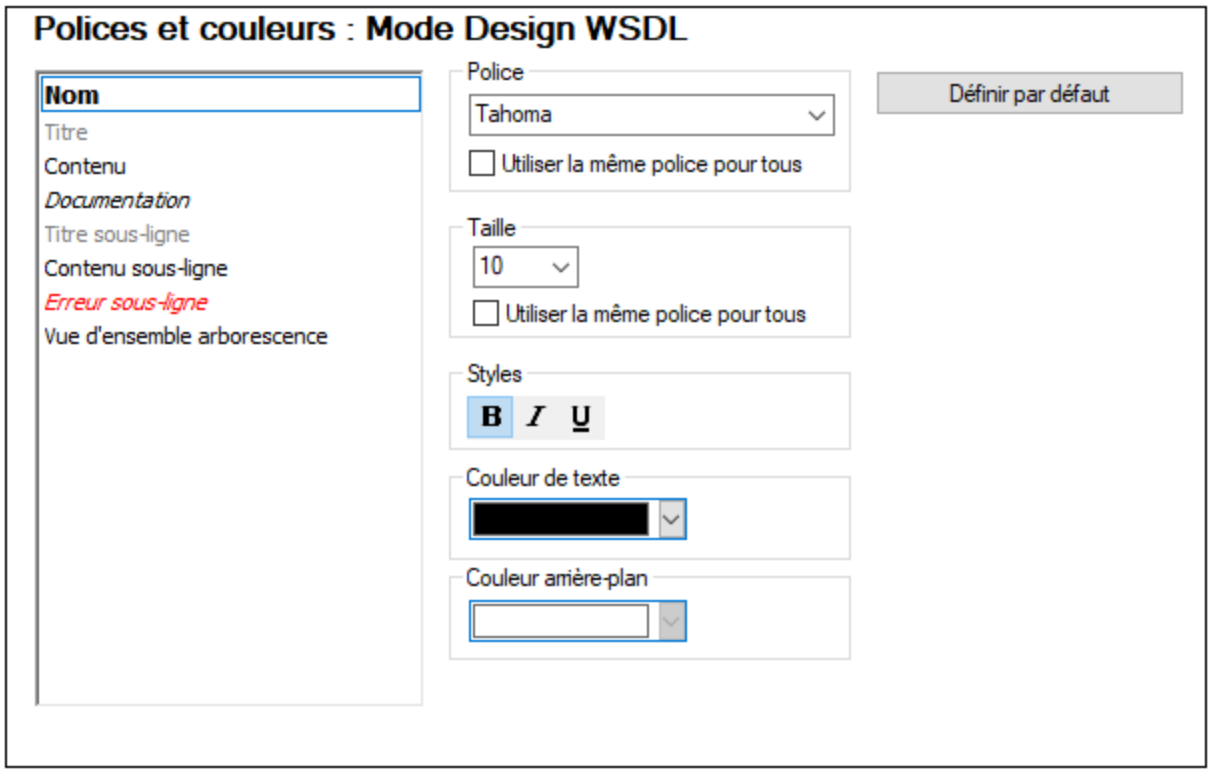
Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

© 2016-2022 Altova GmbH

28.17.15.8.4 Mode Design WSDL

La section **Mode Design WSDL** vous propose des options de personnalisation pour l'apparence des items de texte dans le Mode WSDL (*voir capture d'écran ci-dessous*). Le formatage sera défini pour le thème actif actuel. Pour définir le formatage d'un autre thème, faites de l'autre thème le thème actif.



Vous pouvez définir les propriétés suivantes pour le type d'item de texte sélectionné :

- *Police et taille :* La deuxième police sera aussi utilisée em imprimés de l'Affichage WSDL Design. Si vous souhaitez utiliser la même police et la même taille de police pour tous les types d'item de texte, cochez la case respective *Utiliser la même pour tous*.
- *Style de texte, couleur de texte, arrière-plan de texte :* Définit le style, la couleur et l'arrière-plan des types d'items de texte individuels. Les paramètres actuels sont immédiatement reflétés dans la liste du panneau de gauche, de manière à ce que vous puissiez visualiser l'apparence de l'item de texte

Note :
Le formatage de l'en-tête WSDL et des noms de service dans la documentation générée du document WSDL est prélevé à partir du formatage des types d'item de texte *Doc.Schema Header(2)* et *Doc.Element Header(2)*, respectivement du Mode Conception de schéma.

Définir par défaut

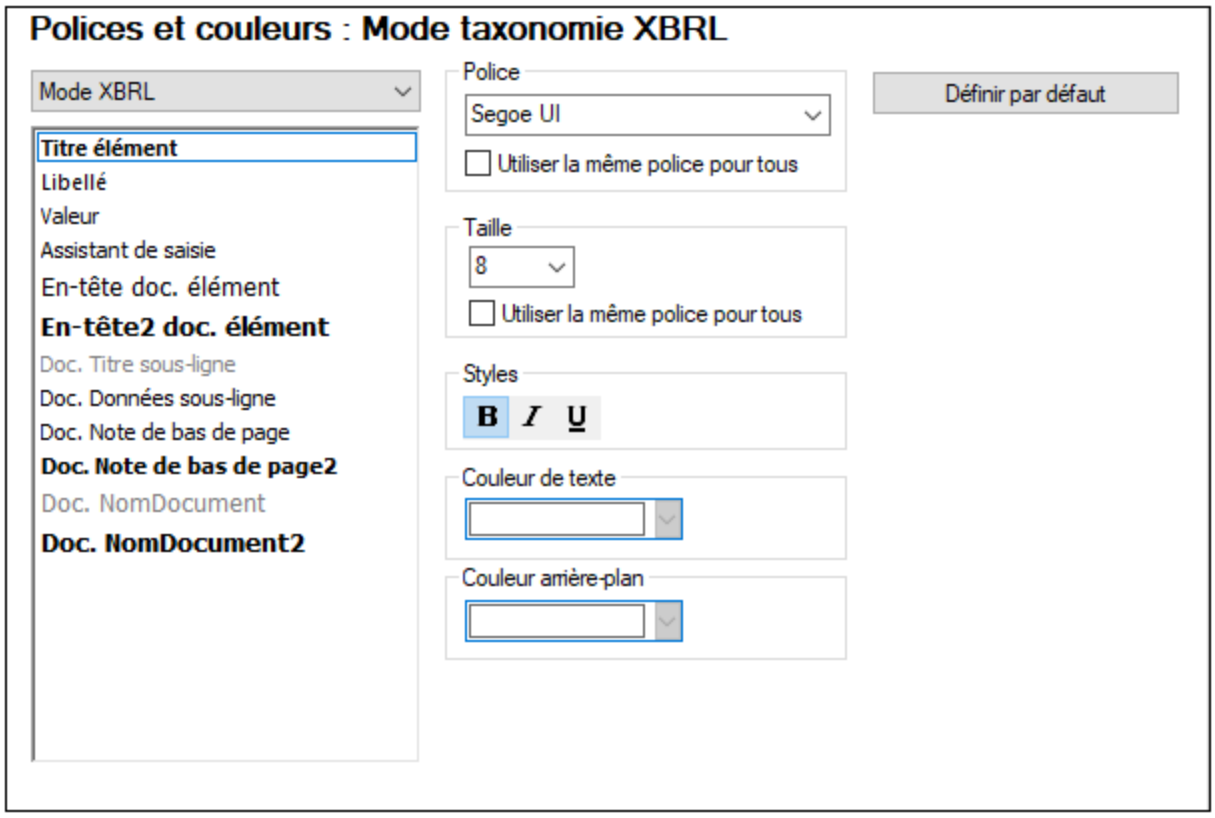
La touche **Définir par défaut** réinitialise les polices aux paramètres d'installation d'origine.

Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.8.5 Mode Taxonomie XBRL

La section **Mode Taxonomie XBRL** vous propose des options de personnalisation pour l'apparence des items de texte dans XBRL View (*voir capture d'écran ci-dessous*). Le formatage sera défni pour le thème actif actuel. Pour définir le formatage d'un autre thème, faites de l'autre thème le thème actif.



Vous pouvez définir les propriétés suivantes pour le type d'item de texte sélectionné :

- *Police et taille :* La police sélectionnée sera également utilisée en imprimés de l'Affichage de taxonomie XBRL. Si vous souhaitez utiliser la même police et la même taille de police pour tous les types d'item de texte, cochez la case respective *Utiliser la même pour tous*.
- *Style de texte, couleur de texte, arrière-plan de texte :* Définit le style, la couleur et l'arrière-plan des types d'items de texte individuels. Les paramètres actuels sont immédiatement reflétés dans la liste du panneau de gauche, de manière à ce que vous puissiez visualiser l'apparence de l'item de texte
- Les options de prévisualisation de la Table XBRL permettent des paramètres pour les composants des tables XBRL, qui peuvent être consultés en tant que tables dans l'onglet *Table* du Mode XBRL.

Définir par défaut

La touche **Définir par défaut** réinitialise les polices aux paramètres d'installation d'origine.

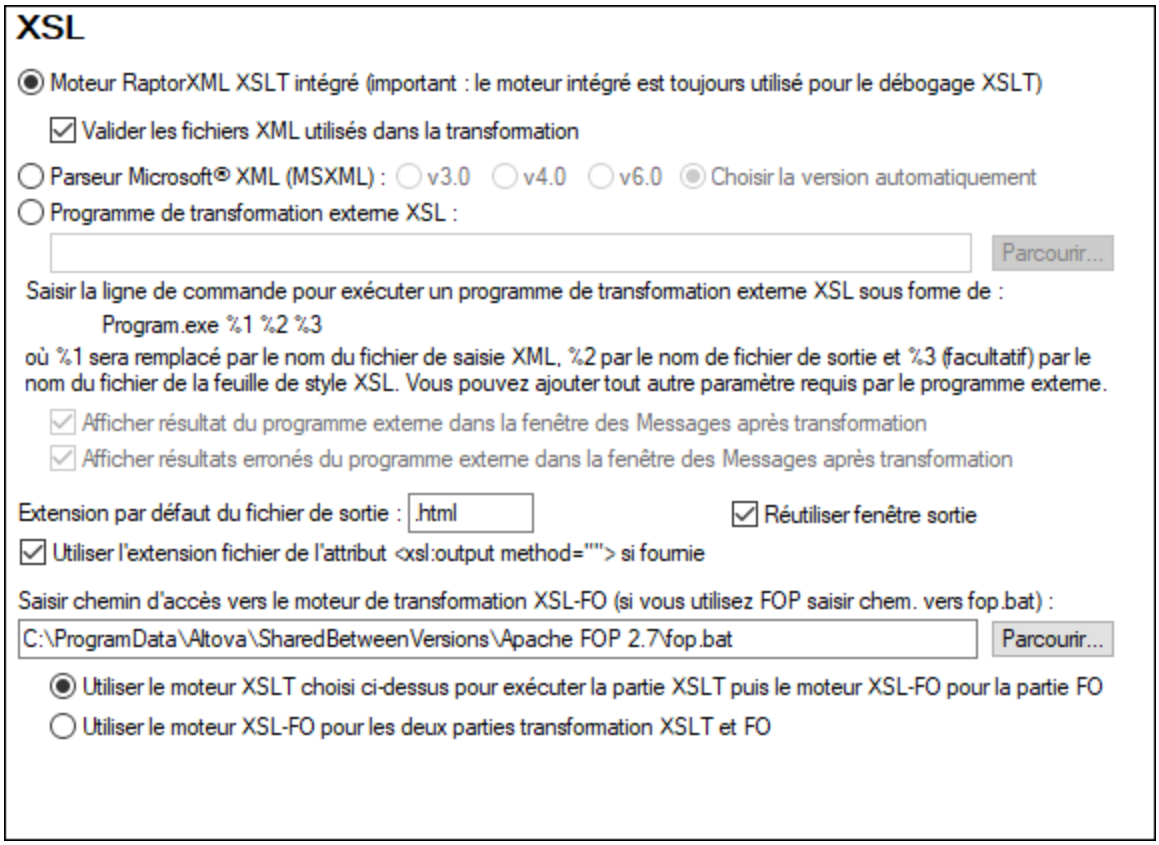
© 2016-2022 Altova GmbH

Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.9 XSL

La section **XSL** (*capture d'écran ci-dessous*) vous permet de définir des options pour les transformations XSLT et les transformations XSL-FO effectuées depuis l'application. Vous pouvez choisir un moteur XSLT préféré pour les transformations dans XMLSpy. Les options pertinentes pour chaque moteur sont affichées dans la section des paramètres du *Moteur* lorsque vous sélectionnez un moteur. Les paramètres *Fichier de sortie* et les paramètres *Transformation XSL-FO* qui sont disponibles sont les mêmes pour tous les moteurs. La capture d'écran ci-dessous affiche les paramètres lorsqu'un moteur built-in Altova RaptorXML XSLT est utilisé.



Paramètres de moteur

Vous pouvez établir un processeur XSLT pour exécuter les transformations XSLT lorsque la commande **Transformation XSLT** est invoquée.

Vous pouvez choisir parmi une des options de moteur XSLT suivantes :

- Moteur built-in RaptorXML XSLT

- Parseur Microsoft XML (MSXML)
- Moteur externe XSLT

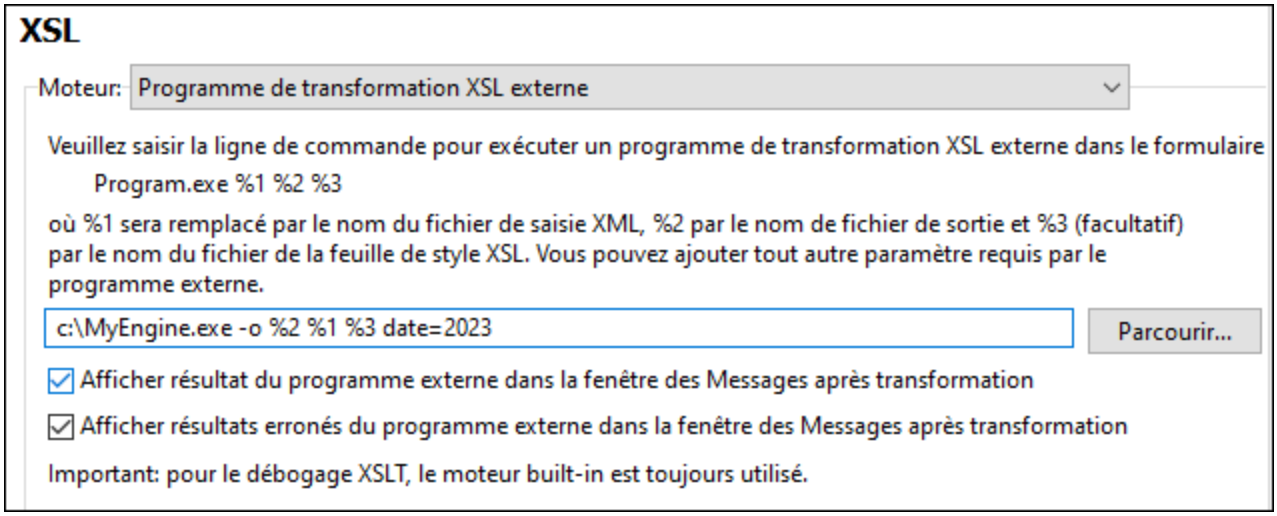
Note: pour un débogage XSLT dans XMLSpy, le moteur built-in RaptorXML XSLT est toujours utilisé — même si un autre moteur XSLT est sélectionné ici pour les transformations.

Moteur Altova RaptorXML XSLT
XMLSpy contient les moteurs Altova RaptorXML XSLT 1.0, XSLT 2.0 et XSLT 3.0 que vous pouvez utiliser pour les transformations XSLT. Le moteur XSLT approprié (1.0, 2.0 ou 3.0) est utilisé (conformément à la valeur de l'attribut `version` de l'élément `xsl:stylesheet` ou `xsl:transform`). Cela vaut aussi bien pour les transformations XSLT que pour le débogage XSLT utilisant le Débogueur XSLT/XQuery de XMLSpy.

Si vous souhaitez valider les fichiers XML files utilisés dans la transformation, sélectionnez l'option *Valider* (*voir la capture d'écran ci-dessus*).

Parseur Microsoft XML (MSXML)
Un ou plus de parseurs de MSXML 3.0, 4.0 ou 6.0 seront préinstallés sur votre appareil. Si vous savez quelle version installée vous voulez utilisez, vous pourriez la sélectionner. Autrement, vous devriez laisser la sélection de la version automatiquement à XMLSpy. (L'option *Choisir version automatiquement* est active par défaut.) Dans ce cas, XMLSpy tente de choisir la version disponible la plus récente.

Moteur externe XSLT
Choisissez un processeur externe XSLT de votre choix en saisissant le chemin à son fichier exécutable.



Vous devez spécifier le string de ligne de commande que le processeur externe XSLT utilise pour exécuter une transformation. Vous pouvez créer le string de ligne de la commande avec les composants suivants :

`%1` = document XML à traiter
`%2` = Fichier de sortie à générer
`%3` = Feuille de style XSLT à utiliser (si le document XML ne contient pas de référence à la feuille de style)

 © 2016-2022 Altova GmbH

Par exemple, vous avez un processeur qui utilise le modèle de commande suivant pour exécuter une transformation XSLT :

```
myxsltengine.exe -o <output.xml> <input.xml> <stylesheet.xslt> <param-name>=<param-value>?
```

Ensuite, dans XMLSpy, créez la ligne de commande utilisant les variables dans les emplacements corrects. Par exemple :

```
c:\MyEngine.exe -o %2 %1 %3 date=2023
```

XMLSpy vous enverra les fichiers d'entrée corrects au moteur externe pour le traitement renverra le/s fichier/s de sortie vers un emplacement de sortie s'il est précisé et/ou vers une fenêtre d'application.

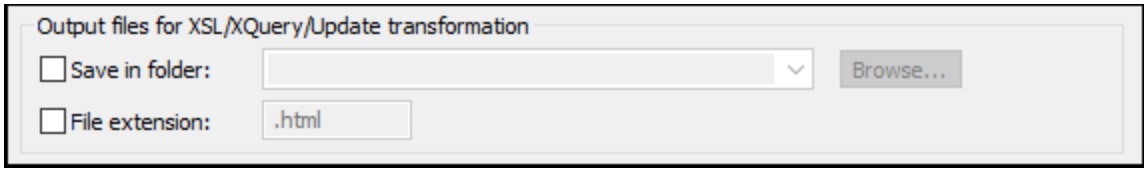
Cochez les cases respectives pour montrer les messages de sortie et d'erreur du programme externe dans la fenêtre Messages de XMLSpy.

Note : les paramètres définis dans le dialogue Paramètres d'entrée XSLT de XMLSpy sont uniquement passés dans les moteurs XSLT Altova internes. Ils ne sont pas passés dans un autre Moteur XSLT configuré en tant que processeur XSLT par défaut.

Paramètres fichier Output

Les options suivantes sont disponibles :

- *Extension de fichier par défaut :* Définit une extension de fichier par défaut pour les fichiers de sortie, qui peuvent être substitués par l'extension de fichier nommée dans l'élément XSLT **`xsl:output`** (*voir le dernier item de la liste*).
- *Réutiliser la fenêtre de sortie :* Entraîne des transformations subséquentes pour afficher le document de résultat dans la même fenêtre de sortie. Si le fichier d'entrée XML appartient à un projet et que l'option *Réutiliser la fenêtre de sortie* est désactivée, le paramètre prend effet si le chemin de fichier de sortie *Enregistrer dans le dossier* (*capture d'écran ci-dessous*) se trouvant dans les propriétés de projet pertinentes est **aussi** désactivé.



- *Utilisez l'extension de fichier de l'élément xsl:output:* Sélectionne si l'extension de fichier spécifié dans l'élément **`xsl:output`** de la feuille de style XSLT substituerait l'extension par défaut spécifiée dans la première option de cette liste.

Paramètres de transformation XSL-FO

Les documents FO sont traités avec un processeur FO, et le chemin vers l'exécutable du moteur FO doit être spécifié dans zone de texte pour le moteur de transformation XSL-FO. La transformation est effectuée à l'aide de la commande de menu **Transformation XSL/XQuery | XSL-FO**. Si le fichier source (le document actif lorsque la commande est exécutée dans l'IDE) est un document XSL-FO, le moteur FO est invoqué pour la transformation. Si le document source est un document XML, une transformation XSLT est requise pour tout d'abord convertir le document XML en un document XSL-FO. Cette transformation XSLT peut être effectuée soit par le biais du moteur XSLT que vous avez spécifié en tant que moteur par défaut pour l'application (*voir ci-*

dessus), ou par le moteur XSLT qui pourrait être intégré dans le moteur FO que vous avez spécifié en tant que le moteur FO par défaut pour l'application. Pour effectuer un choix entre ces deux options, cliquez sur le bouton radio approprié.

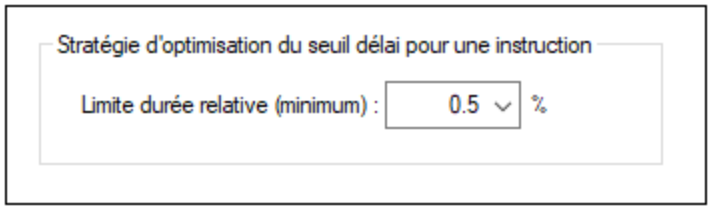
Note : À moins d'avoir désélectionné l'option pour installer le moteur FOP du Projet Apache XML, celui-ci sera installé dans le dossier `C:\ProgramData\Altova\SharedBetweenVersions`. En cas d'installation, le chemin y menant sera automatiquement saisi dans le champ d'entrée du Moteur XSL-FO. Vous pouvez déterminer le chemin vers n'importe lequel des moteurs FO que vous souhaitez utiliser. Veuillez noter, néanmoins, que le même chemin sera utilisé par d'autres produits Altova qui utilisent les moteurs FO et comportent des paramètres pour sélectionner le processeur FO (StyleVision et Authentic Desktop).

Enregistrer et quitter

Après avoir défini les paramètres, cliquez sur **OK** pour terminer.

28.17.15.9.1 Optimiseur de vitesse

La section **XSL: Optimiseur de vitesse** (*capture d'écran ci-dessous*) vous permet de définir des options pour l'Optimiseur de vitesse XSL.



Il est possible de déterminer un seuil de temps pour des instructions XSLT uniques dans une feuille de style XSLT. Les valeurs s'étendent de 0,1% du temps de transformation total à 99% du temps total. Si une instruction prend plus de temps à être exécutée que le temps spécifié en tant que le seuil, l'analyse d'optimisation est invoquée. Sinon, aucune analyse n'est exécutée. Si l'analyse d'optimisation échoue, il se peut que le seuil de temps soit trop élevé. Vous devrez éventuellement le baisser.

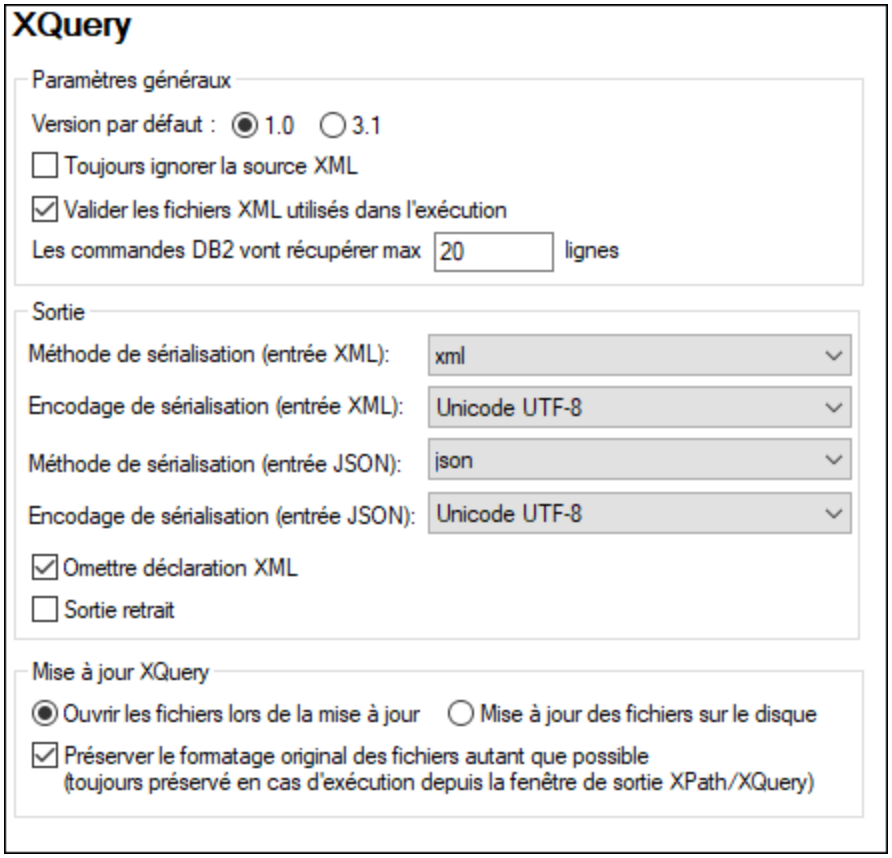
Enregistrer et quitter

Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer.

28.17.15.10 XQuery

La section **XQuery** (*capture d'écran ci-dessous*) définit des options liées à l'édition et à l'exécution des documents XQuery et XQuery Update.

© 2016-2022 Altova GmbH



Options générales XQuery

Les options suivantes sont disponibles :

- *Sérialisation :* la sérialisation se réfère à la manière dont le texte est écrit dans le document de sortie. Vous pouvez choisir la méthode de sérialisation (adaptative*, HTML, JSON, texte, XHTML, ou XML) et l'encodage de sérialisation de la sortie pour les différents types d'entrée. La sérialisation de sortie peut être sélectionnée séparément pour l'entrée XML et l'entrée JSON. (*Note :* la méthode adaptative permet de traiter un document d'instance sans erreur ; elle détermine automatiquement la méthode de sérialisation sur la base du document d'entrée.)
- *Omettre déclaration XML :* Omet la déclaration XML dans le document (de sortie) sérialisé.
- *Sortie retrait :* Insère un retrait dans le document de sortie pour montrer la hiérarchie du document.
- *Toujours ignorer la source XML :* lorsqu'un document XQuery est exécuté, XMLSpy peut proposer une source XML avec laquelle exécuter le document XQuery. L'invite est un dialogue qui vous permet de chercher le fichier XML. Choisir cette option pour sauter le dialogue et exécuter directement le document XQuery. Si cette option est sélectionnée, le document XQuery devrait pouvoir exécuter correctement sans être obligé de passer par un document XML. Il se peut que cela se produise soit parce qu'aucun document XML n'est requis, soit parce que les données XML sont accédées par le biais de fonctions se trouvant dans le document XQuery.
- *Valider les fichiers XML :* Valide les fichiers XML qui sont utilisés dans l'exécution des documents XQuery. Les fichiers XML invalides sont marqués et le document XQuery n'est pas traité.

- *Récupération de lignes DB2 :* dans les affichages qui montrent les données DB, vous pouvez spécifier le nombre maximum de lignes à extraire. XMLSpy reconnaît les extensions de fichier `.xqr` en tant que fichiers XQuery-pour-DB.
- *Version par défaut XQuery :* spécifie la version de moteur XQuery à utiliser pour l'exécution de documents XQuery qui ne possèdent pas un mot-clé `version`. Cela s'applique aussi bien aux documents XQuery et XQuery Update, et sélectionne le moteur XQuery par défaut à utiliser.

Options XQuery Update

Les options XQuery Update suivantes sont disponibles :

- *Mise à jour :* lorsqu'un fichier XQuery Update est exécuté, les fichiers cibles XML peuvent soit être mis à jour directement ou être ouverts dans XMLSpy puis mis à jour dans la mémoire. L'option *Ouvrir les fichiers lors de la mise à jour* vous permet de revoir les mises à jour et d'enregistrer le fichier sur le disque ou bien de rejeter les mises à jour (en fermant le fichier sans enregistrer).
- *Préserver le formatage original :* Préserve le formatage original du document mis à jour autant que possible.

Enregistrer et quitter

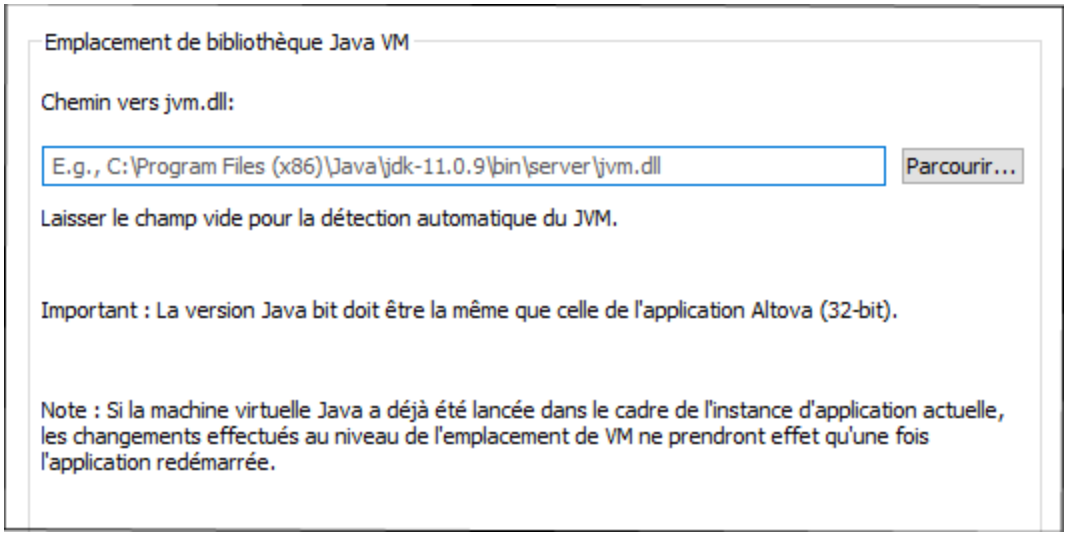
Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer.

28.17.15.11 Java

Dans l'onglet *Java* (*voir la capture d'écran ci-dessous*), vous pouvez saisir en option un chemin vers une Java VM (Machine Virtuelle) sur votre système de fichier. Veuillez noter que le fait d'ajouter un chemin Java VM personnalisé n'est pas toujours nécessaire. Par défaut, XMLSpy tente de détecter le chemin Java VM automatiquement en lisant (dans cet ordre) le registre Windows et la variable d'environnement JAVA_HOME. Le chemin personnalisé ajouté dans ce dialogue prendra la priorité sur tout autre chemin Java VM détecté automatiquement.

Vous devrez éventuellement ajouter un chemin Java VM personnalisé, par exemple si vous utilisez une machine virtuelle Java qui ne possède pas de programme d'installation et ne crée pas d'entrées de registre (par exemple, OpenJDK d'Oracle). Vous pourrez également définir ce chemin si vous souhaitez contourner, pour une raison quelconque, tout chemin Java VM détecté automatiquement par XMLSpy.

© 2016-2022 Altova GmbH



Veuillez noter les points suivants :

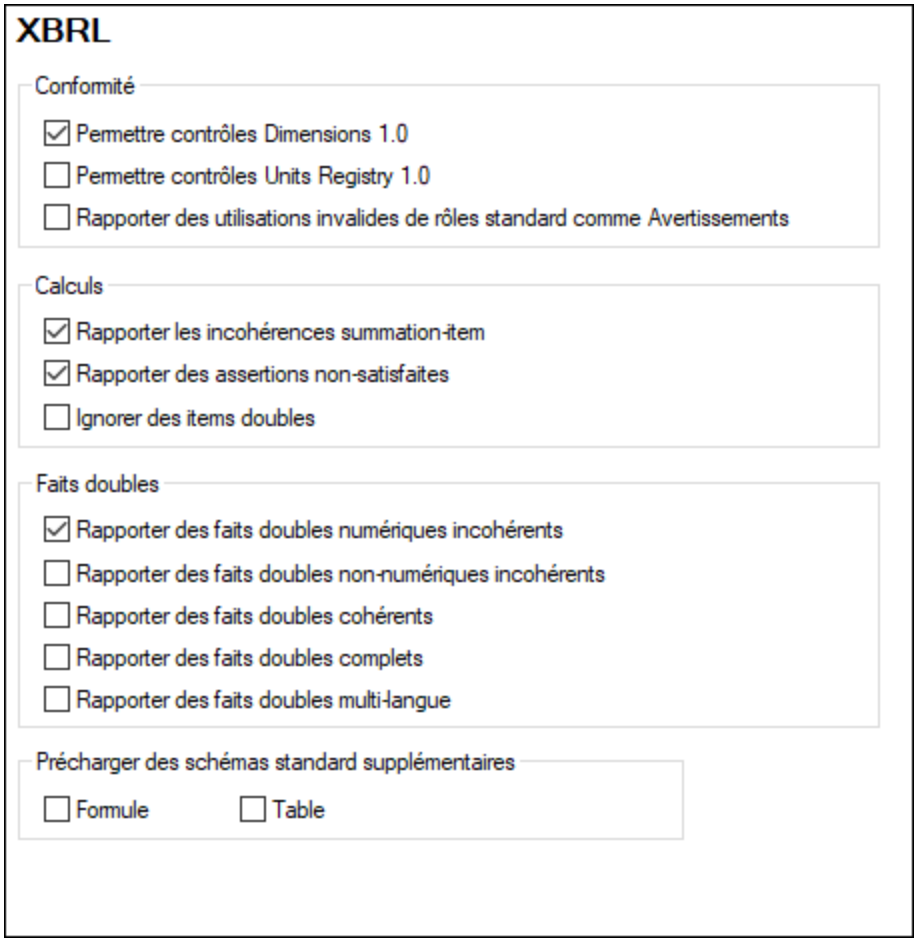
- Le chemin Java VM est partagé entre les applications de desktop (pas serveur) Altova. Par conséquent, si vous le modifiez dans une application, il s'appliquera automatiquement à toutes les autres applications Altova.
- Le chemin doit pointer vers le fichier **`jvm.dll`** provenant du répertoire **`\bin\server`** ou **`\bin\client`**, par rapport au répertoire sur lequel le JDK a été installé.
- La plateforme XMLSpy (32-bit, 64-bit) doit être la même que celle du JDK.
- Une fois avoir modifié le chemin Java VM, vous devrez éventuellement redémarrer XMLSpy pour que les nouveaux paramètres prennent effet.

La modification du chemin Java VM touche les zones suivantes :

- Connectivité JDBC
- Fonctions d'extension Java pour XSLT/XPath

28.17.15.12 XBRL

La section **XBRL** fournit des options pour la validation de documents et le traitement de documents d'instance XBRL et XULE, inline XBRL et pour ajouter et gérer les packages de taxonomie XBRL.



Validation XBRL

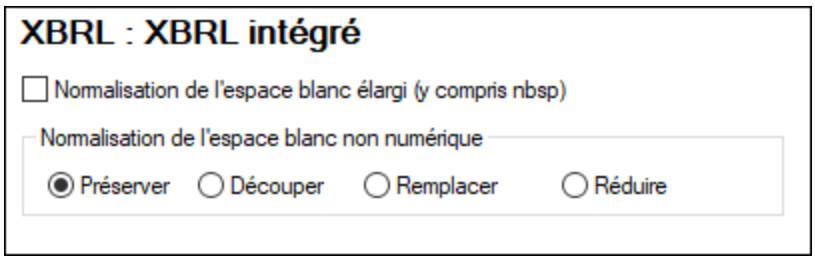
Au niveau supérieur des options XBRL (*voir la capture d'écran ci-dessus*), vous pouvez spécifier les options de validation suivantes :

- La mise en œuvre de vérifications de conformité avec Dimensions 1.0 et Units Registry 1.0. Il existe également une option de rapporter l'utilisation invalide de rôles standard comme avertissements.
- S'il faut faire rapport de calculs qui impliquent des (i) inconsistances lorsque des items d'addition sont calculés, (ii) des assertions qui ne sont pas satisfaites.
- La non-considération de doublons (doublons complets et doublons consistants) qui sont impliqués dans les calculs. Là où le fait est dupliqué, le fait le plus précis est utilisé alors que les autres sont ignorés.
- Rapports de doublons. Pour plus d'information sur la classification des doublons, voir la spécification de la gestion des faits dupliqués dans XBRL et Inline XBRL 1.0.
- S'il faut précharger des schémas standard pour les formules et/ou les tables.

© 2016-2022 Altova GmbH

28.17.15.12.1 XBRL Inline

L'onglet Inline XBRL (*voir la capture d'écran ci-dessous*) offre des options pour la normalisation de l'espace blanc (caractères d'espace, onglets, sauts de ligne et retour de chariot) dans XBRL Inline.



Les options sont les suivantes :

- Si *Normalisation des espaces blancs étendue* est sélectionnée, alors les espaces non cassants sont également considérés être des espaces blancs.
- *Préserver* garde les espaces blancs inchangés.
- *Découper* supprime les espaces blancs des deux côtés des strings.
- *Remplacer* toutes les occurrences d'onglet, de sauts de ligne et de retour de chariot sont remplacées par un espace.
- *Compresser* est une extension de *Remplacer* dans le sens qu'un remplacement est effectué, puis tous les espaces contigus sont compressés en un simple espace.
- *Réaliser les vérifications du manuel de rapport ESEF :* Permet les vérifications de conformité avec le Manuel de rapport ESEF lorsqu'un document Inline XBRL est validé. Pour plus d'information sur ESEF, voir la page web du European Single Electronic Format (ESEF).

28.17.15.12.2 Packs de taxonomie

Un pack de taxonomie XBRL est un archive zippé qui contient une copie hors ligne d'une taxonomie. Le pack contient un fichier XML de catalogue qui mappe les URI aux emplacement de fichier de la taxonomie, et qui rend donc la taxonomie disponible hors ligne pour les applications. Les règles qui spécifient comment les packs de taxonomie doivent être structurés et intégrés sont détaillées dans la Recommendations de packages de taxonomie sous XBRL.org.

Une fois avoir téléchargé un pack de taxonomie, vous pouvez configurer XMLSpy pour identifier et utiliser automatiquement le fichier de catalogue du point d'entrée du pack. Pour ce faire, ajouter le pack à la liste des packs de taxonomie actifs. Les fichiers de catalogue des packs actifs seront ensuite utilisés pour localiser des ressources pour les opérations comme une validation XBRL. Il existe deux types de packs de taxonomie que vous pouvez ajouter :

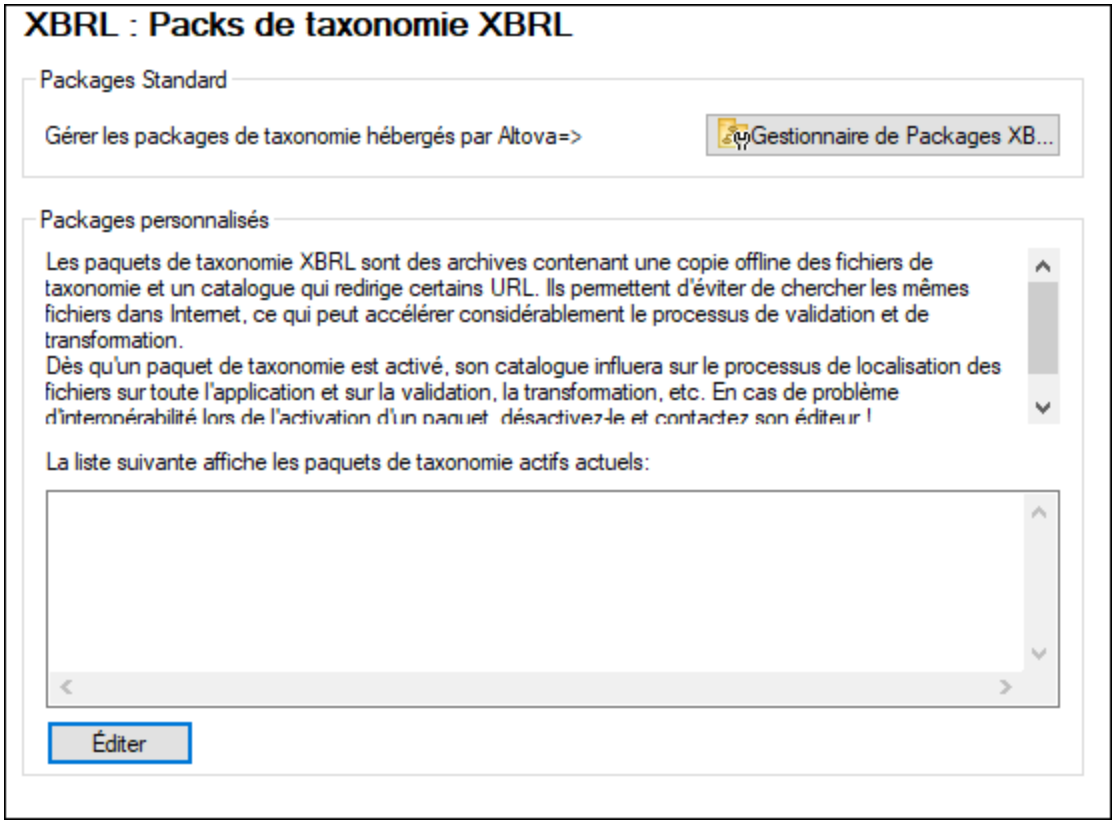
- des packs standdard, qui peuvent être gérés facilement en toute convivialité pour tous les produits de Altova en utilisant le Manager de taxonomie XBRL de Altova (voir la description ici).
- des packs personnalisés, que vous ajoutez et gérez par le biais du dialogue.

Note : Une ressource pointée par le fichier de catalogue d'un pack actif sera utilisée pour toutes les opérations XMLSpy qui nécessitent cette ressource. Si une telle ressource est différente d'une autre manière que la ressource qui a été utilisée précédemment par XMLSpy, alors les des erreurs peuvent

survenir lorsque les opérations sont exécutées. Pour plus d'informations, voir l'avertissement dans l'encadré gris en bas de cette page.

Ajouter et gérer des packs de taxonomie

Choisir **Outils | Options | Packs de taxonomie** pour afficher le volet Packs de Taxonomie (voir la *capture d'écran ci-dessous*).

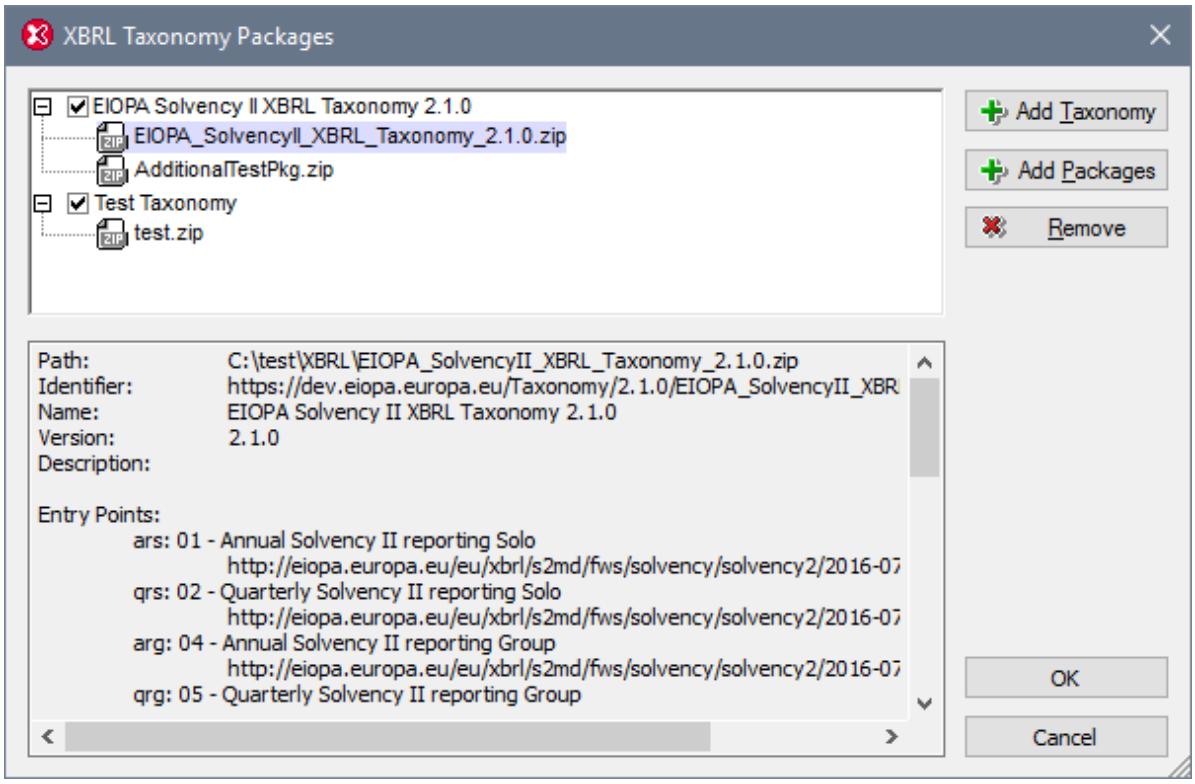


Pour ajouter un pack de taxonomie standard, cliquez sur **Manager de taxonomie XBRL**. Pour toute instruction sur comment utiliser le Manager de taxonomie XBRL de Altova, voir sa description ici.

Pour ajouter un pack personnalisé, suivez les étapes ci-dessous.

1. Cliquez sur **Éditer** (*voir capture d'écran ci-dessus*) pour afficher le dialogue Packs de Taxonomie XBRL (*capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



2. Cliquez sur **Ajouter Taxonomie**, puis cherchez l'emplacement du pack de taxonomie, le choisir et cliquer sur **Ouvrir**. (Vous pouvez aussi choisir plusieurs packs à ajouter en une fois.) Le pack sera ajouté à la liste de pack de taxonomie dans le dialogue. La liste est affichée en tant qu'arborescence à deux niveau. Le premier niveau indique la taxonomie ; le second niveau montre les packs de cette taxonomie. La case à cocher située à gauche d'une entrée de taxonomie indique si cette taxonomie est active ou pas. Une taxonomie ajoutée récemment sera active par défaut.
3. Cliquer sur **OK** pour terminer. Les packs récemment ajoutés seront affichés dans le volet Packs de taxonomie du dialogue Options (*première capture d'écran ci-dessus*).

Noter les points suivants :

- Si vous souhaitez ajouter un pack supplémentaire à une taxonomie, procéder comme suit : choisir la taxonomie dans le dialogue Packs de taxonomie XBRL (*capture d'écran ci-dessus*), puis ajouter les packs supplémentaires par le biais de la touche **Ajouter Packs**. Les packs ajoutés seront affichés dans le deuxième niveau de cette taxonomie.
- Lorsqu'un pack de taxonomie est sélectionné dans la liste dans le volet supérieur du dialogue Packs de taxonomie XBRL, ses détails (y compris son emplacement hors ligne) sont affichés dans le volet inférieur du dialogue (*voir capture d'écran ci-dessus*).
- Pour désactiver une taxonomie, décocher sa case. Si vous désactivez une taxonomie, ses fichiers de catalogue ne seront pas utilisés. Une désactivation est utile si, par exemple, vous souhaitez passer entre deux versions d'une taxonomie.
- Vous pouvez supprimer un pack en le sélectionnant et en cliquant sur **Supprimer**.
- Les applications Altova suivantes prennent en charge l'Enregistrement de Pack de Taxonomie : XMLSpy, MapForce et StyleVision. Le pack de taxonomie est commun à toutes ses applications. Si vous éditez la liste dans une application, la liste modifiée sera également affichée dans d'autres applications. Si vous éditez la liste de pack dans une application et qu'une autre application est

ouverte au même moment, l'autre application affichera une alerte vous demandant si vous souhaitez recharger la liste de pack pour refléter la modification.

Avertissement : les catalogues de pack peuvent rediriger vers des ressources incompatibles
Une ressource pointée par le/s fichier/s de catalogue d'un pack actif sera utilisé pour toutes les opérations XMLSpy qui nécessitent cette ressource. Un exemple d'une telle ressource serait le Schéma XML, qui est utilisé aussi bien pour la validation XML que pour la validation XBRL. Si la ressource hors ligne localisée par le fichier de catalogue du pack est incompatible avec votre environnement existant, des erreurs peuvent survenir. Dans ce cas, désactivez le pack de taxonomie et contactez les créateurs du pack avec l'information d'erreur.

28.17.15.12.3 XULE

L'onglet XBRL XULE (*capture d'écran ci-dessous*) propose des options pour le traitement XULE.

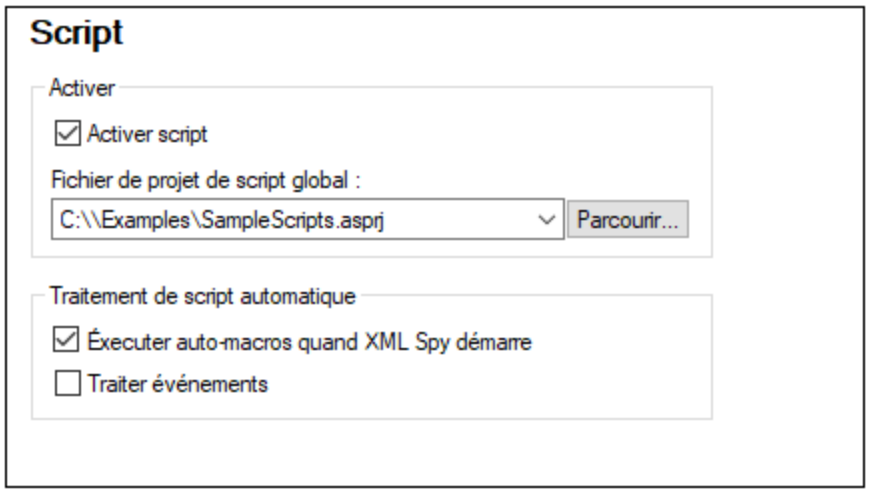


- *Ignorer faits doubles :* un fait double se produit généralement dans du Inline XBRL, lorsque le même fait est noté plus d'une fois dans le code HTML. Cette option spécifie que le fait dupliqué est produit une seule fois.
- *Sortie :* lorsqu'un document XULE est traité avec la commande **XBRL | Execute XULE**, la sortie peut être générée dans : (i) la fenêtre Messages, ou (ii) un nouveau document qui est affiché dans XMLSpy et stocké temporairement dans la mémoire ; ce document peut être stocké dans le fichier avec la commande **Fichier | Enregistrer sous**.

28.17.15.13 Script

La section **Script** (*capture d'écran ci-dessous*) vous permet d'activer l'Environnement de script lors du démarrage de l'application. Cocher la case *Activer Script* pour ce faire. Vous pouvez ensuite spécifier le fichier de Projet de script global (*voir capture d'écran ci-dessous*).

© 2016-2022 Altova GmbH



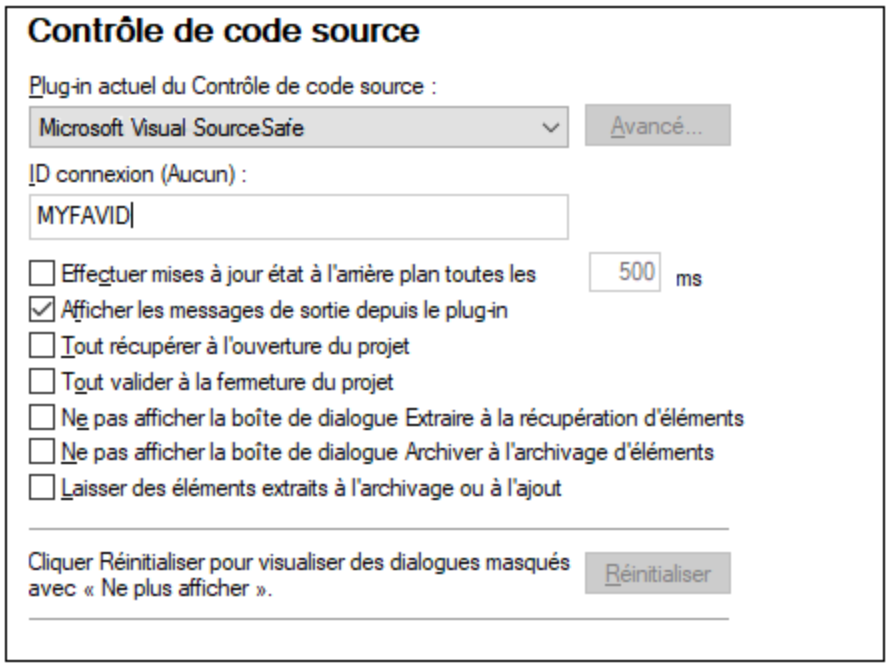
Pour définir un projet de script global pour XMLSpy, cocher la case *Activer Script* puis chercher le fichier de Projet de script global Altova (`.asprj`). Vous pouvez aussi spécifier : (i) si les Auto-macros contenus dans le projet de script doivent être exécutés automatiquement au démarrage de XMLSpy, et (ii) si les scripts de gestionnaire d'événement d'application dans le projet doivent être exécutés automatiquement ou pas ; cocher ou décocher les cases selon vos besoins.

Enregistrer et quitter

Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer. Les macros contenus dans le Projet de script global seront ensuite affichés dans le sous-menu de la commande **Macros**.

28.17.15.14 Contrôle de Source

La section **Contrôle de source** (*capture d'écran ci-dessous*) vous permet de spécifier le fournisseur de contrôle de source et les paramètres et l'ID de logon par défaut pour chaque fournisseur de contrôle de source.



Plugin Contrôle de source

Le plugin de contrôle de source actuel peut être sélectionné parmi les systèmes de contrôle de source installés. Ces systèmes sont listés dans la liste déroulante de la liste de choix. Une fois avoir sélectionné le contrôle de source requis, spécifier l'ID de login nécessaire dans le champ de saisie suivant. La touche **Avancé** ouvre un dialogue spécifique au plugin de contrôle de source sélectionné, dans lequel vous pouvez définir des paramètres pour ce plugin de contrôle de source. Ces paramètres sont différents pour les différents plugin de contrôle de source.

Préférences de l'utilisateur

Une gamme de préférences de l'utilisateur est disponible, y compris :

- Les mises à jour de statut peuvent être exécutées en arrière-plan au bout d'un intervalle défini par l'utilisateur, ou bien, elles peuvent être éteintes complètement. Des bases de données de contrôle de source très volumineuses peuvent consommer des ressources de CPU et de réseau considérables. Le système peut néanmoins être accéléré en désactivant les mises à jour en arrière-plan du statut ou en augmentant les intervalles entre elles.
- Lors de l'ouverture et de la fermeture de projets, les fichiers peuvent être automatiquement extraits ou archivés, respectivement.
- L'affichage des dialogues Extraire et Archiver peut être suppressé.
- La touche **Réinitialiser** est activée si vous avez coché/activé l'option *Ne plus afficher* dans un des dialogues. Cliquer sur la touche **Réinitialiser** pour réactiver l'invite *Ne plus afficher*.

Enregistrer et quitter

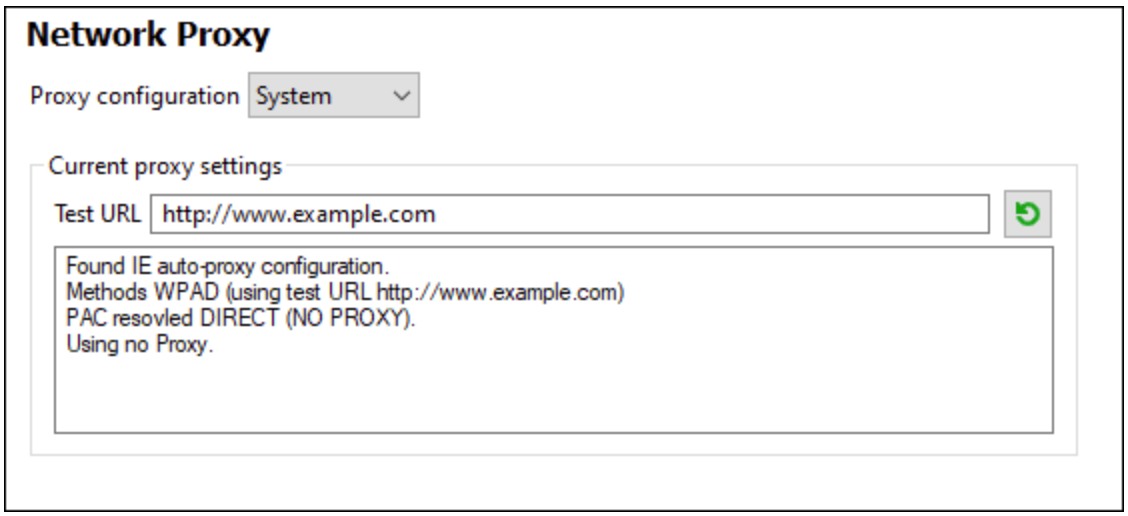
Une fois avoir effectué la configuration, cliquer sur **OK** pour terminer.

© 2016-2022 Altova GmbH

28.17.15.15 Proxy de réseau

La section *Proxy de réseau* vous permet de configurer des paramètres de proxy personnalisés. Ces paramètres ont une incidence sur la manière dont l'application se connecte à Internet (par exemple, pour une validation XML). Par défaut, l'application utilise les paramètres proxy du système, dans la plupart des cas, vous n'aurez donc pas à changer les paramètres de proxy. Si vous souhaitez, toutefois, définir un proxy réseau alternatif, dans la zone de liste de *configuration Proxy*, sélectionnez *Automatic* ou *Manual* pour configurer les paramètres en conséquence.

Note : les paramètres de proxy de réseau sont partagés parmi toutes les applications de Altova MissionKit. Donc, si vous modifiez les paramètres dans une application, toutes les applications de MissionKit en seront touchées.

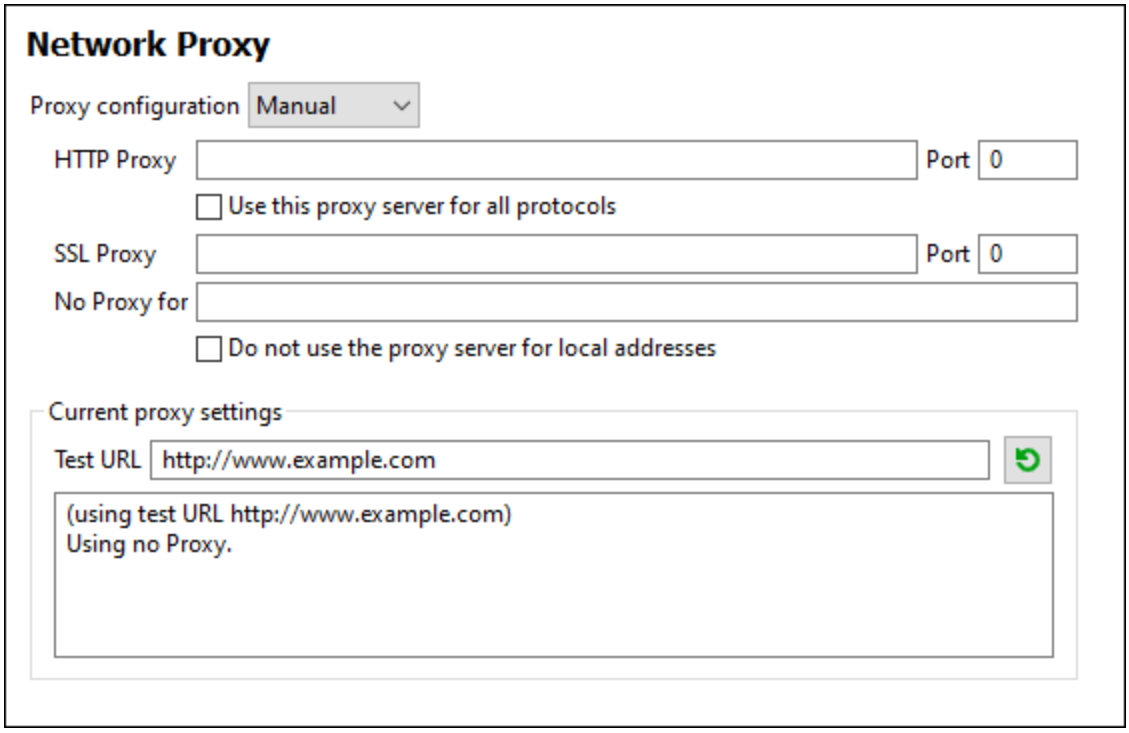


Utiliser les paramètres de proxy de système
Utilise les paramètres Internet Explorer (IE) configurables par le biais des paramètres de proxy du système. Effectue également une requête des paramètres configurés avec **`netsh.exe winhttp`**.

Configuration de proxy automatique
Les options suivantes sont possibles :

- *Paramètres auto-détection :* Consulte un script WPAD (**`http://wpad.LOCALDOMAIN/wpad.dat`**) par le biais de DHCP ou DNS, et utilise ce script pour une configuration proxy.
- *URL de script :* Spécifie une URL HTTP dans un script configuration-auto-proxy (**`.pac`**) qui doit être utilisé pour cette configuration de proxy.
- *Recharger :* Réinitialise et recharge la configuration automatique de proxy actuelle. Cette action requiert Windows 8 ou plus, et peut prendre jusqu'à 30 sec avant de prendre effet.

Configuration de proxy manuelle
Spécifier manuellement le nom d'hôte et le port entièrement qualifiés pour les proxies des protocoles respectifs. Un scheme pris en charge peut être inclus dans le nom d'hôte (par exemple : **`http://hostname`**). Il n'est pas exigé que le schéma soit le même que le protocole respectif si le proxy prend en charge le schéma.



Les options suivantes sont possibles :

- *Proxy HTTP :* Utilise le nom d'hôte spécifié et le port pour le protocole HTTP. Si *Utiliser le serveur proxy pour tous les protocoles* est sélectionné, alors le proxy HTTP spécifié est utilisé pour tous les protocoles.
- *Proxy SSL :* Utilise le nom d'hôte spécifié et le port pour le protocole SSL.
- *Pas de proxy pour :* Une liste séparée par point-virgule (`;`) de nom d'hôtes entièrement qualifiés, de noms de domaine, ou d'adresses IP pour des hôtes qui doivent être utilisés sans un proxy. Les adresses IP ne doivent pas être abrégées et les adresses IPv6 doivent être entourées de crochets (par exemple : `[2606:2800:220:1:248:1893:25c8:1946]`). Les noms de domaine doivent commencer avec un point (par exemple : `.example.com`).
- *Ne pas utiliser le serveur proxy pour les adresses locales :* Si cochées, ajoute `<local>` à la liste *Pas de proxy pour*. Si cette option est sélectionnée, les éléments suivants n'utiliseront pas le proxy : (i) `127.0.0.1`, (ii) `[::1]`, (iii) tous les noms d'hôte ne contenant pas de point (`.`).

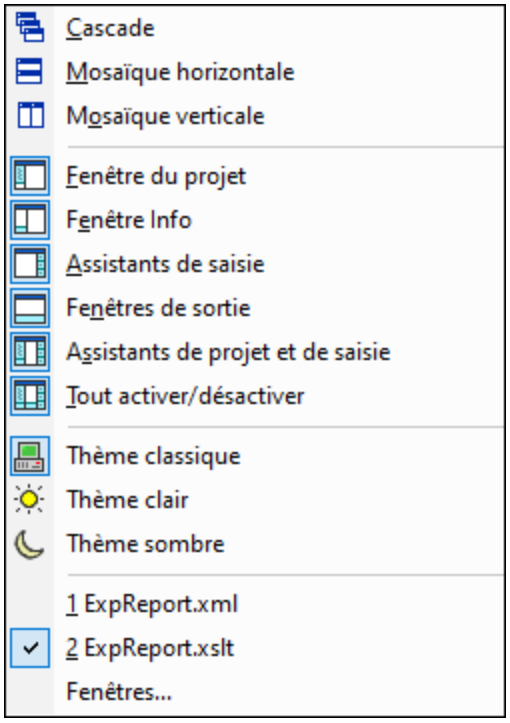
Paramètres de proxy actuels
Fournit un journal verbeux de la détection de proxy. Il peut être réinitialisé avec la touche **Réinitialiser** située à droite du champ *Tester URL* (par exemple, en changeant l'URL de test, ou lorsque les paramètres de proxy ont été modifiés).

- *URL test :* Une URL test peut être utilisée pour voir quel proxy est utilisé pour cette URL spécifique. Aucun E/S n'est effectué avec cette URL. Ce champ ne doit pas être vide si configuration-auto-proxy est utilisé (soit par le biais de *Utiliser paramètres de proxy de système* soit *Configuration proxy automatique*).

© 2016-2022 Altova GmbH

28.18 Menu Fenêtre

Le menu **Fenêtre** contient des commandes qui vous laissent organiser des fenêtres d'application individuelle et de document au sein de la GUI. Vous pouvez mettre en cascade ou en mosaïque des fenêtres de document ouvertes et vous pouvez arranger des assistants de saisie et des fenêtres de sortie, de même que les masquer.



Cascade, Mosaïque horizontale/verticale

La commande **Cascade** arrange les fenêtres de document de telle façon qu'elles sont décalées en séquence de l'arrière vers l'avant.

Les **Mosaïque horizontale** et **Mosaïque verticale** arrangent les fenêtres de documents ouverts et non minimisés pour qu'elles soient redimensionnées en mosaïques qui sont toutes visibles dans la fenêtre d'application.

Fenêtre de projet, fenêtre d'info, assistants de saisie, fenêtres de sortie

Ces commandes activent/désactivent l'affichage respectivement des Fenêtre de projet, fenêtre d'Info, assistants de saisie et fenêtres de sortie.

Chacune de ces fenêtres est une fenêtre ancrable. Glisser sur la barre de titre de la fenêtre la détache de sa position actuelle et en fait une fenêtre flottante. Cliquer à droite de la barre de titre pour permettre l'ancrage ou dissimuler la fenêtre.

Projet et Aides à la saisie

Cette commande vous permet d'activer et de désactiver l'affichage de la Fenêtre de Projet et les Assistants de saisie ensemble. Ainsi, vous ne devez pas activer/désactiver l'affichage de ces fenêtres individuellement.

Tous activés/désactivés

Cette commande vous permet d'activer et de désactiver toutes les fenêtres (*liste ci-dessous*).

- Fenêtre de projet
- Fenêtre d'info
- Assistants de saisie
- Fenêtre Sortie

Ceci est utile si vous voulez masquer toutes les fenêtres non document rapidement, pour avoir un espace d'affichage maximum pour le(s) document(s) sur lesquels vous travaillez.

Thèmes

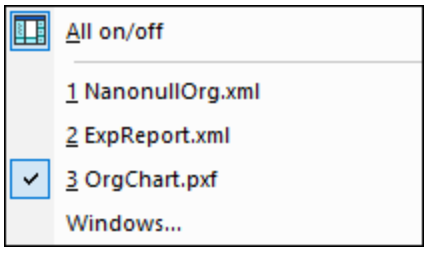
XMLSpy vous offre un choix des trois thèmes recensés ci-dessous. Lorsque vous sélectionnez un thème, il est appliqué immédiatement.

- Classique (le défaut)
- Clair
- Sombre

Pour le thème actif actuel, vous pouvez personnaliser le formatage des composants de texte individuel d'un type de document. Ceci se fait dans les onglets *Polices et couleurs* du dialogue des Options (**Outils | Options**).

Liste de fenêtres actuellement ouvertes

Cette liste montre toutes les fenêtres ouvertes actuellement, et vous permet de basculer rapidement de l'une à l'autre.

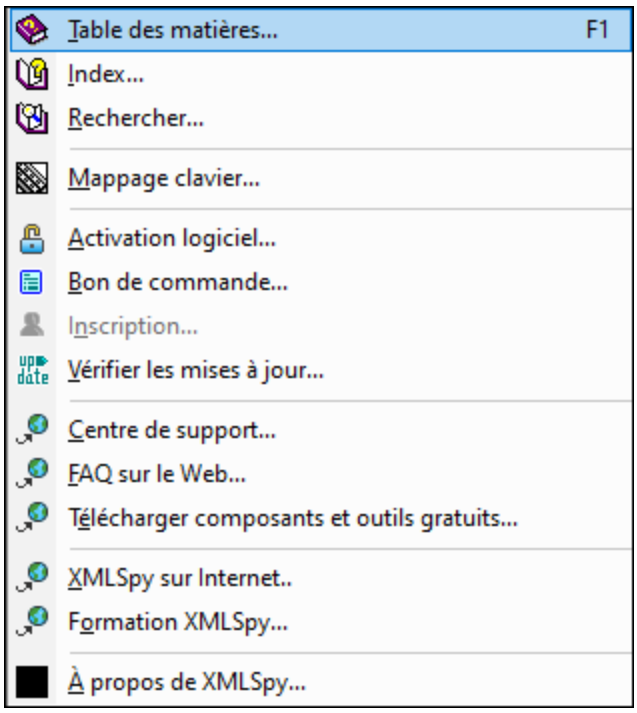


Vous pouvez aussi utiliser les raccourcis clavier **Ctrl+F6** pour passer dans les fenêtres ouvertes.

© 2016-2022 Altova GmbH

28.19 Menu Aide

Le menu **Aide** contient des commandes permettant d'obtenir de l'aide ou des informations supplémentaires sur XMLSpy, et des liens vers des pages d'information et d'assistance sur le serveur web d'Altova.



Le menu **Aide** contient aussi le dialogue Inscription, qui vous permet de saisir votre code clé de licence une fois que vous avez acheté le produit.

28.19.1 Table des matières, Index, Rechercher

- Sommaire

  Ouvre le manuel d'aide sur écran de XMLSpy avec le Sommaire affiché dans le panneau de gauche de la fenêtre Aide. Le Sommaire présente un aperçu de l'ensemble du document Aide. Cliquer sur une entrée dans le Sommaire pour vous rendre dans cette section.

- Index

  Ouvre le manuel d'aide sur écran de XMLSpy avec l'Index de clavier affiché dans le panneau de gauche de la fenêtre Aide. L'index regroupe les mots-clés et vous permet de naviguer vers un thème en double-cliquant le mot-clé. Si un mot-clé est lié à plus d'un thème, une liste de ces thèmes s'affichera.

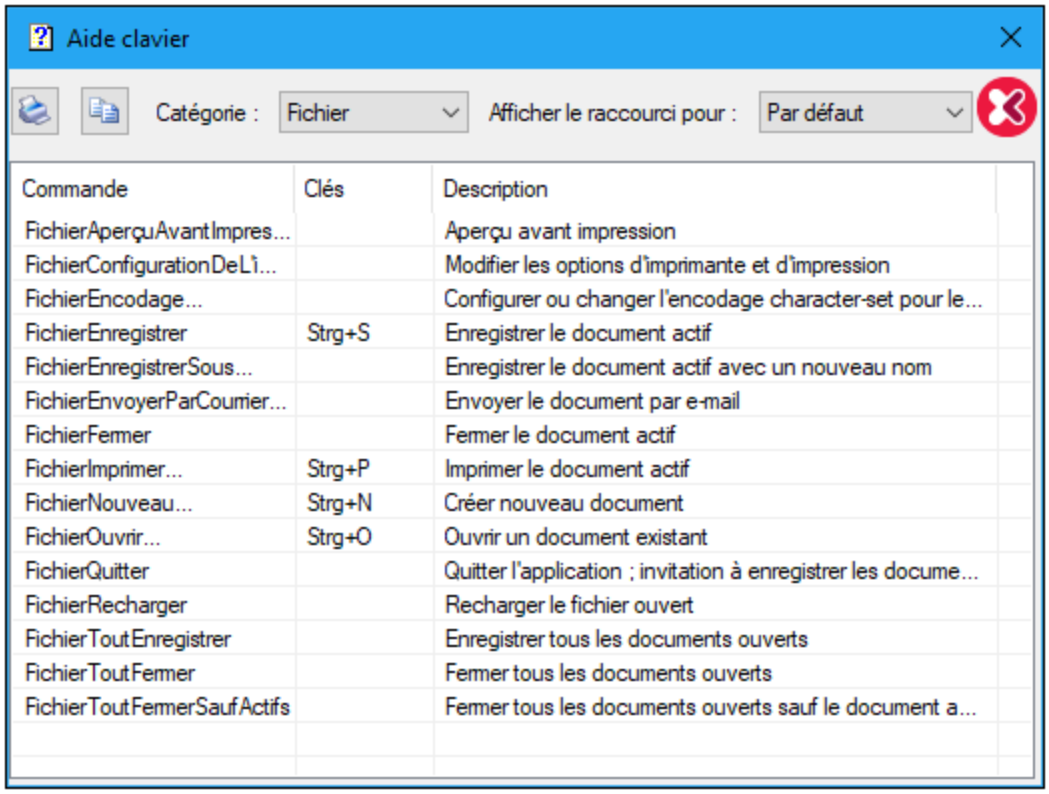
- Recherche

  Ouvre le manuel d'aide sur écran de XMLSpy avec le dialogue Recherche affiché dans le panneau de gauche de la fenêtre Aide. Pour rechercher un terme, saisir le terme dans le champ de saisie et appuyer

(i) sur Entrée ou (ii) sur List topics. Le système d'aide effectuera une recherche complète dans l'ensemble de la documentation d'aide et retourne une liste des rubriques trouvées. Double-cliquer sur un élément pour l'afficher.

28.19.2 Mappage clavier

La commande **Aide | Mappage clavier** déclenche l'ouverture d'une fenêtre d'information. Celle-ci contient une liste menu par menu de toutes les commandes dans XMLSpy. Les commandes de menu sont listées avec une description et des touches de raccourcis pour la commande.



Pour consulter les commandes dans un menu particulier, sélectionner le nom de menu dans la liste de choix Catégorie. Vous pouvez imprimer la commande en cliquant sur l'icône d'imprimante.

Veuillez noter les points suivants concernant les raccourcis :

- Certaines commandes (et leurs raccourcis) sont uniquement utilisables dans un certain mode. Par exemple, la plupart des commandes dans le menu XML sont utilisables uniquement dans le Mode Grille. D'autres commandes (comme **Fichier | Enregistrer** ou **XML | Vérifier la bonne formation**) sont disponibles dans plusieurs modes.
- Autres raccourcis intéressants : par exemple, **Shift+F10** permet d'amener le menu contextuel dans le Mode Texte et le Mode Schéma ; **Ctrl+E** lorsque le curseur se trouve dans un élément démarrer ou terminer dans le Mode Texte déplace le curseur à la fin ou au début de la balise, respectivement.

© 2016-2022 Altova GmbH

- Dans l'onglet Clavier du dialogue Personnaliser, vous pouvez aussi définir vos propres raccourcis pour les différentes commandes de menu.

28.19.3 Activation, Bon de commande, Inscription, Mises à jour

Activation logiciel

Mettre sous licence votre produit
Après avoir téléchargé votre logiciel de produits Altova, vous pourrez acquérir une licence - ou l'activer - en utilisant soit une clé d'évaluation gratuite ou en achetant une clé de licence permanente.

- ***Licence d'évaluation gratuite.*** Lorsque vous lancez le logiciel pour la première fois après l'avoir téléchargé et installé, le dialogue **Activation du logiciel** s'ouvrira. Vous y trouverez un bouton pour demander une licence d'évaluation gratuite. Saisir votre nom, entreprise et votre adresse e-mail dans le dialogue qui apparaît et cliquer sur **Requête.** Un fichier de licence est envoyé à l'adresse e-mail que vous avez saisi et devrait arriver dans quelques minutes. Enregistrer le fichier de licence dans un endroit approprié.

  Lorsque vous cliquez sur **Requête**, un champ de saisie apparaît en bas du dialogue Requête. Ce champ prend le chemin vers le fichier de licence. Chercher ou saisir le champ vers le fichier de licence et cliquer sur **OK**. (Dans le dialogue d'**Activation du logiciel**, vous pouvez aussi cliquer sur **Charger une Nouvelle Licence** pour accéder à un dialogue dans lequel le chemin vers le fichier de licence est saisi.) Le logiciel sera débloqué pour une période de 30 jours.

- ***Clé de licence permanente.*** Le dialogue **Activation du logiciel** contient un bouton pour acheter une clé de licence permanente. Cliquer sur ce bouton pour vous rendre à la boutique en ligne d'Altova, où vous pourrez acheter une clé de licence permanente pour votre produit. Votre licence vous sera envoyée par e-mail sous forme d'un fichier de licence contenant vos données de licence.

  Il existe trois types de licences permanentes : *installée*, *utilisateur simultané*, et *utilisateur nommé*. Une licence installée déverrouille le logiciel sur un seul ordinateur. Si vous achetez une licence installée pour*N* ordinateurs, la licence permettra une utilisation du logiciel sur jusqu'à *N* ordinateurs. Une licence utilisateur concomitant pour *N* utilisateurs concomitants permet à *N* utilisateurs d'exécuter le logiciel simultanément. (Le logiciel peut être installé sur 10N ordinateurs.) Une licence utilisateur nommé autorise un utilisateur spécifique d'utiliser le logiciel sur jusqu'à 5 ordinateurs différents. Pour activer votre logiciel, cliquer sur **Charger une Nouvelle licence**, et, dans le dialogue qui apparaît, chercher ou saisir le chemin vers le fichier de licence et cliquer sur **OK**.

Note : en ce qui concerne les licences utilisateurs multiples, chaque utilisateur sera invité à saisir son nom dans le champ Nom.

Votre e-mail de licence et les différents moyens de mise sous licence de votre produit Altova (activation) à votre disposition
L'e-mail de licence que vous avez reçu de la part d'Altova contient votre fichier de licence en pièce jointe. Le fichier package a une extension de fichier `.altova_licenses`.

Pour activer votre produit Altova, vous pouvez choisir une des étapes suivantes :

- Enregistrez le fchier de licence (`.altova_licenses`) vers un emplacement approprié, double-cliquez sur le fichier de licence, saisissez tout détail requis dans le dialogue qui apparaît, et terminez en cliquant sur **Appliquer clés**.
- Enregistrez le fichier de licence (`.altova_licenses`) vers un emplacement approprié. Dans votre produit Altova, sélectionnez la commande de menu **Aide | Activation Logiciel**, puis **Télécharger une nouvelle licence**. Recherchez ou saisissez le chemin du fichier de licence, et cliquez sur **OK**.
- Enregistrez le fichier de licence (`.altova_licenses`) vers un emplacement approprié et chargez-le depuis cet emplacement vers le pool de licences de votre Altova LicenseServer Vous pouvez ensuite : (i) acquérir la licence depuis votre produit Altova via le dialogue d'Activation du logiciel du produit *(voir ci-dessous)* ou (ii) attribuer la licence au produit depuis l'Altova LicenseServer. *Pour plus d'informations concernant la mise sous licence via LicenseServer, lire le reste de cette rubrique.*

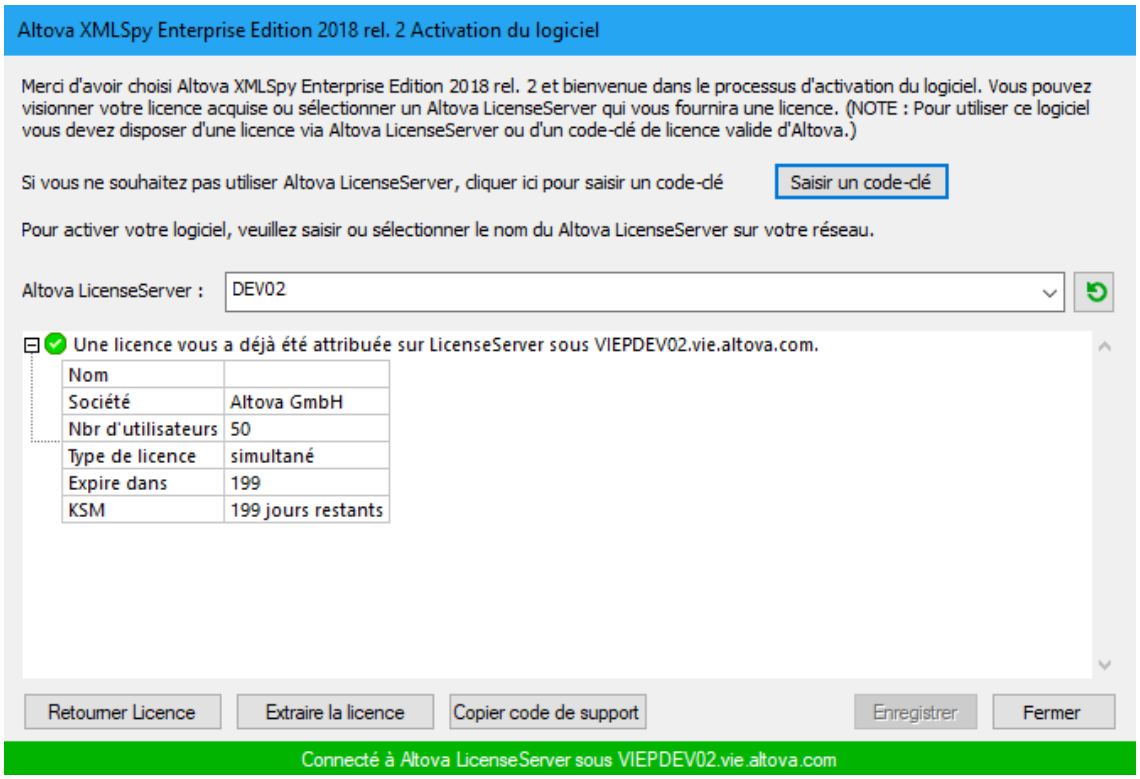
Le dialogue **Activation du logiciel** (*capture d'écran ci-dessous*) peut être accédé à tout moment en cliquant sur la commande **Aide | Activation du logiciel**.

Activer votre logiciel
Vous pouvez activer le logiciel en inscrivant le dialogue Activation du logiciel ou en enregistrant la licence par le biais du Serveur de licence Altova (*voir les détails ci-dessous*).

- *Enregistrant la licence dans le dialogue d'Activation du logiciel*. Dans le dialogue, cliquez sur **Charger une Nouvelle Licence**, puis cherchez et sélectionnez le fichier de licence. Cliquez sur **OK** pour confirmer le chemin vers le fichier de licence et confirmez toutes les données que vous avez saisies (votre nom dans le cas de licences multi-utilisateur). Terminez en cliquant **Enregistrer**.

- *Mise sous licence par le biais du Serveur de licence Altova sur votre réseau :* Pour acquérir une licence par le biais d'un Altova LicenseServer sur votre réseau, cliquez sur **Utiliser Altova LicenseServer**, situé en bas du dialogue **Activation du logiciel**. Choisissez l'appareil sur lequel le LicenseServer que vous souhaitez utiliser a été installé. Veuillez noter que l'auto-découverte des License Servers fonctionne par le biais d'une diffusion envoyée sur le LAN. Puisque les diffusions sont limitées à un sous-réseau, License Server doit se trouver sur le même sous-réseau que l'appareil client pour la découverte automatique afin de fonctionner. Si elle ne fonctionne pas, saisir le nom du serveur. L'Altova LicenseServer doit disposer d'une licence pour votre produit Altova dans son pool de licence. Si une licence est disponible dans le pool de LicenseServer, cela sera indiqué dans le dialogue d'**Activation du logiciel** (*la capture d'écran ci-dessous affiche le dialogue dans Altova XMLSpy*). Cliquez sur **Enregistrer** pour acquérir la licence.

© 2016-2022 Altova GmbH



Une fois qu'une licence spécifique aux appareils (aka installée) a été acquise depuis LicenseServer, elle ne peut pas être retournée au LicenseServer pour une période de sept jours. Après cette période, vous pouvez rendre la licence installée (cliquer sur **Retourner licence**) de manière à ce que la licence puisse être acquise depuis LicenseServer par un autre client. (Néanmoins, un administrateur de LicenseServer, peut désattribuer à tout moment une licence acquise par le biais de la Web UI du LicenseServer). Veuillez noter qu'un renvoi de la licence n'est applicable qu'aux seules licences sur appareil installées, pas aux licences concurrentes.

Extraire la licence

Vous pouvez consulter une licence du pool de licence pour une période de jusqu'à 30 jours pour que la licence puisse être stockée sur l'appareil de produit. Cela vous permet de travailler hors ligne, ce qui peut être utile, par exemple, si vous souhaitez travailler dans un environnement où vous ne pourrez pas accéder à votre Altova LicenseServer (par exemple, si votre produit Altova est installé sur un ordinateur portable et que vous vous trouvez en déplacement). Tant que la licence est extraite, LicenseServer affiche la licence comme étant utilisée ; elle ne peut donc pas être utilisée par une autre machine. La licence passe automatiquement à l'état d'archivage lorsque la période d'extraction expire. En alternative, une licence extraite peut être archivée à tout moment par le biais du bouton **Archiver** du dialogue d'**Activation du logiciel**.

Pour extraire une licence, procédez comme suit : (i) dans le dialogue d'A**ctivation du logiciel**, cliquez sur **Extraire licence** (*voir la capture d'écran ci-dessus*); (ii) dans le dialogue d'**extraction de la licence** qui apparaît, sélectionnez la période d'extraction que vous souhaitez et cliquez sur **Extraire**. La licence sera extraite. Après avoir extrait la licence, deux choses se produisent : (i) Le dialogue d'**Activation du logiciel** affichera les informations d'extraction, y compris l'heure à laquelle l'extraction expirera, (ii) le bouton **Extraire licence** dans le dialogue se transforme en un bouton **Archiver**. Vous pouvez archiver la licence à nouveau à tout moment en cliquant sur

Archiver. Étant donné que la licence passe automatiquement au statut Archiver à l'issue de la période d'extraction, assurez-vous que la période d'extraction que vous avez choisie couvre bien la période pendant laquelle vous travaillerez hors ligne.

Les enregistrements de licence doivent correspondre à la même version majeure du produit duquel la licence a été extraite. Donc veillez à enregistrer une licence avant de mettre à niveau votre produit Altova vers la prochaine version majeure.

Note : afin de pouvoir effectuer des extractions de licence, la fonction d'extraction doit être activée sur le LicenseServer. Si la fonction n'a pas été activée, vous recevrez un message d'erreur à cet effet lorsque vous essayez de faire le « check out ». Dans ce cas, veuillez contacter votre administrateur de LicenseServer.

Copier code de support
Cliquer sur **Copier code de support** pour copier des détails de licence dans le presse-papiers. Il s'agit des données que vous devrez fournir en cas de demande d'assistance avec le formulaire d'assistance en ligne.

Altova LicenseServer offre aux administrateurs IT un aperçu en temps réel de toutes les licences Altova sur un réseau, avec les détails de chaque licence, ainsi que les attributions clients et l'utilisation client des licences. L'avantage d'utiliser LicenseServer réside donc dans les fonctions administratives qu'il offre pour la gestion de licence à large volume d'Altova. Altova LicenseServer est disponible gratuitement depuis le site web Altova. Pour plus d'informations concernant Altova LicenseServer et la mise sous licence par le biais d'Altova LicenseServer, voir la documentation Altova LicenseServer.

- Formulaire de commande

  Lorsque vous êtes prêt pour commander une version de licence du produit de logiciel, vous pouvez soit utiliser la touche **Acheter une clé de licence permanente** dans le dialogue **Activation du logiciel** (*voir la section précédente*) ou la commande **Formulaire de commande** pour continuer vers la boutique en ligne Altova sécurisée.

- Inscription

  Ouvre la page d'enregistrement du produit Altova dans un onglet de votre navigateur. L'enregistrement de votre logiciel Altova vous aidera à vous assurer de toujours rester à jour avec les dernières informations du produit.

- Vérifier les mises à jour

  Contrôle sur le serveur Altova si une version plus récente que la vôtre est actuellement disponible et, dans l'affirmative, affiche un message approprié.

28.19.4 Autres commandes

- Centre de support

  Un lien qui vous mènera vers le Centre de support Altova sur Internet. Le Centre de support contient des FAQ, des forums de discussion pour toute sorte de problèmes et l'accès à l'équipe de support technique

© 2016-2022 Altova GmbH

d'Altova.

- FAQ sur le web

  Un lien menant à la base de données FAQ d'Altova sur Internet. La base de données FAQ est constamment mise à jour à la suite des problèmes rapportés par les clients.

- Télécharger les composants et les outils gratuits

  Un lien menant au Centre de téléchargement des composants Altova sur Internet. À partir de là, vous pouvez télécharger une variété de logiciels complémentaires à utiliser avec des produits Altova. Ces logiciels vont de processeurs XSLT et XSL-FO à des Plateformes de serveur d'application. Les logiciels disponibles dans le Centre de téléchargement des composants sont généralement gratuits.

- XMLSpy sur Internet

  Un lien menant au site web Altova sur Internet. Vous pouvez en apprendre plus sur XMLSpy, les technologies et produits liés le le site web Altova.

- Formation XMLSpy

  Un lien menant à la page de Formation en ligne sur le site web Altova. Ici, vous pouvez choisir parmi une série de cours en ligne tenus par des formateurs experts Altova.

- À propos de XMLSpy

  Affiche la fenêtre d'accueil et le numéro de version de votre produit. Si vous utilisez la version 64-bit de XMLSpy, cela est indiqué par le suffixe (x64) placé après le nom de l'application. Il n'y a pas de suffixe pour la version 32-bit.

28.20 Ligne de commande

Certaines actions XMLSpy peuvent être réalisées depuis la ligne de commande. Ces commandes sont listées ci-dessous :

Ouvrir un fichier

xmlspy.exe file.xml

Ouvre le fichier file.xml dans XMLSpy

Ouvrir de multiples fichiers

XSLT1 file1.xml file2.xml

Ouvre les fichiers file1.xml et file2.xml dans XMLSpy

Attribue un fichier SPS vers un fichier XML pour l'édition du mode Authentic

XSLT1 myxml.xml /sps mysps.sps

Ouvre le fichier myxml.xml en mode Authentic avec mysps.sps comme son fichier SPS. L'indicateur /sps spécifie que le fichier SPS qui suit doit être utilisé avec un fichier XML qui précède l'indicateur /sps (pour l'édition du mode Authentic).

Ouvrir un nouveau modèle de fichier XML par le biais d'un fichier SPS

xmlspy.exe mysps.sps

Ouvrir le fichier XML dans Authentic View. L'affichage sera basé sur le SPS et le nouveau fichier XML aura une structure squelletique basée sur le schéma SPS. Le nom du nouveau fichier XML doit être assigné lorsque vous enregistrez le fichier XML.

Ouvrir un fichier SPS comme document XML dans le mode Texte

xmlspy.exe /raw mysps.sps

Ouvrir le fichier mysps.sps comme document XML dans le mode Texte. L'indicateur /raw spécifie que le fichier SPS qui suit est à éditer comme fichier XML.

© 2016-2022 Altova GmbH

29 Programmers' Reference

XMLSpy is an Automation Server: It exposes programmable objects to other applications called Automation Clients. An Automation Client can directly access the objects and functionality that the Automation Server makes available. So, an Automation Client of XMLSpy can use, for example, the XML validation functionality of XMLSpy. As a consequence, developers can enhance their applications with the ready-made functionality of XMLSpy.

The programmable objects of XMLSpy are made available to Automation Clients via the Application API of XMLSpy, which is a COM API. The Application API of XMLSpy will also be called Application API for short from now onwards. The object model of the Application API and a complete description of all the available objects are provided in this documentation (see the section Application API).

Execution environments

The Application API can be accessed from within the following environments:

- Scripting Editor
- IDE Plug-ins
- External programs
- ActiveX Integration

Each of these environments is described briefly below.

Scripting Editor: Customizing and modifying XMLSpy functionality
You can customize your installation of XMLSpy by modifying and adding functionality to it. You can also create Forms for user input and modify the user interface so that it contains new menu commands and toolbar shortcuts. All these features are achieved by writing scripts that interact with objects of the Application API. To aid you in carrying out these tasks efficiently, XMLSpy offers you an in-built Scripting Editor. A complete description of the functionality available in the Scripting Editor and how it is to be used is given in the Scripting Editor section of this documentation. The supported programming languages are **JScript** and **VBScript**.

IDE Plug-ins: Creating plug-ins for XMLSpy
XMLSpy enables you to create your own plug-ins and integrate them into XMLSpy. You can do this using XMLSpy's special interface for plug-ins. A description of how to create plug-ins is given in the section XMLSpy IDE Plug-ins.

An application object gets passed to most methods that must be implemented by an IDE plug-in and gets called by the application. Typical languages used to implement an IDE plug-in are **C#** and **C++**. For more information, see the section XMLSpy IDE Plugins.

External programs

Additionally, you can manipulate XMLSpy with external scripts. For example, you could write a script to open XMLSpy at a given time, then open an XML file in XMLSpy, validate the file, and print it out. External scripts would again make use of the Application API to carry out these tasks. For a description of the Application API, see the section Application API.

Using the Application API from outside XMLSpy requires an instance of XMLSpy to be started first. How this is done depends on the programming language used. See the section, Programming Languages, for information about individual languages.

Essentially, XMLSpy will be started via its COM registration. Then the `Application` object associated with the XMLSpy instance is returned. Depending on the COM settings, an object associated with an already running XMLSpy can be returned. Any programming language that supports creation and invocation of COM objects can be used. The most common of these are listed below.

- JScript and VBScript script files have a simple syntax and are designed to access COM objects. They can be run directly from a DOS command line or with a double click on Windows Explorer. They are best used for simple automation tasks.
- C# is a a full-fledged programming language that has a wide range of existing functionality. Access to COM objects can be automatically wrapped using C#..
- C++ provides direct control over COM access but requires relatively larger amounts of code than the other languages.
- Java: Altova products come with native Java classes that wrap the Application API and provide a full Java look-and-feel.
- Other programming languages that make useful alternatives are: Visual Basic for Applications, Perl, and Python.

ActiveX Integration

A special case of accessing the Application API is via the XMLSpy ActiveX control. This feature is only available if the XMLSpy integration package is installed. Every ActiveX Control has a property that returns a corresponding COM object for its underlying functionality. The manager control provides an `Application` object, the document control a `Document` object, and the placeholder object, in cases where it contains the project tree, returns the `Project` object. The methods supported by these objects are exactly as described in the Interfaces section of the Application API. Care must be taken not to use methods that do not make sense in the context of ActiveX control integration. For details see ActiveX Integration.

About Programmers' Reference

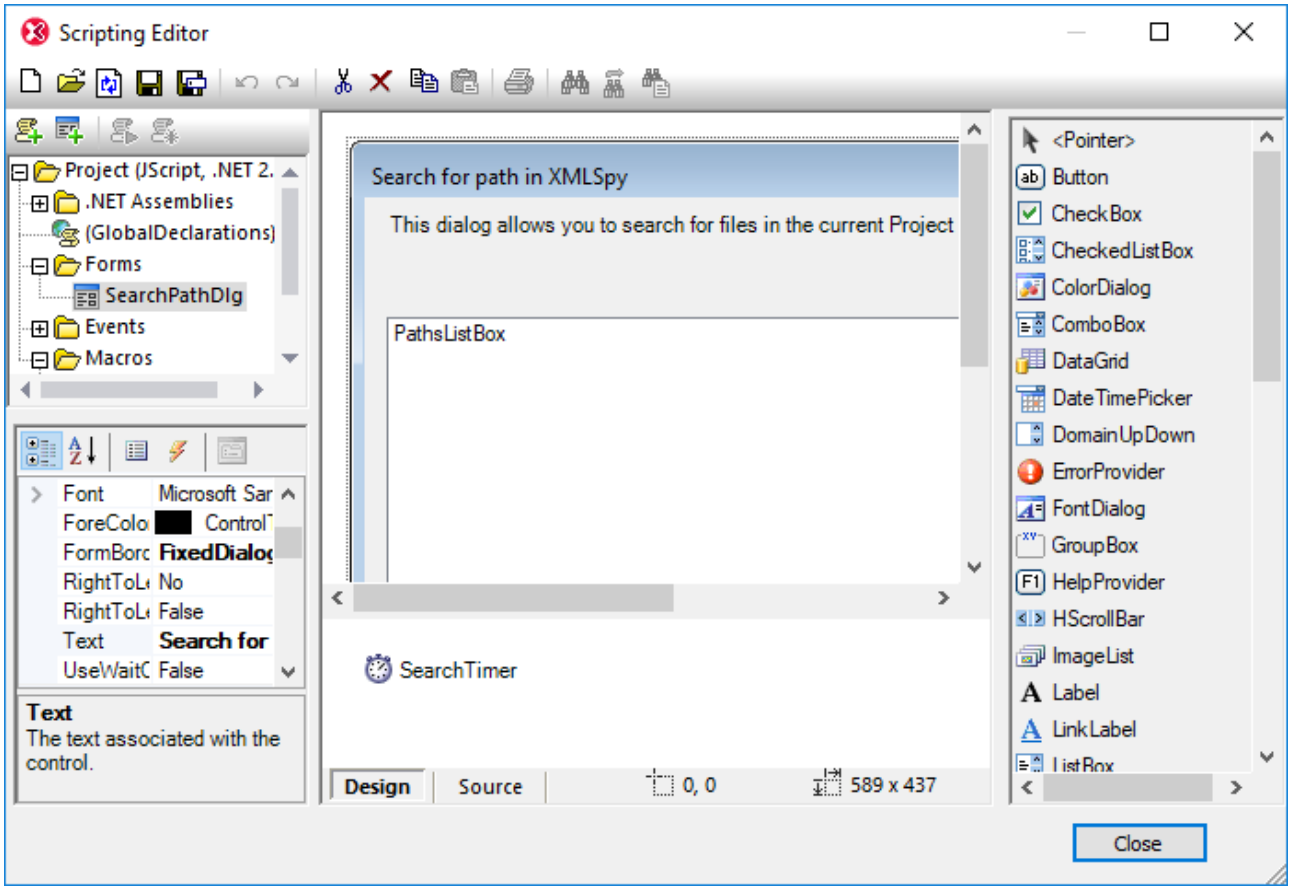
The documentation contained in the Programmers' Reference for XMLSpy consists of the following sections:

- Scripting Editor: a user reference for the Scripting Environment available in XMLSpy
- IDE Plug-ins: a description of how to create plug-ins for XMLSpy
- Application API: a reference for the Application API
- ActiveX Integration: a guide and reference for how to integrate the XMLSpy GUI and XMLSpy functionality using an ActiveX control

© 2016-2022 Altova GmbH

29.1 Scripting Editor

Scripting Editor is a development environment built into XMLSpy from where you can customize the functionality of XMLSpy with the help of JScript or VBScript scripts. For example, you can add a new menu item to perform a custom project task, or you can have XMLSpy trigger some behavior each time when a document is opened or closed. To make this possible, you create scripting projects—files with .asprj extension (Altova Scripting Project).



Scripting Editor

Scripting projects typically include one or several macros—these are programs that perform miscellaneous custom tasks when invoked. You can run macros either explicitly from a menu item (or a toolbar button, if configured), or you can set up a macro to run automatically whenever XMLSpy starts. The scripting environment also integrates with the XMLSpy COM API. For example, your VBScript or JScript scripts can handle application or document events such as starting or shutting down XMLSpy, opening or closing a project, and so on. Scripting projects can include Windows Forms that you can design visually, in a way similar to Visual Studio. In addition, several built-in commands are available that help you instantiate and use .NET classes from VBScript or JScript code.

Once your scripting project is complete, you can enable it either globally in XMLSpy, or only for specific projects.

Scripting Editor requires .NET Framework 2.0 or later to be installed before XMLSpy is installed.

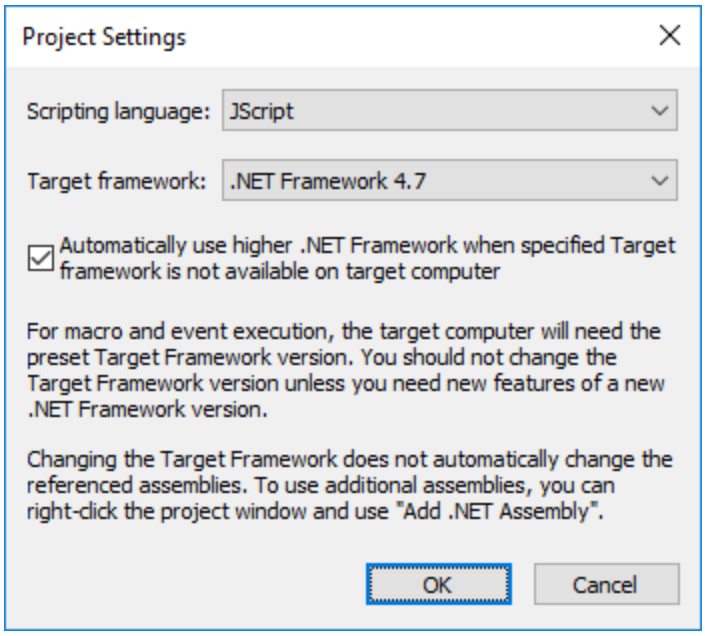
29.1.1 Creating a Scripting Project

All scripts and scripting information created in the Scripting Editor are stored in Altova Scripting styleclass="Code Bold" (.asprj files). A scripting project may contain macros, application event handlers, and forms (which can have their own event handlers). In addition, you can add global variables and functions to a "Global Declarations" script—this makes such variables and functions accessible across the entire project.

To start a new project, run the menu command **Tools | Scripting Editor**.

The languages supported for use in a scripting project are JScript and VBScript (not to be confused with Visual Basic, which is not supported). These scripting engines are available by default on Windows and have no special requirements to run. You can select a scripting language as follows:

1. Right-click the **Project** item in the upper-left pane, and select **Project settings** from the context menu.
2. Select a language (JScript or VBScript), and click **OK**.



From the Project settings dialog box above, you can also change the target .NET Framework version. This is typically necessary if your scripting project requires features available in a newer .NET Framework version. Note that any clients using your scripting project will need to have the same .NET Framework version installed (or a later compatible version).

By default, a scripting project references several .NET assemblies, like `System`, `System.Data`, `System.Windows.Forms`, and others. If necessary, you can import additional .NET assemblies, including assemblies from .NET Global Assembly Cache (GAC) or custom .dll files. You can import assemblies as follows:

1. Statically, by adding them manually to the project. Right-click **Project** in the top-left pane, and select **Add .NET Assembly** from the context menu.

 © 2016-2022 Altova GmbH

2. Dynamically, at runtime, by calling the `CLR.LoadAssembly` command from the code.

You can create multiple scripting projects if necessary. You can save a scripting project to the disk, and then load it back into the Scripting Editor later. To do this, use the standard Windows buttons available in the toolbar: **New**, **Open**, **Save**, **Save As**. Once the scripting project has been tested and is ready for deployment, you can load it into XMLSpy and run any of its macros or event handlers. For more information, see Enabling Scripts and Macros.

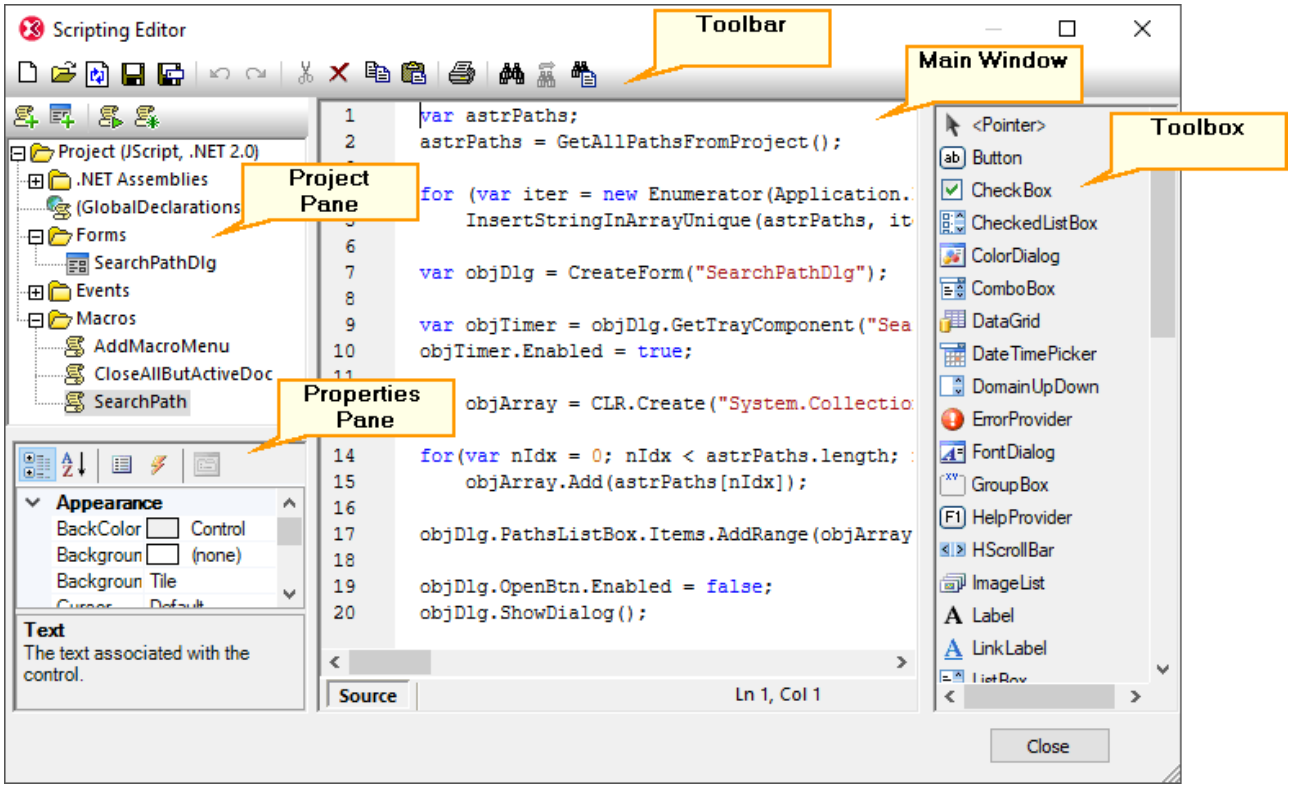
You can also find an example scripting project at the following path: **C:\Users\<user>\Documents\Altova\XMLSpy2023\Examples\SampleScripts.asprj**.

The next sections focus on the parts that your scripting project may need: global declarations, macros, forms, and events.

29.1.1.1 Overview of the Environment

The Scripting Editor consists of the following parts:

- Toolbar
- Project pane
- Properties pane
- Main window
- Toolbox



Toolbar

The toolbar includes standard Windows file management commands (**New**, **Open**, **Save**, **Save As**) commands and editor commands (**Copy**, **Cut**, **Delete**, **Paste**). When editing source code, the **Find** and **Replace** commands are additionally available, as well as the **Print** command.

Project pane

The project pane helps you view and manage the structure of the project. A scripting project consists of several components that can work together and may be created in any order:

- A *"Global Declarations"* script. As the name suggests, this script stores information available globally across the project. You can declare in this script any variables or functions that you need to be available in all forms, event handler scripts, and macros.
- *Forms.* Forms are typically necessary to collect user input, or provide some informative dialog boxes. A form is invoked by a call to it either within a function (in the Global Declarations script) or directly in a macro.
- *Events.* The "Events" folder displays XMLSpy application events provided by the COM API. To write a script that will be executed when an event occurs, double-click any event, and then type the handling code in the editor. The application events should not be confused with form events; the latter are handled at form level, as further detailed below.
- *Macros*. A macro is a script that can be invoked either on demand from a context menu or be executed automatically when XMLSpy starts. Macros do not have parameters or return values. A macro can access all variables and functions declared in the Global Declarations script and it can also display forms.

Right-click any of the components to see the available context menu commands and their shortcuts. Double-click any file (such as a form or a script) to open it in the main window.

The toolbar buttons provide the following quick commands:

	**New macro**	Adds a new macro to the project, in the **Macros** directory.
	**New form**	Adds a new form to the project, in the **Forms** directory.
	**Run macro**	Runs the selected macro.
	**Debug macro**	Runs the selected macro in debug mode.

Properties pane

The Properties pane is very similar to the one in Visual Studio. It displays the following:

- Form properties, when a form is selected
- Object properties, when an object in a form is selected
- Form events, when a form is selected
- Object events, when an object in a form is selected

To switch between the properties and events of the selected component, click the **Properties**  or **Events**  buttons, respectively.

© 2016-2022 Altova GmbH

The **Categorized** and **Alphabetical** icons display the properties or events either organized by category or organized in ascending alphabetical order.

When a property or event is selected, a short description of it is displayed at the bottom of the Properties pane.

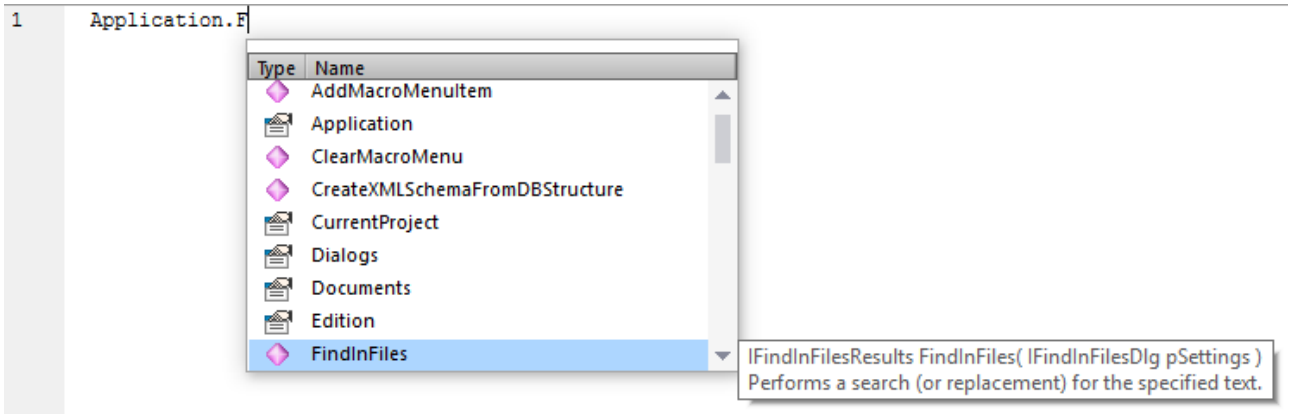
Main window

The main window is the working area where you can enter source code or modify the design of the form. When editing forms, you can work in two tabs: the **Design** tab and the **Source** tab. The **Design** tab shows the layout of the form, while the **Source** tab contains the source code such as handler methods for the form events.

The source code editor provides code editing aids such as syntax coloring, source code folding, highlighting of starting and ending braces, zooming, autocompletion suggestions, bookmarks.

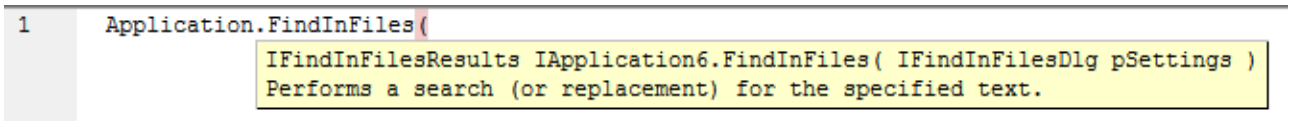
Autocompletion suggestions

JScript and VBScript are untyped languages, so autocompletion is limited to COM API names and XMLSpy built-in commands. The full method or property signature is shown next to the autocompletion entry helper.



If names start with `objDocument`, `objProject`, `objXMLData`, or `objAuthenticRange`, members of the corresponding interface will be shown.

Placing the mouse over a known method or property displays its signature (and documentation if available), for example:



The auto-completion entry helper is normally shown automatically during editing, but it can also be obtained on demand by pressing **Ctrl+Space**.

Bookmarks

- To set or remove a bookmark, click inside a line, and then press **Ctrl+F2**
- To navigate to the next bookmark, press **F2**
- To navigate to the previous bookmark, press **Shift+F2**
- To delete all bookmarks, press **Ctrl+Shift+F2**

Zooming in/out

- To zoom in or out, hold the **Ctrl** key pressed and then press the "+" or "-" keys or rotate the mouse wheel.

Text view settings

To trigger text settings, right-click inside the editor, and select **Text View Settings** from the context menu.

Font settings

To change the font, right-click inside the editor, and select **Text View Font** from the context menu.

Toolbox

The Toolbox contains all the objects that are available for designing forms, such as buttons, text boxes, combo boxes, and so on.

To add a Toolbox item to a form:

1. Create or open a form and make sure that the **Design** tab is selected.
2. Click the Toolbox object (for example, **Button**), and then click at the location in the form where you wish to insert it. Alternatively, drag the object directly onto the form.

Some objects such as `Timer` are not added to the Form but are created in a tray at the bottom of the main window. You can select the object in the tray and set properties and event handlers for the object from the Properties pane. For an example of handling tray components from the code, see Handling form events.

You can also add registered ActiveX controls to the form. To do this, right-click the Toolbox area and select **Add ActiveX Control** from the context menu.

29.1.1.2 Global Declarations

The "Global Declarations" script is present by default in any scripting project; you do not need to create it explicitly. Any variables or functions that you add to this script are considered global across the entire project. Consequently, you can refer to such variables and functions from any of the project's macros and events. The following is an example of a global declarations script that imports the `System.Windows.Forms` namespace into the project. To achieve that, the code below invokes the `CLR.Import` command built into Scripting Editor.

```
// import System.Windows.Forms namespace for all macros, forms and events:
CLR.Import( "System.Windows.Forms" );
```

Note: Every time a macro is executed or an event handler is called, the global declarations are re-initialized.

29.1.1.3 Macros

Macros are scripts that contain JScript (or VBScript, depending on your project's language) statements, such as variable declarations and functions.

© 2016-2022 Altova GmbH

If your projects should use macros, you can add them as follows: right-click inside the Project pane, select **Add Macro** from the context menu, and then enter the macro's code in the main form. The code of a macro could be as simple as an alert, for example:

```
alert("Hello, I'm a macro!");
```

More advanced macros can contain variables and local functions. Macros can also contain code that invokes forms from the project. The listing below illustrates an example of a macro that shows a form. It is assumed that this form has already been created in the "Forms" folder and has the name "SampleForm", see also Forms.

```
// display a form
ShowForm( "SampleForm" );
```

In the code listing above, `ShowForm` is a command built into Scripting Editor. For reference to other similar commands that you can use to work with forms and .NET objects, the Built-in Commands.

You can add multiple macros to the same project, and you can designate any macro as "auto-macro". When a macro is designated as "auto-macro", it runs automatically when XMLSpy starts. To designate a macro as auto-macro, right-click it, and select **Set as Auto-Macro** from the context menu.

Only one macro can be run at a time. After a macro (or event) is executed, the script is closed and global variables lose their values.

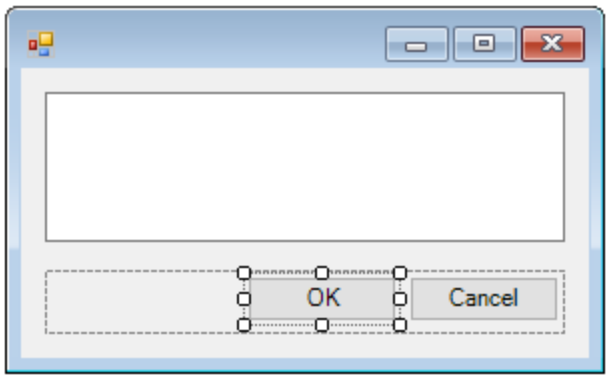
To run a macro directly in Script Editor, click **Run Macro** . To debug a macro using the Visual Studio debugger, click **Debug Macro** . For information about enabling and running macros in XMLSpy, see Enabling Scripts and Macros.

29.1.1.4 Forms

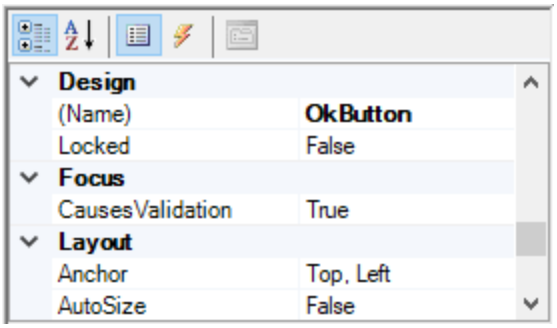
Forms are particularly useful if you need to collect input data from users or display data to users. A form can contain miscellaneous controls to facilitate this, such as buttons, check boxes, combo boxes, and so on.

To add a form, right-click inside the Project pane, and then select **Add Form** from the context menu. To add a control to a form, drag it from the Toolbox available to the right side of Scripting Editor and drop it onto the form.

You can change the position and size of the controls directly on the form, by using the handles that appear when you click any control, for example:



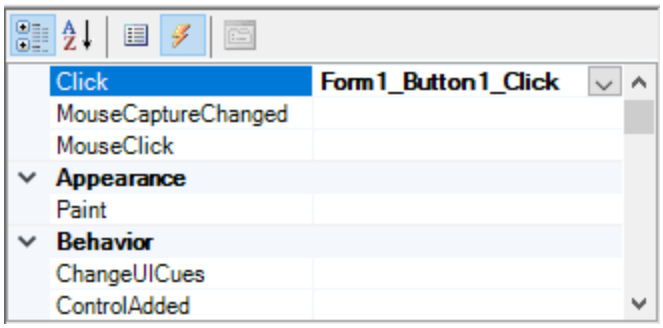
All form controls have properties that you can easily adjust in the Properties pane. To do this, first select the control on the form, and then edit the required properties in the Properties pane.



Handling form events

Each form control also exposes various events to which your scripting project can bind. For example, you might want to invoke some XMLSpy COM API method whenever a button is clicked. To create a function that binds to a form event, do the following:

1. In the Properties pane, click **Events** .
2. In the **Action** column, double-click the event where you need the method (for example, in the image below, the handled event is "Click").



© 2016-2022 Altova GmbH

You can also add handler methods by double-clicking a control on the form. For example, double-clicking a button in the form design generates a handler method for the "Click" event of that button.

Once the body of the handler method is generated, you can type code that handles this event, for example:

```
//Occurs when the component is clicked.
function MyForm_ButtonClick( objSender, e_EventArgs )
{
   alert("A button was clicked");
}
```

To display a work-in-progress form detached from the Scripting Editor, right-click the form, and select **Test Form** from the context menu. Note that the **Test Form** command just displays the form; the form's events (such as button clicks) are still disabled. To have the form react to events, call it from a macro, for example:

```
// Instantiate and display a form
ShowForm( "SampleForm" );
```

Accessing form controls

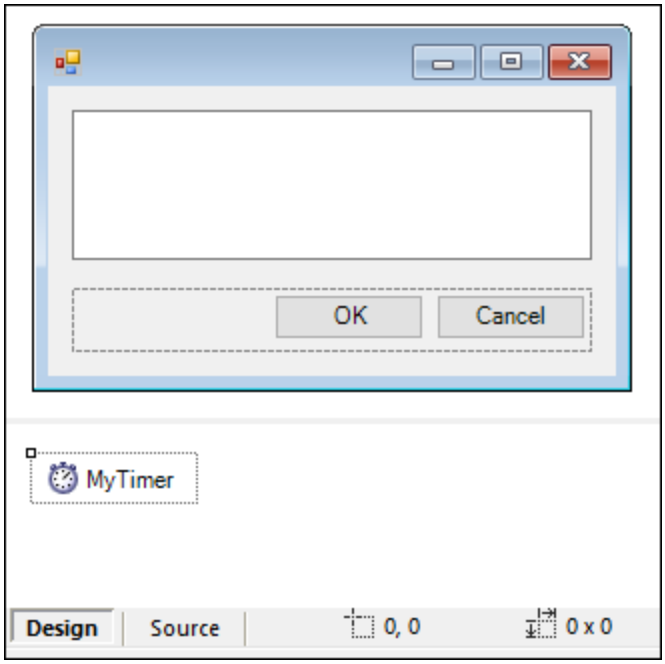
You can access any components on a form from your code by using field access syntax. For example, suppose there is a form designed as follows:

```
// MyForm
//   ButtonPanel
//      OkButton
//      CancelButton
//   TextEditor
//      AxMediaPlayer1
// TrayComponents
//   MyTimer
```

The code below shows how to instantiate the form, access some of its controls using field access syntax, and then display the form:

```
// Instantiate the form
var objForm = CreateForm("MyForm");
// Disable the OK button
objForm.ButtonPanel.OkButton.Enabled = false;
// Change the text of TextEditor
objForm.TextEditor.Text = "Hello";
// Show the form
objForm.ShowDialog();
```

When you add certain controls such as timers to the form, they are not displayed on the form; instead, they are shown as tray components at the base of the form design, for example:



To access controls from the tray, use the `GetTrayComponent` method on the form object, and supply the name of the control as argument. In this example, to get a reference to `MyTimer` and enable it, use the following code:

```
var objTimer = objForm.GetTrayComponent("MyTimer");
objTimer.Enabled = true;
```

For ActiveX Controls, you can access the underlying COM object via the OCX property:

```
var ocx = lastform.AxMediaPlayer1.OCX; // get underlying COM object
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

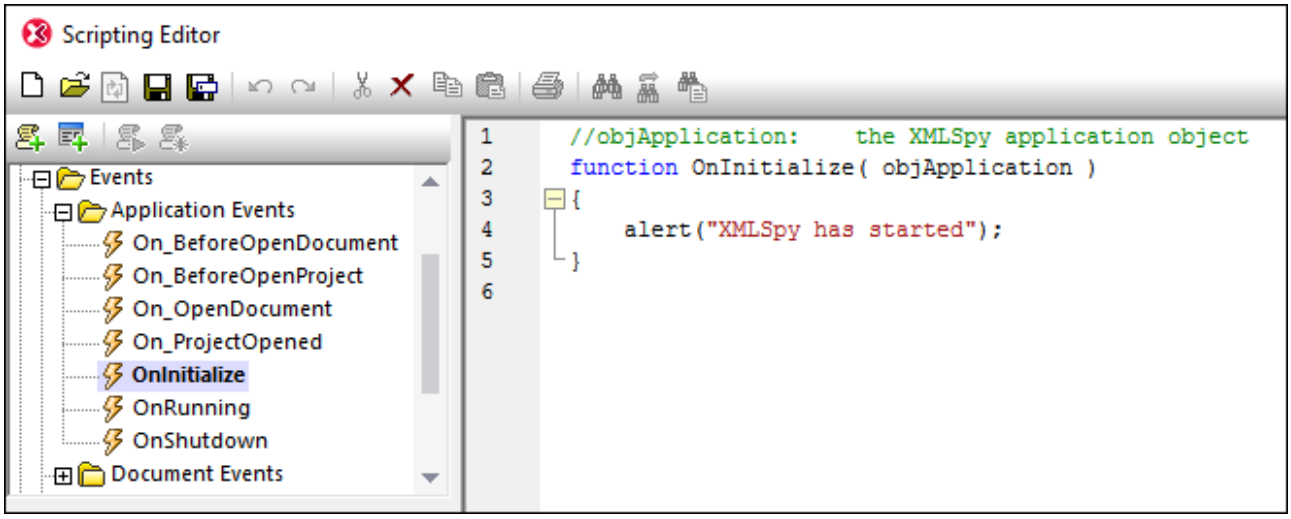
29.1.1.5 Events

Your scripting project may optionally include scripts that handle XMLSpy events such as opening, closing, or saving a document, starting or closing XMLSpy, adding an element to a diagram, and others. These events are provided by the XMLSpy COM API, and you can find them in the "Events" folder of your scripting project. Note that these events are XMLSpy-specific, as opposed to form events. Events are organized into folders as follows:

- Application Events
- Document Events
- AuthenticView Events
- GridView Events
- TextView Evets

© 2016-2022 Altova GmbH

To create an event handler script, right-click an event, and select **Open** from the context menu (or double-click the event). The event handler script is displayed in the main window, where you can start editing it. For example, the event handler illustrated below displays an alert each time XMLSpy starts:



Note the following:

- The `alert` command is applicable to JScript. The VBScript equivalent is `MsgBox`. See also alert.
- The name of the event handler function must not be changed; otherwise, the event handler script will not be called.
- In order for events to be processed, the **Process Events** check box must be selected when you enable the scripting project in XMLSpy. For more information, see Enabling Scripts and Macros.

You can optionally define local variables and helper functions within event handler scripts, for example:

```
var local;

function OnInitialize( objApplication )
{
   local = "OnInitialize";
   Helper();
}

function Helper()
{
   alert("I'm a helper function for " + local);
}
```

29.1.1.6 JScript Programming Tips

Below are a few JScript programming tips that you may find useful while developing a scripting project in XMLSpy Scripting Editor.

Out parameters

Out parameters from methods of the.NET Framework require special variables in JScript. For example:

```
var dictionary =
CLR.Create("System.Collections.Generic.Dictionary<System.String,System.String>");
dictionary.Add("1", "A");
dictionary.Add("2", "B");

// use JScript method to access out-parameters
var strOut = new Array(1);
if ( dictionary.TryGetValue("1", strOut) ) // TryGetValue will set the out parameter
  alert( strOut[0] ); // use out parameter
```

Integer arguments

.NET Methods that require integer arguments should not be called directly with JScript number objects which are floating point values. For example, instead of:

```
var objCustomColor = CLR.Static("System.Drawing.Color").FromArgb(128,128,128);
```

use:

```
var objCustomColor =
CLR.Static("System.Drawing.Color").FromArgb(Math.floor(128),Math.floor(128),Math.floor(12
8));
```

Iterating .NET collections

To iterate .NET collections, the JScript Enumerator as well as the .NET iterator technologies can be used, for example:

```
// iterate using the JScript iterator
var itr = new Enumerator( coll );
for ( ; !itr.atEnd(); itr.moveNext() )
  alert( itr.item() );

// iterate using the .NET iterator
var itrNET = coll.GetEnumerator();
while( itrNET.MoveNext() )
  alert( itrNET.Current );
```

.NET templates

.NET templates can be instantiated as shown below:

```
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
```


© 2016-2022 Altova GmbH

or

```
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary<String,Dictionary<String,String>>" );
```

.NET enumeration values

.NET enumeration values are accessed as shown below:

```
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch;
```

Enumeration literals

The enumeration literals from the XMLSpy API can be accessed as shown below (there is no need to know their numerical value).

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

29.1.1.7 Example Scripting Project

A demo project that illustrates scripting with XMLSpy is available at the following path: **C:\Users\<user>\Documents\Altova\XMLSpy2023\Examples\SampleScripts.asprj**. This scripting project consists of a few macros and a Windows form.

To load the scripting project into Scripting Editor:

1. On the **Tools** menu, click **Scripting Editor**.
2. Click **Open** and browse for the **SampleScripts.asprj** file from the path above.

The project contains several macros in the "Macros" directory.

Macro	Description
AddMacroMenu	This macro adds a new menu item to XMLSpy, by invoking the `Application.AddMacroMenuItem` method of the COM API. The first argument of the `AddMacroMenuItem` method is the name of the macro to be added (in this example, "CloseAllButActiveDoc") and the second argument is the display text for the menu item. Whenever this macro is run, a new menu command called "CloseAllButActiveDoc") is added under the **Tools** menu. To clear macro menu items created previously, either restart XMLSpy or create a macro that calls the `Application.ClearMacroMenu` API method.
CloseAllButActiveDocument	When executed, the macro iterates though the currently open documents in XMLSpy and closes all of them, except for the active document.

SearchPath	This macro displays a form that lets users perform search for files within the current project. The form is available in the "Forms" directory, where you can view its design and the associated event handlers. The `GetAllPathsFromProject()` method returns all the file paths that belong to the currently opened project, as an array. The definition of this method is in the **GlobalDeclarations** script of the project. The `InsertStringInArrayUnique` method ensures that only unique paths are added to the array. Next, the form is initialized with CreateForm. Finally, the array is converted to a .NET type with the help of the CLR.Create method and the form is populated with the resulting `ArrayList` collection. The **Open** button of the form has a handler that calls the `Application.Documents.OpenFile` API method to open the currently selected file.

To enable the scripting project as global XMLSpy scripting project:

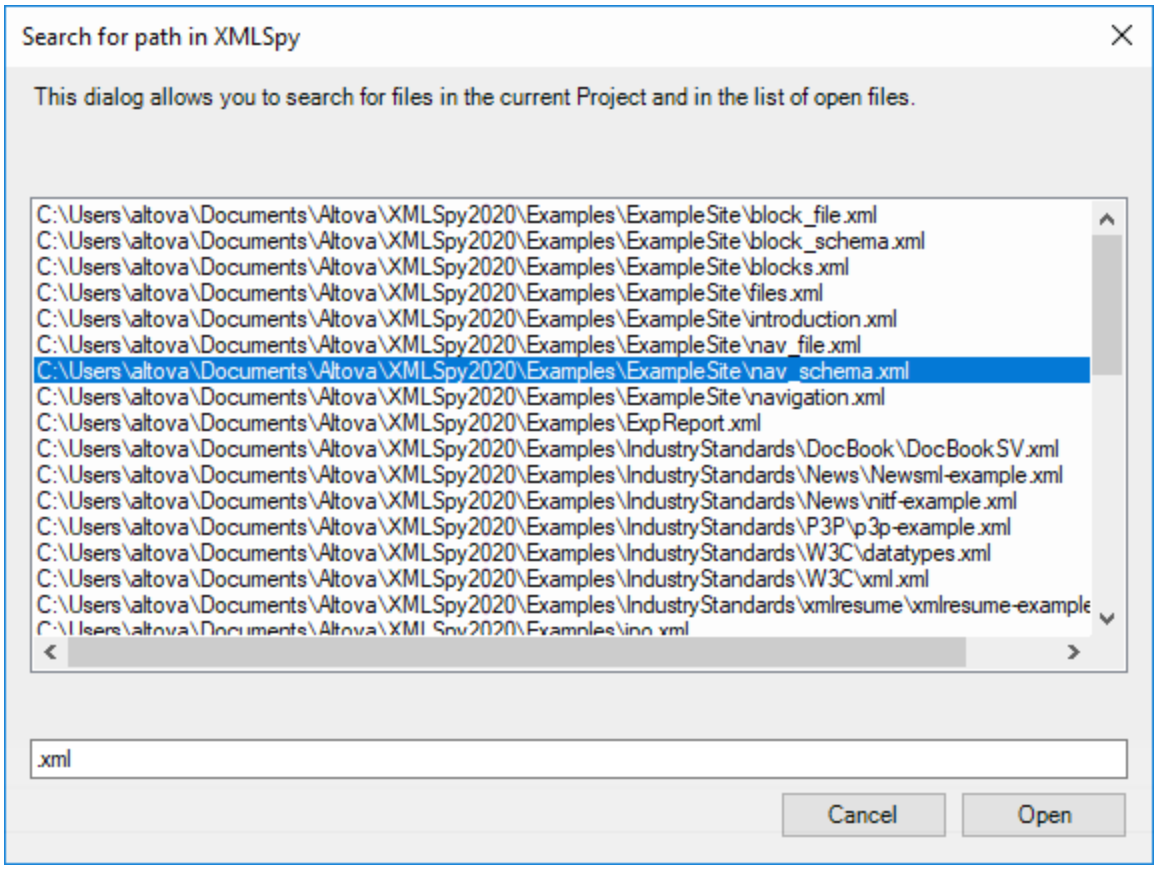
1. On the **Tools** menu, click **Options**.
2. Click the **Scripting** tab.
3. Under "Global scripting project file", click **Browse** and select the **SampleScripts.asprj** file from the path above.
4. This scripting project does not have auto-macros and application event handlers; therefore, you don't need to select either the **Run auto-macros...** or **Process events** check boxes.
5. Click **Apply**.

At this stage, several new menu items (one for each macro) become available under the **Tools | Macros** menu.

To run the "SearchPath" macro:

1. Open an XMLSpy project that contains several files (in this example, **C:\Users\<user>\Documents\Altova\XMLSpy2023\Examples\Examples.spp**).
2. On the **Tools** menu, click **Macros**, and then click **Search Path**.
3. Type the search term (in this example, ".xml").

© 2016-2022 Altova GmbH



As shown above, all file names that contain the search term are now listed. You can click any element in the list, and then click **Open** to display it in the main editor.

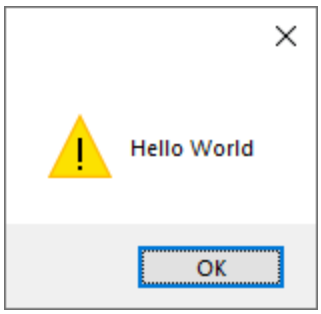
29.1.2 Built-in Commands

This section provides reference to all the commands you can use in the XMLSpy Scripting Editor.

- alert
- confirm
- CLR.Create
- CLR.Import
- CLR.LoadAssembly
- CLR.ShowImports
- CLR.ShowLoadedAssemblies
- CLR.Static
- CreateForm
- doevents
- lastform
- prompt
- ShowForm
- watchdog

29.1.2.1 alert

Displays a message box that shows a given message and the "OK" button. To proceed, the user will have to click "OK".



Signature

For JScript, the signature is:

```
alert(strMessage : String) -> void
```

For VBScript, the signature is:

```
MsgBox(strMessage : String) -> void
```

Example

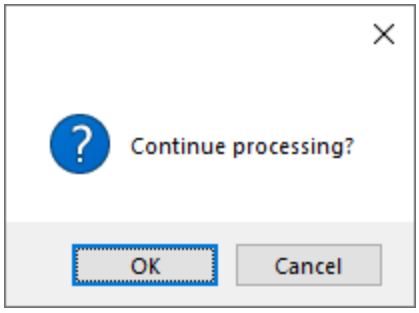
The following JScript code displays a message box with the text "Hello World".

```
alert("Hello World");
```

29.1.2.2 confirm

Opens a dialog box that shows a given message, a confirmation button, and a cancel button. The user will have to click either "OK" or "Cancel" to proceed. Returns a Boolean that represents the user's answer. If the user clicked "OK", the function returns **true**; if the user clicked "Cancel", the function returns **false**.

© 2016-2022 Altova GmbH



Signature

```
confirm(strMessage : String) -> result : Boolean
```

Example (JScript)

```
if ( confirm( "Continue processing?" ) == false )
  alert("You have cancelled this action");
```

Example (VBScript)

```
If ( confirm( "Continue processing?" ) = false ) Then
  MsgBox("You have cancelled this action")
End If
```

29.1.2.3 CLR.Create

Creates a new .NET object instance of the type name supplied as argument. If more than one argument is passed, the successive arguments are interpreted as the arguments for the constructor of the .NET object. The return value is a reference to the created .NET object

Signature

```
CLR.Create(strTypeNameCLR : String, constructor arguments ... ) -> object
```

Example

The following JScript code illustrates how to create instances of various .NET classes.

```
// Create an ArrayList
var objArray = CLR.Create("System.Collections.ArrayList");
// Create a ListViewItem
var newItem = CLR.Create( "System.Windows.Forms.ListViewItem", "NewItemText" );
// Create a List<string>
```

```
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
// Import required namespaces and create a Dictionary object
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary< String, Dictionary< String, String > >" );
```

29.1.2.4 CLR.Import

Imports a namespace. This is the scripting equivalent of C# `using` and VB.Net `imports` keyword. Calling `CLR.Import` makes it possible to leave out the namespace part in subsequent calls like `CLR.Create()` and `CLR.Static()`.

Note: Importing a namespace does not add or load the corresponding assembly to the scripting project. You can add assemblies to the scripting project dynamically (at runtime) in the source code by calling CLR.LoadAssembly.

Signature

```
CLR.Import(strNamespaceCLR : String) -> void
```

Example

Instead of having to use fully qualified namespaces like:

```
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "System.Windows.Forms.MessageBox" ).Show( "Hello " + sName );
}
```

One can import namespaces first and subsequently use the short form:

```
CLR.Import( "System.Windows.Forms" );

if ( ShowForm( "FormName" ) == CLR.Static( "DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "MessageBox" ).Show( "Hello " + sName );
}
```

29.1.2.5 CLR.LoadAssembly

Loads the .NET assembly with the given long assembly name or file path. Returns Boolean **true** if the assembly could be loaded; **false** otherwise.

© 2016-2022 Altova GmbH

Signature

```
CLR.LoadAssembly(strAssemblyNameCLR : String, showLoadErrors : Boolean) -> result :
Boolean
```

Example

The following JScript code attempts to set the clipboard text by loading the required assembly dynamically.

```
// set clipboard text (if possible)
// System.Windows.Clipboard is part of the PresentationCore assembly, so load this
assembly first:
if ( CLR.LoadAssembly( "PresentationCore, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35", true ) )
{
    var clipboard = CLR.Static( "System.Windows.Clipboard" );
    if ( clipboard != null )
        clipboard.SetText( "HelloClipboard" );
}
```

29.1.2.6 CLR.ShowImports

Opens a message box that shows the currently imported namespaces. The user will have to click "OK" to proceed.

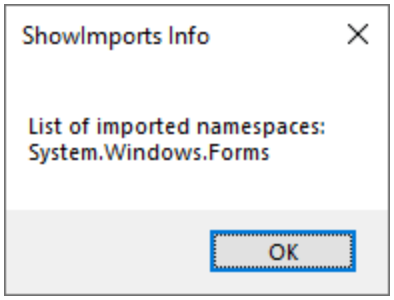
Signature

```
CLR.ShowImports() -> void
```

Example

The following JScript code first imports a namespace, and then displays the list of imported namespaces:

```
CLR.Import( "System.Windows.Forms");
CLR.ShowImports();
```



29.1.2.7 CLR.ShowLoadedAssemblies

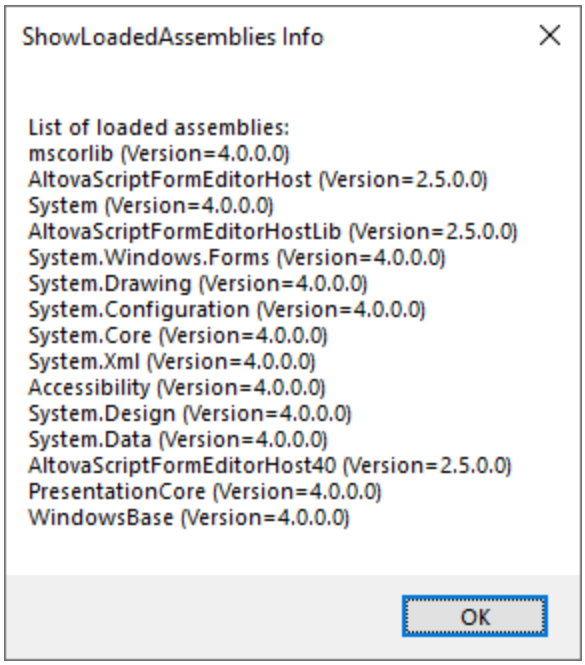
Opens a message box that shows the currently loaded assemblies. The user will have to click "OK" to proceed.

Signature

```
CLR.ShowLoadedAssemblies() -> void
```

Example

```
CLR.ShowLoadedAssemblies();
```



29.1.2.8 CLR.Static

Returns a reference to a static .NET object. You can use this function to get access to .NET types that have no instances and contain only static members.

Signature

```
CLR.Static(strTypeNameCLR : String) -> object
```


© 2016-2022 Altova GmbH

Example (JScript)

```
// Get the value of a .NET Enum into a variable
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch

// Set the value of the Windows clipboard
var clipboard = CLR.Static( "System.Windows.Clipboard" );
clipboard.SetText( "HelloClipboard" );

// Check the buttons pressed by the user on a dialog box
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

29.1.2.9 CreateForm

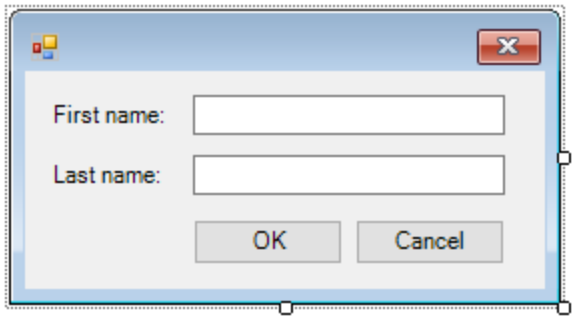
Instantiates the `Form` object identified by the name supplied as argument. The form must exist in the "Forms" folder of the scripting project. Returns the form object (`System.Windows.Forms.Form`) corresponding to the given name, or `null` if no form with such name exists.

Signature

```
CreateForm (strFormName : String) -> System.Windows.Forms.Form | null
```

Example

Let's assume that a form called "FormName" exists in the scripting project.



The following JScript code instantiates the form with some default values and displays it to the user.

```
var myForm = CreateForm( "FormName" );
if ( myForm != null )
{
   myForm.textboxFirstName.Text = "Daniela";
   myForm.textboxLastName.Text = "Heidegger";
```

```
   var dialogResult = myForm.ShowDialog();
}
```

The `dialogResult` can subsequently be evaluated as follows:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

Note: The code above will work only if the **DialogResult** property of the "OK" and "Cancel" buttons is set correctly from the Properties pane (for example, it must be **OK** for the "OK" button).

29.1.2.10 doevents

Processes all Windows messages currently in the message queue.

Signature

```
doevents() -> void
```

Example (JScript)

```
for ( i=0; i < nLongLastingProcess; ++i )
{
    // do long lasting process

    doevents(); // process Windows messages; give UI a chance to update
}
```

29.1.2.11 lastform

This is a global field that returns a reference to the last form object that was created via `CreateForm()` or `ShowForm()`.

Signature

```
lastform -> formObj : System.Windows.Forms.Form
```

Example

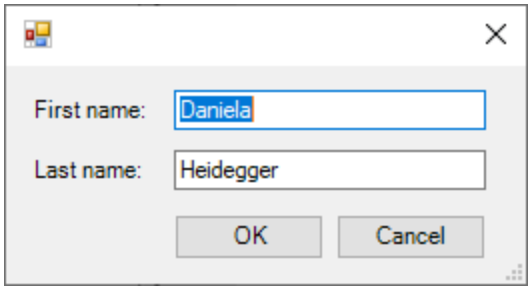
The following JScript code shows the form "FormName" as a dialog box.

```
CreateForm( "FormName" );
if ( lastform != null )
```


© 2016-2022 Altova GmbH

```
{
    lastform.textboxFirstName.Text = "Daniela";
    lastform.textboxLastName.Text = "Heidegger";
    var dialogResult = lastform.ShowDialog();
}
```

The values of both textbox controls are initialized with the help of `lastform`.



29.1.2.12 prompt

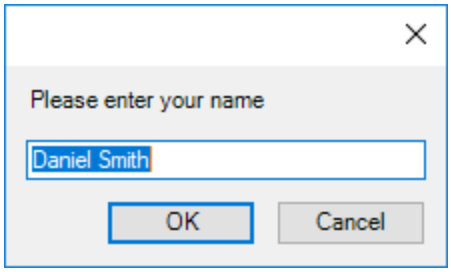
Opens a dialog box that shows a message and a textbox control with a default answer. This can be used to let the user input a simple string value. The return value is a string that contains the textbox value or null if the user selected "Cancel".

Signature

```
prompt(strMessage : String, strDefault : String) -> val : String
```

Example

```
var name = prompt( "Please enter your name", "Daniel Smith" );
if ( name != null )
    alert( "Hello " + name + "!" );
```



29.1.2.13 ShowForm

Instantiates a new form object from the given form name and immediately shows it as dialog box. The return value is an integer that represents the generated `DialogResult` (`System.Windows.Forms.DialogResult`). For the list of possible values, refer to the documentation of the `DialogResult` Enum (https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.dialogresult?view=netframework-4.8).

Signature

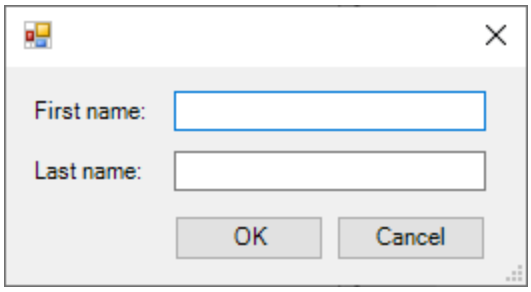
```
ShowForm(strFormName : String) -> result : Integer
```

Example

The following JScript code

```
var dialogResult = ShowForm( "FormName" );
```

Shows the form "FormName" as a dialog box:



The `DialogResult` can subsequently be evaluated, for example:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

Note: The code above will work only if the **DialogResult** property of the "OK" and "Cancel" buttons is set correctly from the Properties pane (for example, it must be **OK** for the "OK" button).

29.1.2.14 watchdog

Long running CPU-intensive scripts may ask the user if the script should be terminated. The `watchdog()` method is used to disable or enable this behavior. By default, the watchdog is enabled.

Calling `watchdog(true)` can also be used to reset the watchdog. This can be useful before executing long running CPU-intensive tasks to ensure they have the maximum allowed script processing quota.

© 2016-2022 Altova GmbH

Signature

```
watchdog(bEnable : boolean) -> void
```

Example

```
watchdog( false );  // disable watchdog - we know the next statement is CPU intensive but
it will terminate for sure
doCPUIntensiveScript();
watchdog( true );   // re-enable watchdog
```

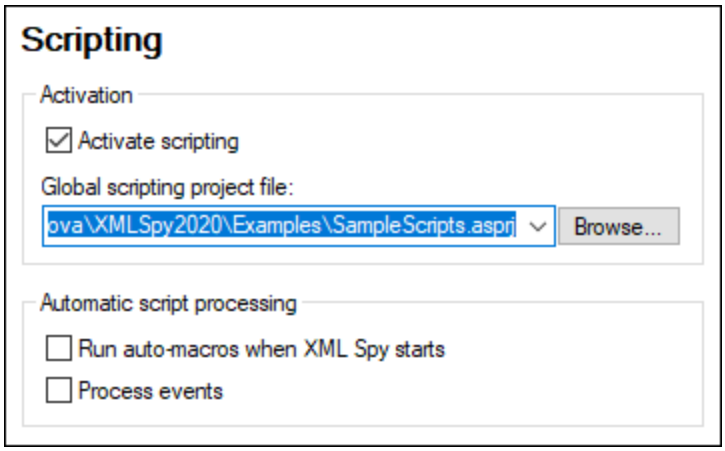
29.1.3 Enabling Scripts and Macros

Once a scripting project is complete and tested, you can use it in the following ways:

1. As the global scripting project for XMLSpy. This means that all the scripts and macros from the scripting project are available to XMLSpy.
2. At project level. This means that a reference to the .asprj file is saved together with the XMLSpy project. When the XMLSpy project is opened, its associated scripts and macros can be called.

To set a scripting project as global:

1. On the **Tools** menu, click **Options**.
2. Click the **Scripting** tab.
3. Select the **Activate scripting** check box and browse for the .asprj file to be used as global scripting project.



You can optionally enable the following additional script processing options:

Run auto-macros when XMLSpy starts	If you select this check box, any macros that were set as "Auto-macro" in the project will be triggerred automatically when XMLSpy starts.

Process events	Select this check box if your scripts bind to any application events. Clear the check box to prevent the scripts from reacting to events.

To enable a scripting project at project level:

1. Open the project.
2. On the **Project** menu, click **Script Settings**.
3. Select the **Activate project scripts** check box and browse for the .asprj file.

The **Run-auto macros...** check box has the same meaning as already described above.

29.1.3.1 Running Macros

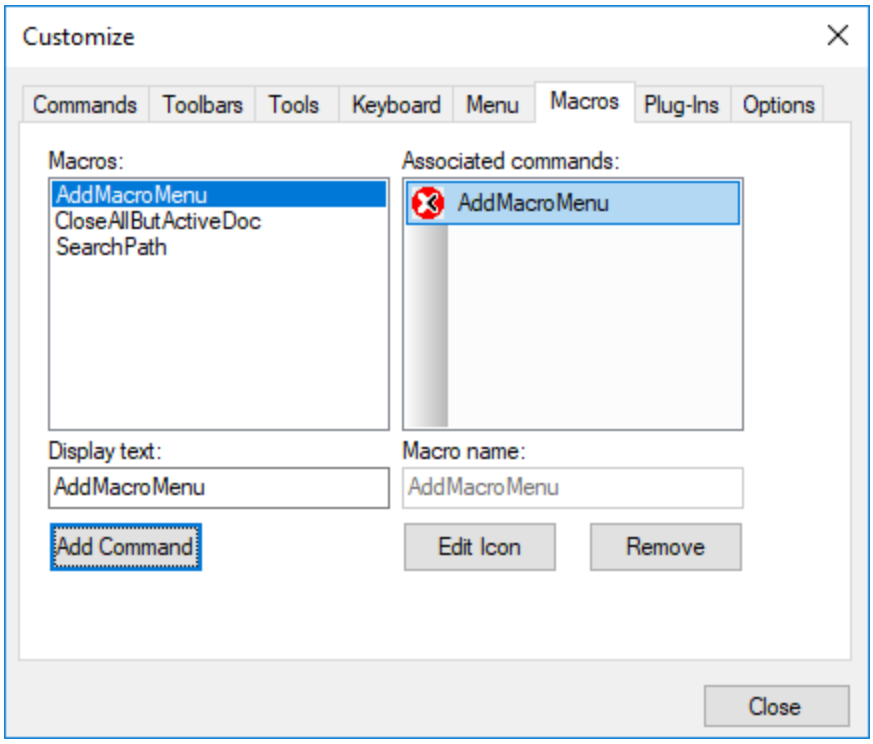
When a scripting project is active in XMLSpy, any macros available in that project are displayed in the **Tools | Macros** menu. Therefore, you can run a macro at any time, by triggering the respective menu command, for example **Tools | Macros | <SomeMacro>**.

Macros that were configured as auto-macros will run automatically whenever XMLSpy starts, provided that this behavior is enabled from options, as described in Enabling Scripts and Macros.

For convenience, you can create toolbar buttons for macros, as follows:

1. On the **Tools** menu, click **Customize**.
2. Click the **Macros** tab. Any macros that are available at application level (in the *global* scripting project) are listed.
3. Click **Add Command**.

© 2016-2022 Altova GmbH



4. Optionally, click **Edit icon** and draw a new icon for the new macro. You can also assign a shortcut to the macro, from the **Keyboard** tab.
5. Drag the macro from the **Associated commands** pane onto the toolbar where you would like it to appear.

To remove a macro from a toolbar:

1. On the **Tools** menu, click **Customize**.
2. Click the **Macros** tab.
3. Drag the macro from the toolbar where it appears back into the **Associated commands** pane.

29.2 IDE Plugins

XMLSpy allows you to create your own IDE plug-ins and integrate them into XMLSpy.

Use plug-ins to:

- Configure your version of XMLSpy, add commands through menus, icons, buttons etc.
- React to events from XMLSpy.
- Run your specific code within XMLSpy with access to the complete XMLSpy API

XMLSpy expects your plug-in to implement the IXMLSpyPlugIn interface. C# and C++ examples are included with your installation package and are located in the `XMLSpy2023\Examples\IDEPlugin` folder of your XMLSpy installation.

Windows 7, 8, 10, 11	`C:/Users/<username>/Documents`

See ATL sample files for an example using C++.

29.2.1 Registration of IDE PlugIns

XMLSpy maintains a specific key in the Registry where it stores all registered IDE plug-ins:

HKEY_CURRENT_USER\Software\Altova\XML Spy\PlugIns

All values of this key are treated as references to registered plug-ins and must conform to the following format:

Value name:	ProgID of the plug-in
Value type:	must be REG_SZ
Value data:	CLSID of the component

Each time the application starts the values of the "PlugIns" key is scanned, and the registered plug-ins are loaded.

Register plug-in manually

To register a plug-in manually, use the "Customize" dialog box of the XMLSpy "Tools" menu. Use the "Add Plug-In..." button to specify the DLL that implements your plug-in. XMLSpy registers the DLL as a COM server and adds the corresponding entry in its "PlugIns" key.

If you experience problems with manual registration you can check if the CLSID of your plug-in is correctly registered in the "PlugIns" key. If this is not the case, the name of your plug-in DLL was probably not sufficiently unique. Use a different name or perform direct registration.

Register plug-in directly

A plug-in can be directly registered as an IDE plug-in by first registering the DLL and then adding the appropriate value to the "PlugIns" key of XMLSpy during plug-in setup for example. The new plug-in will be activated the next time XMLSpy is launched.

© 2016-2022 Altova GmbH

Creating plug-ins

Source code for sample plug-ins has been provided in the application's (My) Documents folder: `Examples\IDEPlugin` folder. To build a plug-in from such source code, do the following:

1. Open the solution you want to build as a plug-in in Visual Studio.
2. Build the plug-in with the command in the Build menu.
3. The plug-in's DLL file will be created in the `Bin` or `Debug` folder. This DLL file is the file that must be added as a plug-in (*see above*).

29.2.2 ActiveX Controls

ActiveX controls are supported. Any IDE PlugIn which is also an ActiveX control will be displayed in a Dialog Control Bar. A sample PlugIn that is also an ActiveX control is included in the `IDEPlugin` folder in the `Examples` folder of your application folder.

29.2.3 Configuration XML

The IDE plug-in allows you to change the user interface (UI) of XMLSpy. This is done by describing each separate modification using an XML data stream. The XML configuration is passed to XMLSpy using the GetUIModifications method of the IXMLSpyPlugIn interface.

The XML file containing the UI modifications for the IDE PlugIn, must have the following structure:

```
<ConfigurationData>
	<ImageFile>path To image file</ImageFile>
	<Modifications>
		<Modification>
			...
		</Modification>
		...
	</Modifications>
</ConfigurationData>
```

You can define icons or toolbar buttons for the new menu items which are added to the UI of XMLSpy by the plug-in. The path to the file containing the images is set using the ImageFile element. Each image must be 16 x 16 pixels using max. 256 colors. The image references must be arranged from left to right in a single (<ImageFile>...) line. The rightmost image index value is zero.

The Modifications element can have any number of Modification child elements. Each Modification element defines a specific change to the standard UI of XMLSpy. Starting with version 4.3, it is also possible to remove UI elements from XMLSpy.

Structure of Modification elements

All Modification elements consist of the following two child elements:

```
<Modification>
	<Action>Type of action</Action>
```

```
        <UIElement Type="type of UI element">
        </UIElement>
</Modification>
```

Valid values for the Action element are:

Add - to add the following UI element to XMLSpy
Hide - to hide the following UI element in XMLSpy
Remove - to remove the UI element from the "Commands" list box, in the customize dialog

You can combine values of the Action element e.g. "Hide Remove"

The UIElement element describes any new or existing UI element for XMLSpy. Possible elements are currently: new toolbars, buttons, menus or menu items. The **type** attribute defines which UI element is described by the XML element.

Common UIElement children
The ID and Name elements are valid for all different types of XML UIElement fragments. It is, however, possible to ignore one of the values for a specific type of UIElement. For example, Name is ignored for a separator.

```
<ID></ID>
<Name></Name>
```

If UIElement describes an existing element of the UI. The value of the ID element is predefined by XMLSpy. Normally these ID values are not known to the public. If the XML fragment describes a new part of the UI, then the ID is arbitrary and the value should be less than 1000. The Name element sets the textual value. Existing UI elements can be identified just by name, for example, menus and menu items with associated sub menus. For new UI elements, the Name element sets the caption (for example, the title of a toolbar) or text for a menu item.

Toolbars and Menus
To define a toolbar it is necessary to specify the ID and/or the name of the toolbar. An existing toolbar can be specified using only the name or ID (if the latter is known). To create a **new** toolbar both values must be set. The **type** attribute must be equal to "ToolBar".

```
<UIElement Type="ToolBar">
        <ID>1</ID>
        <Name>TestPlugIn</Name>
</UIElement>
```

To specify an XMLSpy menu you need two parameters:

- The ID of the menu bar which contains the menu. If no XML documents are open in the main window, the menu bar ID is 128. If one or more XML documents are open, the menu bar ID is 129.
- The menu name. Menus do not have an associated ID value. The following example defines the "Edit" menu of the menu bar which is active, when at least one XML document is open:

```
<UIElement Type="Menu">
        <ID>129</ID>
        <Name>Edit</Name>
</UIElement>
```

An additional element is used if you want to create a new menu. The Place element defines the position of the new menu in the menu bar:

© 2016-2022 Altova GmbH

```
<UIElement Type="Menu">
	<ID>129</ID>
	<Name>PlugIn Menu</Name>
	<Place>12</Place>
</UIElement>
```

A value of -1 for the Place element sets the new button or menu item at the end of the menu or toolbar.

Commands
If you add a new command (through a toolbar button or a menu item), the UIElement fragment can contain any of these sub elements:

```
<MacroName></MacroName>
<Info></Info>
<ImageID></ImageID>
```

If MacroName is specified, XMLSpy searches for a macro with the same name in the scripting environment and executes it each time this command is processed. The Info element contains a short description string which is displayed in the status bar when the mouse pointer is over the associated command (button or menu item). ImageID defines the index of the icon in the image file. Please note that all icons are stored in one image file.

To define a toolbar button create an UIElement with this structure:

```
<UIElement Type="ToolBarItem">
	<!--don't reuse local IDs even the commands do the same-->
	<ID>5</ID>
	<Name>Open file from repository...</Name>
	<!--Set Place To -1 If this is the first button To be inserted-->
	<Place>-1</Place>
	<ImageID>0</ImageID>
	<ToolBarID>1</ToolBarID>
	<!--instead of the toolbar ID the toolbar name could be used-->
	<ToolBarName>TestPlugIn</ToolBarName>
</UIElement>
```

Additional elements to declare a toolbar button are Place, ToolBarID and ToolBarName. ToolBarID and ToolBarName are used to identify the toolbar which contains the new or existing button. The textual value of ToolBarName is case-sensitive. The (UIElement) **type** attribute must be "ToolBarItem".

To define a menu item, the elements MenuID, Place and Parent are available in addition to the standard elements used to declare a command. MenuID can be either 128 or 129. Please see "Toolbars and Menus" for more information on these values.

The Parent element is used to identify the **menu** where the new menu entry should be inserted. As sub menu items have no unique Windows ID, we need some other way to identify the parent of the menu item.

The value of the Parent element is a path to the menu item. The text value of the Parent element, must equal the **parent menu name** of the submenu, where the submenu name is separated by a colon. If the menu has no parent, because it is not a submenu, add a colon to the beginning of the name. The **type** attribute must be set to "MenuItem". Example for an UIElement defining a menu item:

```
<UIElement Type="MenuItem">
	<!--the following element is a Local command ID-->
```

```
        <ID>3</ID>
        <Name>Open file from repository...</Name>
        <Place>-1</Place>
        <MenuID>129</MenuID>
        <Parent>:PlugIn Menu</Parent>
        <ImageID>0</ImageID>
    </UIElement>
```

XMLSpy makes it possible to add toolbar separators and menus if the value of the ID element is set to 0.

29.2.4 ATL sample files

The following pages show how to create a simple XMLSpy IDE plug-in DLL using ATL. To build the DLL it is necessary to know about ATL, the wizards that generate new ATL objects, as well as MS VisualStudio.

To access the API the implementation imports the Type Library of XMLSpy. The code reads various properties and calls methods using the smart pointers provided by the #import statement.

In addition, the sample code uses the MFC class CString and the ATL conversion macros such as W2T.

At a glance the steps to create an ATL DLL are as follows:

1. Open VisualStudio and select "New..." from the "File" menu.
2. Select the "styleclass="Code Bold"" tab.
3. Select "ATL COM AppWizard" and type in a project name.
4. Select "Support for MFC" if you want to use MFC classes, or if you want to create a project for the sample code.

   Having created the project files you can add an ATL object to implement the IXMLSpyPlugIn interface:

1. Select "New ATL Object..." from the "Insert" menu.
2. Select "Simple Object" from the wizard and click "Next".
3. Type in a name for the object.
4. On the "Attributes" tab, select "Custom" for the type of interface, and disable Aggregation.

These steps produce the skeleton code for the implementation of the IDE plug-in interface. Please see the following pages on how to modify the code and achieve some basic functionality.

29.2.4.1 Interface description (IDL)

The IDL of the newly created ATL object contains a declaration for one COM interface.

- This interface declaration must be replaced by the declaration of IXMLSpyPlugIn as shown below.
- The IDL must also contain the definition of the SPYUpdateAction enumeration.
- Replace the generated default interface name, (created by the wizard) with "IXMLSpyPlugIn" in the coclass declaration. The IDL should then look something like the example code below:

Having created the ATL object, you then need to implement the IDE plug-in interface of XMLSpy.

```
import "oaidl.idl";
```


© 2016-2022 Altova GmbH

```
import "ocidl.idl";

// ----- please insert the following block into your IDL file -----
 typedef enum {
  spyEnable = 1,
  spyDisable = 2,
  spyCheck = 4,
  spyUncheck = 8
 } SPYUpdateAction;

// ----- end insert block ----


// ----- E.g. Interface entry automatically generated by the ATL wizard -----
// [
//   object,
//   uuid(AB7CD86A-8145-429A-A1F3-270692EO8AFC),

//   helpstring("IXMLSpyPlugIn Interface")
//   pointer_default(unique)
// ]
// interface IXMLSpyPlugIn : IUnknown
// {
// };

// ----- end automatically generated Interface Entry


// ----- replace the Interface Entry (shown above) generated for you by the ATL wizard,
with the following block -----

 [
  odl,
  uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85),
  helpstring("IXMLSpyPlugIn Interface")
 ]
 interface IXMLSpyPlugIn : IUnknown
  {
   HRESULT _stdcall OnCommand([in] long nID, [in] IDispatch* pXMLSpy);

   HRESULT _stdcall OnUpdateCommand([in] long nID, [in] IDispatch* pXMLSpy, [out, retval]
SPYUpdateAction* pAction);

   HRESULT _stdcall OnEvent([in] long nEventID, [in] SAFEARRAY(VARIANT)* arrayParameters,
[in] IDispatch* pXMLSpy, [out, retval] VARIANT* pReturnValue);

   HRESULT _stdcall GetUIModifications([out, retval] BSTR* pModificationsXML);

   HRESULT _stdcall GetDescription([out, retval] BSTR* pDescription);
  };

// ----- end replace block -----
```

```
// ----- The code below is automatically generated by the ATL wizard and will look slightly
different in your case -----

 [
 uuid(24FE0D1B-3FC0-494E-B36E-1D4CE412B014),
 version(1.0),
 helpstring("XMLSpyIDEPlugInDLL 1.0 Type Library")
 ]
 library XMLSPYIDEPLUGINDLLLib
 {
 importlib("stdole32.tlb");
 importlib("stdole2.tlb");

 [
  uuid(3800E791-7F6B-4ACD-9E32-2AC184444501),
  helpstring("XMLSpyIDEPlugIn Class")
 ]
 coclass XMLSpyIDEPlugIn
 {
  [default] interface IXMLSpyPlugIn;   // ----- define IXMLSpyPlugIn as the default
interface -----
 };
};
```

29.2.4.2 Class definition

In the class definition of the ATL object, several changes must be made. The class has to derive from IXMLSpyPlugIn, the "Interface Map" needs an entry for IXMLSpyPlugIn, and the methods of the IDE plug-in interface must be declared:

```
#ifndef __XMLSPYIDEPLUGIN_H_
#define __XMLSPYIDEPLUGIN_H_

#include "resource.h"       // main symbols

/////////////////////////////////////////////////////////////////////////////
// CXMLSpyIDEPlugIn
class ATL_NO_VTABLE CXMLSpyIDEPlugIn :
 public CComObjectRootEx<CComSingleThreadModel>,
 public CComCoClass<CXMLSpyIDEPlugIn, &CLSID_XMLSpyIDEPlugIn>,
 public IXMLSpyPlugIn
{
public:
 CXMLSpyIDEPlugIn()
 {
 }

DECLARE_REGISTRY_RESOURCEID(IDR_XMLSPYIDEPLUGIN)
DECLARE_NOT_AGGREGATABLE(CXMLSpyIDEPlugIn)

DECLARE_PROTECT_FINAL_CONSTRUCT()
```


© 2016-2022 Altova GmbH

```
BEGIN_COM_MAP(CXMLSpyIDEPlugIn)
 COM_INTERFACE_ENTRY(IXMLSpyPlugIn)
END_COM_MAP()

// IXMLSpyIDEPlugIn
public:
 virtual HRESULT _stdcall OnCommand(long nID, IDispatch* pXMLSpy);

 virtual HRESULT _stdcall OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction);

 virtual HRESULT _stdcall OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue);

 virtual HRESULT _stdcall GetUIModifications(BSTR* pModificationsXML);

 virtual HRESULT _stdcall GetDescription(BSTR* pDescription);
};

#endif //__XMLSPYIDEPLUGIN_H_
```

29.2.4.3 Implementation

The code below shows a simple implementation of an XMLSpy IDE plug-in. It adds a menu item and a separator (available with XMLSpy) to the Tools menu. Inside the OnUpdateCommand() method, the new command is only enabled when the active document is displayed using the Grid View. The command searches for the XML element which has the current focus, and opens any URL starting with "http://", from the textual value of the element.

```
////////////////////////////////////////////////////////////////////////////
// CXMLSpyIDEPlugIn

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

HRESULT CXMLSpyIDEPlugIn::OnCommand(long nID, IDispatch* pXMLSpy)
{
 USES_CONVERSION;

 if(nID == 1) {
  IApplicationPtripSpyApp;

  if(pXMLSpy) {
   if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication),(void **)&ipSpyApp))) {
    IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

    // we assume that grid view is active
    if(ipDocPtr) {
     IGridViewPtr ipGridPtr = ipDocPtr->GridView;
```

```
     if(ipGridPtr){
      IXMLDataPtr  ipXMLData = ipGridPtr->CurrentFocus;

      CString strValue = W2T(ipXMLData->TextValue);

      if(!strValue.IsEmpty() && (strValue.Left(7) == _T("http://")))
       ::ShellExecute(NULL,_T("open"),W2T(ipXMLData->TextValue),NULL,NULL,SW_SHOWNORMAL);
     }
    }
   }
  }
 }

 return S_OK;
}


HRESULT CXMLSpyIDEPlugIn::OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction)
{
 *pAction = spyDisable;

 if(nID == 1) {
  IApplicationPtripSpyApp;

  if(pXMLSpy) {
   if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication),(void **)&ipSpyApp)))     {
    IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

    // only enable if grid view is active
    if((ipDocPtr != NULL) && (ipDocPtr->CurrentViewMode == spyViewGrid))
     *pAction = spyEnable;
   }
  }
 }

 return S_OK;
}


HRESULT CXMLSpyIDEPlugIn::OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue)
{
 return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::GetUIModifications(BSTR* pModificationsXML)
{
 CComBSTR bstrMods = _T(" \
    <ConfigurationData>\
     <Modifications>");
 // add "Open URL..." to Tools menu
 bstrMods.Append (_T(" \
```


© 2016-2022 Altova GmbH

```
      <Modification> \
       <Action>Add</Action> \
       <UIElement type=\"MenuItem\"> \
        <ID>1</ID> \
        <Name>Open URL...</Name> \
        <Place>0</Place> \
        <MenuID>129</MenuID> \
        <Parent>:Tools</Parent> \
       </UIElement> \
      </Modification>"));
 // add Seperator to Tools menu
 bstrMods.Append (_T(" \
      <Modification> \
       <Action>Add</Action> \
       <UIElement type=\"MenuItem\"> \
        <ID>0</ID> \
        <Place>1</Place> \
        <MenuID>129</MenuID> \
        <Parent>:Tools</Parent> \
       </UIElement> \
      </Modification>"));
 // finish modification description
 bstrMods.Append (_T(" \
     </Modifications> \
    </ConfigurationData>"));

 return bstrMods.CopyTo(pModificationsXML);
}


HRESULT CXMLSpyIDEPlugIn::GetDescription(BSTR* pDescription)
{
 CComBSTR bstrDescr = _T("ATL C++ XMLSpy IDE PlugIn;This PlugIn demonstrates the
implementation of a simple ATL DLL as a IDE PlugIn for XMLSpy.");
 return bstrDescr.CopyTo(pDescription);
}
```

29.2.5 IXMLSpyPlugIn

See also

Methods
OnCommand
OnUpdateCommand
OnEvent
GetUIModifications
GetDescription

Description

If a DLL is added to XMLSpy as an IDE plug-in, it is necessary that it registers a COM component that answers to an IXMLSpyPlugIn interface with the reserved uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85), for it to be recognized as a plug-in.

29.2.5.1 OnCommand

See also

Declaration: `OnCommand(nID as long, pXMLSpy as IDispatch)`

Description

The `OnCommand()` method of the interface implementation, is called each time a command added by the the IDE plug-in (menu item or toolbar button) is processed. nID stores the command ID defined by the `ID` element of the respective `UIElement`.

`pXMLSpy` holds a reference to the dispatch interface of the `Application` object of XMLSpy.

Example

```
Public Sub IXMLSpyPlugIn_OnCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
    If (Not (pXMLSpy Is Nothing)) Then
        Dim objDlg
        Dim objDoc As XMLSpyLib.Document
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "XML Files (*.xml)|*.xml|All Files (*.*)|*.*||"
            objDlg.FilterIndex = 1
            objDlg.ShowOpen

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.Documents.OpenFile(objDlg.FileName, False)
                Set objDoc = Nothing
            End If
        End If

        If nID = 4 Or nID = 6 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "All Files (*.*)|*.*||"
            objDlg.Flags = cdlOFNPathMustExist
            objDlg.ShowSave

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.ActiveDocument

                If Not (objDoc Is Nothing) Then
                    objDoc.SetPathName objDlg.FileName
                    objDoc.Save
                    Set objDoc = Nothing
                End If
            End If
        End If
```


© 2016-2022 Altova GmbH

```
        Set objSpy = Nothing
    End If
End Sub
```

29.2.5.2 OnUpdateCommand

See also

Declaration: `OnUpdateCommand(nID as long, pXMLSpy as IDispatch) as SPYUpdateAction`

Description

The `OnUpdateCommand()` method is called each time the visible state of a button or menu item needs to be set. `nID` stores the command ID defined by the `ID` element of the respective `UIElement`.

`pXMLSpy` holds a reference to the dispatch interface of the `Application` object.

Possible return values to set the update state are:

```
spyEnable         = 1
spyDisable        = 2
spyCheck          = 4
spyUncheck        = 8
```

Example

```
Public Function IXMLSpyPlugIn_OnUpdateCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
As SPYUpdateAction
    IXMLSpyPlugIn_OnUpdateCommand = spyDisable

    If (Not (pXMLSpy Is Nothing)) Then
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
        End If
        If nID = 4 Or nID = 6 Then
            If objSpy.Documents.Count > 0 Then
                IXMLSpyPlugIn_OnUpdateCommand = spyEnable
            Else
                IXMLSpyPlugIn_OnUpdateCommand = spyDisable
            End If
        End If
    End If
End Function
```

29.2.5.3 OnEvent

See also

Declaration: `OnEvent(nEventID as long, arrayParameters as SAFEARRAY(VARIANT), pXMLSpy as IDispatch) as VARIANT`

Description

`OnEvent()` is called each time an event is raised from XMLSpy.

Possible values for `nEventID` are:

On_BeforeStartEditing	= 1
On_EditingFinished	= 2
On_FocusChanged	= 3
On_Beforedrag	= 4
On_BeforeDrop	= 5
On_OpenProject	= 6
On_OpenDocument	= 7
On_CloseDocument	= 8
On_SaveDocument	= 9

Events available since XMLSpy 4r4:

On_DocEditDragOver	= 10
On_DocEditDrop	= 11
On_DocEditKeyDown	= 12
On_DocEditKeyUp	= 13
On_DocEditKeyPressed	= 14
On_DocEditMouseMove	= 15
On_DocEditButtonUp	= 16
On_DocEditButtonDown	= 17
On_DocEditContextMenu	= 18
On_DocEditPaste	= 19
On_DocEditCut	= 20
On_DocEditCopy	= 21
On_DocEditClear	= 22
On_DocEditSelectionChanged	= 23

Events available since XMLSpy 2004:

On_DocEditDragOver	= 10

© 2016-2022 Altova GmbH

Events available since XMLSpy 2004r4 (type library version 1.4):

On_BeforeOpenProject	= 25
On_BeforeOpenDocument	= 26
On_BeforeSaveDocument	= 27
On_BeforeCloseDocument	= 28
On_ViewActivation	= 29
On_DocEditKeyboardEvent	= 30
On_DocEditMouseEvent	= 31

Events available since XMLSpy 2006 SP1 (type library version 1.5):

On_BeforeValidate	= 32

Events available since XMLSpy 2007 (type library version 1.6):

On_BeforeShowSuggestions	= 33
On_ProjectOpened	= 34
On_Char	= 35

Events available since XMLSpy 2009 (type library version 2.2):

On_Initialize	= 36
On_Running	= 37
On_Shutdown	= 38

Events available since XMLSpy 2012 (type library version 2.8):

On_AuthenticBeforeSave	= 39
On_AuthenticContextMenuActivated	= 40
On_AuthenticLoad	= 41
On_AuthenticToolbarButtonClicked	= 42
On_AuthenticToolbarButtonExecuted	= 43
On_AuthenticUserAddedXMLNode	= 44

The names of the events are the same as they appear in the Scripting Environment of XMLSpy. For IDE plug-ins the names used are immaterial. The events are identified using the `ID` value.

`arrayParameters` is an array which is filled with the parameters of the currently raised event. Order, type and meaning of the single parameters are available through the scripting environment of XMLSpy. The events module of a scripting project, contains predefined functions for all events prior to version 4.4. The parameters passed to the predefined functions are identical to the array elements of the `arrayParameters` parameter.

Events raised from the Authentic View of XMLSpy do not pass any parameters directly. An "event" object is used instead. The event object can be accessed through the Document object of the active document.

`pXMLSpy` holds a reference to the dispatch interface of the `Application` object of XMLSpy.

If the return value of `OnEvent()` is set, then neither the IDE plug-in, nor an event handler inside of the scripting environment will get this event afterwards. Please note that all IDE plug-ins get/process the event before the Scripting Environment does.

29.2.5.4 GetUIModifications

See also

Declaration: `GetUIModifications()` as String

Description

The `GetUIModifications()` method is called during initialization of the plug-in, to get the configuration XML data that defines the changes to the UI of XMLSpy. The method is called when the plug-in is loaded for the first time, and at every start of XMLSpy.

See also Configuration XML for a detailed description how to change the UI.

Example

```
Public Function IXMLSpyPlugIn_GetUIModifications() As String
    ' GetUIModifications() gets the XML file with the specified modifications of
    ' the UI from the config.xml file in the plug-in folder
    Dim strPath As String
    strPath = App.Path

    If Len(strPath) > 0 Then
        Dim fso As New FileSystemObject
        Dim file As file

        Set file = fso.GetFile(strPath & "\config.xml")

        If (Not (file Is Nothing)) Then
            Dim stream As TextStream
            Set stream = file.OpenAsTextStream(ForReading)

            ' this replaces the token '**path**' from the XML file with
            ' the actual installation path of the plug-in to get the image file
            Dim strMods As String
            strMods = stream.ReadAll
            strMods = Replace(strMods, "**path**", strPath)

            IXMLSpyPlugIn_GetUIModifications = strMods
        Else
            IXMLSpyPlugIn_GetUIModifications = ""
        End If
    End If
End Function
```

29.2.5.5 GetDescription

See also

Declaration: `GetDescription()` as String

Description

© 2016-2022 Altova GmbH

`GetDescription()` is used to define the description string for the plug-in entries visible in the Customize dialog box.

Example

```
Public Function IXMLSpyPlugIn_GetDescription() As String
    IXMLSpyPlugIn_GetDescription = "Sample Plug-in for XMLSpy;This Plug-in demonstrates the
implementation of a simple VisualBasic DLL as a Plug-in for XMLSpy."
End Function
```

29.3 Application API

The COM-based API of XMLSpy (also called the Application API from now on) enables other applications to use the functionality of XMLSpy. As a result, it is possible to automate a wide range of tasks, from validating an XML file to modifying complex XML content (with the XMLData interface).

XMLSpy and its Application API follow the common specifications for automation servers set out by Microsoft. It is possible to access the methods and properties of the Application API from common development environments, such as those using C#, C++, VisualBasic, and Delphi, and with scripting languages like JScript and VBScript.

Execution environments for the Application API

The Application API can be accessed from the following execution environments:

- External programs (described below and in the Overview part of this section)
- From within the built-in Scripting Editor of XMLSpy. For a description of the scripting environment, see the section, Scripting Editor.
- XMLSpy allows you to create and integrate your own plug-ins into the application using a special interface for plug-ins. A description of how to create plug-ins is given in the section IDE Plug-ins.
- Via an ActiveX Control, which is available if the integration package is installed. For more information, see the section ActiveX Integration.

External programs

In the Overview part of this section, we describe how the functionality of XMLSpy can be accessed and automated from external programs.

Using the Application API from outside XMLSpy requires an instance of XMLSpy to be started first. How this is done depends on the programming language used. See the section, Programming Languages, for information about individual languages.

Essentially, XMLSpy will be started via its COM registration. Then the `Application` object associated with the XMLSpy instance is returned. Depending on the COM settings, an object associated with an already running XMLSpy can be returned. Any programming language that supports creation and invocation of COM objects can be used. The most common of these are listed below.

- JScript and VBScript script files have a simple syntax and are designed to access COM objects. They can be run directly from a DOS command line or with a double click on Windows Explorer. They are best used for simple automation tasks.
- C# is a full-fledged programming language that has a wide range of existing functionality. Access to COM objects can be automatically wrapped using C#..
- C++ provides direct control over COM access but requires relatively larger amounts of code than the other languages.
- Java: Altova products come with native Java classes that wrap the Application API and provide a full Java look-and-feel.
- Other programming languages that make useful alternatives are: Visual Basic for Applications, Perl, and Python.

© 2016-2022 Altova GmbH

Programming points

The following limitations must be considered in your client code:

- Be aware that if your client code crashes, instances of XMLSpy may still remain in the system.
- Don't hold references to objects in memory longer than you need them, especially those from the `XMLData` interface. If the user interacts between two calls of your client, then there is no guarantee that these references are still valid.
- Don't forget to disable dialogs if the user interface is not visible.
- See Error handling in JScript (and in C# and Java) for details of how to avoid annoying error messages.
- Free references explicitly if you are using C# or C++.

This documentation

This documentation section about the Application API is broadly divided into two parts.

- The first part consists of an Overview, which describes the object model for the API and explains how the API is accessed via various programming languages.
- The second part is a reference section (Interfaces and Enumerations) that contains descriptions of the interface objects of the Application API.

29.3.1 Overview

This overview of the Application API is organized as follows:

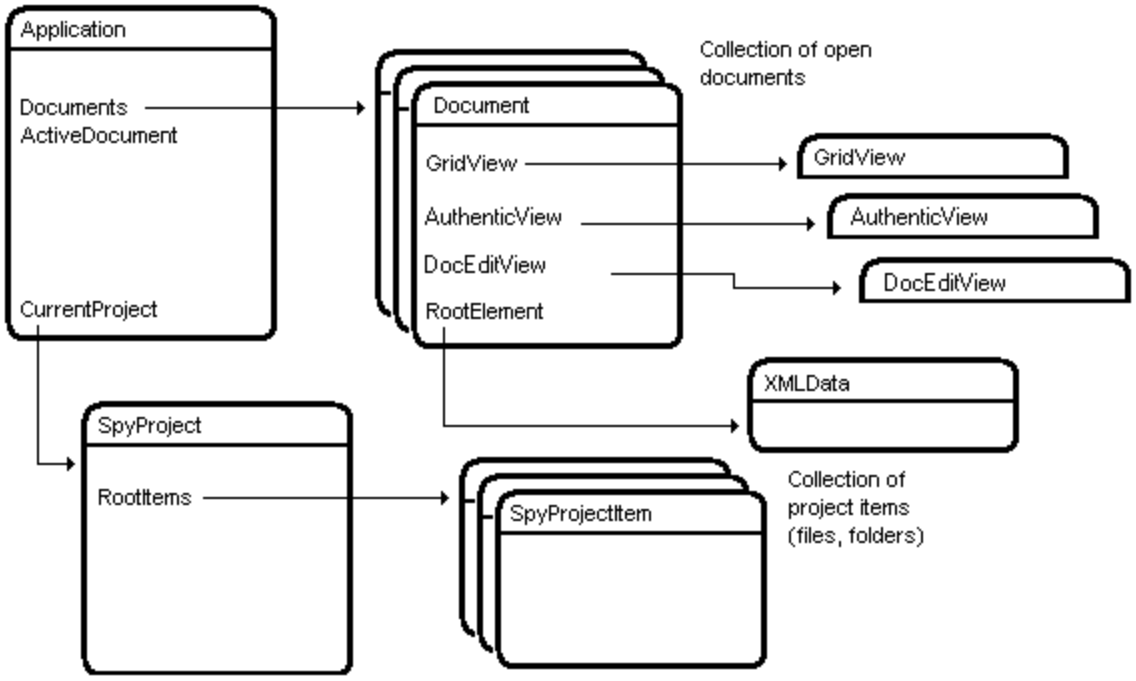
- The Object Model describes the relationships between the objects of the Application API.
- Programming Languages explains how the most commonly used programming languages (JScript, VBScript, C#, and Java) can be used to access the functionality of the Application API. Code listings from the example files supplied with your application package are used to describe basic mechanisms.
- The DOM and XMLData explains the relationship between the Application API's XMLData interface and the DOM.
- Obsolete: Authentic View Row Operations supplies information about obsolete objects for Authentic View table row operations.
- Obsolete: Authentic View Editing Operations supplies information about obsolete objects for Authentic View editing operations.

29.3.1.1 Object Model

The starting point for every application which uses the Application API is the `Application` object. This object contains general methods like import/export support and references to the open documents and any open project.

The `Application` object is created differently in various programming languages. In scripting languages such as JScript or VBScript, this involves calling a function which initializes the application's COM object. For examples, see the Programming Languages section.

The picture below shows the links between the main objects of the Application API:



The application object consists of the following parts:

1. Document collection and reference to the active document.
2. Reference to current project and methods for creating and opening projects.
3. Methods to support the export to and import from databases, text files, and Word documents.
4. URL management.
5. Methods for macro menu items.

Once you have created an Application object you can start using the functionality of XMLSpy. In most cases, you either open a project and access the documents from there or you directly open a document via the Documents interface.

29.3.1.2 Programming Languages

Programming languages differ in the way they support COM access. A few examples for the most frequently used languages (*links below*) will help you get started. The code listings in this section show how basic functionality can be accessed. The files in the `API` subfolder of the `Examples` folder can be used to test this functionality:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples`

JScript

The JScript listings demonstrate the following basic functionality:

 © 2016-2022 Altova GmbH

- Start application or attach to a running instance
- Simple document access
- Iteration
- Error handling
- Events
- Import and export of data

VBScript

VBScript is different than JScript only syntactically; otherwise it works in the same way. The link below goes to an example of how VBScript can be used. For more information, refer to the JScript examples.

- Events: Shows how events are handled using VBScript.

C#

C# can be used to access the Application API functionality. The code listings show how to access the API for certain basic functionality.

- Start XMLSpy: Starts XMLSpy, which is registered as an automation server, or activates the program if XMLSpy is already running.
- Open OrgChart.pxf: Locates one of the example documents installed with XMLSpy and opens it. If this document is already open it becomes the active document.
- OnDocumentOpened Event On/Off: Shows how to listen to XMLSpy events. When turned on, a message box will pop up after a document has been opened.
- Open ExpReport.xml: Opens another example document.
- Toggle View Mode: Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- Validate: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- Shutdown XMLSpy: Stops XMLSpy.

Java

The XMLSpy API can be accessed from Java code. The Java sub-section of this section explains how some basic XMLSpy functionality can be accessed from Java code. It is organized into the following sub-sections:

- Mapping Rules for the Java Wrapper
- Example Java Project
- Application Startup and Shutdown
- Simple Document Access
- Iterations
- Use of Out-Parameters
- Event Handlers

29.3.1.2.1 JScript

This section contains listings of JScript code that demonstrate the following basic functionality:

- Start application or attach to a running instance
- Simple document access
- Iteration
- Error handling
- Events
- Import and export of data

Example files

The code listings in this section are available in example files that you can test as is or modify to suit your needs. The JScript example files are located in the `JScript` folder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples`

The example files can be run in one of two ways:

- *From the command line:* Open a command prompt window, change the directory to the path above, and type the name of one of the example scripts (for example, `Start.js`).
- *From Windows Explorer:* In Windows Explorer, browse for the JScript file and double-click it.

The script is executed by Windows Script Host that is packaged with Windows operating system. For more information about Windows Script Host, refer to MSDN documentation (https://msdn.microsoft.com).

29.3.1.2.1.1 Start Application

The JScript below starts the application and shuts it down. If the COM object of the 32-bit XMLSpy cannot be found, the code attempts to get the COM object of the 64-bit application; otherwise, an error is thrown. If an instance of the application is already running, the running instance will be returned.

Note : Pour XMLSpy 32-bit, le nom enregistré ou l'identificateur programmatique (ProgId) de l'objet COM est `XMLSpy.Application`. Pour XMLSpy 64-bit, le nom est `XMLSpy_x64.Application`. Néanmoins, n'oubliez pas que le programme d'appel accédera aux entrée de registre CLASSES dans sa propre ruche de registre, ou groupe (32-bit ou 64-bit). C'est pourquoi, si vous exécutez des scripts en utilisant l'invite de commande standard et Windows Explorer sur Windows 64-bit, les entrées de registre 64-bit seront accédées, et pointeront vers le XMLSpy 64-bit. Pour cette raison, si les deux XMLSpy 32-bit et 64-bit sont installés, une gestion spéciale est nécessaire afin de pouvoir appeler le XMLSpy 32-bit. Par exemple, si on part du principe que Windows Scripting Host est le programme d'appel, procéder comme suit :

1. Changer le répertoire actuel à **C:\Windows\SysWOW64**.
2. Dans la ligne de commande, saisir **wscript.exe** suivi du chemin vers le script que vous souhaitez exécuter, par exemple :

```
wscript.exe "C:\Users\...
\Documents\Altova\StyleVision2023\StyleVisionExamples\API\JScript\start.js"wscript.exe
"C:\Users\...\Documents\Altova\XMLSpy2023\Examples\API\JScript\start.js"
```


© 2016-2022 Altova GmbH

```
// Initialize application's COM object. This will start a new instance of the application
and
// return its main COM object. Depending on COM settings, the main COM object of an
already
// running application might be returned.

try {   objSpy = WScript.GetObject("", "XMLSpy.Application");   }
catch(err) {}

if( typeof( objSpy ) == "undefined" )
{
   try   {   objSpy = WScript.GetObject("", "XMLSpy_x64.Application")   }
   catch(err)
   {
      WScript.Echo( "Can't access or create XMLSpy.Application" );
      WScript.Quit();
   }
}

// if newly started, the application will start without its UI visible. Set it to
visible.
objSpy.Visible = true;

WScript.Echo(objSpy.Edition + " has successfully started. ");

objSpy.Visible = false;      // will shutdown application if it has no more COM
connections
//objSpy.Visible = true;   // will keep application running with UI visible
```

The JScript code listed above is available in the sample file `Start.js` (see Example Files).

29.3.1.2.1.2 Simple Document Access

After you have started the application as shown in Start Application, you will most likely want to programmatically open a document in order to work with it. The JScript code listing below illustrates how to open two documents from the XMLSpy Examples folder and set one of them as the active document.

```
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objSpy.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\XMLSpy" + majorVersionYear + "\\Examples\\";

// Tell XMLSpy to open two documents. No dialogs
objDoc1 = objSpy.Documents.OpenFile(strExampleFolder + "OrgChart.pxf", false);
objSpy.Documents.OpenFile(strExampleFolder + "ExpReport.xml", false);
```

```
// The document currently active can be easily located.
objDoc2 = objSpy.ActiveDocument;

// Let us make sure that the document is shown in grid view.
objDoc2.SwitchViewMode(0);   // SPYViewModes.spyViewGrid = 0

// Now switch back to the document opened first
objDoc1.SetActiveDocument();
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see Example Files).

29.3.1.2.1.3 Iteration

The JScript listing below shows how to iterate through the open documents. It is assumed that you have already started the application and opened some documents as shown in the previous sections.

```
// go through all open documents using a JScript Enumerator
bRequiresSaving = false;
for (var iterDocs = new Enumerator(objSpy.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
   if (iterDocs.item().IsModified)
      bRequiresSaving = true;

   var strErrorText = new Array(1);
   var nErrorNumber = new Array(1);
   var errorData = new Array(1);

   if (!iterDocs.item().IsValid(strErrorText, nErrorNumber, errorData))
   {
       var text = strErrorText;
      // access that XMLData object only if filled in
       if (errorData[0] != null)
         text += "(" + errorData[0].Name + "/" + errorData[0].TextValue + ")";

      WScript.Echo("Document \"" + iterDocs.item().Name +"\" validation error[" +
nErrorNumber + "]: " + text);
   }
   else
   {
      // The COM call succeeded and the document is valid.
      WScript.Echo("Document \"" + iterDocs.item().Name + "\" is valid.");
   }
}

// go through all open documents using index-based access to the document collection
for (i = objSpy.Documents.Count; i > 0; i--)
   objSpy.Documents.Item(i).Close(false);
```


© 2016-2022 Altova GmbH

The JScript code listed above is available in the sample file `DocumentAccess.js` (see Example Files).

29.3.1.2.1.4 Error Handling

The Application API returns errors in two different ways:

- The `HRESULT` returned by every API method
- The `IErrorInfo` interface of the Application API

Every API method returns an `HRESULT`. This return value gives the caller information about errors during execution of the method. If the call was successful, the return value is `S_OK`. The `HRESULT` option is commonly used in C/C++ programs.

However, programming languages such as VisualBasic and scripting languages (and other high-level development environments) don't give the programmer access to the `HRESULT` return of a COM call. Such languages use the `IErrorInfo` interface, which is also supported by the Application API. If an error occurs, the Application API creates a new object that implements the `IErrorInfo` interface. The information provided by the `IErrorInfo` interface is imported by the development environment into its own error-handling mechanism.

For example, the JScript code listing below causes an error to be thrown by incorrectly declaring an array. Additional information about the error object is provided by its properties `number` and `description`.

```
try {
   var arr = new Array(-1);
}
catch (err) {
   WScript.Echo("Error : (" + (err.number & 0xffff) + ")" + err.description);
}
```

29.3.1.2.1.5 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using JScript.

The method `WScript.ConnectObject` is used to receive events.

```
// The event-handler function
function DocEvent_OnBeforeCloseDocument(objDocument)
{
   WScript.Echo("Received event - before closing document");
}

// Create or connect to XMLSpy (or Authentic Desktop)
try
```

```
{
   // Create the environment and XMLSpy (or Authentic Desktop)
   objWshShell = WScript.CreateObject("WScript.Shell");
   objFSO = WScript.CreateObject("Scripting.FileSystemObject");
   objSpy = WScript.GetObject("", "XMLSpy.Application");

// If only Authentic Desktop is installed (and XMLSpy is not installed) use:
// objSpy = WScript.GetObject("", "AuthenticDesktop.Application")

}
catch(err)
   { WScript.Echo ("Can't create WScript.Shell object or XMLSpy"); }

// Create document object and connect to its events
objSpy.Visible = true;
majorVersionYear = objSpy.MajorVersion + 1998
docPath = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\\Altova\
\XMLSpy" + majorVersionYear + "\\Examples\\ExpReport.xml";
objDoc = objSpy.Documents.OpenFile (docPath, false);
WScript.ConnectObject(objDoc, "DocEvent_");

// Keep running while waiting for the event
// In the meanwhile close this document in XMLSpy (or Authentic Desktop) manually
WScript.Echo ("Sleeping for 10 seconds ...");
WScript.Sleep (10000);

objDoc = null;
WScript.Echo ("Stopped listening for event");
objSpy.Quit();
```

29.3.1.2.1.6 Import and Export of Data

Before you implement your import and export tasks with the Application API, it is good practice to test the connections, parameters, SQL queries and so on in XMLSpy. In this way you are able to verify the results and make quick adjustments to import or export parameters. Most of the methods for importing and exporting data are placed in the `Application` object; the remaining functions are accessible via the `Document` interface.

There is some preparatory work necessary before the actual import or export can be started. Every import/export job consists of two parts. You need to define a connection to your data and the specific behavior for the import/export process. In case of an import, the connection is either a database, a text-file, or a Word document. The main issue is which data (columns) to imported into XMLSpy. In case of an export, the connection is either a database or a text file. Specify which data (elements of the XML file) and additional parameters (for example, automatic key generation or number of sub-levels) to use from the XML-structure.

The properties in the `DatabaseConnection`, `TextImportExportSettings` and `ExportSettings` interfaces have default values. See the corresponding descriptions in the `Interfaces` chapter for further information.

The sub-sections of this section describe each of these operations in detail.

- Import from Database
- Export to Database

© 2016-2022 Altova GmbH

- Import from Text
- Export to Text

Given below are the steps to establish a connection to an existing database for import:

1. Use a `DatabaseConnection` object and set the following:
   - The method `Application.GetDatabaseSettings` returns a new object for a database connection: `objImpSettings = objSpy.GetDatabaseSettings();`
   - You have to set either an ADO connection string, `objImpSettings.ADOConnection = strADOConnection` or the path to an existing database file: `objImpSettings.File = strExampleFolder + "Tutorial\\Company.mdb";`
   - To complete the settings you create an SQL SELECT statement to define the data to be queried: `objImpSettings.SQLSelect = "SELECT * FROM Address";`
2. Call `Application.GetDatabaseImportElementList` to get a collection of the resulting columns of the SQL query: `objElementList = objSpy.GetDatabaseImportElementList(objImpSettings);` This collection gives you the opportunity to control which columns should be imported and specify the datatype of the new elements. Each item of the collection represents one column to import. If you remove an item, the corresponding column will not be imported. You can additionally modify the `ElementListItem.ElementKind` property to set the datatype of the XML elements for each column. Please consider that `GetDatabaseImportElementList()` executes the SQL query and could initiate a time-consuming call. To avoid this, it is possible to pass a null-pointer as the second parameter to `ImportFromDatabase();` this imports all columns as plain XML elements.
3. Start the import with `Application.ImportFromDatabase`: `objImpDocFromDB = objSpy.ImportFromDatabase(objImpSettings,objElementList);`

```
// Locate examples via USERPROFILE shell variable.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objSpy.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\XMLSpy" + majorVersionYear + "\\Examples\\";

try
{
   // specify the source of data import
   objImpSettings = objSpy.GetDatabaseSettings();
   objImpSettings.File = strExampleFolder + "Tutorial\\Company.mdb";
   objImpSettings.SQLSelect = "SELECT * FROM Address";

   // column filter
   objElementList = objSpy.GetDatabaseImportElementList(objImpSettings);

   // import into a new XML file
   objImpDocFromDB = objSpy.ImportFromDatabase(objImpSettings,objElementList);
}
catch(err)
{
   WScript.Echo("Error importing from database.\n\n" +
                "Error: " + (err.number & 0xffff) + "\n" +
                "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files).

To export data to a database, carry out the steps below:

1. Use a DatabaseConnection object and set the necessary properties. All properties except SQLSelect are important for the export. ADOConnection or File defines the target for the output. You need to set only one of them.
2. Fill an ExportSettings object with the required values. These properties are the same options as those available in the export dialog of XMLSpy. Select the menu option **Convert | Export to Text files/Database** to see the options and try a combination of export settings. After that it is easy to transfer these settings to the properties of the interface. Call Application.GetExportSettings to get an ExportSettings object: objExpSettings = objSpy.GetExportSettings()
3. Build an element list with Document.GetExportElementList. The element list enables you to eliminate XML elements from the export process. It also gives you information about the record and field count in the RecordCount and FieldCount properties. Set the ExportSettings.ElementList property to this collection. It is possible to set the element list to null/Nothing (default) to export all elements.
4. Call Document.ExportToDatabase to execute the export. The description of the ExportToDatabase method contains also a code example for a database export.

```
   // set the behaviour of the export with ExportSettings
   objExpSettings = objSpy.GetExportSettings()

   //set the destination with DatabaseConnection
   objDB = objSpy.GetDatabaseSettings();
   objDB.CreateMissingTables = true;
   objDB.CreateNew = true;
   objDB.File = "C:\\Temp\\Export.mdb";

try
{
   objImpDocFromDB.ExportToDatabase(objImpDocFromDB.RootElement, objExpSettings, objDB);
}
catch(err)
{
   WScript.Echo("Error exporting to database.\n\n" +
                     "Error: " + (err.number & 0xffff) + "\n" +
                     "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file ImportExport.js (see Example Files).

Importing data from a text file is similar to the import from a database. You must use other interfaces (described in steps 1 to 3 below) with different methods and properties:

1. Use a TextImportExportSettings object and set the properties: The method Application.GetTextImportExportSettings returns a new object to specify a text file for import: objImpSettings = objSpy.GetTextImportExportSettings(); You have to set at least the ImportFile property to the path of the file for the import. Another important property is HeaderRow. Set it to False if the text file does not contain a leading line as a header row: objImpSettings.ImportFile = strExampleFolder + "Tutorial\\Shapes.txt";
2. Call Application.GetTextImportElementList to get a collection of all columns inside the text file: objElementList = objSpy.GetTextImportElementList(objImpSettings);

© 2016-2022 Altova GmbH

3. Start the import with Application.ImportFromText: `objImpDocFromText = objSpy.ImportFromText(objImpSettings,objElementList);`

```
try
{
   // specify the source of data import
   objImpSettings = objSpy.GetTextImportExportSettings();
   objImpSettings.ImportFile = strExampleFolder + "Tutorial\\Shapes.txt";
   objImpSettings.HeaderRow = false;

   // column filter
   objElementList = objSpy.GetTextImportElementList(objImpSettings);

   // import into a new XML file
   objImpDocFromText = objSpy.ImportFromText(objImpSettings,objElementList);
}
catch(err)
{
   WScript.Echo("Error importing from text file.\n\n" +
                    "Error: " + (err.number & 0xffff) + "\n" +
                    "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files).

To export data to text, carry out the steps below:

1. Use a TextImportExportSettings object and set the necessary properties.
2. Fill an ExportSettings object with the required values. See Item 2 in Export to database.
3. Build an element list with Document.GetExportElementList. See Item 3 in Export to database.
4. Call Document.ExportToText to execute the export.

```
objExpSettings = objSpy.GetExportSettings();
objExpSettings.ElementList =
objImpDocFromText.GetExportElementList(objImpDocFromText.RootElement, objExpSettings);

objTextExp = objSpy.GetTextImportExportSettings();
objTextExp.HeaderRow = true;
objTextExp.DestinationFolder = "C:\\Temp";

try
{

       objImpDocFromText.ExportToText(objImpDocFromText.RootElement, objExpSettings,
objTextExp);
}
catch(err)
{
   WScript.Echo("Error exporting to text.\n\n" +
                    "Error: " + (err.number & 0xffff) + "\n" +
```

```
                    "Description: " + err.description);
}
```

The JScript code listed above is available in the sample file `ImportExport.js` (see Example Files).

29.3.1.2.2 VBScript

VBScript is syntactically different than JScript but works in the same way. This section contains a listing showing how events are used with VBScript and an example.

For information about other functionality, refer to the JScript examples listed below:

- Start application or attach to a running instance
- Simple document access
- Iteration
- Error handling
- Import and export of data

29.3.1.2.2.1 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using VBScript.

The method `WScript.ConnectObject` is used to receive events.

To run this code, paste it into a file with .vbs extension, and either double-click in Windows Explorer, or run it from a command prompt.

```
' the event handler function
Function DocEvent_OnBeforeCloseDocument(objDocument)
   Call WScript.Echo("received event - before closing document")
End Function

' create or connect to XmlSpy
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set objSpy = WScript.GetObject("", "XMLSpy.Application")
' If only Authentic is installed (and XMLSpy is not installed) use:
' Set objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
' If only XMLSpy 64-bit is intalled, use:
' Set objSpy = WScript.GetObject("", "XMLSpy_x64.Application")

' create document object and connect to its events
objSpy.Visible = True
```


© 2016-2022 Altova GmbH

```
' Find out user's personal folder and locate one of the installed examples.
personalFolder = objWshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear  &
"\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' open a document
Set objDoc = objSpy.Documents.OpenFile (docPath, False)
Call WScript.ConnectObject(objDoc, "DocEvent_")

' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
Call WScript.Echo ("sleeping for 10 seconds ...")
Call WScript.Sleep (10000)

Set objDoc = Nothing
Call WScript.Echo ("stopped listening for event")
Call objSpy.Quit
```

Note : Pour XMLSpy 32-bit, le nom enregistré ou l'identificateur programmatique (ProgId) de l'objet COM est `XMLSpy.Application`. Pour XMLSpy 64-bit, le nom est `XMLSpy_x64.Application`. Néanmoins, n'oubliez pas que le programme d'appel accédera aux entrée de registre CLASSES dans sa propre ruche de registre, ou groupe (32-bit ou 64-bit). C'est pourquoi, si vous exécutez des scripts en utilisant l'invite de commande standard et Windows Explorer sur Windows 64-bit, les entrées de registre 64-bit seront accédées, et pointeront vers le XMLSpy 64-bit. Pour cette raison, si les deux XMLSpy 32-bit et 64-bit sont installés, une gestion spéciale est nécessaire afin de pouvoir appeler le XMLSpy 32-bit. Par exemple, si on part du principe que Windows Scripting Host est le programme d'appel, procéder comme suit :

1. Changer le répertoire actuel à **C:\Windows\SysWOW64**.
2. Dans la ligne de commande, saisir **wscript.exe** suivi du chemin vers le script que vous souhaitez exécuter, par exemple :

```
wscript.exe "C:\Users\...
\Documents\Altova\StyleVision2023\StyleVisionExamples\API\JScript\start.js"wscript.exe
"C:\Users\...\Documents\Altova\XMLSpy2023\Examples\API\JScript\start.js"
```

29.3.1.2.2.2 Example: Using Events

Authentic View supports event connection on a per-object basis. Implementation of this feature is based on COM connection points and is available in environments that support this mechanism.

The following example is a VBScript code example that shows how to use events from within a VBScript project.

```
' ---------------------------------------------------------------------
' VBScript example that demonstrates how to use events.
```

```
' ------------------------------------------------------------------------

' Event handler for OnSelectionChanged event of AuthenticView
Function AuthenticViewEvent_OnSelectionChanged(objAuthenticRange)
    If objAuthenticRange.FirstTextPosition <> objAuthenticRange.LastTextPosition Then
        Call WScript.Echo("Selection: " & objAuthenticRange.Text & vbNewLine & vbNewLine
& "Close this dialog.")
    Else
        Call WScript.Echo("Cursor position: " & objAuthenticRange.FirstTextPosition &
vbNewLine & vbNewLine & "Close this dialog.")
    End If
End Function

' Start/access XMLSpy and connect to its automation interface.
Set WshShell = WScript.CreateObject("WScript.Shell")
Set objSpy = GetObject("", "XMLSpy.Application")
' Make the UI of XMLSpy visible.
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed XMLSpy examples.
personalFolder = WshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear  &
"\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' Create object to access windows file system and test if the our document exists.
Set fso = CreateObject("Scripting.FileSystemObject")
If fso.FileExists(docPath) Then
    ' open the document
    Call objSpy.Documents.OpenFile(docPath, False)
    set objDoc = objSpy.ActiveDocument

    ' switch active document to authentic view
    objDoc.SwitchViewMode 4 ' spyViewAuthentic

    ' Register for connection point events on the authentic view of the active document.
    ' Any function with a valid event name prefixed with "AuthenticViewEvent_" will
    ' be called when the corresponding event gets triggered on the specified object.
    set objView = objDoc.AuthenticView
    Call WScript.ConnectObject(objView, "AuthenticViewEvent_")
    Call WScript.Echo("Events are connected." & vbNewLine & vbNewLine & "Now set or move
the cursor in XMLSpy." & vbNewLine & vbNewLine & "Close this dialog to shut down
XMLSpy.")

    ' To disconnect from the events delete the reference to the object.
    set objView = Nothing
Else
    Call WScript.Echo("The file " & docPath & " does not exist.")
End If

' shut down XMLSpy when this script ends
objSpy.Visible = False
```

 © 2016-2022 Altova GmbH

29.3.1.2.3 C#

The C# programming language can be used to access the Application API functionality. You could use Visual Studio 2012/2013/2015/2017/2019/2022 to create the C# code, saving it in a Visual Studio project. Create the project as follows:

1. In Microsoft Visual Studio, add a new project using **File | New | Project**.
2. Add a reference to the XMLSpy Type Library by clicking **Project | Add Reference**. The Add Reference dialog appears. Browse for the XMLSpy Type Library component, which is located in the XMLSpy application folder, and add it.
3. Enter the code you want.
4. Compile the code and run it.

Example C# project

Your XMLSpy package contains an example C# project, which is located in the `API\C#` subfolder of the `Examples` folder :

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples`

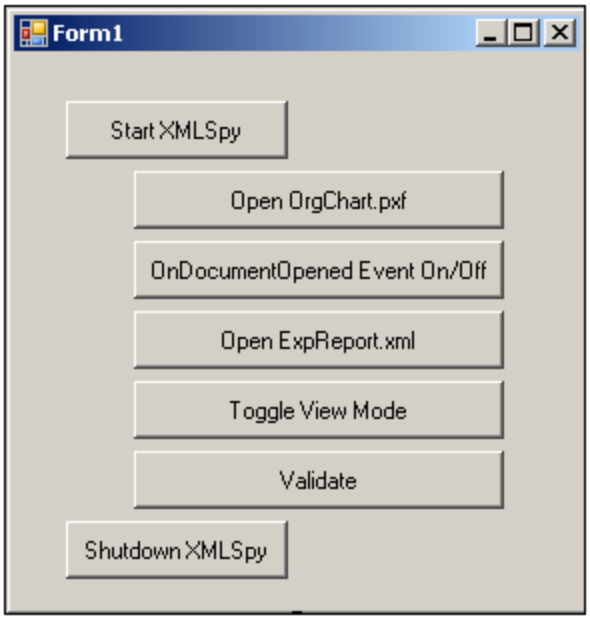
You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022. The code listing below shows how basic application functionality can be used. This code is similar to the example C# project in the API Examples folder of your application package, but might differ slightly.

Platform configuration

If you have a 64-bit operating system and are using a 32-bit installation of XMLSpy, you must add the x86 platform in the solution's Configuration Manager and build the sample using this configuration. A new x86 platform (for the active solution in Visual Studio) can be created in the New Solution Platform dialog (**Build | Configuration Manager | Active solution platform | <New…>**).

What the code listing below does

The example code listing below creates a simple user interface (*screenshot below*) with buttons that invoke basic XMLSpy operations:



- Start XMLSpy: Starts XMLSpy, which is registered as an automation server, or activates the application if it is already running.
- Open OrgChart.pxf: Locates one of the example documents installed with XMLSpy and opens it. If this document is already open it becomes the active document.
- OnDocumentOpened Event On/Off: Shows how to listen to XMLSpy events. When turned on, a message box will pop up after a document has been opened.
- Open ExpReport.xml: Opens another example document.
- Toggle View Mode: Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- Validate: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- Shut down XMLSpy: Stops XMLSpy.

You can modify the code (of the code listing below or of the example C# project in the API Examples folder) in any way you like and run it.

Compiling and running the example

In the API Examples folder, double-click the file `AutomateXMLSpy_VS2008.sln` or the file `AutomateXMLSpy_VS2010.sln` (to open in Visual Studio 2012/2013/2015/2017/2019/2022). Alternatively the file can be opened from within Visual Studio (with **File | Open | Project/Solution**). To compile and run the example, select **Debug | Start Debugging** or **Debug | Start Without Debugging**.

Code listing of the example

Given below is the C# code listing of the basic functionality of the form (`Form1.cs`) created in the `AutomateXMLSpy` example. Note that the code listed below might differ slightly from the code in the API Examples form.The listing below is commented for ease of understanding. Parts of the code are also presented separately in the sub-sections of this section, according to the Application API functionality they access.

The code essentially consists of a series of handlers for the buttons in the user interface shown in the screenshot above.

© 2016-2022 Altova GmbH

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // An instance of XMLSpy accessed via its automation interface
        XMLSpyLib.Application XMLSpy;

        // Location of examples installed with XMLSpy
        String strExamplesFolder;

        private void Form1_Load(object sender, EventArgs e)
        {
            // Locate examples installed with XMLSpy
            // REMARK: You might need to adapt this if you have a different major version
of the product
            strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My
Documents\\Altova\\XMLSpy2012\\Examples\\";
        }

        // Handler for the "Start XMLSpy" button
        private void StartXMLSpy_Click(object sender, EventArgs e)
        {
            if (XMLSpy == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If no XMLSpy instance is open, create one and make it visible
                XMLSpy = new XMLSpyLib.Application();
                XMLSpy.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If an instance of XMLSpy is already running, make sure it's visible
                if (!XMLSpy.Visible)
                    XMLSpy.Visible = true;
            }
        }

        // Handler for the "Open OrgChart.pxf" button
        private void openOrgChart_Click(object sender, EventArgs e)
        {
            // Make sure there's a running XMLSpy instance, and that it's visible
            StartXMLSpy_Click(null, null);

            // Open one of the example files installed with the product
            XMLSpy.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
        }
```

```
        // Handler for the "Open ExpReport.xml" button
        private void openExpReport_Click(object sender, EventArgs e)
        {
            // Make sure there's a running XMLSpy instance, and that it's visible
            StartXMLSpy_Click(null, null);

            // Open one of the sample files installed with the product.
            XMLSpy.Documents.OpenFile(strExamplesFolder + "ExpReport.xml", false);
        }

        // Handler for the "Toggle View Mode" button
        private void toggleView_Click(object sender, EventArgs e)
        {
            // Make sure there's a running XMLSpy instance, and that it's visible
            StartXMLSpy_Click(null, null);

            // Iterate through all open documents and toggle view between Text View and
Authentic View
            foreach (XMLSpyLib.Document doc in XMLSpy.Documents)
                if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewText)
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
                else
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewText);
        }

        // Handler for the "Shutdown XMLSpy" button
        // Shut down the application instance by explicitly releasing the COM object
        private void shutdownXMLSpy_Click(object sender, EventArgs e)
        {
            if (XMLSpy != null)
            {
                // Allow shutdown of XMLSpy by releasing the UI
                XMLSpy.Visible = false;

                // Explicitly release the COM object
                try
                {
                    while (System.Runtime.InteropServices.Marshal.ReleaseComObject(XMLSpy)
> 0) ;
                }
                finally
                {
                    // Disallow subsequent access to this object
                    XMLSpy = null;
                }
            }
        }

        // Handler for button "Validate"
        private void validate_Click(object sender, EventArgs e)
        {
            // COM errors are returned to C# as exceptions. We use a try/catch block to
handle them.
            try
            {
```

 © 2016-2022 Altova GmbH

```
                // Method 'IsValid' is one of the few functions that uses output parameters
                // Use 'object' type for these parameters
                object strErrorText = "";
                object nErrorNumber = 0;
                object errorData = null;

                if (!XMLSpy.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref
errorData))
                {
                    // The COM call succeeded but the document is not valid
                    // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is not valid.");
                    listBoxMessages.Items.Add("\tErrorText  : " + strErrorText);
                    listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
                    listBoxMessages.Items.Add("\tElement    : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
                }
                else
                {
                    // The COM call succeeded and the document is valid
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is valid.");
                }
            }
            catch (Exception ex)
            {
                // The COM call was not successful
                // Probably no application instance has been started or no document is
open.
                listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
            }
        }



        // Event handler for OnDocumentOpened event
        private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
        {
            MessageBox.Show("Document " + i_ipDocument.Name + " was opened!");
        }

        // Remember if the event handler is currently registered.
        private bool bEventHandlerIsRegistered = false;

        // Handler for button 'OnDocuemntOpened Event On/Off
        private void toggleOnDocumentOpenedEvent_Click(object sender, EventArgs e)
        {
            if (XMLSpy != null)
            {
                if (bEventHandlerIsRegistered)
                    XMLSpy.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
```

```
                    XMLSpy.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);

                bEventHandlerIsRegistered = !bEventHandlerIsRegistered;
            }
        }
    }
}
```

29.3.1.2.3.1 Add Reference to XMLSpy API

Add the application's type library as a reference in a .NET project as follows: With the .NET project open, click **Project | Add Reference**. Then browse for the type library, which is called `XMLSpy.tlb`, and is located in the XMLSpy application folder.

Then declare a variable to access the XMLSpy API:

```
// An instance of XMLSpy is accessed via its automation interface.
   XMLSpyLib.Application XMLSpy;
```

29.3.1.2.3.2 Application Startup and Shutdown

In the code snippets below, the methods `StartXMLSpy_Click` and `ShutdownXMLSpy_Click` are those assigned to buttons in the AutomateXMLSpy example that, respectively, start up and shut down the application. This example is located in the C# folder of the API Examples folder (*see the file Form1.cs*):

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples`

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Starting XMLSpy

The following code snippet from the AutomateXMLSpy example shows how to start up the application.

```
// Handler for the "Start XMLSpy" button
private void StartXMLSpy_Click(object sender, EventArgs e)
{
    if (XMLSpy == null)
    {
        Cursor.Current = Cursors.WaitCursor;

        // If no XMLSpy instance is running, we create one and make it visible
        XMLSpy = new XMLSpyLib.Application();
        XMLSpy.Visible = true;

        Cursor.Current = Cursors.Default;
    }
```

 © 2016-2022 Altova GmbH

```
            else
            {
                // If an instance of XMLSpy is already running, make sure it's visible
                if (!XMLSpy.Visible)
                    XMLSpy.Visible = true;
            }
        }
```

Shutting down XMLSpy

The following code snippet from the AutomateXMLSpy example shows how to shut down the application.

```
        // Handler for the "Shutdown XMLSpy" button
        // Shut down the application instance by explicitly releasing the COM object
         private void shutdownXMLSpy_Click(object sender, EventArgs e)
         {
             if (XMLSpy != null)
             {
                 // Allow shutdown of XMLSpy by releasing the UI
                 XMLSpy.Visible = false;

                 // Explicitly release COM object
                 try
                 {
                     while (System.Runtime.InteropServices.Marshal.ReleaseComObject(XMLSpy)
> 0) ;
                 }
                 finally
                 {
                     // Disallow subsequent access to this object
                     XMLSpy = null;
                 }
             }
         }
```

29.3.1.2.3.3 Opening Documents

The code snippets below (from the AutomateXMLSpy example) show how two files are opened via two separate methods assigned to two buttons in the user interface. Both methods use the same Application API access mechanism: XMLSpy.Documents.OpenFile(string, boolean).

The AutomateXMLSpy example (*see the file Form1.cs*) is located in the C# folder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples`

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
        // Handler for the "Open OrgChart.pxf" button
         private void openOrgChart_Click(object sender, EventArgs e)
         {
             // Make sure there's a running XMLSpy instance, and that it's visible
             StartXMLSpy_Click(null, null);

             // Open a file from the Examples folder installed with the product
             XMLSpy.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
         }

         // Handler for the "Open ExpReport.xml" button
         private void openExpReport_Click(object sender, EventArgs e)
         {
             // Make sure there's a running XMLSpy instance, and that it's visible
             StartXMLSpy_Click(null, null);

             // Open a file from the Examples folder installed with the product
             XMLSpy.Documents.OpenFile(strExamplesFolder + "ExpReport.xml", false);
         }
```

The file opened last will be the active file.

29.3.1.2.3.4 Iterating through Open Documents

The code snippet below (from the AutomateXMLSpy example; *see the file Form1.cs*) shows how to iterate through open documents. A condition is then tested within the iteration loop, and the document view is switched between Text View and Authentic View.

```
        // Handler for the "Toggle view mode" button
         private void toggleView_Click(object sender, EventArgs e)
         {
             // Make sure there's a running XMLSpy instance, and that it's visible
             StartXMLSpy_Click(null, null);

             // Iterate through open documents and toggle current view between text and
authentic view.
             foreach (XMLSpyLib.Document doc in XMLSpy.Documents)
                 if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewText)
                     doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
                 else
                     doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewText);
         }
```

The AutomateXMLSpy example example is located in the C# folder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

© 2016-2022 Altova GmbH

29.3.1.2.3.5 Errors and COM Output Parameters

The code snippet below (from the AutomateXMLSpy example) shows how to handle errors and COM output parameters. The method XMLSpy.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref errorData) uses output parameters that are used, in the code snippet below, to generate an error-message text.

The AutomateXMLSpy example (*see the file Form1.cs*) is located in the C# folder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
        // Handler for button "Validate"
        private void validate_Click(object sender, EventArgs e)
        {
            // COM errors are returned to C# as exceptions. We use a try/catch block to
handle them.
            try
            {
                // Method 'IsValid' is one of the few functions that uses output parameters
                // Use 'object' type for these parameters
                object strErrorText = "";
                object nErrorNumber = 0;
                object errorData = null;

                if (!XMLSpy.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref
errorData))
                {
                    // The COM call succeeded but the document is not valid
                    // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is not valid.");
                    listBoxMessages.Items.Add("\tErrorText  : " + strErrorText);
                    listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
                    listBoxMessages.Items.Add("\tElement    : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
                }
                else
                {
                    // The COM call succeeded and the document is valid
                    listBoxMessages.Items.Add("Document " + XMLSpy.ActiveDocument.Name + "
is valid.");
                }
            }
            catch (Exception ex)
```

```
            {
                // The COM call was not successful
                // Probably no application instance has been started or no document is
open.
                listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
            }
        }
```

29.3.1.2.3.6 Events

The code snippet below (from the AutomateXMLSpy example) lists the code for two event handlers. The AutomateXMLSpy example (*see the file Form1.cs*) is located in the C# folder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples`

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

Code snippet

```
        // Event handler for OnDocumentOpened event
        private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
        {
            MessageBox.Show("Document " + i_ipDocument.Name + " was opened!");
        }

        // Remember if the event handler is currently registered.
        private bool bEventHandlerIsRegistered = false;

        // Handler for button 'OnDocuemntOpened Event On/Off
        private void toggleOnDocumentOpenedEvent_Click(object sender, EventArgs e)
        {
            if (XMLSpy != null)
            {
                if (bEventHandlerIsRegistered)
                    XMLSpy.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
                    XMLSpy.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);

                bEventHandlerIsRegistered = !bEventHandlerIsRegistered;
            }
        }
```


© 2016-2022 Altova GmbH

29.3.1.2.4 Java

The Application API can be accessed from Java code. To allow accessing the XMLSpy automation server directly from Java code, the libraries listed below must reside in the `classpath`. They are installed in the folder: `JavaAPI` in the XMLSpy application folder.

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (`AltovaAutomation_x64.dll` in the case of 64-bit versions)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `XMLSpyAPI.jar`: Java classes that wrap the XMLSpy automation interface
- `XMLSpyAPI_JavaDoc.zip`: a `Javadoc` file containing help documentation for the Java API

Note: In order to use the Java API, the DLL and Jar files must be on the Java Classpath.

Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details of the example Java project, see the section, Example Java Project.

Rules for mapping the Application API names to Java

The rules for mapping between the Application API and the Java wrapper are as follows:

- ***Classes and class names***
  For every interface of the XMLSpy automation interface a Java class exists with the name of the interface.

- ***Method names***
  Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with `get` and `set` can be used. If a property does not support write-access, no setter method is available. Example: For the `Name` property of the `Document` interface, the Java methods `getName` and `setName` are available.

- ***Enumerations***
  For every enumeration defined in the automation interface, a Java enumeration is defined with the same name and values.

- ***Events and event handlers***
  For every interface in the automation interface that supports events, a Java interface with the same name plus '`Event`' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus '`DefaultHandler`'. For example:
  `Application`: Java class to access the application
  `ApplicationEvents`: Events interface for the Application
  `ApplicationEventsDefaultHandler`: Default handler for `ApplicationEvents`

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
Document, method SetEncoding	setFileEncoding
AuthenticView, method Goto	gotoElement
AuthenticRange, method Goto	gotoElement
AuthenticRange, method Clone	cloneRange

This section

This section explains how some basic XMLSpy functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project
- Application Startup and Shutdown
- Simple Document Access
- Iterations
- Use of Out-Parameters
- Event Handlers

29.3.1.2.4.1 Example Java Project

The XMLSpy installation package contains an example Java project, located in the the API\Java subfolder of the Examples folder :

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<username>\Documents\Altova\XMLSpy\2023\Examples

This folder contains Java examples for the XMLSpy API. You can test it directly from the command line using the batch file BuildAndRun.bat, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

File list

The Java examples folder contains all the files required to run the example project. These files are listed below. If you are using a 64-bit version of the application, some filenames contain _x64 in the name. These filenames are indicated with (_x64).

AltovaAutomation(_x64).dll	Java-COM bridge: DLL part
AltovaAutomation.jar	Java-COM bridge: Java library part
XMLSpyAPI.jar	Java classes of the XMLSpy API
RunXMLSpy.java	Java example source code
BuildAndRun.bat	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
.classpath	Eclipse project helper file
.project	Eclipse project file
XMLSpyAPI_JavaDoc.zip	Javadoc file containing help documentation for the Java API

© 2016-2022 Altova GmbH

What the example does

The example starts up XMLSpy and performs a few operations, including opening and closing documents. When done, XMLSpy stays open. You must close it manually.

- Start XML Spy: Starts XMLSpy, which is registered as an automation server, or activates XMLSpy if it is already running.
- Open OrgChart.pxf: Locates one of the example documents installed with XMLSpy and opens it.
- Iteration and Changing the View Mode: Changes the view of all open documents to Text View. The code also shows how to iterate through open documents.
- Iteration, validation, output parameters: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- Event Handling: Shows how to handle XMLSpy events.
- Shut down XMLSpy: Shuts down XMLSpy.

You can modify the example in any way you like and run it.

Running the example from the command line

To run the example from the command line, open a command prompt window, go to the Java folder of the API Examples folder (*see above for location*), and then type:

```
buildAndRun.bat "<Path-to-the-Java-bin-folder>"
```

The Java binary folder must be that of a JDK 1.5 or later installation on your computer. Press the **Return** key. The Java source in `RunXMLSpy.java` will be compiled and then executed.

Loading the example in Eclipse

Open Eclipse and use the **Import | Existing Projects into Workspace** command to add the Eclipse project file (`.project`) located in the Java folder of the API Examples folder (*see above for location*). The project `RunXMLSpy` will then appear in your Package Explorer or Navigator. Select the project and then the command **Run as | Java Application** to execute the example.

Note: You can select a class name or method of the Java API and press F1 to get help for that class or method.

Java source code listing

The Java source code in the example file `RunXMLSpy.java` is listed below with comments.

```
001 // Access general JAVA-COM bridge classes
002 import com.altova.automation.libs.*;
003
004 // Access XMLSpy Java-COM bridge
005 import com.altova.automation.XMLSpy.*;
006 import com.altova.automation.XMLSpy.Enums.SPYViewModes;
007
008 /**
```

```
009  * An example that starts XMLSpy COM server and performs view operations on it
010  * Feel free to extend
011  */
012 public class RunXMLSpy
013 {
014   public static void main(String[] args)
015   {
016     // An instance of the application.
017     Application xmlSpy = null;
018
019     // Instead of COM error handling, use Java exception mechanism
020     try
021       {
022       // Start XMLSpy as COM server
023       xmlSpy = new Application();
024
025       // COM servers start up invisible, so make it visible
026       xmlSpy.setVisible(true);
027
028       // Locate samples installed with the product
029       String strExamplesFolder =
030       System.getenv("USERPROFILE") + "\\My Documents\\Altova\\XMLSpy2012\\Examples\\";
031
032       // Open two example files
033       xmlSpy.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
034       xmlSpy.getDocuments().openFile(strExamplesFolder + "ExpReport.xml", false);
035
036       // Iterate through open documents and set view mode to 'Text'.
037       for (Document doc:xmlSpy.getDocuments())
038       if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
039           doc.switchViewMode(SPYViewModes.spyViewText);
040
041       // An alternative iteration mode is index-based
042       // COM indices are typically zero-based
043       Documents documents = xmlSpy.getDocuments();
044       for (int i = 1; i <= documents.getCount();
045           i++)
046           {
047             Document doc = documents.getItem(i);
048
049       // Validation is one of the few methods to have output parameters.
050       // The class JVariant is the correct type for parameters in these cases.
051       // To get values back mark them with the by-reference flag.
052       JVariant validationErrorText = new
053
054       JVariant.JStringVariant("");
055
056       validationErrorText.setByRefFlag();
057       JVariant validationErrorCount = new
058
059       JVariant.JIntVariant(0);
060
061       validationErrorCount.setByRefFlag();
062       JVariant validationErrorXMLData = new
063
064       JVariant.JIDispatchVariant(0);
```

 © 2016-2022 Altova GmbH

```
065
066       validationErrorXMLData.setByRefFlag();
067       if (!doc.isValid(validationErrorText, validationErrorCount,
validationErrorXMLData))
068           System.out.println("Document" + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
069         else
070           System.out.println("Document" + doc.getName() + " is wellformed.");
071       }
072
073       // The following lines attach to the document events using a default
implementation
074       // for the events and override one of its methods.
075       // If you want to override all document events it is better to derive your
listener class
076       // from DocumentEvents and implement all methods of this interface.
077       Document doc = xmlSpy.getActiveDocument();
078       doc.addListener(new
079
080 DocumentEventsDefaultHandler()
081       {
082         @Override
083         public boolean
084
085 onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
086         {
087           System.out.println("Document
088
089 " + i_ipDoc.getName() + " requested closing.");
090
091           // Allow closing of document
092           return true;
093         }
094       });
095       doc.close(true);
096       doc = null;
097
098       System.out.println("Watch XMLSpy!");
099     }
100     catch (AutomationException e)
101     {
102       // e.printStackTrace();
103     }
104     finally
105     {
106       // Make sure that XMLSpy can shut down properly.
107       if (xmlSpy != null)
108           xmlSpy.dispose();
109
110       // Since the COM server was made visible and still is visible,
111       // it will keep running, and needs to be closed manually.
112       System.out.println("Now close XMLSpy!");
113     }
114   }
115 }
```

29.3.1.2.4.2 Application Startup and Shutdown

The code listings below show how the application can be started up and shut down.

Application startup

Before starting up the application, the appropriate classes must be imported (*see below*).

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access XMLSpy Java-COM bridge
05 import com.altova.automation.XMLSpy.*;
06 import com.altova.automation.XMLSpy.Enums.SPYViewModes;
07
08 /**
09  * An example that starts XMLSpy COM server and performs view operations on it
10  * Feel free to extend
11  */
12 public class RunXMLSpy
13 {
14   public static void main(String[] args)
15   {
16     // An instance of the application.
17     Application xmlSpy = null;
18
19     // Instead of COM error handling, use Java exception mechanism
20     try
21     {
22         // Start XMLSpy as COM server
23           xmlSpy = new Application();
24         // COM servers start up invisible, so make it visible
25           xmlSpy.setVisible(true);
26
27 ...
28     }
29   }
30 }
```

Application shutdown

The application can be shut down as shown below.

```
01 {
02       // Allow shutdown of XMLSpy by releasing the UI.
03       xmlSpy.setVisible(true);
04
05       // Make sure that XMLSpy can shut down properly.
06       if (xmlSpy != null)
07         xmlSpy.dispose();
08
09       // Since the COM server was made visible and still is visible,
10       // it will keep running, and needs to be closed manually.
11       System.out.println("Now close XMLSpy!");
```


© 2016-2022 Altova GmbH

```
12 }
```

29.3.1.2.4.3 Simple Document Access

The code listing below shows how to open a document.

```
1  // Locate samples installed with the product
2  String strExamplesFolder =
3  System.getenv("USERPROFILE") + "\\My Documents\\Altova\\XMLSpy2012\\Examples\\";
4
5
6  // Open file
7  xmlSpy.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
```

29.3.1.2.4.4 Iterations

The listing below shows how to iterate through open documents.

```
01  // Iterate through open documents and set view mode to 'Text'.
02  for (Document doc:xmlSpy.getDocuments())
03    if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
04              doc.switchViewMode(SPYViewModes.spyViewText);
05
06  // An alternative iteration mode is index-based
07  // COM indices are typically zero-based
08  Documents documents = xmlSpy.getDocuments();
09     for (int i = 1; i <= documents.getCount();
10         i++)
11       {
12         Document doc = documents.getItem(i);
13               ...
14       }
```

29.3.1.2.4.5 Use of Out-Parameters

The code listing below iterates through open documents and validates each of them. For each validation, a message is generated using the output parameters of the Validation method.

```
01 // An alternative iteration mode is index-based
02 // COM indices are typically zero-based
03 Documents documents = xmlSpy.getDocuments();
04 for (int i = 1; i <= documents.getCount();
05       i++)
06     {
```

```
07         Document doc = documents.getItem(i);
08
09 // Validation is one of the few methods to have output parameters.
10 // The class JVariant is the correct type for parameters in these cases.
11 // To get values back mark them with the by-reference flag.
12 JVariant validationErrorText = new
13
14 JVariant.JStringVariant("");
15
16 validationErrorText.setByRefFlag();
17       JVariant validationErrorCount = new
18
19 JVariant.JIntVariant(0);
20
21 validationErrorCount.setByRefFlag();
22       JVariant validationErrorXMLData = new
23
24 JVariant.JIDispatchVariant(0);
25
26 validationErrorXMLData.setByRefFlag();
27       if (!doc.isValid(validationErrorText,
28
29         validationErrorCount, validationErrorXMLData))
30         System.out.println("Document
31
32         " + doc.getName() + " is not wellformed - " +
33
34         validationErrorText.getStringValue());
35       else
36         System.out.println("Document
37
38         " + doc.getName() + " is wellformed.");
39     }
```

29.3.1.2.4.6 Event Handlers

The listing below shows how to listen for and use events.

```
01 // The following lines attach to the document events using a default implementation
02 // for the events and override one of its methods.
03 // If you want to override all document events it is better to derive your listener
class
04 // from DocumentEvents and implement all methods of this interface.
05
06 Document doc = xmlSpy.getActiveDocument();
07 doc.addListener(new DocumentEventsDefaultHandler()
08     {
09       @Override
10       public boolean
11       onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
12       {
13         System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
```


© 2016-2022 Altova GmbH

```
14
15          // Allow closing of document
16          return true;
17        }
18      });
19 doc.close(true);
20 doc = null;
```

29.3.1.3 The DOM and XMLData

The `XMLData` interface gives you full access to the XML structure behind the current document with less methods than DOM and is much simpler. The `XMLData` interface is a minimalist approach to reading and modifying existing, or newly created XML data. You might however, want to use a DOM tree because you can access one from an external source or you just prefer the MSXML DOM implementation.

The `ProcessDOMNode()` and `ProcessXMLDataNode()` functions provided below convert any segments of an XML structure between XMLData and DOM.

To use the `ProcessDOMNode()` function:
- pass the root element of the DOM segment you want to convert in objNode and
- pass the plugin object with the CreateChild() method in objCreator

To use the `ProcessXMLDataNode()` function:
- pass the root element of the XMLData segment in objXMLData and
- pass the DOMDocument object created with MSXML in xmlDoc

```
////////////////////////////////////////////////////////////////
// DOM To XMLData conversion
Function ProcessDOMNode(objNode,objCreator)
{
       var objRoot;
       objRoot = CreateXMLDataFromDOMNode(objNode,objCreator);

       If(objRoot)   {
              If((objNode.nodeValue != Null) && (objNode.nodeValue.length > 0))
                     objRoot.TextValue = objNode.nodeValue;
                     // add attributes
              If(objNode.attributes) {
                     var Attribute;
                     var oNodeList = objNode.attributes;

                     For(var i = 0;i < oNodeList.length; i++) {
                            Attribute = oNodeList.item(i);

                            var newNode;
                            newNode = ProcessDOMNode(Attribute,objCreator);

                            objRoot.AppendChild(newNode);
                     }
              }
              If(objNode.hasChildNodes)   {
```

```
                    try {
                           // add children
                           var Item;
                           oNodeList = objNode.childNodes;

                           For(var i = 0;i < oNodeList.length; i++) {
                                  Item = oNodeList.item(i);

                                  var newNode;
                                  newNode = ProcessDOMNode(Item,objCreator);

                                  objRoot.AppendChild(newNode);
                           }
                    }
                    catch(err) {
                    }
             }
       }
       Return objRoot;
}

Function CreateXMLDataFromDOMNode(objNode,objCreator)
{
       var bSetName = True;
       var bSetValue = True;

       var nKind = 4;

       switch(objNode.nodeType)    {
             Case 2:nKind = 5;break;
             Case 3:nKind = 6;bSetName = False;break;
             Case 4:nKind = 7;bSetName = False;break;
             Case 8:nKind = 8;bSetName = False;break;
             Case 7:nKind = 9;break;
       }
       var objNew = Null;
       objNew = objCreator.CreateChild(nKind);

       If(bSetName)
             objNew.Name = objNode.nodeName;

       If(bSetValue && (objNode.nodeValue != Null))
             objNew.TextValue = objNode.nodeValue;

       Return objNew;
}
////////////////////////////////////////////////////////////
// XMLData To DOM conversion

Function ProcessXMLDataNode(objXMLData,xmlDoc)
{
       var objRoot;
       objRoot = CreateDOMNodeFromXMLData(objXMLData,xmlDoc);

       If(objRoot)   {
             If(IsTextNodeEnabled(objRoot) && (objXMLData.TextValue.length > 0))
```


© 2016-2022 Altova GmbH

```
                objRoot.appendChild(xmlDoc.createTextNode(objXMLData.TextValue));

        If(objXMLData.HasChildren)  {
                try {
                        var objChild;
                        objChild = objXMLData.GetFirstChild(-1);

                        While(True)   {
                                If(objChild)  {
                                        var newNode;
                                        newNode = ProcessXMLDataNode(objChild,xmlDoc);

                                        If(newNode.nodeType == 2)   {
                                                // child node is an attribute
                                                objRoot.attributes.setNamedItem(newNode);
                                        }
                                        Else
                                                objRoot.appendChild(newNode);
                                }
                                objChild = objXMLData.GetNextChild();
                        }
                }
                catch(err) {
                }
        }
    }
    Return objRoot;
}

Function CreateDOMNodeFromXMLData(objXMLData,xmlDoc)
{
    switch(objXMLData.Kind)     {
            Case 4:Return xmlDoc.createElement(objXMLData.Name);
            Case 5:Return xmlDoc.createAttribute(objXMLData.Name);
            Case 6:Return xmlDoc.createTextNode(objXMLData.TextValue);
            Case 7:Return xmlDoc.createCDATASection(objXMLData.TextValue);
            Case 8:Return xmlDoc.createComment(objXMLData.TextValue);
            Case 9:Return
xmlDoc.createProcessingInstruction(objXMLData.Name,objXMLData.TextValue);
    }

    Return xmlDoc.createElement(objXMLData.Name);
}
Function IsTextNodeEnabled(objNode)
{
    switch(objNode.nodeType) {
            Case 1:
            Case 2:
            Case 5:
            Case 6:
            Case 11:Return True;
    }
    Return False;
}
```

29.3.1.4 Obsolete: Authentic View Row operations

If the schema on which an XML document is based specifies that an element is repeatable, such a structure can be represented in Authentic View as a table. When represented as a table, rows and their contents can be manipulated individually, thereby allowing you to manipulate each of the repeatable elements individually. Such row operations would be performed by an external script.

If an external script is to perform row operations then two steps must occur:

- The first step checks whether the cursor is currently in a row using a property. Such a check could be, for example, `IsRowInsertEnabled`, which returns a value of either `TRUE` or `FALSE`.
- If the return value is `TRUE` then a row method, such as `RowAppend`, can be called. (`RowAppend` has no parameters and returns no value.)

The following is a list of properties and methods available for table operations. Each property returns a `BOOL`, and the methods have no parameter.

Property	Method	Table operations
IsRowInsertEnabled	**RowInsert**, superseded by `AuthenticRange.InsertRow`	Insert row operation
IsRowAppendEnabled	**RowAppend**, superseded by `AuthenticRange.AppendRow`	Append row operation
IsRowDeleteEnabled	**RowDelete**, superseded by `AuthenticRange.DeleteRow`	Delete row operation
IsRowMoveUpEnabled	**RowMoveUp**, superseded by `AuthenticRange.MoveRowUp`	Move XML data up one row
IsRowMoveDownEnabled	**RowMoveDown**, superseded by `AuthenticRange.MoveRowDown`	Move XML data down one row
IsRowDuplicateEnabled	**RowDuplicate**, superseded by `AuthenticRange.DuplicateRow`	Duplicate currently selected row

29.3.1.5 Obsolete: Authentic View Editing operations

When XML data is displayed as data in Authentic View, it is possible to manipulate individual elements using standard editing operations such as cut, copy, and paste. However, not all XML data nodes can be edited. So, in order to carry out an editing operation, first a property is used to test whether editing is possible, and then a method is called to perform the editing operation.

The only method that does not have a test is the method `EditSelectAll`, which automatically selects all elements displayed in the document.

The following is a list of properties and methods that perform editing operations. Each property returns a `BOOL`, and the methods have no parameter.

© 2016-2022 Altova GmbH

Property	Method	Editing operation
IsEditUndoEnabled	**EditUndo**, superseded by AuthenticView.Undo	Undo an editing operation
IsEditRedoEnabled	**EditRedo**, superseded by AuthenticView.Redo	Redo an editing operation
IsEditCopyEnabled	**EditCopy**, superseded by AuthenticRange.Copy	Copy selected text to the clipboard
IsEditCutEnabled	**EditCut**, superseded by AuthenticRange.Cut	Cut selected text to the clipboard
IsEditPasteEnabled	**EditPaste**, superseded by AuthenticRange.Paste	Paste from clipboard to current cursor position
IsEditClearEnabled	**EditClear**, superseded by AuthenticRange.Delete	Clear selected text from XML document

29.3.2 Interfaces

Object Hierarchy

Application
- SpyProject
  - SpyProjectItems
    - SpyProjectItem
- Documents
  - Document
    - GridView
    - AuthenticView
      - AuthenticRange
      - AuthenticDataTransfer (previously DocEditDataTransfer)
    - OldAuthenticView (previously DocEditView, **now obsolete**, superseded by AuthenticView and AuthenticRange)
      - AuthenticSelection (previously DocEditSelection, **now obsolete**, superseded by AuthenticRange)
      - AuthenticEvent (previously DocEditEvent, **now obsolete**)
      - AuthenticDataTransfer (previously DocEditDataTransfer)
    - TextView
    - XMLData
- Dialogs
  - CodeGeneratorDlg
  - FileSelectionDlg
  - SchemaDocumentationDlg
  - GenerateSampleXMLDlg
  - DTDSchemaGeneratorDlg
  - FindInFilesDlg
  - WSDLDocumentationDlg
  - WSDL20DocumentationDlg
  - XBRLDocumentationDlg
- DatabaseConnection
- ExportSettings
- TextImportExportSettings

ElementList
ElementListItem

Enumerations

Description
This chapter contains the reference of the XMLSpy 1.5 Type Library.

Most of the given examples are written in VisualBasic. These code snippets assume that there is a variable defined and set, called **objSpy of type Application**. There are also some code samples written in JavaScript.

29.3.2.1 Application

Methods
GetDatabaseImportElementList
GetDatabaseSettings
GetDatabaseTables
ImportFromDatabase
CreateXMLSchemaFromDBStructure

GetTextImportElementList
GetTextImportExportSettings
ImportFromText

ImportFromWord

ImportFromSchema

GetExportSettings

NewProject
OpenProject

AddMacroMenuItem
ClearMacroMenu

ShowForm

ShowApplication

URLDelete
URLMakeDirectory

AddXSLT_XQParameter
GetXSLT_XQParameterCount
GetXSLT_XQParameterName
GetXSLT_XQParameterXPath
RemoveXSLT_XQParameter

FindInFiles

© 2016-2022 Altova GmbH

Quit

Properties
Application
Parent

ActiveDocument
Documents

CurrentProject

Dialogs

WarningNumber
WarningText

Status
MajorVersion
MinorVersion
Edition
IsAPISupported
ServicePackVersion

Description
Application is the root for all other objects. It is the only object you can create by CreateObject (VisualBasic) or other similar COM related functions.

Example

```
Dim objSpy As Application
Set objSpy = CreateObject("XMLSpy.Application")
```

29.3.2.1.1 Events

29.3.2.1.1.1 OnBeforeOpenDocument

Event: OnBeforeOpenDocument(*objDialog* as FileSelectionDlg)

Description
This event gets fired whenever a document gets opened via the OpenFile or OpenURL menu command. It is sent after a document file has been selected but before the document gets opened. The file selection dialog object is initialized with the name of the selected document file. You can modify this selection. To continue the opening of the document leave the FileSelectionDlg.DialogAction property of *io_objDialog* at its default value spyDialogOK. To abort the opening of the document set this property to spyDialogCancel.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

Function On_BeforeOpenDocument(*objDialog*)
End Function

XMLSpy scripting environment - JScript:
function On_BeforeOpenDocument(*objDialog*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (26, ...) // nEventId = 26

29.3.2.1.1.2 OnBeforeOpenProject

Event: OnBeforeOpenProject(*objDialog* as FileSelectionDlg)

Description
This event gets fired after a project file has been selected but before the project gets opened. The file selection dialog object is initialized with the name of the selected project file. You can modify this selection. To continue the opening of the project leave the FileSelectionDlg.DialogAction property of *io_objDialog* at its default value spyDialogOK. To abort the opening of the project set this property to spyDialogCancel.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
Function On_BeforeOpenProject(*objDialog*)
End Function

XMLSpy scripting environment - JScript:
function On_BeforeOpenProject(*objDialog*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (25, ...) // nEventId = 25

29.3.2.1.1.3 OnDocumentOpened

Event: OnDocumentOpened(*objDocument* as Document)

Description
This event gets fired whenever a document opens in XMLSpy. This can happen due to opening a file with the OpenFile or OpenURL dialog, creating a new file or dropping a file onto XMLSpy. The new document gets passed as parameter. The operation cannot be canceled.

Examples
Given below are examples of how this event can be scripted.

© 2016-2022 Altova GmbH

XMLSpy scripting environment - VBScript:
Function On_OpenDocument(*objDocument*)
End Function

XMLSpy scripting environment - JScript:
function On_OpenDocument(*objDocument*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (7, ...) // nEventId = 7

29.3.2.1.1.4 OnProjectOpened

Event: OnProjectOpened(*objProject* as SpyProject)

Description
This event gets fired whenever a project gets opened in XMLSpy. The new project gets passed as parameter.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
Function On_OpenProject(*objProject*)
End Function

XMLSpy scripting environment - JScript:
function On_OpenProject(*objProject*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (6, ...) // nEventId = 6

29.3.2.1.2 ActiveDocument

Property: ActiveDocument as Document

Description
Reference to the active document. If no document is open, ActiveDocument is null (nothing).

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.3 AddMacroMenuItem

Method: AddMacroMenuItem(*strMacro* as String,*strDisplayText* as String)

Description
Adds a menu item to the **Tools** menu. This new menu item invokes the macro defined by strMacro. See also Example Scripting Project.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1108 Number of macro items is limited to 16 items.

29.3.2.1.4 AddXSLT_XQParameter

Method: AddXSLT_XQParameter(name as String, XPath as String)

Description
Adds an XSLT or XQuery parameter. The parameter's name and value are the two arguments of the method.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.
1124 The XPath expression is not set.
1125 Not a QName.
1126 The specified XPath is not valid. Reason for invalidity appended.
1127 A parameter with the submitted name already exists.

29.3.2.1.5 Application

Property: Application as Application (read-only)

Description
Accesses the XMLSpy application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.6 ClearMacroMenu

Method: ClearMacroMenu()

Return Value

© 2016-2022 Altova GmbH

None

Description
Removes from the **Tools** menu those menu items that were added by calling `AddMacroMenuItem`. See also Example Scripting Project.

Errors
1111 The application object is no longer valid.

29.3.2.1.7 CreateXMLSchemaFromDBStructure

Method: CreateXMLSchemaFromDBStructure(pImportSettings as DatabaseConnection, pTables as ElementList)

Description
CreateXMLSchemaFromDBStructure creates from a database specified in pImportSettings for the defined tables in pTables new XML Schema document(s) describing the database tables structure.

The parameter pTables specifies which table structures the XML Schema document should contain. This parameter can be NULL, specifying that all table structures will be exported.

See also GetDataBaseTables.

Errors
1112 Invalid database specified.
1120 Database import failed.

29.3.2.1.8 CurrentProject

Property: CurrentProject as SpyProject

Description
Reference to the active document. If no project is open, CurrentProject is null (nothing).

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.9 Dialogs

Property: Dialogs as Dialogs (read-only)

Description
Access the built-in dialogs of XMLSpy.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.10 Documents

Property: Documents as Documents

Description
Collection of all open documents.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.11 Edition

Property: Edition as String

Description
Returns the edition of the application, for example `Altova XMLSpy Enterprise Edition` for the Enterprise edition.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.12 FindInFiles

Method: FindInFiles(*pSettings* as FindInFilesDlg) as FindInFilesResults

Description
Returns a FindInFilesResults object containing information about the files that matched the specified settings.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.13 GetDatabaseImportElementList

Method: GetDatabaseImportElementList(*pImportSettings* as DatabaseConnection) as ElementList

Description
The function returns a collection of ElementListItems where the properties ElementListItem.Name contain the names of the fields that can be selected for import and the properties ElementListItem.ElementKind are

© 2016-2022 Altova GmbH

initialized either to *spyXMLDataAttr* or *spyXMLDataElement*, depending on the value passed in DatabaseConnection.AsAttributes. This list serves as a filter to what finally gets imported by a future call to ImportFromDatabase. Use ElementList.RemoveElement to exclude fields from import.

Properties mandatory to be filled out for the database connection are one of DatabaseConnection.File, DatabaseConnection.ADOConnection and DatabaseConnection.ODBCConnection, as well as DatabaseConnection.SQLSelect. Use the property DatabaseConnection.AsAttributes to initialize ElementListItem.ElementKind of the resulting element list to either *spyXMLDataAttr* or *spyXMLDataElement*, respectively.

Example
See example at ImportFromDatabase.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from database failed.
1112 Invalid database specified.
1114 Select statement is missing.
1119 database element list import failed.

29.3.2.1.14 GetDatabaseSettings

Method: GetDatabaseSettings() as DatabaseConnection

Description
GetDatabaseSettings creates a new object of database settings. The object is used to specify database connection parameters for the methods GetDatabaseTables, GetDatabaseImportElementList, ImportFromDatabase, ImportFromSchema and ExportToDatabase.

Example
See example of ImportFromDatabase.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.15 GetDatabaseTables

Method: GetDatabaseTables(*pImportSettings* as DatabaseConnection) as ElementList

Description
GetDatabaseTables reads the table names from the database specified in *pImportSettings*. Properties mandatory to be filled out for the database connection are one of DatabaseConnection.File, DatabaseConnection.ADOConnection and DatabaseConnection.ODBCConnection. All other properties are ignored.
The function returns a collection of ElementListItems where the properties ElementListItem.Name contain the names of tables stored in the specified database. The remaining properties of ElementListItem are unused.

Errors

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1112 Invalid database specified.
1113 Error while reading database table information.
1118 Database table query failed.

Example

```
        Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings
objImpSettings.ADOConnection = TxtADO.Text

'store table names in list box
ListTables.Clear

Dim objList As ElementList
Dim objItem As ElementListItem
On Error GoTo ErrorHandler
Set objList = objSpy.GetDatabaseTables(objImpSettings)

        For Each objItem In objList
                ListTables.AddItem objItem.Name
        Next
```

29.3.2.1.16 GetExportSettings

Method: GetExportSettings()as ExportSettings (read-only)

Description
GetExportSettings creates a new object of common export settings. This object is used to pass the parameters to the export functions and defines the behaviour of the export calls. See also the export functions from Document and the examples at Import and Export.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.17 GetTextImportElementList

Method: GetTextImportElementList(pImportSettings as TextImportExportSettings) as ElementList

Description
GetTextImportElementList retrieves importing information about the text-file as specified in pImportSettings. The function returns a collection of ElementListItems where the properties ElementListItem.Name contain the names of the fields found in the file. The values of remaining properties are undefined.

If the text-file does not contain a column header, set `pImportSettings.HeaderRow` to `false`. The resulting element list will contain general column names like 'Field1' and so on.

See also Import and export of data.

© 2016-2022 Altova GmbH

Errors

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from database failed.
1115 Error during text element list import. Cannot create parser for import file.
1116 Error during text element list import.

Example

```
' ------------------------------------------------------
' VBA client code fragment - import selected fields from text file
' ------------------------------------------------------
        Dim objImpSettings As TextImportExportSettings
        Set objImpSettings = objSpy.GetTextImportExportSettings

        objImpSettings.ImportFile = "C:\ImportMe.txt"
        objImpSettings.HeaderRow = False

        Dim objList As ElementList
        Set objList = objSpy.GetTextImportElementList(objImpSettings)

        'exclude first column
        objList.RemoveItem 1

        Dim objImpDoc As Document
        On Error Resume Next
        Set objImpDoc = objSpy.ImportFromText(objImpSettings, objList)
        CheckForError
```

29.3.2.1.18 GetTextImportExportSettings

Method: GetTextImportExportSettings() as TextImportExportSettings (read-only)

Description

GetTextImportExportSettings creates a new object of common import and export settings for text files. See also the example for Application.GetTextImportElementList and Import and Export.

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.19 GetXSLT_XQParameterCount

Method: GetXSLT_XQParameterCount() as Long

Description

Returns the number of XSLT and XQuery parameters.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.20 GetXSLT_XQParameterName

Method: GetXSLT_XQParameterName(index as Long) as String

Description
Returns the name of the XSLT or XQuery parameter identified by the supplied index.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.21 GetXSLT_XQParameterXPath

Method: GetXSLT_XQParameterXPath(index as Long) as String

Description
Returns the XPath expression of the XSLT or XQuery parameter identified by the supplied index.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.22 ImportFromDatabase

Method: ImportFromDatabase(pImportSettings as DatabaseConnection,pElementList as ElementList) as Document

Return Value
Creates a new document containing the data imported from the database.

Description
ImportFromDatabase imports data from a database as specified in pImportSettings and creates a new document containing the data imported from the database. Properties mandatory to be filled out are one of DatabaseConnection.File, DatabaseConnection.ADOConnection or DatabaseConnection.ODBCConnection and DatabaseConnection.SQLSelect. Additionally, you can use DatabaseConnection.AsAttributes, DatabaseConnection.ExcludeKeys, DatabaseConnection.IncludeEmptyElements and NumberDateTimeFormat to further parameterize import.

© 2016-2022 Altova GmbH

The parameter pElementList specifies which fields of the selected data gets written into the newly created document, and which are created as elements and which as attributes. This parameter can be NULL, specifying that all selected fields will be imported as XML elements.

See GetDatabaseSettings and GetDatabaseImportElementList for necessary steps preceding any import of data from a database.

Errors

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from database failed.
1112 Invalid database specified.
1114 Select statement is missing.
1117 Transformation to XML failed.
1120 Database import failed.

Example

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings

objImpSettings.ADOConnection = strADOConnection
objImpSettings.SQLSelect = "SELECT * FROM MyTable"

Dim objDoc As Document
On Error Resume Next
Set objDoc = objSpy.ImportFromDatabase(objImpSettings,
objSpy.GetDatabaseImportElementList(objImpSettings))
' CheckForError here
```

29.3.2.1.23 ImportFromSchema

Method: ImportFromSchema(*pImportSettings* as DatabaseConnection,*strTable* as String,*pSchemaDoc* as Document) as Document

Return Value

Creates a new document filled with data from the specified database as specified by the schema definition in *pSchemaDoc*.

Description

ImportFromSchema imports data from a database specified in pImportSettings. Properties mandatory to be filled out are one of DatabaseConnection.File, DatabaseConnection.ADOConnection or DatabaseConnection.ODBCConnection. Additionally, you can use DatabaseConnection.AsAttributes, DatabaseConnection.ExcludeKeys and NumberDateTimeFormat to further parameterize import. All other properties get ignored.

ImportFromSchema does not use an explicit SQL statement to select the data. Instead, it expects a structure definition of the document to create in form of an XML schema document in *pSchemaDoc*. From this definition the database select statement is automatically deduced. Specify in *strTable* the table name of the import root that will become the root node in the new document.

See GetDatabaseSettings and GetDatabaseTables for necessary steps preceding an import from a database based on a schema definition. To create the schema definition file use command 'create database schema' from the 'convert' menu of XMLSpy.

Errors

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from database failed.
1112 Invalid database specified.
1120 Database import failed.
1121 Could not create validator for the specified schema.
1122 Failed parsing schema for database import.

29.3.2.1.24 ImportFromText

Method: ImportFromText(*pImportSettings* as TextImportExportSettings,*pElementList* as ElementList) as Document

Description

ImportFromText imports the text file as specified in pImportSettings. The parameter pElementList can be used as import filter. Either pass the list returned by a previous call to GetTextImportElementList or null to import all columns. To avoid import of unnecessary columns use ElementList.RemoveElement to remove the corresponding field names from `pElementList` before calling `ImportFromText`.
The method returns the newly created document containing the imported data. This document is the same as the active document of XMLSpy.

See also Import and export of data.

Errors

1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1107 Import from text file failed.
1117 Transformation to XML failed.

Example

```
' ------------------------------------------------------
' VBA client code fragment - import from text file
' ------------------------------------------------------
        Dim objImpSettings As TextImportExportSettings
        Set objImpSettings = objSpy.GetTextImportExportSettings

        objImpSettings.ImportFile = strFileName
        objImpSettings.HeaderRow = False

        Dim objImpDoc As Document
        On Error Resume Next
        Set objImpDoc = objSpy.ImportFromText(objImpSettings,
objSpy.GetTextImportElementList(objImpSettings))

        CheckForError
```

 © 2016-2022 Altova GmbH

29.3.2.1.25 ImportFromWord

Method: ImportFromWord(*strFile* as String) as Document

Description
ImportFromWord imports the MS-Word Document strFile into a new XML document.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
Import from document failed.

29.3.2.1.26 IsAPISupported

Property: IsAPISupported as Boolean

Description
Returns whether the API is supported in this version or not.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.27 MajorVersion

Property: MajorVersion as Integer

Description
Returns the application version's major number, for example `15` for 2013 versions, and `16` for 2014 versions..

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.28 MinorVersion

Property: MinorVersion as Integer

Description
Returns the application version's minor number.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.29 NewProject

Method: NewProject(*strPath* as String,*bDiscardCurrent* as Boolean)

Description
NewProject creates a new project.

If there is already a project open that has been modified and bDiscardCurrent is false, then NewProject() fails.

Errors
1111 The application object is no longer valid.
1102 A project is already open but *bDiscardCurrent* is *true*.
1103 Creation of new project failed.

29.3.2.1.30 OpenProject

Method: OpenProject(*strPath* as String,*bDiscardCurrent* as Boolean,*bDialog* as Boolean)

Parameters
strPath
Path and file name of the project to open. Can be empty if bDialog is true.

bDiscardCurrent
Discard currently open project and possibly lose changes.

bDialog
Show dialogs for user input.

Return Value
None

Description
OpenProject opens an existing project. If there is already a project open that has been modified and bDiscardCurrent is false, then OpenProject() fails.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.
1101 Cannot open specified project.
1102 A project is already open but *bDiscardCurrent* is *true*.

29.3.2.1.31 Parent

Property: Parent as Application (read-only)

Description

© 2016-2022 Altova GmbH

Accesses the XMLSpy application object.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.32 Quit

Method: Quit()

Return Value
None

Description
This method terminates XMLSpy. All modified documents will be closed without saving the changes. This is also true for an open project.

If XMLSpy was automatically started as an automation server by a client program, the application will not shut down automatically when your client program shuts down if a project or any document is still open. Use the Quit method to ensure automatic shut-down.

Errors
1111 The application object is no longer valid.

29.3.2.1.33 ReloadSettings

Method: ReloadSettings

Return Value

Description
The application settings are reloaded from the registry.

Available with TypeLibrary version 1.5

Errors
1111 The application object is no longer valid.

29.3.2.1.34 RemoveXSLT_XQParameter

Method: RemoveXSLT_XQParameter(index as Long)

Description
Removes the XSLT or XQuery parameter identified by the supplied index.

Errors
1111 The application object is no longer valid.

1100 Invalid address for the return parameter was specified.

29.3.2.1.35 RunMacro

Method: RunMacro(*strMacro* as String)

Return Value

Description
Calls the specified macro either from the project scripts (if present) or from the global scripts.

Available with TypeLibrary version 1.5

Errors
1111 The application object is no longer valid.

29.3.2.1.36 ScriptingEnvironment

Property: ScriptingEnvironment as IUnknown (read-only)

Description
Reference to any active scripting environment. This property makes it possible to access the TypeLibrary of the XMLSpyFormEditor.exe application which is used as the current scripting environment.

Available with TypeLibrary version 1.5

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.37 ServicePackVersion

Property: ServicePackVersion as Long

Description
Returns the Service Pack version number of the application. Eg: 1 for 2010 R2 SP1

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.38 ShowApplication

Method: ShowApplication(*bShow* as Boolean)

Return Value

© 2016-2022 Altova GmbH

None

Description
The method shows (bShow = True) or hides (bShow = False) XMLSpy.

Errors
1110 The application object is no longer valid.

29.3.2.1.39 ShowFindInFiles

Method: ShowFindInFiles(pSettings as FindInFilesDlg) as Boolean

Return Value
Returns false if the user pressed the Cancel button, true otherwise.

Description
Displays the FindInFiles dialog preset with the given settings. The user modifications of the settings are stored in the passed dialog object.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.1.40 ShowForm

Method: ShowForm(*strFormName* as String) as Long

Return Value
Returns zero if the user pressed a Cancel button or the form calls TheView.Cancel().

Description
Displays the form strFormName.

Forms, event handlers and macros can be created with the Scripting Environment. Select "Switch to scripting environment" from the **Tools** menu to invoke the Scripting Environment.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.1.41 Status

Property: Status as ENUMApplicationStatus

Description
Returns the current status of the running application.

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.42 URLDelete

Method: URLDelete(*strURL* as String,*strUser* as String,*strPassword* as String)

Return Value
None

Description
The method deletes the file at the URL strURL.

Errors
1111 The application object is no longer valid.
1109 Error deleting file at specified URL.

29.3.2.1.43 URLMakeDirectory

Method: URLMakeDirectory(*strURL* as String,*strUser* as String,*strPassword* as String)

Return Value
None

Description
The method creates a new directory at the URL strURL.

Errors
1111 The application object is no longer valid.
1100 Invalid parameter specified.

29.3.2.1.44 Visible

Property: Visible as VARIANT_BOOL

Description
Sets or gets the visibility attribute of XMLSpy. This standard automation property makes usage of ShowApplication obsolete.

Errors
1110 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.1.45 WarningNumber

Property: WarningNumber as integer

Description
Some methods fill the property WarningNumber with additional information if an error occurs.

Currently just Documents.OpenFile fills this property.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.1.46 WarningText

Property: WarningText as String

Description
Some methods fill the property WarningText with additional information if an error occurs.

Currently just Documents.OpenFile fills this property.

Errors
1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

29.3.2.2 AuthenticContextMenu

The context menu interface provides the means for the user to customize the context menus shown in Authentic. The interface has the methods listed in this section.

29.3.2.2.1 CountItems

Method: CountItems () nItems as long

Return Value
Returns the number of menu items.

Errors
2501 Invalid object.

29.3.2.2.2 DeleteItem

Method: DeleteItem `(IndexPosition as long)`

Return Value
Deletes the menu item that has the index position submitted in the first parameter.

Errors
2501 Invalid object
2502 Invalid index

29.3.2.2.3 GetItemText

Method: GetItemText `(IndexPosition as long)` MenuItemName as string

Return Value
Gets the name of the menu item located at the index position submitted in the first parameter.

Errors
2501 Invalid object
2502 Invalid index

29.3.2.2.4 InsertItem

Method: InsertItem `(IndexPosition as long, MenuItemName as string, MacroName as string)`

Return Value
Inserts a user-defined menu item at the position in the menu specified in the first parameter and having the name submitted in the second parameter. The menu item will start a macro, so a valid macro name must be submitted.

Errors
2501 Invalid object
2502 Invalid index
2503 No such macro
2504 Internal error

29.3.2.2.5 SetItemText

Method: SetItemText `(IndexPosition as long, MenuItemName as string)`

© 2016-2022 Altova GmbH

Return Value
Sets the name of the menu item located at the index position submitted in the first parameter.

Errors
2501 Invalid object
2502 Invalid index

29.3.2.3 AuthenticDataTransfer

Renamed from DocEditDataTransfer to AuthenticDataTransfer

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage—except for AuthenticDataTransfer—is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces. All functionality available up to now in DocEditView, DocEditSelection, DocEditEvent and DocEditDataTransfer is now available via AuthenticView, AuthenticRange and AuthenticDataTransfer. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Methods

getData

Properties

dropEffect
ownDrag
type

Description

The events OnDragOver and OnBeforeDrop provide information about the object being dragged with an instance of type AuthenticDataTransfer. It contains a description of the dragged object and its content. The latter is available either as string or a pointer to a COM object supporting the IUnkown interface.

29.3.2.3.1 dropEffect

Property: dropEffect as long

Description

The property stores the drop effect from the default event handler. You can set the drop effect if you change this value and return TRUE for the event handler (or set AuthenticEvent.cancelBubble to TRUE if you are still using the now obsolete AuthenticEvent interface).

Errors
2101 Invalid address for the return parameter was specified.

29.3.2.3.2 getData

Method: getData() as Variant

Description
Retrieve the data associated with the dragged object. Depending on AuthenticDataTransfer.type, that data is either a string or a COM interface pointer of type IUnknown.

Errors
2101 Invalid address for the return parameter was specified.

29.3.2.3.3 ownDrag

Property: ownDrag as Boolean (read-only)

Description
The property is TRUE if the current dragging source comes from inside Authentic View.

Errors
2101 Invalid address for the return parameter was specified.

29.3.2.3.4 type

Property: type as String (read-only)

Description
Holds the type of data you get with the DocEditDataTransfer.getData method.

Currently supported data types are:

OWN	data from Authentic View itself
TEXT	plain text
UNICODETEXT	plain text as UNICODE

Errors
2101 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.4 AuthenticEventContext

The `EventContext` interface gives access to many properties of the context in which a macro is executed.

29.3.2.4.1 EvaluateXPath

Method: EvaluateXPath (strExpression as string) as strValue as string

Return Value
The method evaluates the XPath expression in the context of the node within which the event was triggered and returns a string.

Description
`EvaluateXPath()` executes an XPath expression with the given event context. The result is returned as a string, in the case of a sequence it is a space-separated string.

Errors
2201 Invalid object.
2202 No context.
2209 Invalid parameter.
2210 Internal error.
2211 XPath error.

29.3.2.4.2 GetEventContextType

Method: GetEventContextType `()` Type as `AuthenticEventContextType` enumeration

Return Value
Returns the context node type.

Description
`GetEventContextType` allows the user to determine whether the macro is in an XML node or in an XPath atomic item context. The enumeration `AuthenticEventContextType` is defined as follows:

authenticEventContextXML,
authenticEventContextAtomicItem,
authenticEventContextOther

If the context is a normal XML node, the `GetXMLNode()` function gives access to it (returns `NULL` if not).

Errors
2201 Invalid object.
2202 No context.
2209 Invalid parameter.

29.3.2.4.3 GetNormalizedTextValue

Method: GetNormalizedTextValue() strValue as string

Return Value
Returns the value of the current node as string

Errors
2201 Invalid object.
2202 No context.
2203 Invalid context
2209 Invalid parameter.

29.3.2.4.4 GetVariableValue

Method: GetVariableValue(strName as string) strValue as string

Return Value
Gets the value of the variable submitted as the parameter.

Description
GetVariableValue gets the variable's value in the scope of the context.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

Errors
2201 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2209 Invalid parameter

29.3.2.4.5 GetXMLNode

Method: GetXMLNode() Node as XMLData object

Return Value
Returns the context XML node or NULL

Errors

© 2016-2022 Altova GmbH

2201 Invalid object.
2202 No context.
2203 Invalid context
2209 Invalid parameter.

29.3.2.4.6 IsAvailable

Method: IsAvailable`()` as Boolean

Return Value
Returns true if `EventContext` is set, false otherwise.

Errors
2201 Invalid object.

29.3.2.4.7 SetVariableValue

Method: SetVariableValue`(strName as string, strValue as string)`

Return Value
Sets the value (second parameter) of the variable submitted in the first parameter.

Description
`SetVariableValue` sets the variable's value in the scope of the context.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

Errors
2201 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2207 Variable read-only
2208 No modification allowed

29.3.2.5 AuthenticRange

The first table lists the properties and methods of AuthenticRange that can be used to navigate through the document and select specific portions.

Properties	Methods	
Application	Clone	MoveBegin
FirstTextPosition	CollapsToBegin	MoveEnd
FirstXMLData	CollapsToEnd	NextCursorPosition
FirstXMLDataOffset	ExpandTo	PreviousCursorPosition
LastTextPosition	Goto	Select
LastXMLData	GotoNext	SelectNext
LastXMLDataOffset	GotoPrevious	SelectPrevious
Parent	IsEmpty	SetFromRange
	IsEqual	

The following table lists the content modification methods, most of which can be found on the right/button mouse menu.

Properties	Edit operations	Dynamic table operations
Text	Copy	AppendRow
	Cut	DeleteRow
	Delete	DuplicateRow
	IsCopyEnabled	InsertRow
	IsCutEnabled	IsFirstRow
	IsDeleteEnabled	IsInDynamicTable
	IsPasteEnabled	IsLastRow
	Paste	MoveRowDown
		MoveRowUp

The following methods provide the functionality of the Authentic entry helper windows for range objects.

Operations of the entry helper windows

Elements	Attributes	Entities
CanPerformActionWith	GetElementAttributeValue	GetEntityNames
CanPerformAction	GetElementAttributeNames	InsertEntity
PerformAction	GetElementHierarchy	
	HasElementAttribute	
	IsTextStateApplied	
	SetElementAttributeValue	

Description
AuthenticRange objects are the 'cursor' selections of the automation interface. You can use them to point to any cursor position in the Authentic view, or select a portion of the document. The operations available for AuthenticRange objects then work on this selection in the same way, as the corresponding operations of the user interface do with the current user interface selection. The main difference is that you can use an arbitrary

© 2016-2022 Altova GmbH

number of AuthenticRange objects at the same time, whereas there is exactly one cursor selection in the user interface.

To get to an initial range object use AuthenticView.Selection, to obtain a range corresponding with the current cursor selection in the user interface. Alternatively, some trivial ranges are accessible via the read/only properties AuthenticView.DocumentBegin, AuthenticView.DocumentEnd, and AuthenticView.WholeDocument. The most flexible method is AuthenticView.Goto, which allows navigation to a specific portion of the document within one call. For more complex selections, combine the above with the various navigation methods on range objects listed in the first table on this page.

Another method to select a portion of the document is to use the position properties of the range object. Two positioning systems are available and can be combined arbitrarily:

- **Absolute** text cursor positions, starting with position 0 at the document beginning, can be set and retrieved for the beginning and end of a range. For more information see FirstTextPosition and LastTextPosition. This method requires complex internal calculations and should be used with care.

- The **XMLData** element and a text position inside this element, can be set and retrieved for the beginning and end of a range. For more information see FirstXMLData, FirstXMLDataOffset, LastXMLData, and LastXMLDataOffset. This method is very efficient but requires knowledge of the underlying document structure. It can be used to locate XMLData objects and perform operations on them otherwise not accessible through the user interface.

Modifications to the document content can be achieved by various methods:

- The Text property allows you to retrieve the document text selected by the range object. If set, the selected document text gets replaced with the new text.
- The standard document edit functions Cut, Copy, Paste and Delete.
- Table operations for tables that can grow dynamically.
- Methods that map the functionality of the Authentic entry helper windows.
- Access to the XMLData objects of the underlying document to modify them directly.

29.3.2.5.1 AppendRow

Method: AppendRow()as Boolean

Description
If the beginning of the range is inside a dynamic table, this method inserts a new row at the end of the selected table. The selection of the range is modified to point to the beginning of the new row. The function returns *true* if the append operation was successful, otherwise *false*.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------------------
' Scripting environment - VBScript
' Append row at end of current dynamically growable table
' ----------------------------------------------------
```

```
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
        objRange.AppendRow
        ' objRange points to beginning of new row
        objRange.Select
End If
```

29.3.2.5.2 Application

Property: Application as Application (read-only)

Description
Accesses the XMLSpy application object.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.3 CanPerformAction

Method: CanPerformAction (*eAction* as SPYAuthenticActions, *strElementName* as String) as Boolean

Description
CanPerformAction and its related methods enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content, without having to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If the location can be found, the method returns *True*, otherwise it returns *False*.

HINT: To find out all valid element names for a given action, use CanPerformActionWith.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.
2007 Invalid action was specified.

Examples
See PerformAction.

29.3.2.5.4 CanPerformActionWith

Method: CanPerformActionWith (*eAction* as SPYAuthenticActions, *out_arrElementNames* as Variant)

Description

© 2016-2022 Altova GmbH

PerformActionWith and its related methods, enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without having to know exactly where the modification will take place.

This method returns an array of those element names that the specified action can be performed with.

HINT: To apply the action use CanPerformActionWith.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.
2007 Invalid action was specified.

Examples

See PerformAction.

29.3.2.5.5 Clone

Method: Clone() as AuthenticRange

Description

Returns a copy of the range object.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.6 CollapsToBegin

Method: CollapsToBegin() as AuthenticRange

Description

Sets the end of the range object to its begin. The method returns the modified range object.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.7 CollapsToEnd

Method: CollapsToEnd() as AuthenticRange

Description

Sets the beginning of the range object to its end. The method returns the modified range object.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.8 Copy

Method: Copy() as Boolean

Description
Returns *False* if the range contains no portions of the document that may be copied.
Returns *True* if text, and in case of fully selected XML elements the elements as well, has been copied to the copy/paste buffer.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.9 Cut

Method: Cut() as Boolean

Description
Returns *False* if the range contains portions of the document that may not be deleted.
Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document and saved in the copy/paste buffer.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.10 Delete

Method: Delete() as Boolean

Description
Returns *False* if the range contains portions of the document that may not be deleted.
Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.11 DeleteRow

Method: DeleteRow() as Boolean

© 2016-2022 Altova GmbH

Description
If the beginning of the range is inside a dynamic table, this method deletes the selected row. The selection of the range gets modified to point to the next element after the deleted row. The function returns *true,* if the delete operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples
```
' ---------------------------------------------------------
' Scripting environment - VBScript
' Delete selected row from dynamically growing table
' ---------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we are in a table
If objRange.IsInDynamicTable Then
        objRange.DeleteRow
End If
```

29.3.2.5.12 DuplicateRow

Method: DuplicateRow() as Boolean

Description
If the beginning of the range is inside a dynamic table, this method inserts a duplicate of the current row after the selected one. The selection of the range gets modified to point to the beginning of the new row. The function returns *true* if the duplicate operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples
```
' ---------------------------------------------------------
' Scripting environment - VBScript
' duplicate row in current dynamically growable table
' ---------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
        objRange.DuplicateRow
        ' objRange points to beginning of new row
        objRange.Select
End If
```

29.3.2.5.13 EvaluateXPath

Method: EvaluateXPath (strExpression as string) strValue as string

Return Value
The method returns a string

Description
`EvaluateXPath()` executes an XPath expression with the context node being the beginning of the range selection. The result is returned as a string, in the case of a sequence it is a space-separated string. If XML context node is irrelevant, the user may provide any node, like `AuthenticView.XMLDataRoot`.

Errors

2001	Invalid object
2005	Invalid parameter
2008	Internal error
2202	Missing context node
2211	XPath error

29.3.2.5.14 ExpandTo

Method: ExpandTo (*eKind* as SPYAuthenticElementKind), as AuthenticRange

Description
Selects the whole element of type eKind, that starts at, or contains, the first cursor position of the range. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Range expansion would be beyond end of document.
2005	Invalid address for the return parameter was specified.

29.3.2.5.15 FirstTextPosition

Property: FirstTextPosition as Long

Description
Set or get the left-most text position index of the range object. This index is always less or equal to LastTextPosition. Indexing starts with 0 at document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decrementing the test position by 1 has the same effect as the cursor-left key.

If you set FirstTextPosition to a value greater than the current LastTextPosition, LastTextPosition gets set to the new FirstTextPosition.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

© 2016-2022 Altova GmbH

Errors

2001 The authentic range object, or its related view object is not valid.
2005 Invalid address for the return parameter was specified.
2006 A text position outside the document was specified.

Examples

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.isEqual(objAuthenticView.WholeDocument) Then
        MsgBox "Test using direct text cursor positioning was ok"
Else
        MsgBox "Ooops!"
End If
```

29.3.2.5.16 FirstXMLData

Property: FirstXMLData as XMLData

Description

Set or get the first XMLData element in the underlying document that is partially, or completely selected by the range. The exact beginning of the selection is defined by the FirstXMLDataOffset attribute.

Whenever you set FirstXMLData to a new data object, FirstXMLDataOffset gets set to the first cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set FirstXMLData / FirstXMLDataOffset selects a position greater then the current LastXMLData / LastXMLDataOffset, the latter gets moved to the new start position.

HINT: You can use the FirstXMLData and LastXMLData properties to directly access and manipulate the underlying XML document in those cases where the methods available with the AuthenticRange object are not sufficient.

Errors

2001 The authentic range object, or its related view object is not valid.

2005 Invalid address for the return parameter was specified.
2008 Internal error
2009 The XMLData object cannot be accessed.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' show name of currently selected XMLData element
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objXmlData
Set objXMLData = objAuthenticView.Selection.FirstXMLData
' authentic view adds a 'text' child element to elements
' of the document which have content. So we have to go one
' element up.
Set objXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name
```

29.3.2.5.17 FirstXMLDataOffset

Property: FirstXMLDataOffset as Long

Description

Set or get the cursor position offset inside FirstXMLData element for the beginning of the range. Offset positions are based on the characters returned by the Text property, and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on screen. Although the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.

- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If FirstXMLData / FirstXMLDataOffset selects a position after the current LastXMLData / LastXMLDataOffset, the latter gets moved to the new start position.

Errors

2001 The authentic range object, or its related view object is not valid.
2005 Invalid offset was specified.
Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
```


© 2016-2022 Altova GmbH

```
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0  ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData  ' select only one element
objRange.LastXMLDataOffset = -1  ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
        objRange.Select()
Else
        MsgBox "Oops"
End If
```

29.3.2.5.18 GetElementAttributeNames

Method: GetElementAttributeNames (*strElementName* as String, *out_arrAttributeNames* as Variant)

Description
Retrieve the names of all attributes for the enclosing element with the specified name. Use the element/attribute pairs, to set or get the attribute value with the methods GetElementAttributeValue and SetElementAttributeValue.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2005	Invalid element name was specified.
	Invalid address for the return parameter was specified.

Examples
See SetElementAttributeValue.

29.3.2.5.19 GetElementAttributeValue

Method: GetElementAttributeValue (*strElementName* as String, *strAttributeName* as String) as String

Description
Retrieve the value of the attribute specified in strAttributeName, for the element identified with strElementName. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use GetElementAttributeNames, or HasElementAttribute.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid element name was specified.
Invalid attribute name was specified.
Invalid address for the return parameter was specified.

Examples

See SetElementAttributeValue.

29.3.2.5.20 GetElementHierarchy

Method: GetElementHierarchy (*out_arrElementNames* as Variant)

Description

Retrieve the names of all XML elements that are parents of the current selection. Inner elements get listed before enclosing elements. An empty list is returned whenever the current selection is not inside a single XMLData element.

The names of the element hierarchy, together with the range object uniquely identify XMLData elements in the document. The attributes of these elements can be directly accessed by GetElementAttributeNames, and related methods.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

C# Examples

```
' ----------------------------------------
' C#
' ----------------------------------------

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            XMLSpyLib.Application app = new XMLSpyLib.Application();

            app.ShowApplication(true);

            XMLSpyLib.AuthenticView view = app.ActiveDocument.AuthenticView;
            XMLSpyLib.AuthenticRange range = view.DocumentBegin;

            object o = null;
            range.GetElementHierarchy(ref o);

            object[] elements = (object[])o;
```


© 2016-2022 Altova GmbH

```
            foreach (string e in elements)
            {
                Console.WriteLine(e);
            }
        }
    }
}
```

Also see: SetElementAttributeValue.

29.3.2.5.21 GetEntityNames

Method: GetEntityNames (*out_arrEntityNames* as Variant)

Description
Retrieve the names of all defined entities. The list of retrieved entities is independent of the current selection, or location. Use one of these names with the InsertEntity function.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples
See: GetElementHierarchy and InsertEntity.

29.3.2.5.22 GetVariableValue

Method: GetVariableValue`(strName as string) strVal as string`

Return Value
Gets the value of the variable named as the method's parameter.

Errors
2001 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2209 Invalid parameter

29.3.2.5.23 Goto

Method: Goto (*eKind* as SPYAuthenticElementKind, *nCount* as Long, *eFrom* as SPYAuthenticDocumentPosition) as AuthenticRange

Description
Sets the range to point to the beginning of the nCount element of type eKind. The start position is defined by the parameter eFrom.

Use positive values for nCount to navigate to the document end. Use negative values to navigate to the beginning of the document. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2004	Target lies before begin of document.
2005	Invalid element kind specified.
	Invalid start position specified.
	Invalid address for the return parameter was specified.

29.3.2.5.24 GotoNext

Method: GotoNext (*eKind* as SPYAuthenticElementKind) as AuthenticRange

Description
Sets the range to the beginning of the next element of type eKind. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document word-by-word
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
	objRange.GotoNext(spyAuthenticWord).Select
	If ((Err.number - vbObjecterror) = 2003) Then
		bEndOfDocument = True
		Err.Clear
	ElseIf (Err.number <> 0) Then
```


© 2016-2022 Altova GmbH

```
            Err.Raise  ' forward error
        End If
Wend
```

29.3.2.5.25 GotoNextCursorPosition

Method: GotoNextCursorPosition() as AuthenticRange

Description
Sets the range to the next cursor position after its current end position. Returns the modified object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2005	Invalid address for the return parameter was specified.

29.3.2.5.26 GotoPrevious

Method: GotoPrevious (*eKind* as SPYAuthenticElementKind) as AuthenticRange

Description
Sets the range to the beginning of the element of type eKind which is before the beginning of the current range. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2004	Target lies before beginning of document.
2005	Invalid element kind specified. Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
        objRange.GotoPrevious(spyAuthenticTag).Select
        If ((Err.number - vbObjecterror) = 2004) Then
                bBeginOfDocument = True
                Err.Clear
```

```
        ElseIf (Err.number <> 0) Then
                Err.Raise  ' forward error
        End If
Wend
```

29.3.2.5.27 GotoPreviousCursorPosition

Method: GotoPreviousCursorPosition() as AuthenticRange

Description
Set the range to the cursor position immediately before the current position. Returns the modified object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2004	Target lies before begin of document.
2005	Invalid address for the return parameter was specified.

29.3.2.5.28 HasElementAttribute

Method: HasElementAttribute (*strElementName* as String, *strAttributeName* as String) as Boolean

Description
Tests if the enclosing element with name strElementName, supports the attribute specified in strAttributeName.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2005	Invalid element name was specified. Invalid address for the return parameter was specified.

29.3.2.5.29 InsertEntity

Method: InsertEntity (*strEntityName* as String)

Description
Replace the ranges selection with the specified entity. The specified entity must be one of the entity names returned by GetEntityNames.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2005	Unknown entry name was specified.

Examples

```
' ----------------------------------------------------
' Scripting environment - VBScript
' Insert the first entity in the list of available entities
' ----------------------------------------------------
Dim objRange
```


© 2016-2022 Altova GmbH

```
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we get the names of all available entities as they
' are shown in the entry helper of XMLSpy
Dim arrEntities
objRange.GetEntityNames arrEntities

' we insert the first one of the list
If UBound(arrEntities) >= 0 Then
        objRange.InsertEntity arrEntities(0)
Else
        MsgBox "Sorry, no entities are available for this document"
End If
```

29.3.2.5.30 InsertRow

Method: InsertRow() as Boolean

Description
If the beginning of the range is inside a dynamic table, this method inserts a new row before the current one. The selection of the range gets modified to point to the beginning of the newly inserted row. The function returns *true* if the insert operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------------------------
' Scripting environment - VBScript
' Insert row at beginning of current dynamically growing table
' ----------------------------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
        objRange.InsertRow
        ' objRange points to beginning of new row
        objRange.Select
  End If
```

29.3.2.5.31 IsCopyEnabled

Property: IsCopyEnabled as Boolean (read-only)

Description
Checks if the copy operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.32 IsCutEnabled

Property: IsCutEnabled as Boolean (read-only)

Description
Checks if the cut operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.33 IsDeleteEnabled

Property: IsDeleteEnabled as Boolean (read-only)

Description
Checks if the delete operation is supported for this range.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.34 IsEmpty

Method: IsEmpty() as Boolean

Description
Tests if the first and last position of the range are equal.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.35 IsEqual

Method: IsEqual (*objCmpRange* as AuthenticRange) as Boolean

Description
Tests if the start and end of both ranges are the same.

Errors
2001 One of the two range objects being compared, is invalid.

© 2016-2022 Altova GmbH

2005 Invalid address for a return parameter was specified.

29.3.2.5.36 IsFirstRow

Property: IsFirstRow as Boolean (read-only)

Description
Test if the range is in the first row of a table. Which table is taken into consideration depends on the extend of the range. If the selection exceeds a single row of a table, the check is if this table is the first element in an embedding table. See the entry helpers of the user manual for more information.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.37 IsInDynamicTable

Method: IsInDynamicTable() as Boolean

Description
Test if the whole range is inside a table that supports the different row operations like 'insert', 'append', duplicate, etc.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.38 IsLastRow

Property: IsLastRow as Boolean (read-only)

Description
Test if the range is in the last row of a table. Which table is taken into consideration depends on the extend of the range. If the selection exceeds a single row of a table, the check is if this table is the last element in an embedding table. See the entry helpers of the user manual for more information.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.39 IsPasteEnabled

Property: IsPasteEnabled as Boolean (read-only)

Description
Checks if the paste operation is supported for this range.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.40 IsSelected

Property: IsSelected as Boolean

Description

Returns true() if selection is present. The selection range still can be empty: that happens when e.g. only the cursor is set.

29.3.2.5.41 IsTextStateApplied

Method: IsTextStateApplied (*i_strElementName* as String) as Boolean

Description

Checks if all the selected text is embedded into an XML Element with name i_strElementName. Common examples for the parameter i_strElementName are "strong", "bold" or "italic".

Errors

2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.42 LastTextPosition

Property: LastTextPosition as Long

Description

Set or get the rightmost text position index of the range object. This index is always greater or equal to FirstTextPosition. Indexing starts with 0 at the document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decreasing the test position by 1 has the same effect as the cursor-left key.

If you set LastTextPosition to a value less then the current FirstTextPosition, FirstTextPosition gets set to the new LastTextPosition.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

Errors

2001 The authentic range object, or its related view object is not valid.
2005 Invalid address for the return parameter was specified.
2006 A text position outside the document was specified.

Examples

© 2016-2022 Altova GmbH

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.isEqual(objAuthenticView.WholeDocument) Then
        MsgBox "Test using direct text cursor positioning was ok"
Else
        MsgBox "Oops!"
End If
```

29.3.2.5.43 LastXMLData

Property: LastXMLData as XMLData

Description
Set or get the last XMLData element in the underlying document that is partially or completely selected by the range. The exact end of the selection is defined by the LastXMLDataOffset attribute.

Whenever you set LastXMLData to a new data object, LastXMLDataOffset gets set to the last cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set LastXMLData / LastXMLDataOffset, select a position less then the current FirstXMLData / FirstXMLDataOffset, the latter gets moved to the new end position.

HINT: You can use the FirstXMLData and LastXMLData properties to directly access and manipulate the underlying XML document in those cases, where the methods available with the AuthenticRange object are not sufficient.

Errors

2001	The authentic range object, or its related view object is not valid.
2005	Invalid address for the return parameter was specified.
2008	Internal error
2009	The XMLData object cannot be accessed.

29.3.2.5.44 LastXMLDataOffset

Property: LastXMLDataOffset as Long

Description
Set or get the cursor position inside LastXMLData element for the end of the range.

Offset positions are based on the characters returned by the Text property and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on the screen. Although, the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If LastXMLData / LastXMLDataOffset selects a position before FirstXMLData / FirstXMLDataOffset, the latter gets moved to the new end position.

Errors

2001	The authentic range object, or its related view object is not valid.
2005	Invalid offset was specified.
	Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------------
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' ---------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0  ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData  ' select only one element
objRange.LastXMLDataOffset = -1  ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
```


© 2016-2022 Altova GmbH

```
        objRange.Select()
Else
        MsgBox "Ooops"
End If
```

29.3.2.5.45 MoveBegin

Method: MoveBegin (*eKind* as SPYAuthenticElementKind, *nCount* as Long) as AuthenticRange

Description
Move the beginning of the range to the beginning of the nCount element of type eKind. Counting starts at the current beginning of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The end of the range stays unmoved, unless the new beginning would be larger than it. In this case, the end is moved to the new beginning. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2004	Target lies before beginning of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

29.3.2.5.46 MoveEnd

Method: MoveEnd (*eKind* as SPYAuthenticElementKind, *nCount* as Long) as AuthenticRange

Description
Move the end of the range to the begin of the nCount element of type eKind. Counting starts at the current end of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The beginning of the range stays unmoved, unless the new end would be less than it. In this case, the beginning gets moved to the new end. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2004	Target lies before begin of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

29.3.2.5.47 MoveRowDown

Method: MoveRowDown() as Boolean

Description

If the beginning of the range is inside a dynamic table and selects a row which is not the last row in this table, this method swaps this row with the row immediately below. The selection of the range moves with the row, but does not otherwise change. The function returns *true* if the move operation was successful, otherwise *false*.

Errors
2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.48 MoveRowUp

Method: MoveRowUp() as Boolean

Description
If the beginning of the range is inside a dynamic table and selects a row which is not the first row in this table, this method swaps this row with the row above. The selection of the range moves with the row, but does not change otherwise. The function returns *true* if the move operation was successful, otherwise *false*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.49 Parent

Property: Parent as AuthenticView (read-only)

Description
Access the view that owns this range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.5.50 Paste

Method: Paste() as Boolean

Description
Returns *False* if the copy/paste buffer is empty, or its content cannot replace the current selection.

Otherwise, deletes the current selection, inserts the content of the copy/paste buffer, and returns *True*.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.5.51 PerformAction

Method: PerformAction (*eAction* as SPYAuthenticActions, *strElementName* as String) as Boolean

Description
PerformAction and its related methods, give access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without a need to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If no such location can be found, the method returns *False*. Otherwise, the document gets modified and the range points to the beginning of the modification.

HINT: To find out element names that can be passed as the second parameter use CanPerformActionWith.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2005 Invalid address for the return parameter was specified.
2007 Invalid action was specified.

Examples
```
' ----------------------------------------
' Scripting environment - VBScript
' Insert the innermost element
' ----------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' we determine the elements that can be inserted at the current position
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
If UBound(arrElements) >= 0 Then
        objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
        ' objRange now points to the beginning of the inserted element
        ' we set a default value and position at its end
        objRange.Text = "Hello"
        objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
        MsgBox "Can't insert any elements at current position"
End If
```

29.3.2.5.52 Select

Method: Select()

Description
Makes this range the current user interface selection. You can achieve the same result using:
'*objRange.Parent.Selection = objRange*'

Errors
2001 The authentic range object or its related view object is no longer valid.

Examples
```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' set current selection to end of document
objAuthenticView.DocumentEnd.Select()
```

29.3.2.5.53 SelectNext

Method: SelectNext (*eKind* as SPYAuthenticElementKind) as AuthenticRange

Description
Selects the element of type eKind after the current end of the range. The method returns the modified range object.

Errors
2001 The authentic range object, or its related view object is no longer valid.
2003 Target lies after end of document.
2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples
```
' ---------------------------------------------
' Scripting environment - VBScript
' Scan through the whole document word-by-word
' ---------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
        objRange.SelectNext(spyAuthenticWord).Select
        If ((Err.number - vbObjecterror) = 2003) Then
                bEndOfDocument = True
                Err.Clear
        ElseIf (Err.number <> 0) Then
                Err.Raise  ' forward error
        End If
```


© 2016-2022 Altova GmbH

```
Wend
```

29.3.2.5.54 SelectPrevious

Method: GotoPrevious (*eKind* as SPYAuthenticElementKind) as AuthenticRange

Description
Selects the element of type eKind before the current beginning of the range. The method returns the modified range object.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2004	Target lies before begin of document.
2005	Invalid element kind specified.
	Invalid address for the return parameter was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' ----------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
        objRange.SelectPrevious(spyAuthenticTag).Select
        If ((Err.number - vbObjecterror) = 2004) Then
                bBeginOfDocument = True
                Err.Clear
        ElseIf (Err.number <> 0) Then
                Err.Raise  ' forward error
        End If
Wend
```

29.3.2.5.55 SetElementAttributeValue

Method: SetElementAttributeValue (*strElementName* as String, *strAttributeName* as String, *strAttributeValue* as String)

Description

Set the value of the attribute specified in strAttributeName for the element identified with strElementName. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use GetElementAttributeNames, or HasElementAttribute.

Errors

2001 The authentic range object or its related view object is no longer valid.
2005 Invalid element name was specified.
Invalid attribute name was specified.
Invalid attribute value was specified.

Examples

```
' ----------------------------------------
' Scripting environment - VBScript
' Get and set element attributes
' ----------------------------------------
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we find out all the elements below the beginning of the range
Dim arrElements
objRange.GetElementHierarchy arrElements

If IsArray(arrElements) Then
	If UBound(arrElements) >= 0 Then
		' we use the top level element and find out its valid attributes
		Dim arrAttrs()
		objRange.GetElementAttributeNames arrElements(0), arrAttrs

		If UBound(arrAttrs) >= 0 Then
			' we retrieve the current value of the first valid attribute
			Dim strAttrVal
			strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))
			msgbox "current value of " & arrElements(0) & "//" & arrAttrs(0) & " is: " & strAttrVal

			' we change this value and read it again
			strAttrVal = "Hello"
			objRange.SetElementAttributeValue arrElements(0), arrAttrs(0), strAttrVal
			strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))
			msgbox "new value of " & arrElements(0) & "//" & arrAttrs(0) & " is: " & strAttrVal
		End If
	End If
End If
```

29.3.2.5.56 SetFromRange

Method: SetFromRange (*objSrcRange* as AuthenticRange)

Description

© 2016-2022 Altova GmbH

Sets the range object to the same beginning and end positions as objSrcRange.

Errors

2001 One of the two range objects, is invalid.
2005 Null object was specified as source object.

29.3.2.5.57 SetVariableValue

Method: SetVariableValue`(strName as string, strValue as string)`

Return Value
Sets the value (second parameter) of the variable named in the first parameter.

Errors

2201 Invalid object.
2202 No context.
2204 No such variable in scope
2205 Variable cannot be evaluated
2206 Variable returns sequence
2207 Variable read-only
2208 No modification allowed

29.3.2.5.58 Text

Property: Text as String

Description
Set or get the textual content selected by the range object.

The number of characters retrieved are not necessarily identical, as there are text cursor positions between the beginning and end of the selected range. Most document elements support an end cursor position different to the beginning cursor position of the following element. Drop-down lists maintain only one cursor position, but can select strings of any length. In the case of radio buttons and check boxes, the text property value holds the string of the corresponding XML element.

If the range selects more then one element, the text is the concatenation of the single texts. XML entities are expanded so that '&' is expected as '&'.

Setting the text to the empty string, does not delete any XML elements. Use Cut, Delete or PerformAction instead.

Errors

2001 The authentic range object or its related view object is no longer valid.
2005 Invalid address for a return parameter was specified.

29.3.2.6 AuthenticView

Properties	Methods	Events
Application	Goto	OnBeforeCopy
AsXMLString	IsRedoEnabled	OnBeforeCut
DocumentBegin	IsUndoEnabled	OnBeforeDelete
DocumentEnd	Print	OnBeforeDrop
Event	Redo	OnBeforePaste
MarkupVisibility	Undo	OnDragOver
Parent	UpdateXMLInstanceEntities	OnKeyBoardEvent
Selection		OnMouseEvent
XMLDataRoot		OnSelectionChanged
WholeDocument		

Description

AuthenticView and its child objects AuthenticRange and AuthenticDataTransfer provide you with an interface for Authentic View, which allow easy and consistent modification of document contents. These interfaces replace the following interfaces which are marked now as **obsolete:**

OldAuthenticView (old name was DocEditView)
AuthenticSelection (old name was DocEditSelection, superseded by AuthenticRange)
AuthenticEvent (old name was DocEditEvent)

AuthenticView gives you easy access to specific features such as printing, the multi-level undo buffer, and the current cursor selection, or position.

AuthenticView uses objects of type AuthenticRange to make navigation inside the document straight-forward, and to allow for the flexible selection of logical text elements. Use the properties DocumentBegin, DocumentEnd, or WholeDocument for simple selections, while using the Goto method for more complex selections. To navigate relative to a given document range, see the methods and properties of the AuthenticRange object.

Examples

```
' ---------------------------------------
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' ---------------------------------------
Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
	' we have an active document, now check for view mode
	If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
		If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
			MsgBox "Active document does not support authentic view mode"
		Else
			' now it is safe to access the authentic view object
			Dim objAuthenticView
			Set objAuthenticView = objDocument.AuthenticView
```


© 2016-2022 Altova GmbH

```
                    ' now use the authentic view object

                End If
        End If
Else
        MsgBox "No document is open"
End If
```

29.3.2.6.1 Events

29.3.2.6.1.1 OnBeforeCopy

Event: OnBeforeCopy() as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeCopy()
        ' On_AuthenticBeforeCopy = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeCopy()
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (21, ...) // nEventId = 21

Description
This event gets triggered before a copy operation gets performed on the document. Return *True* (or nothing) to allow copy operation. Return *False* to disable copying.

29.3.2.6.1.2 OnBeforeCut

Event: OnBeforeCut() as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeCut()
        ' On_AuthenticBeforeCut = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeCut()
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent (20, ...) // nEventId = 20

Description
This event gets triggered before a cut operation gets performed on the document. Return *True* (or nothing) to allow cut operation. Return *False* to disable operation.

29.3.2.6.1.3 OnBeforeDelete

Event: OnBeforeDelete() as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeDelete()
	' On_AuthenticBeforeDelete = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeDelete()
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (22, ...) // nEventId = 22

Description
This event gets triggered before a delete operation gets performed on the document. Return *True* (or nothing) to allow delete operation. Return *False* to disable operation.

29.3.2.6.1.4 OnBeforeDrop

Event: OnBeforeDrop (*i_nXPos* as Long, *i_nYPos* as Long, *i_ipRange* as AuthenticRange, *i_ipData* as cancelBoolean

Scripting environment - VBScript:
```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
	' On_AuthenticBeforeDrop = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (11, ...) // nEventId = 11

Description
This event gets triggered whenever a previously dragged object gets dropped inside the application window. All event related information gets passed as parameters.

© 2016-2022 Altova GmbH

The first two parameters specify the mouse position at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return *False* to cancle the drop operation. Return *True* (or nothing) to continue normal operation.

Examples

```
' --------------------------------------------------------------------------
' VB code snippet - connecting to object level events
' --------------------------------------------------------------------------
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnBeforeDrop
' event of object objView
Private Function objView_OnBeforeDrop(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
                                      ByVal i_ipRange As IAuthenticRange,
                                      ByVal i_ipData As IAuthenticDataTransfer) As Boolean

   If (Not i_ipRange Is Nothing) Then
      MsgBox ("Dropping on content is prohibited");
      Return False;
   Else
      Return True;
   End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

29.3.2.6.1.5 OnBeforePaste

Event: OnBeforePaste (*objData* as Variant, *strType* as String) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticBeforePaste(objData, strType)
        ' On_AuthenticBeforePaste = False  ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforePaste(objData, strType)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (19, ...) // nEventId = 19

Description
This event gets triggered before a paste operation gets performed on the document. The parameter *strType* is one of "TEXT", "UNICODETEXT" or "IUNKNOWN". In the first two cases *objData* contains a string representation of the object that will be pasted. In the later case, *objData* contains a pointer to an IUnknown COM interface.

Return *True* (or nothing) to allow paste operation. Return *False* to disable operation.

29.3.2.6.1.6 OnBeforeSave

Event: OnBeforeSave (SaveAs flag) as Boolean

Description: `OnBeforeSave` gives the opportunity to e.g. warn the user about overwriting the existing XML document, or to make the document read-only when specific circumstances are not met. The event will be fired before the file dialog is shown.

29.3.2.6.1.7 OnDragOver

Event: OnDragOver (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent, *objRange* as AuthenticRange, *objData* as AuthenticDataTransfer) as Boolean

Scripting environment - VBScript:
```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
        ' On_AuthenticDragOver = False  ' to disable operation
End Function
```

Scripting environment - JScript:
```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent (10, ...) // nEventId = 10

Description
This event gets triggered whenever an object from within or outside of Authentic View gets dragged with the mouse over the application window. All event related information gets passed as parameters.

The first three parameters specify the mouse position, the mouse button status and the status of the virtual keys at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

 © 2016-2022 Altova GmbH

Return *False* to cancel the drag operation. Return *True* (or nothing) to continue normal operation.

Examples

```
' ---------------------------------------------------------------------
' VB code snippet - connecting to object level events
' ---------------------------------------------------------------------
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnDragOver
' event of object objView
Private Function objView_OnDragOver(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
                                    ByVal i_eMouseEvent As SPYMouseEvent,
                                    ByVal i_ipRange As IAuthenticRange,
                                    ByVal i_ipData As IAuthenticDataTransfer) As Boolean

   If (((i_eMouseEvent And spyShiftKeyDownMask) <> 0) And
            (Not i_ipRange Is Nothing)) Then
      MsgBox ("Floating over element " & i_ipRange.FirstXMLData.Parent.Name);
   End If

   Return True;
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

29.3.2.6.1.8 OnKeyboardEvent

Event: OnKeyboardEvent (*eKeyEvent* as SPYKeyEvent, *nKeyCode* as Long, *nVirtualKeyStatus* as Long) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
        ' On_AuthenticKeyboardEvent = True  ' to cancel bubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
        // return true;  /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (30, ...) // nEventId = 30
```

Description
This event gets triggered for *WM_KEYDOWN*, *WM_KEYUP* and *WM_CHAR* Windows messages.

The actual message type is available in the *eKeyEvent* parameter. The status of virtual keys is combined in the parameter *nVirtualKeyStatus*. Use the bit-masks defined in the enumeration datatype SPYVirtualKeyMask, to test for the different keys or their combinations.

29.3.2.6.1.9 OnLoad

Event: OnLoad ()

Description: `OnLoad` can be used e.g. to restrict some AuthenticView functionality, as shown in the example below:

```
function On_AuthenticLoad(  )
{
    // We are disabling all entry helpers in order to prevent user from manipulating XML tree
    AuthenticView.DisableElementEntryHelper();
    AuthenticView.DisableAttributeEntryHelper();

    // We are also disabling the markup buttons for the same purpose
    AuthenticView.SetToolbarButtonState( 'AuthenticMarkupSmall', authenticToolbarButtonDisabled );
    AuthenticView.SetToolbarButtonState( 'AuthenticMarkupLarge', authenticToolbarButtonDisabled );
    AuthenticView.SetToolbarButtonState( 'AuthenticMarkupMixed', authenticToolbarButtonDisabled );
}
```

In the example the status of the Markup Small, Markup Large, Markup Mixed toolbar buttons are manipulated with the help of button identifiers. See complete list.

29.3.2.6.1.10 OnMouseEvent

Event: OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent, *objRange* as AuthenticRange) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
        ' On_AuthenticMouseEvent = True  ' to cancel bubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
        // return true;  /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (31, ...) // nEventId = 31
```

Description
This event gets triggered for every mouse movement and mouse button Windows message.

© 2016-2022 Altova GmbH

The actual message type and the mouse buttons status, is available in the *eMouseEvent* parameter. Use the bit-masks defined in the enumeration datatype SPYMouseEvent to test for the different messages, button status, and their combinations.

The parameter *objRange* identifies the part of the document found at the current mouse cursor position. The range objects always selects a complete tag of the document. (This might change in future versions, when a more precise positioning mechanism becomes available). If no selectable part of the document is found at the current position, the range object is *null*.

29.3.2.6.1.11 OnSelectionChanged

Event: OnSelectionChanged (*objNewSelection* as AuthenticRange)

Scripting environment - VBScript:
Function On_AuthenticSelectionChanged (*objNewSelection*)
End Function

Scripting environment - JScript:
function On_AuthenticSelectionChanged (*objNewSelection*)
{
}

IDE Plugin:
IXMLSpyPlugIn.OnEvent (23, ...) // nEventId = 23

Description
This event gets triggered whenever the selection in the user interface changes.

Examples
```
' ---------------------------------------------------------------------
' VB code snippet - connecting to object level events
' ---------------------------------------------------------------------
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnSelectionChanged
' event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As XMLSpyLib.IAuthenticRange)
    MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

29.3.2.6.1.12 OnToolbarButtonClicked

Event: OnToolbarButtonClicked (Button identifier)

Description: `OnToolbarButtonClicked` is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. The list of predefined button identifiers is below:

- AuthenticPrint
- AuthenticPrintPreview
- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticClear
- AuthenticMarkupHide
- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticReload
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCellRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCellBelow
- AuthenticTableSplitCellHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCellVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable
- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown

© 2016-2022 Altova GmbH

- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

For custom buttons the user might add his own identifiers. Please, note that the user must take care, as the identifiers are not checked for uniqueness. The same identifiers can be used to identify buttons in the `Set/GetToolbarState()` COM API calls. By adding code for different buttons, the user is in the position to completely redefine the AuthenticView toolbar behavior, adding own methods for table manipulation, etc.

29.3.2.6.1.13 OnToolbarButtonExecuted

Event: OnToolbarButtonExecuted (Button identifier)

Description: `OnToolbarButtonClicked` is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. See the list of predefined button identifiers.

`OnToolbarButtonExecuted` is fired after the toolbar action was executed. It is useful e.g. to add update code, as shown in the example below:

```
//event fired when a toolbar button action was executed
function On_AuthenticToolbarButtonExecuted( varBtnIdentifier )
{
    // After whatever command user has executed - make sure to update toolbar button states
    UpdateOwnToolbarButtonStates();
}
```

In this case `UpdateOwnToolbarButtonStates` is a user function defined in the Global Declarations.

29.3.2.6.1.14 OnUserAddedXMLNode

Event: OnUserAddedXMLNode (XML node)

Description: `OnUserAddedXMLNode` will be fired when the user adds an XML node as a primary action. This happens in the situations, where the user clicks on

- auto-add hyperlinks (see example OnUserAddedXMLNode.sps)
- the Insert…, Insert After…, Insert Before… context menu items
- Append row, Insert row toolbar buttons
- Insert After…, Insert Before… actions in element entry helper (outside StyleVision)

The event doesn't get fired on Duplicate row, or when the node was added externally (e.g. via COM API), or on Apply (e.g. Text State Icons), or when in XML table operations or in DB operations.

The event parameter is the XML node object, which was added giving the user an opportunity to manipulate the XML node added. An elaborate example for an event handler can be found in the `OnUserAddedXMLNode.sps` file.

29.3.2.6.2 Application

Property: Application as Application (read-only)

Description
Accesses the XMLSpy application object.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.3 AsXMLString

Property: AsXMLString as String

Description
Returns or sets the document content as an XML string. Setting the content to a new value does not change the schema file or sps file in use. If the new XMLString does not match the actual schema file error 2011 gets returned.

Errors
2000 The authentic view object is no longer valid.
2011 AsXMLString was set to a value which is no valid XML for the current schema file.

29.3.2.6.4 ContextMenu

Property: ContextMenu() as ContextMenu

Description
The property `ContextMenu` gives access to customize the context menu. The best place to do it is in the event handler `OnContextMenuActivated`.

Errors
2000 Invalid object.
2005 Invalid parameter.

29.3.2.6.5 CreateXMLNode

Method: CreateXMLNode (*nKind* as SPYXMLDataKind) as XMLData

Return Value
The method returns the new XMLData object.

Description
To create a new XMLData object use the CreateXMLNode() method.

© 2016-2022 Altova GmbH

Errors
2000 Invalid object.
2012 Cannot create XML node.

29.3.2.6.6 DisableAttributeEntryHelper

Method: DisableAttributeEntryHelper()

Description
DisableAttributeEntryHelper() disables the attribute entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors
2000 Invalid object.

29.3.2.6.7 DisableElementEntryHelper

Method: DisableElementEntryHelper()

Description
DisableElementEntryHelper() disables the element entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors
2000 Invalid object.

29.3.2.6.8 DisableEntityEntryHelper

Method: DisableEntityEntryHelper()

Description
DisableEntityEntryHelper() disables the entity entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors
2000 Invalid object.

29.3.2.6.9 DocumentBegin

Property: DocumentBegin as AuthenticRange (read-only)

Description
Retrieve a range object that points to the beginning of the document.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.10 DocumentEnd

Property: DocumentEnd as AuthenticRange (read-only)

Description
Retrieve a range object that points to the end of the document.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.11 DoNotPerformStandardAction

Method: DoNotPerformStandardAction ()

Description
`DoNotPerformStandardAction()` serves as cancel bubble for macros, and stops further execution after macro has finished.

Errors
2000 Invalid object.

29.3.2.6.12 EvaluateXPath

Method: EvaluateXPath (XMLData as XMLData, strExpression as string) strValue as string

Return Value
The method returns a string

Description
`EvaluateXPath()` executes an XPath expression with the given XML context node. The result is returned as a string, in the case of a sequence it is a space-separated string.

Errors
2000 Invalid object.
2005 Invalid parameter.
2008 Internal error.
2013 XPath error.

© 2016-2022 Altova GmbH

29.3.2.6.13 Event

Property: Event as AuthenticEvent (read-only)

Description
This property gives access to parameters of the last event in the same way as OldAuthenticView.event does. Since all events for the scripting environment and external clients are now available with parameters this Event property should only be used from within IDE-Plugins.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.14 EventContext

Property: EventContext() as EventContext

Description
`EventContext` property gives access to the running macros context. See the `EventContext` interface description for more details.

Errors
2000 Invalid object.

29.3.2.6.15 GetToolbarButtonState

Method: GetToolbarButtonState (`ButtonIdentifier` as string) as `AuthenticToolbarButtonState`

Return Value
The method returns `AuthenticToolbarButtonState`

Description
`Get/SetToolbarButtonState` queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier (see list above). One usage is to disable toolbar buttons permanently. Another usage is to put `SetToolbarButtonState` in the `OnSelectionChanged` event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the listed enumerations.

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors
2000 Invalid object.

2005 Invalid parameter.
2008 Internal error.
2014 Invalid button identifier.

29.3.2.6.16 Goto

Method: Goto (*eKind* as SPYAuthenticElementKind, *nCount* as Long, *eFrom* as *SPYAuthenticDocumentPosition*) as *AuthenticRange*

Description
Retrieve a range object that points to the beginning of the *nCount* element of type *eKind*. The start position is defined by the parameter *eFrom*. Use positive values for *nCount* to navigate to the document end. Use negative values to navigate towards the beginning of the document.

Errors
2000 The authentic view object is no longer valid.
2003 Target lies after end of document.
2004 Target lies before beginning of document.
2005 Invalid element kind specified.
The document position to start from is not one of *spyAuthenticDocumentBegin* or *spyAuthenticDocumentEnd*.
Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

On Error Resume Next
Dim objRange
' goto beginning of first table in document
Set objRange = objAuthenticView.Goto (spyAuthenticTable, 1, spyAuthenticDocumentBegin)
If (Err.number = 0) Then
        objRange.Select()
Else
        MsgBox "No table found in document"
End If
```

29.3.2.6.17 IsRedoEnabled

Property: IsRedoEnabled as Boolean (read-only)

Description
True if redo steps are available and Redo is possible.

Errors

© 2016-2022 Altova GmbH

2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.18 IsUndoEnabled

Property: IsUndoEnabled as Boolean (read-only)

Description
True if undo steps are available and Undo is possible.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.19 MarkupVisibility

Property: MarkupVisibility as SPYAuthenticMarkupVisibility

Description
Set or get current visibility of markup.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid enumeration value was specified.
Invalid address for the return parameter was specified.

29.3.2.6.20 Parent

Property: Parent as Document (read-only)

Description
Access the document shown in this view.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.21 Print

Method: Print (*bWithPreview* as Boolean, *bPromptUser* as Boolean)

Description

Print the document shown in this view. If *bWithPreview* is set to *True*, the print preview dialog pops up. If *bPromptUser* is set to *True*, the print dialog pops up. If both parameters are set to *False*, the document gets printed without further user interaction.

Errors
2000 The authentic view object is no longer valid.

29.3.2.6.22 Redo

Method: Redo() as Boolean

Description
Redo the modification undone by the last undo command.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.23 Selection

Property: Selection as AuthenticRange

Description
Set or get current text selection in user interface.

Errors
2000 The authentic view object is no longer valid.
2002 No cursor selection is active.
2005 Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' if we are the end of the document, re-start at the beginning
If (objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then
        objAuthenticView.Selection = objAuthenticView.DocumentBegin
Else
        ' objAuthenticView.Selection = objAuthenticView.Selection.GotoNextCursorPosition()
        ' or shorter:
         objAuthenticView.Selection.GotoNextCursorPosition().Select
End If
```


© 2016-2022 Altova GmbH

29.3.2.6.24 SetToolbarButtonState

Method: SetToolbarButtonState (`ButtonIdentifier` as string, `AuthenticToolbarButtonState` state)

Description
`Get/SetToolbarButtonState` queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier (see list above). One usage is to disable toolbar buttons permanently. Another usage is to put `SetToolbarButtonState` in the `OnSelectionChanged` event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the listed enumerations.

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors
2000 Invalid object.
2008 Internal error.
2014 Invalid button identifier.

29.3.2.6.25 Undo

Method: Undo() as Boolean

Description
Undo the last modification of the document from within this view.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.26 UpdateXMLInstanceEntities

Method: UpdateXMLInstanceEntities()

Description
Updates the internal representation of the declared entities, and refills the entry helper. In addition, the validator is reloaded, allowing the XML file to validate correctly. Please note that this may also cause schema files to be reloaded.

Errors
The method never returns an error.

Example
```
// ----------------------------------------
```

```
// Scripting environment - JavaScript
// ----------------------------------------
if(Application.ActiveDocument && (Application.ActiveDocument.CurrentViewMode == 4))
{
	var objDocType;
	objDocType = Application.ActiveDocument.DocEditView.XMLRoot.GetFirstChild(10);

	if(objDocType)
	{
		var objEntity = Application.ActiveDocument.CreateChild(14);
		objEntity.Name = "child";
		objEntity.TextValue = "SYSTEM \"child.xml\"";
		objDocType.AppendChild(objEntity);

		Application.ActiveDocument.AuthenticView.UpdateXMLInstanceEntities();
	}
}
```

29.3.2.6.27 WholeDocument

Property: WholeDocument as AuthenticRange (read-only)

Description
Retrieve a range object that selects the whole document.

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.6.28 XMLDataRoot

Property: XMLDataRoot as XMLData (read-only)

Description
Returns or sets the top-level XMLData element of the current document. This element typically describes the document structure and would be of kind spyXMLDataXMLDocStruct, spyXMLDataXMLEntityDocStruct or spyXMLDataDTDDocStruct..

Errors
2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

29.3.2.7 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Properties and Methods

© 2016-2022 Altova GmbH

Standard automation properties
Application
Parent

Programming language selection properties
ProgrammingLanguage
TemplateFileName

Settings for C++ code
CPPSettings_DOMType
CPPSettings_LibraryType
CPPSettings_UseMFC
CPPSettings_GenerateVC6ProjectFile
CPPSettings_GenerateVSProjectFile

Settings for C# code
CSharpSettings_ProjectType

Dialog handling for above code generation properties
PropertySheetDialogAction

Output path selection properties
OutputPath
OutputPathDialogAction

Presentation of result
OutputResultDialogAction

Description
Use this object to configure the generation of program code for schema files. The method GenerateProgramCode expects a CodeGeneratorDlg as parameter to configure code generation as well as the associated user interactions.

29.3.2.7.1 Application

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.7.2 CompatibilityMode (obsolete)

Property: CompatibilityMode as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Set to true to generate code compatible to XMLSpy 2005R3. Set to false to use newly added code-generation features.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.3 CPPSettings_DOMType

Property: CPPSettings_DOMType as SPYDOMType
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.4 CPPSettings_GenerateVC6ProjectFile

Property: CPPSettings_GenerateVC6ProjectFile as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

© 2016-2022 Altova GmbH

29.3.2.7.5 CPPSettings_GenerateGCCMakefile

Property: CPPSettings_GenerateGCCMakefile as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Creates makefiles to compile the generated code under Linux with GCC.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.6 CPPSettings_GenerateVSProjectFile

Property: CSharpSettings_GenerateVSProjectFile as SPYProjectType
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code. Only `spyVisualStudio2005Project (=4)` and `spyVisualStudio2008Project (=5)` and `spyVisualStudio2010Project (=6)` are valid project types.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.7 CPPSettings_LibraryType

Property: CPPSettings_LibraryType as SPYLibType
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.8 CPPSettings_UseMFC

Property: CPPSettings_UseMFC as Boolean
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines one of the settings that configure generation of C++ code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.9 CSharpSettings_ProjectType

Property: CSharpSettings_ProjectType as SPYProjectType
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines the only setting to configure generation of C# code.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.10 OutputPath

Property: OutputPath as String
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Selects the base directory for all generated code.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.7.11 OutputPathDialogAction

Property: OutputPathDialogAction as SPYDialogAction
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines how the sub-dialog for selecting the code generation output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the OutputPath property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors
2200 The object is no longer valid.

© 2016-2022 Altova GmbH

2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.12 OutputResultDialogAction

Property: OutputResultDialogAction as SPYDialogAction
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines how the sub-dialog that asks to show the result of the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors
2200 The object is no longer valid.
2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.13 Parent

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.7.14 ProgrammingLanguage

Property: ProgrammingLanguage as ProgrammingLanguage
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Selects the output language for the code to be generated.

CAUTION: Setting this property to one of C++, C# or Java, changes the property TemplateFileName to the appropriate template file delivered with XMLSpy as well. If you want to generate C++, C# or Java code based on your own templates, set first the programming language and then select your template file.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.7.15 PropertySheetDialogAction

Property: PropertySheetDialogAction as SPYDialogAction
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Defines how the sub-dialog that configures the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current values as defaults. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

2200	The object is no longer valid.
2201	Invalid action passed as parameter or an invalid address was specified for the return parameter.

29.3.2.7.16 TemplateFileName

Property: TemplateFileName as String
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Selects the code generation template file. XMLSpy comes with template files for C++, C# or Java in the SPL folder of your installation directory.

Setting this property to one of the code generation template files of your XMLSpy installation automatically sets the ProgrammingLanguage property to its appropriate value.

Errors

2200	The object is no longer valid.
2201	Invalid address for the return parameter was specified.

29.3.2.8 DatabaseConnection

Properties for import and export
File or
ADOConnection or
ODBCConnection

Properties for import only
DatabaseKind
SQLSelect
AsAttributes
ExcludeKeys
IncludeEmptyElements
NumberDateTimeFormat
NullReplacement
CommentIncluded

© 2016-2022 Altova GmbH

Properties for export only
CreateMissingTables
CreateNew
TextFieldLen
DatabaseSchema

Properties for XML Schema from DB Structure generation
PrimaryKeys
ForeignKeys
UniqueKeys
SchemaExtensionType
SchemaFormat
ImportColumnsType

Description
DatabaseConnection specifies the parameters for the database connection.

Please note that the properties of the DatabaseConnection interface are referring to the settings of the import and export dialogs of XMLSpy.

29.3.2.8.1 ADOConnection

Property: ADOConnection as String

Description
The property ADOConnection contains a connection string. Either use this property or ODBCConnection or File to refer to a database.

Errors
No error codes are returned.

Example

```
Dim objSpyConn As DatabaseConnection
Set objSpyConn = objSpy.GetDatabaseSettings

Dim objADO As DataLinks
Set objADO = CreateObject("DataLinks")

If Not (objADO Is Nothing) Then
        Dim objConn As Connection
        Set objConn = objADO.PromptNew
        objSpyConn.ADOConnection = objConn.ConnectionString
End If
```

29.3.2.8.2 AsAttributes

Property: AsAttributes as Boolean

Description
Set AsAttributes to true if you want to initialize all import fields to be imported as attributes. Default is false and will initialize all fields to be imported as elements. This property is used only in calls to Application.GetDatabaseImportElementList.

Errors
No error codes are returned.

29.3.2.8.3 CommentIncluded

Property: CommentIncluded as Boolean

Description
This property tells whether additional comments are added to the generated XML. Default is true. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.4 CreateMissingTables

Property: CreateMissingTables as Boolean

Description
If CreateMissingTables is true, tables which are not already defined in the export database will be created during export. Default is true. This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.5 CreateNew

Property: CreateNew as Boolean

Description
Set CreateNew true if you want to create a new database on export. Any existing database will be overwritten. See also DatabaseConnection.File. Default is false. This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.6 DatabaseKind

Property: DatabaseKind as SPYDatabaseKind

Description

© 2016-2022 Altova GmbH

Select the kind of database that gets access. The default value is spyDB_Unspecified(7) and is sufficient in most cases. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.7 DatabaseSchema

Property: DatabaseSchema as String

Description
This property specifies the Schema used for export in Schema aware databases. Default is "". This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.8 ExcludeKeys

Property: ExcludeKeys as Boolean

Description
Set ExcludeKeys to true if you want to exclude all key columns from the import data. Default is false. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.9 File

Property: File as String

Description
The property File sets the path for the database during export or import. This property can only be used in conjunction with a Microsoft Access database. Either use this property or ODBCConnection or ADOConnection to refer to the database.

See also Import and Export.

Errors
No error codes are returned.

29.3.2.8.10 ForeignKeys

Property: ForeignKeys as Boolean

Description
Specifies whether the Foreign Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.8.11 ImportColumnsType

Property: ImportColumnsType as SPYImportColumnsType

Description
Defines if column information from the DB is saved as element or attribute in the XML Schema. Default is as element. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.8.12 IncludeEmptyElements

Property: IncludeEmptyElements as Boolean

Description
Set IncludeEmptyElements to false if you want to exclude all empty elements. Default is true. This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.13 NullReplacement

Property: NullReplacement as String

Description
This property contains the text value that is used during import for empty elements (null values). Default is "". This property is used only when importing from databases.

Errors
No error codes are returned.

29.3.2.8.14 NumberDateTimeFormat

Property: NumberDateTimeFormat as SPYNumberDateTimeFormat

Description
The property NumberDateTimeFormat sets the format of numbers and date- and time-values. Default is spySystemLocale. This property is used only when importing from databases.

© 2016-2022 Altova GmbH

Errors
No error codes are returned.

29.3.2.8.15 ODBCConnection

Property: ODBCConnection as String

Description
The property ODBCConnection contains a ODBC connection string. Either use this property or ADOConnection or File to refer to a database.

Errors
No error codes are returned.

29.3.2.8.16 PrimaryKeys

Property: PrimaryKeys as Boolean

Description
Specifies whether the Primary Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.8.17 SchemaExtensionType

Property: SchemaExtensionType as SPYSchemaExtensionType

Description
Defines the Schema extension type used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

See also Create XML Schema from DB Structure.

Errors
No error codes are returned.

29.3.2.8.18 SchemaFormat

Property: SchemaFormat as SPYSchemaFormat

Description
Defines the Schema format used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

See also Create XML Schema from DB Structure.

Errors
No error codes are returned.

29.3.2.8.19 SQLSelect

Property: SQLSelect as String

Description
The SQL query for the import is stored in the property SQLSelect. This property is used only when importing from databases. See also Import and Export.

Errors
No error codes are returned.

29.3.2.8.20 TextFieldLen

Property: TextFieldLen as long

Description
The property TextFieldLen sets the length for created text fields during the export. Default is 255. This property is used only when exporting to databases.

Errors
No error codes are returned.

29.3.2.8.21 UniqueKeys

Property: UniqueKeys as Boolean

Description
Specifies whether the Unique Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors
No error codes are returned.

29.3.2.9 Dialogs

Properties and Methods

Standard automation properties
Application
Parent

Various dialog objects

© 2016-2022 Altova GmbH

CodeGeneratorDlg
FileSelectionDlg
SchemaDocumentationDlg
GenerateSampleXMLDlg
DTDSchemaGeneratorDlg
FindInFilesDlg
WSDLDocumentationDlg
WSDL20DocumentationDlg
XBRLDocumentationDlg

Description
The Dialogs object provides access to different built-in dialogs of XMLSpy. These dialog objects allow to initialize the fields of user dialogs before they get presented to the user or allow to simulate complete user input by your program.

29.3.2.9.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
2300 The object is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.2 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: CodeGeneratorDlg as CodeGeneratorDlg (read-only)

Description
Get a new instance of a code generation dialog object. You will need this object to pass the necessary parameters to the code generation methods. Initial values are taken from last usage of the code generation dialog.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.3 FileSelectionDlg

Property: FileSelectionDlg as FileSelectionDlg (read-only)

Description
Get a new instance of a file selection dialog object.

File selection dialog objects are passed to you with the some events that signal opening or saving of documents and projects.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.4 Parent

Property: Parent as Application (read-only)

Description
Access the XMLSpy application object.

Errors
2300 The object is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.5 SchemaDocumentationDlg

Property: SchemaDocumentationDlg as SchemaDocumentationDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of schema documentation. See Document.GenerateSchemaDocumentation for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.6 GenerateSampleXMLDlg

Property: GenerateSampleXMLDlg as GenerateSampleXMLDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of a sample XML based on a W3C schema or DTD. See GenerateSampleXML for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.9.7 DTDSchemaGeneratorDlg

Property: DTDSchemaGeneratorDlg as DTDSchemaGeneratorDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of a schema or DTD. See Document.GenerateDTDOrSchemaEx for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.8 FindInFilesDlg

Property: FindInFilesDlg as FindInFilesDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes the search (or replacement) of strings in files. See Application.FindInFiles for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.9 WSDLDocumentationDlg

Property: WSDLDocumentationDlg as WSDLDocumentationDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of WSDL documentation. See Document.GenerateWSDLDocumentation for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.10 WSDL20DocumentationDlg

Property: WSDL20DocumentationDlg as WSDL20DocumentationDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See Document.GenerateWSD20LDocumentation for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.9.11 XBRLDocumentationDlg

Property: XBRL20DocumentationDlg as XBRL20DocumentationDlg (read-only)

Description
Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See Document.GenerateXBRLDocumentation for its usage.

Errors
2300 The Dialogs object or one of its parents is no longer valid.
2301 Invalid address for the return parameter was specified.

29.3.2.10 Document

Properties and Methods

Standard automation properties
Application
Parent

Various document properties and methods
SetActiveDocument
Encoding
SetEncoding (obsolete)
Suggestions

XML validation
IsValid
SetExternalIsValid

Document conversion and transformation
AssignDTD
AssignSchema
AssignXSL
AssignXSLFO
ConvertDTDOrSchema
ConvertDTDOrSchemaEx
GenerateDTDOrSchema
GenerateDTDOrSchemaEx
FlattenDTDOrSchema
CreateSchemaDiagram
ExecuteXQuery
TransformXSL
TransformXSLEx
TransformXSLFO
GenerateProgramCode (Enterprise Edition only)
GenerateSchemaDocumentation
GenerateSampleXML
GenerateWSDL20Documentation
GenerateWSDLDocumentation

© 2016-2022 Altova GmbH

GenerateXBRLDocumentation
ConvertToWSDL20

Document export
GetExportElementList
ExportToText
ExportToDatabase
CreateDBStructureFromXMLSchema
GetDBStructureList

File saving and naming
FullName
Name
Path
GetPathName (obsolete)
SetPathName (obsolete)
Title
IsModified
Saved
SaveAs
Save
SaveInString
SaveToURL
Close

View access
CurrentViewMode
SwitchViewMode
AuthenticView
GridView
DocEditView (obsolete)

Access to XMLData
RootElement
DataRoot
CreateChild
UpdateViews
StartChanges
EndChanges
UpdateXMLData

Description
Document objects represent XML documents opened in XMLSpy.

Use one of the following properties to access documents that are already open XMLSpy:
Application.ActiveDocument
Application.Documents

Use one of the following methods to open a new document in XMLSpy:
Documents.OpenFile
Documents.OpenURL
Documents.OpenURLDialog
Documents.NewFile

Documents.NewFileFromText
SpyProjectItem.Open
Application.ImportFromDatabase
Application.ImportFromSchema
Application.ImportFromText
Application.ImportFromWord
Document.ConvertDTDOrSchema
Document.GenerateDTDOrSchema

29.3.2.10.1 Events

29.3.2.10.1.1 OnBeforeSaveDocument

Event: OnBeforeSaveDocument(*objDocument* as Document, *objDialog* as FileSelectionDlg)

XMLSpy scripting environment - VBScript:

```
Function On_BeforeSaveDocument(objDocument, objDialog)
End Function

' old handler - now obsolete
' return string to save to new file name
' return empty string to cancel save operation
' return nothing to save to original name
Function On_SaveDocument(objDocument, strFilePath)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeSaveDocument(objDocument, objDialog)
{
}

// old handler - now obsolete
// return string to save to new file name
// return empty string to cancel save operation
// return nothing to save to original name
function On_SaveDocument(objDocument, strFilePath)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (27, ...) // nEventId = 27
```

Description

This event gets fired on any attempt to save a document. The file selection dialog object is initialized with the name chosen for the document file. You can modify this selection. To continue saving the document leave the FileSelectionDlg.DialogAction property of *io_objDialog* at its default value spyDialogOK. To abort saving of the document set this property to spyDialogCancel.

© 2016-2022 Altova GmbH

29.3.2.10.1.2 OnBeforeCloseDocument

Event: OnBeforeCloseDocument(*objDocument* as Document)as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeCloseDocument(objDocument)
	' On_BeforeCloseDocument = False  ' to prohibit closing of document
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeCloseDocument(objDocument)
{
	// return false;  /* to prohibit closing of document */
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (28, ...) // nEventId = 28

Description
This event gets fired on any attempt to close a document. To prevent the document from being closed return false.

29.3.2.10.1.3 OnBeforeValidate

Event: OnBeforeValidate(*objDocument* as Document, *bOnLoading* as Boolean, *bOnCommand* as Boolean) as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
	On_BeforeValidate = bCancelDefaultValidation     'set by the script if necessary
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
{
	return bCancelDefaultValidation   //set by the script if necessary
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (32, ...) // nEventId = 32

Description
This event gets fired before the document is validated. It is possible to suppress the default validation by returning false from the event handler. In this case the script should also set the validation result using the SetExternalIsValid method.

bOnLoading is true if the event is raised on the initial validation on loading the document.

bOnCommand is true whenever the user selected the Validate command from the Toolbar or menu.

Available with TypeLibrary version 1.5

29.3.2.10.1.4 OnCloseDocument

Event: OnCloseDocument(*objDocument* as Document)

XMLSpy scripting environment - VBScript:
Function On_Close Document(*objDocument*)
End Function

XMLSpy scripting environment - JScript:
function On_Close Document(*objDocument*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (8, ...) // nEventId = 8

Description
This event gets fired as a result of closing a document. Do not modify the document from within this event.

29.3.2.10.1.5 OnViewActivation

Event: OnViewActivation(*objDocument* as Document, *eViewMode* as SPYViewModes, *bActivated* as Boolean)

XMLSpy scripting environment - VBScript:
Function On_ViewActivation(*objDocument*, *eViewMode*, *bActivated*)
End Function

XMLSpy scripting environment - JScript:
function On_ViewActivation(*objDocument*, *eViewMode*, *bActivated*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (29, ...) // nEventId = 29

Description
This event gets fired whenever a view of a document becomes visible (i.e. becomes the active view) or invisible (i.e. another view becomes the active view or the document gets closed). However, the first view activation event after a document gets opened cannot be received, since there is no document object to get the event from. Use the Application.OnDocumentOpened event instead.

© 2016-2022 Altova GmbH

29.3.2.10.2 Application

Property: Application as Application (read-only)

Description
Accesses the XMLSpy application object.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.3 AssignDTD

Method: AssignDTD(*strDTDFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the DTD file "strDTDFile" into the document. Note that no error occurs if the file does not exist, or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign a DTD to the document.

29.3.2.10.4 AssignSchema

Method: AssignSchema (*strSchemaFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the schema file "strSchemaFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign a schema file to the document.

29.3.2.10.5 AssignXSL

Method: AssignXSL (*strXSLFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the XSL file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign an XSL file to the document.

29.3.2.10.6 AssignXSLFO

Method: AssignXSLFO (*strXSLFOFile* as String, *bDialog* as Boolean)

Description
The method places a reference to the XSLFO file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true XMLSpy presents a dialog to set the file.

Errors
1400 The object is no longer valid.
1409 You are not allowed to assign an XSL file to the document.

29.3.2.10.7 AsXMLString

Property: AsXMLString as String

Description
This property can be used to get or set the document content.

Errors
1400 The document object is no longer valid.
1404 Cannot create XMLData object.
1407 View mode cannot be switched.

29.3.2.10.8 AuthenticView

Method: AuthenticView as AuthenticView (read-only)

Description
Returns an object that gives access to properties and methods specific to Authentic view. The object returned is only valid if the current document is opened in Authentic view mode. The lifetime of an object ends with the next view switch. Any attempt to access objects or any of its children afterwards will result in an error indicating that the object is invalid.

AuthenticView and DocEditView both provide automation access to the Authentic view mode of XMLSpy. Functional overlap is intentional. A future version of Authentic View will include all functionality of DocEditView and its sub-objects, thereby making usage of DocEditView obsolete.

Errors
1400 The object is no longer valid.
1417 Document needs to be open in authentic view mode.

Examples

```
' ----------------------------------------
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' ----------------------------------------
Dim objDocument
```


© 2016-2022 Altova GmbH

```
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
	' we have an active document, now check for view mode
	If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
		If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
			MsgBox "Active document does not support authentic view mode"
		Else
			' now it is safe to access the authentic view object
			Dim objAuthenticView
			Set objAuthenticView = objDocument.AuthenticView
			' now use the authentic view object

		End If
	End If
Else
	MsgBox "No document is open"
End If
```

29.3.2.10.9 Close

Method: Close (*bDiscardChanges* as Boolean)

Description
To close the document call this method. If bDiscardChanges is true and the document is modified, the document will be closed but not saved.

Errors

1400	The object is no longer valid.
1401	Document needs to be saved first.

29.3.2.10.10 ConvertDTDOrSchema

Method: ConvertDTDOrSchema (*nFormat* as SPYDTDSchemaFormat, *nFrequentElements* as SPYFrequentElements)

Parameters
nFormat
Sets the schema output format to DTD or W3C.

nFrequentElements
Create complex elements as elements or complex types.

Description
ConvertDTDOrSchema takes an existing schema format and converts it into a different format.
For a finer tuning of DTD/XSD conversion, use ConvertDTDOrSchemaEx.

Errors

1400 The object is no longer valid.
1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead*.

29.3.2.10.11 ConvertDTDOrSchemaEx

Method: ConvertDTDOrSchemaEx (*nFormat* as SPYDTDSchemaFormat, *nFrequentElements* as SPYFrequentElements, sOutputPath as String, nOutputPathDialogAction as SPYDialogAction)

Parameters
nFormat
Sets the schema output format to DTD, or W3C.

nFrequentElements
Create complex elements as elements or complex types.

`sOutputPath`
The file path for the newly generated file.

`nOutputPathDialogAction`
Defines the dialog interaction for this call.

Description
`ConvertDTDOrSchemaEx` takes an existing schema format and converts it into a different format.

Errors
1400 The object is no longer valid.
1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead*.

29.3.2.10.12 ConvertToWSDL20

Method: ConvertToWSDL20 (*sFilePath* as String, *bShowDialogs* as Boolean)

Parameters
sFilePath

© 2016-2022 Altova GmbH

This specifies the file name of the converted WSDL. In case the source WSDL includes files which also must be converted, then only the directory part of the given path is used and the file names are generated automatically.

bShowDialogs
Defines whether file/folder selection dialogs are shown.

Description
Converts the WSDL 1.1 document to a WSDL 2.0 file. It will also convert any referenced WSDL files that are referenced from within this document. Note that this functionality is limited to WSDL View only. See Document.CurrentViewMode. and SPYViewModes.

Errors

1400	The document object is no longer valid.
1407	Invalid parameters have been passed or an empty file name has been specified as output target.
1417	The document is not opened in WSDL view, maybe it is not an '.wsdl' file.
1421	Feature is not available in this edition.
1433	WSDL 1.1 to WSDL 2.0 conversion failed.

29.3.2.10.13 CreateChild

Method: CreateChild (*nKind* as SPYXMLDataKind) as XMLData

Return Value
The method returns the new XMLData object.

Description
To create a new XMLData object use the CreateChild() method.

Errors

1400	The object is no longer valid.
1404	Cannot create XMLData object.
1407	Invalid address for the return parameter was specified.

29.3.2.10.14 CreateDBStructureFromXMLSchema

Method: CreateDBStructureFromXMLSchema (*pDatabase* as DatabaseConnection, *pTables* as ElementList, *bDropTableWithExistingName* as Boolean) as String

Description
CreateDBStructureFromXMLSchema exports the given tables to the specified database. The function returns the SQL statements that were necessary to perform the changes.

See also GetDBStructureList.

Errors

1429	Database selection missing.
1430	Document export failed.

29.3.2.10.15 CreateSchemaDiagram

Method: CreateSchemaDiagram (*nKind* as SPYSchemaDefKind, *strName* as String, *strFile* as String)

Return Value
None.

Description
The method creates a diagram of the schema type strName of kind nKind and saves the output file into strFile. Note that this functionality is limited to Schema View only. See Document.CurrentViewMode. and SPYViewModes.

Errors
1400 The object is no longer valid.
1414 Failed to save diagram.
1415 Invalid schema definition type specified.

29.3.2.10.16 CurrentViewMode

Method: CurrentViewMode as SPYViewModes

Description
The property holds the current view mode of the document. See also Document.SwitchViewMode.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.17 DataRoot

Property: DataRoot as XMLData (read-only)

Description
This property provides access to the document's first XMLData object of type *spyXMLDataElement*. This is typically the root element for all document content data. See XMLSpyDocument.RootElement to get the root element of the whole document including XML prolog data. If the CurrentViewMode is not *spyViewGrid* or *spyViewAuthentic* an UpdateXMLData may be necessary to get access to the latest XMLData.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.10.18 DocEditView

Method: DocEditView as DocEditView

Description
Holds a reference to the current Authentic View object.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.
1417 Document needs to be open in authentic view mode.

29.3.2.10.19 Encoding

Property: Encoding as String

Description
This property provides access to the document's encoding value. However, this property can only be accessed when the document is opened in *spyViewGrid*, *spyViewText* or *spyViewAuthentic*. See CurrentViewMode on how to detect that a document's actual view mode.

This property makes the method SetEncoding obsolete.

Possible values are, for example:

8859-1,
8859-2,
ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.
1416 Operation not supported in current view mode.

29.3.2.10.20 EndChanges

Method: EndChanges()

Description

Use the method EndChanges to display all changes since the call to Document.StartChanges.

Errors
1400 The object is no longer valid.

29.3.2.10.21 ExecuteXQuery

Method: ExecuteXQuery (*strXMLFileName* as String)

Description
Execute the XQuery statements contained in the document of the document object. Either an XQuery execution or an XQuery Update is performed depending on the file extension of the document. Use the XML file specified in the argument as the XML target document that the XQuery document processes.

- If the document has an XQuery file extension as defined in the Options dialog of XMLSpy, then an XQuery execution is performed. By default: `.xq`, `.xql`, and `.xquery` are set as XQuery file extensions in XMLSpy.
- If the document has an XQuery Update file extension as defined in the Options dialog of XMLSpy, then an XQuery Update action is performed. By default: `.xqu` is set as an XQuery Update file extension in XMLSpy.

If your XQuery script does not use an XML source, set the parameter `strXMLFileName` to an empty string.

Errors
1400 The document object is no longer valid.
1423 XQuery transformation error.
1424 Not all files required for operation could be loaded. Most likely, the file specified in `strXMLFileName` does not exist or is not valid.

29.3.2.10.22 ExportToDatabase

Method: ExportToDatabase (*pFromChild* as XMLData, *pExportSettings* as ExportSettings, *pDatabase* as DatabaseConnection)

Description
ExportToDatabase exports the XML document starting with the element pFromChild. The parameter pExportSettings defines the behaviour of the export (see Application.GetExportSettings). The parameter pDatabase specifies the destination of the export (see Application.GetDatabaseSettings). `UpdateXMLData()` might be indirectly needed as you have to pass the `XMLData` as parameter to this function.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1416 Error during export.
1429 Database selection missing.
1430 Document export failed.

© 2016-2022 Altova GmbH

Example

```
Dim objDoc As Document
Set objDoc = objSpy.ActiveDocument

'set the behaviour of the export with ExportSettings
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

'set the destination with DatabaseConnection
Dim objDB As DatabaseConnection
Set objDB = objSpy.GetDatabaseSettings

objDB.CreateMissingTables = True
objDB.CreateNew = True
objDB.File = "C:\Export.mdb"

objDoc.ExportToDatabase objDoc.RootElement, objExpSettings, objDB
If Err.Number <> 0 Then
        a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
                "Description: " & Err.Description)
End If
```

29.3.2.10.23 ExportToText

Method: ExportToText (*pFromChild* as XMLData, *pExportSettings* as ExportSettings, *pTextSettings* as TextImportExportSettings)

Description
ExportToText exports tabular information from the document starting at pFromChild into one or many text files. Columns of the resulting tables are generated in alphabetical order of the column header names. Use GetExportElementList to learn about the data that will be exported. The parameter pExportSettings defines the specifics for the export. Set the property ExportSettings.ElementList to the - possibly modified - list returned by GetExportElementList to avoid exporting all contained tables. The parameter pTextSettings defines the options specific to text export and import. You need to set the property TextImportExportSettings.DestinationFolder before you call ExportToText. `UpdateXMLData()` might be indirectly needed as you have to pass the `XMLData` as parameter to this function.

See also Import and export of data.

Errors

1400	The object is no longer valid.
1407	Invalid parameter or invalid address for the return parameter was specified.
1416	Error during export.
1430	Document export failed.

Example

```
' ----------------------------------------------------
' VBA client code fragment - export document to text files
```

```
' ----------------------------------------------------
        Dim objDoc As Document
        Set objDoc = objSpy.ActiveDocument

        Dim objExpSettings As ExportSettings
        Set objExpSettings = objSpy.GetExportSettings
        objExpSettings.ElementList = objDoc.GetExportElementList(
                                                        objDoc.RootElement,
                                                        objExpSettings)

        Dim objTextExp As TextImportExportSettings
        Set objTextExp = objSpy.GetTextExportSettings
        objTextExp.HeaderRow = True
        objTextExp.DestinationFolder = "C:\Exports"

        On Error Resume Next
        objDoc.ExportToText objDoc.RootElement, objExpSettings, objTextExp

        If Err.Number <> 0 Then
                a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) & "Description: "
                & Err.Description)
        End If
```

29.3.2.10.24 FlattenDTDOrSchema

Method: FlattenDTDOrSchema (sOutputPath as String, nOutputPathDialogAction as SPYDialogAction)

Parameters

`sOutputPath`
The file path for the newly generated file.

`nOutputPathDialogAction`
Defines the dialog interaction for this call.

Description

`FlattenDTDOrSchema` takes an existing DTD or schema, generates a flattened file, and saves the generated file at the specified location. In the case of DTDs, flattening removes parameter entities and produces a single DTD from a collection of modules; sections marked `IGNORE` are suppressed and unused parameter entities are deleted. When an XML Schema is flattened, (i) the components of all included schemas are added as global components of the active schema, and (ii) included schemas are deleted.

Errors

1400 The object is no longer valid.
1412 Error during conversion.

© 2016-2022 Altova GmbH

29.3.2.10.25 FullName

Property: FullName as String

Description
This property can be used to get or set the full file name - including the path - to where the document gets saved. The validity of the name is not verified before the next save operation.

This property makes the methods GetPathName and SetPathName obsolete.

Errors
1400 The document object is no longer valid.
1402 Empty string has been specified as full file name.

29.3.2.10.26 GenerateDTDOrSchema

Method: GenerateDTDOrSchema (*nFormat* as SPYDTDSchemaFormat, *nValuesList* as integer, *nDetection* as SPYTypeDetection, *nFrequentElements* as SPYFrequentElements)

Parameters
nFormat
Sets the schema output format to DTD, or W3C.

nValuesList
Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

nDetection
Specifies granularity of simple type detection.

nFrequentElements
Shall the types for all elements be defined as global? Use that value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Description
Use this method to automatically generate a DTD or schema for the current XML document.
For a finer tuning of DTD / schema generation, use GenerateDTDOrSchemaEx.
Note that this functionality is not available in ZIP View only. See Document.CurrentViewMode. and SPYViewModes.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.27 GenerateDTDOrSchemaEx

Method: GenerateDTDOrSchemaEx (*objDlg* as DTDSchemaGeneratorDlg) as Document

Description

Use this method to automatically generate a DTD or schema for the current XML document. A DTDSchemaGeneratorDlg object is used to pass information to the schema/DTD generator. The generation process can be configured to allow user interaction or run without further user input.
Note that this functionality is not available in ZIP View only. See Document.CurrentViewMode. and SPYViewModes.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.28 GenerateProgramCode

Method: GenerateProgramCode (*objDlg* as CodeGeneratorDlg)
Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description
Generate Java, C++ or C# class files from the XML Schema definitions in your document. A CodeGeneratorDlg object is used to pass information to the code generator. The generation process can be configured to allow user interaction or run without further user input.

Errors
1400 The document object is no longer valid.
1407 An empty file name has been specified.
1421 Feature not available in this edition

29.3.2.10.29 GenerateSampleXML

Method: GenerateSampleXML (*objDlg* as GenerateSampleXMLDlg) as Document

Description
Generates a sample XML if the document is a schema or DTD. Use Dialogs.GenerateSampleXMLDlg to get an initialized set of options.

Available with TypeLibrary version 1.5

Errors
1400 The document object is no longer valid.

29.3.2.10.30 GenerateSchemaDocumentation

Method: GenerateSchemaDocumentation (*objDlg* as SchemaDocumentationDlg)

Description
Generate documentation for a schema definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.SchemaDocumentationDlg to get an initialized set of options. As a minimum, you will need to set the property SchemaDocumentationDlg.OutputFile before

© 2016-2022 Altova GmbH

starting the generation process. Note that this functionality is limited to Schema View only. See Document.CurrentViewMode and SPYViewModes.

Errors

1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.
1422 Error during generation

29.3.2.10.31 GenerateWSDL20Documentation

Method: GenerateWSDL20Documentation (*objDlg* as WSDL20DocumentationDlg)

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.WSDL20DocumentationDlg to get an initialized set of options. As a minimum, you will need to set the property WSDL20DocumentationDlg.OutputFile before starting the generation process. Note that this functionality is limited to WSDL View only. See Document.CurrentViewMode and SPYViewModes.

Errors

1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.
1422 Error during generation

29.3.2.10.32 GenerateWSDLDocumentation

Method: GenerateWSDLDocumentation (*objDlg* as WSDLDocumentationDlg)

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.WSDLDocumentationDlg to get an initialized set of options. As a minimum, you will need to set the property WSDLDocumentationDlg.OutputFile before starting the generation process. Note that this functionality is limited to WSDL View only. See Document.CurrentViewMode and SPYViewModes.

Errors

1400 The document object is no longer valid.
1407 Invalid parameters have been passed or an empty file name has been specified as output target.
1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
1421 Feature is not available in this edition.
1422 Error during generation

29.3.2.10.33 GenerateXBRLDocumentation

Method: GenerateXBRLDocumentation (*objDlg* as XBRLDocumentationDlg)

Description
Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use Dialogs.XBRLDocumentationDlg to get an initialized set of options. As a minimum, you will need to set the property XBRLDocumentationDlg.OutputFile before starting the generation process. Note that this functionality is limited to XBRL View only. See Document.CurrentViewMode and SPYViewModes.

Errors

1400	The document object is no longer valid.
1407	Invalid parameters have been passed or an empty file name has been specified as output target.
1417	The document is not opened in schema view, maybe it is not an '.xsd' file.
1421	Feature is not available in this edition.
1422	Error during generation

29.3.2.10.34 GetDBStructureList

Method: GetDBStructureList (*pDatabase* as DatabaseConnection) as ElementList

Description
GetDBStructureList creates a collection of elements from the Schema document for which tables in the specified database are created. The function returns a collection of ElementListItems where the properties ElementListItem.Name contain the names of the tables.

See also CreateDBStructureFromXMLSchema.

Errors

1400	The object is no longer valid.
1427	Failed creating parser for the specified XML.
1428	Export of element list failed.
1429	Database selection missing.

29.3.2.10.35 GetExportElementList

Method: GetExportElementList (*pFromChild* as XMLData, *pExportSettings* as ExportSettings) as ElementList

Description
GetExportElementList creates a collection of elements to export from the document, depending on the settings in pExportSettings and starting from the element pFromChild. The function returns a collection of

© 2016-2022 Altova GmbH

ElementListItems where the properties ElementListItem.Name contain the names of the tables that can be exported from the document. The property ElementListItem.FieldCount contains the number of columns in the table. The property ElementListItem.RecordCount contains the number of records in the table. The property ElementListItem.ElementKind is unused. `UpdateXMLData()` might be indirectly needed as you have to pass the `XMLData` as parameter to this function.

See also Import and export of data.

Errors

1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1427 Failed creating parser for the specified XML.
1428 Export of element list failed.

29.3.2.10.36 GetPathName (obsolete)

Superseded by Document.FullName

```
// ----- javascript sample -----
// instead of:
// strPathName = Application.ActiveDocument.GetPathName();
// use now:
strPathName = Application.ActiveDocument.FullName;
```

Method: GetPathName() as String

Description
The method GetPathName gets the path of the active document.

See also Document.SetPathName (obsolete).

29.3.2.10.37 GridView

Property: GridView as GridView

Description
This property provides access to the grid view functionality of the document.

Errors

1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.
1417 Document needs to be open in enhanced grid view mode.

29.3.2.10.38 IsModified

Property: IsModified as Boolean

Description
True if the document is modified.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.39 IsValid

Method: IsValid (*strError* as Variant) as Boolean

Return Value
True if the document is valid, false if not. To call `IsValid()`, the application GUI must be visible. (If you wish to validate without the GUI being visible, please use Altova RaptorXML Server.)

Description
`IsValid` validates the document against its associated schema or DTD. `strError` gives you the same error message as when you validate the file within the GUI.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1408 Unable to validate file.

29.3.2.10.40 IsValidEx

Method: IsValidEx (*nXSDVersion* as `SPYValidateXSDVersion`, *nErrorLimit* as `int`, *nErrorFormat* as `SPYValidateErrorFormat`, out *strError* as Variant) as Boolean

Return Value
True if the document is valid, false if not.

Description
`IsValidEx` validates the document against its associated schema or DTD.

In parameters:
`nXSDVersion` which is an enumeration value of `SPYValidateXSDVersion` that selects the XSD version to validate against.
`nErrorLimit` which is an integer. Values must be `1` to `999`.

© 2016-2022 Altova GmbH

`nErrorFormat` which is an enumeration value of `SPYValidateErrorFormat` that selects the XSD version to validate against.

Out parameter:
`strError` is the error message, and is the same as that received when validating the file within the GUI.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.
1408 Unable to validate file.

29.3.2.10.41 IsWellFormed

Method: IsWellFormed (*pData* as XMLData, *bWithChildren* as Boolean, *strError* as Variant, *nErrorPos* as Variant, *pBadXMLData* as Variant) as Boolean

Return Value
True if the document is well formed.

Description
IsWellFormed checks the document for well-formedness starting at the element pData.

If the document is not well formed, strError contains an error message, nErrorPos the position in the file and pBadXMLData holds a reference to the element which breaks the well-formedness. These out-parameters are defined as VARIANTs to support scripting languages like VBScript.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

Example

See IsValid.

29.3.2.10.42 Name

Property: Name as String (read-only)

Description
Use this property to retrieve the name - not including the path - of the document file. To change the file name for a document use the property FullName.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.43 Parent

Property: Parent as Documents (read-only)

Description
Access the parent of the document object.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

Property: Parent as Application (read-only)

29.3.2.10.44 Path

Property: Path as String (read-only)

Description
Use this property to retrieve the path - not including the file name - of the document file. To change the file name and path for a document use the property FullName.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.45 RootElement

Property: RootElement as XMLData (read-only)

Description
The property RootElement provides access to the root element of the XML structure of the document including the XML prolog data. To access the first element of a document's content navigate to the first child of kind *spyXMLDataElement* or use the Document.DataRoot property. If the CurrentViewMode is not *spyViewGrid* or *spyViewAuthentic* an UpdateXMLData may be necessary to get access to the latest XMLData.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.46 Save

Method: Save()

Description
The method writes any modifications of the document to the associated file. See also Document.FullName.

Errors

© 2016-2022 Altova GmbH

1400 The document object is no longer valid.
1407 An empty file name has been specified.
1403 Error when saving file, probably the file name is invalid.

29.3.2.10.47 SaveAs

Method: SaveAs (*strFileName* as String)

Description
Save the document to the file specified. If saving was successful, the FullName property gets set to the specified file name.

Errors
1400 The document object is no longer valid.
1407 An empty file name has been specified.
1403 Error when saving file, probably the file name is invalid.

29.3.2.10.48 Saved

Property: Saved as Boolean (read-only)

Description
This property can be used to check if the document has been saved after the last modifications. It returns the negation of IsModified.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.49 SaveInString

Method: SaveInString (*pData* as XMLData, *bMarked* as Boolean) as String

Parameters
pData
XMLData element to start. Set pData to Document.RootElement if you want to copy the complete file.

bMarked
If bMarked is true, only the elements selected in the grid view are copied.

Return Value
Returns a string with the XML data.

Description
SaveInString starts at the element pData and converts the XMLData objects to a string representation. `UpdateXMLData()` might be indirectly needed as you have to pass the `XMLData` as parameter to this function.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.50 SaveToURL

Method: SaveToURL (*strURL* as String, *strUser* as String, *strPassword* as String)

Return Value

Description
SaveToURL() writes the document to the URL strURL. This method does not set the permanent file path of the document.

Errors
1400 The object is no longer valid.
1402 Invalid URL specified.
1403 Error while saving to URL.

29.3.2.10.51 SetActiveDocument

Method: SetActiveDocument()

Description
The method sets the document as the active and brings it to the front.

Errors
1400 The object is no longer valid.

29.3.2.10.52 SetEncoding (obsolete)

Superseded by Document.Encoding

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.SetEncoding("UTF-16");
// use now:
Application.ActiveDocument.Encoding = "UTF-16";
```

Method: SetEncoding (*strEncoding* as String)

Description
SetEncoding sets the encoding of the document like the menu item "File/Encoding..." in XMLSpy. Possible values for strEncoding are, for example:

8859-1,
8859-2,

© 2016-2022 Altova GmbH

ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

29.3.2.10.53 SetExternalIsValid

Method: SetExternalIsValid (*bValid* as Boolean)

Parameters

bValid
Sets the result of an external validation process.

Description
The internal information set by this method is only queried on cancelling the default validation in any OnBeforeValidate handler.

Available with TypeLibrary version 1.5

Errors
1400 The object is no longer valid.

29.3.2.10.54 SetPathName (obsolete)

Superseded by Document.FullName

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.SetPathName("C:\\myXMLFiles\\test.xml");
// use now:
Application.ActiveDocument.FullName = "C:\\myXMLFiles\\test.xml";
```

Method: SetPathName (*strPath* as String)

Description
The method SetPathName sets the path of the active document. SetPathName only copies the string and does not check if the path is valid. All succeeding save operations are done into this file.

29.3.2.10.55 StartChanges

Method: StartChanges()

Description
After StartChanges is executed XMLSpy will not update its editor windows until Document.EndChanges is called. This increases performance of complex tasks to the XML structure.

Errors
1400 The object is no longer valid.

29.3.2.10.56 Suggestions

Property: Suggestions as Array

Description
This property contains the last valid user suggestions for this document. The XMLSpy generated suggestions can be modified before they are shown to the user in the OnBeforeShowSuggestions event.

Errors
1400 The object is no longer valid.
1407 Invalid parameter or invalid address for the return parameter was specified.

29.3.2.10.57 SwitchViewMode

Method: SwitchViewMode (*nMode* as SPYViewModes) as Boolean

Return value
Returns true if view mode is switched.

Description
The method sets the current view mode of the document in XMLSpy. See also Document.CurrentViewMode.

Errors
1400 The object is no longer valid.
1407 Invalid address for the return parameter was specified.
1417 Invalid view mode specified.

29.3.2.10.58 TextView

Property: TextView as TextView

Description
This property provides access to the text view functionality of the document.

Errors
1400 The object is no longer valid.

© 2016-2022 Altova GmbH

1407 Invalid address for the return parameter was specified.

29.3.2.10.59 Title

Property: Title as String (read-only)

Description
Title contains the file name of the document. To get the path and filename of the file use FullName.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.60 TransformXSL

Method: TransformXSL()

Description
TransformXSL processes the XML document via the associated XSL file. See Document.AssignXSL on how to place a reference to a XSL file into the document.

Errors
1400 The document object is no longer valid.
1411 Error during transformation process.

29.3.2.10.61 TransformXSLEx

Method: TransformXSLEx(*nAction* as SPYDialogAction)

Description
TransformXSLEx processes the XML document via the associated XSL file. The parameter specifies whether a dialog asking for the result document name should pop up or not. See Document.AssignXSL on how to place a reference to a XSL file into the document.

Errors
1400 The document object is no longer valid.
1411 Error during transformation process.

29.3.2.10.62 TransformXSLFO

Method: TransformXSLFO()

Description
TransformXSLFO processes the XML document via the associated XSLFO file. See AssignXSLFO on how to place a reference to a XSLFO file into the document. You need to assign a FOP processor to XMLSpy before you can use this method.

Errors
1400 The document object is no longer valid.
1411 Error during transformation process.

29.3.2.10.63 TreatXBRLInconsistenciesAsErrors

Property: TreatXBRLInconsistenciesAsErrors as Boolean

Description
If this is set to `true` the `Document.IsValid()` method will return `false` for XBRL instances containing inconsistencies as defined by the XBRL Specification. The default value of this property is `false`.

Errors
1400 The document object is no longer valid.
1407 Invalid address for the return parameter was specified.

29.3.2.10.64 UpdateViews

Method: UpdateViews()

Description
To redraw the Enhanced Grid View and the Tree View call UpdateViews. This can be important after you changed the XMLData structure of a document. This method does not redraw the text view of XMLSpy.

Errors
1400 The document object is no longer valid.

29.3.2.10.65 UpdateXMLData

Method: UpdateXMLData() as Boolean

Description
The XMLData tree is updated from the current view. Please note that this can fail in case of the TextView if the current XML text is not well-formed. This is not necessary if CurrentViewMode is *spyViewGrid* or *spyViewAuthentic* because these views keep the XMLData updated.

Available with TypeLibrary version 1.5

Errors
1400 The document object is no longer valid.

© 2016-2022 Altova GmbH

29.3.2.11 Documents

Properties
Count
Item

Methods
NewAuthenticFile
NewFile
NewFileFromText
OpenAuthenticFile
OpenFile
OpenURL
OpenURLDialog

Description
This object represents the set of documents currently open in XMLSpy. Use this object to open further documents or iterate through already opened documents.

Examples

```
' ----------------------------------------
' XMLSpy scripting environment - VBScript
' iterate through open documents
' ----------------------------------------
Dim objDocuments
Set objDocuments = Application.Documents

For Each objDoc In objDocuments
        'do something useful with your document
        objDoc.SetActiveDocument()
Next
```

```
// ----------------------------------------
// XMLSpy scripting environment - JScript
// close all open documents
// ----------------------------------------
for (var iter = new Enumerator (Application.Documents);
    ! iter.atEnd();
    iter.moveNext())
{
        // MsgBox ("Closing file " + iter.item().Name);
        iter.item().Close (true);
}
```

29.3.2.11.1 Count

Property: Count as long

Description

Count of open documents.

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

29.3.2.11.2 Item

Method: Item (*n* as long) as Document

Description
Gets the document with the index n in this collection. Index is 1-based.

Errors
1600 Invalid `Documents` object
1601 Invalid input parameter

29.3.2.11.3 NewAuthenticFile

Method: NewAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as Document

Parameters
strSPSPath
The path to the SPS document.

strXMLPath
The new XML document name.

Return Value
The method returns the new document.

Description
NewAuthenticFile creates a new XML file and opens it in Authentic View using SPS design strSPSPath.

29.3.2.11.4 NewFile

Method: NewFile (*strFile* as String, *strType* as String) as Document

Parameters
strFile
Full path of new file.

strType
Type of new file as string (i.e. "xml", "xsd", ...)

© 2016-2022 Altova GmbH

Return Value
Returns the new file.

Description
NewFile creates a new file of type strType (i.e. "xml"). The newly created file is also the ActiveDocument.

29.3.2.11.5 NewFileFromText

Method: NewFileFromText (*strText* as String, *strType* as String) as Document

Parameters
strText
The content of the new document in plain text.

strType
Type of the document to create (i.e. "xml").

Return Value
The method returns the new document.

Description
NewFileFromText creates a new document with strText as its content.

29.3.2.11.6 OpenAuthenticFile

Method: OpenAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as Document

Parameters
strSPSPath
The path to the SPS document.

strXMLPath
The path to the XML document (can be empty).

Return Value
The method returns the new document.

Description
OpenAuthenticFile opens an XML file or database in Authentic View using SPS design strSPSPath.

29.3.2.11.7 OpenFile

Method: OpenFile (*strPath* as String, *bDialog* as Boolean) as Document

Parameters
strPath
Path and file name of file to open.

bDialog
Show dialogs for user input.

Return Value
Returns the opened file on success.

Description
OpenFile opens the file strPath. If bDialog is TRUE, a file-dialog will be displayed.

Example

```
Dim objDoc As Document
Set objDoc = objSpy.Documents.OpenFile(strFile, False)
```

29.3.2.11.8 OpenURL

Method: OpenURL (*strURL* as String, *nURLType* as SPYURLTypes, *nLoading* as SPYLoading, *strUser* as String, *strPassword* as String) as Document

Parameters
strURL
URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

nLoading
Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentification. Can be empty.

Return Value
The method returns the opened document.

Description
OpenURL opens the URL strURL.

29.3.2.11.9 OpenURLDialog

Method: OpenURLDialog (*strURL* as String, *nURLType* as SPYURLTypes, *nLoading* as SPYLoading, *strUser* as String, *strPassword* as String) as Document

Parameters
strURL
URL to open as document.

© 2016-2022 Altova GmbH

nURLType
Type of document to open. Set to -1 for auto detection.

nLoading
Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentification. Can be empty.

Return Value
The method returns the opened document.

Description
OpenURLDialog displays the "open URL" dialog to the user and presets the input fields with the given parameters.

29.3.2.12 DTDSchemaGeneratorDlg

Properties and Methods

Standard automation properties
Application
Parent

DTDSchemaFormat
ValueList
TypeDetection
FrequentElements
MergeAllEqualNamed
ResolveEntities
AttributeTypeDefinition
GlobalAttributes
OnlyStringEnums
MaxEnumLength
OutputPath
OutputPathDialogAction

Description
Use this object to configure the generation of a schema or DTD. The method GenerateDTDOrSchemaEx expects a DTDSchemaGeneratorDlg as parameter to configure the generation as well as the associated user interactions.

29.3.2.12.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.2 AttributeTypeDefinition

Property: AttributeTypeDefinition as SPYAttributeTypeDefinition

Description
Specifies how attribute definitions get merged.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.3 DTDSchemaFormat

Property: DTDSchemaFormat as SPYDTDSchemaFormat

Description
Sets the schema output format to DTD, or W3C.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.4 FrequentElements

Property: FrequentElements as SPYFrequentElements

Description
Shall the types for all elements be defined as global? Use that value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.12.5 GlobalAttributes

Property: GlobalAttributes as Boolean

Description
Shall attributes with same name and type be resolved globally?

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.6 MaxEnumLength

Property: MaxEnumLength as Integer

Description
Specifies the maximum number of characters allowed for enumeration names. If one value is longer than this, no enumeration will be generated.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.7 MergeAllEqualNamed

Property: MergeAllEqualNamed as Boolean

Description
Shall types of all elements with the same name be merged into one type?

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.8 OnlyStringEnums

Property: OnlyStringEnums as Boolean

Description
Specifies if enumerations will be created only for plain strings or all types of values.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.9 OutputPath

Property: OutputPath as String

Description
Selects the file name for the generated schema/DTD.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.10 OutputPathDialogAction

Property: OutputPathDialogAction as SPYDialogAction

Description
Defines how the sub-dialog for selecting the schema/DTD output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the OutputPath property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.11 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.12 ResolveEntities

Property: ResolveEntities as Boolean

Description
Shall all entities be resolved before generation starts? If yes, an info-set will be built.

Errors

© 2016-2022 Altova GmbH

3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.13 TypeDetection

Property: TypeDetection as SPYTypeDetection

Description
Specifies granularity of simple type detection.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.12.14 ValueList

Property: ValueList as Integer

Description
Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

Errors
3000 The object is no longer valid.
3001 Invalid address for the return parameter was specified.

29.3.2.13 ElementList

Properties
Count
Item

Methods
RemoveElement

Description
Element lists are used for different purposes during export and import of data. Depending on this purpose, different properties of ElementListItem are used.

It can hold
- a list of table names returned by a call to Application.GetDatabaseTables,
- a list of field names retuned by a call to Application.GetDatabaseImportElementList or Application.GetTextImportElementList,
- a field name filter list used in Application.ImportFromDatabase and Application.ImportFromText,
- a list of table names and counts for their rows and columns as returned by calls to GetExportElementList or
- a field name filter list used in Document.ExportToDatabase and Document.ExportToText.

29.3.2.13.1 Count

Property: Count as long (read-only)

Description
Count of elements in this collection.

29.3.2.13.2 Item

Method: Item(n as long) as ElementListItem

Description
Gets the element with the index n from this collection. The first item has index 1.

29.3.2.13.3 RemoveElement

Method: RemoveElement(Index as long)

Description
RemoveElement removes the element Index from the collection. The first Item has index 1.

29.3.2.14 ElementListItem

Properties
Name

ElementKind

FieldCount
RecordCount

Description
An element in an ElementList. Usage of its properties depends on the purpose of the element list. For details see ElementList.

29.3.2.14.1 ElementKind

Property: ElementKind as SPYXMLDataKind

Description
Specifies if a field should be imported as XML element (data value of spyXMLDataElement) or attribute (data value of spyXMLDataAttr).

© 2016-2022 Altova GmbH

29.3.2.14.2 FieldCount

Property: FieldCount as long (read-only)

Description
Count of fields (i.e. columns) in the table described by this element. This property is only valid after a call to Document.GetExportElementList.

29.3.2.14.3 Name

Property: Name as String (read-only)

Description
Name of the element. This is either the name of a table or a field, depending on the purpose of th element list.

29.3.2.14.4 RecordCount

Property: RecordCount as long (read-only)

Description
Count of records (i.e. rows) in the table described by this element. This property is only valid after a call to Document.GetExportElementList.

29.3.2.15 ExportSettings

Properties

ElementList

EntitiesToText

ExportAllElements
SubLevelLimit

FromAttributes
FromSingleSubElements
FromTextValues

CreateKeys
IndependentPrimaryKey

Namespace

ExportCompleteXML
StartFromElement

Description
ExportSettings contains options used during export of XML data to a database or text file. See Import and export of data for a general overview.

29.3.2.15.1 CreateKeys

Property: CreateKeys as Boolean

Description
This property turns creation of keys (i.e. primary key and foreign key) on or off. Default is True.

29.3.2.15.2 ElementList

Property: ElementList as ElementList

Description
Default is empty list. This list of elements defines which fields will be exported. To get the list of available fields use Document.GetExportElementList. It is possible to prevent exporting columns by removing elements from this list with ElementList.RemoveElement before passing it to Document.ExportToDatabase or Document.ExportToText.

29.3.2.15.3 EntitiesToText

Property: EntitiesToText as Boolean

Description
Defines if XML entities should be converted to text or left as they are during export. Default is True.

29.3.2.15.4 ExportAllElements

Property: ExportAllElements as Boolean

Description
If set to true, all elements in the document will be exported. If set to false, then ExportSettings.SubLevelLimit is used to restrict the number of sub levels to export. Default is true.

29.3.2.15.5 ExportCompleteXML

Property: ExportCompleteXML as Boolean

Description
Defines whether the complete XML is exported or only the element specified by StartFromElement and its children. Default is True.

© 2016-2022 Altova GmbH

29.3.2.15.6 FromAttributes

Property: FromAttributes as Boolean

Description
Set FromAttributes to false if no export data should be created from attributes. Default is True.

29.3.2.15.7 FromSingleSubElements

Property: FromSingleSubElements as Boolean

Description
Set FromSingleSubElements to false if no export data should be created from elements. Default is True.

29.3.2.15.8 FromTextValues

Property: FromTextValues as Boolean

Description
Set FromTextValues to false if no export data should be created from text values. Default is True.

29.3.2.15.9 IndependentPrimaryKey

Property: IndependentPrimaryKey as Boolean

Description
Turns creation of independent primary key counter for every element on or off. If ExportSettings.CreateKeys is False, this property will be ignored. Default is True.

29.3.2.15.10 Namespace

Property: Namespace as SPYExportNamespace

Description
The default setting removes all namespace prefixes from the element names. In some database formats the colon is not a legal character. Default is spyNoNamespace.

29.3.2.15.11 StartFromElement

Property: StartFromElement as String

Description
Specifies the start element for the export. This property is only considered when ExportCompleteXML is false.

29.3.2.15.12 SubLevelLimit

Property: SubLevelLimit as Integer

Description
Defines the number of sub levels to include for the export. Default is 0. This property is ignored if ExportSettings.ExportAllElements is true.

29.3.2.16 FileSelectionDlg

Properties and Methods

Standard automation properties
Application
Parent

Dialog properties
FullName

Acceptance or cancellation of action that caused event
DialogAction

Description
The dialog object allows you to receive information about an event and pass back information to the event handler in the same way as with a user dialog. Use the FileSelectionDlg.FullName to select or modify the file path and set the FileSelectionDlg.DialogAction property to cancel or agree with the action that caused the event.

29.3.2.16.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
2400 The object is no longer valid.
2401 Invalid address for the return parameter was specified.

29.3.2.16.2 DialogAction

Property: DialogAction as SPYDialogAction

Description
If you want your script to perform the file selection operation without any user interaction necessary, simulate user interaction by either setting the property to *spyDialogOK(0)* or *spyDialogCancel(1)*.

© 2016-2022 Altova GmbH

To allow your script to fill in the default values but let the user see and react on the dialog, use the value *spyDialogUserInput(2)*. If you receive a FileSelectionDlg object in an event handler, *spyDialogUserInput(2)* is not supported and will be interpreted as *spyDialogOK(0)*.

Errors

2400 The object is no longer valid.
2401 Invalid value for dialog action or invalid address for the return parameter was specified.

29.3.2.16.3 FullName

Property: FullName as String

Description
Access the full path of the file the gets selected by the dialog. Most events that pass a FileSelectionDlg object to you allow you modify this value and thus influence the action that caused the event (e.g. load or save to a different location).

Errors

2400 The object is no longer valid.
2401 Invalid address for the return parameter was specified.

29.3.2.16.4 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors

2400 The object is no longer valid.
2401 Invalid address for the return parameter was specified.

29.3.2.17 FindInFilesDlg

Properties and Methods

Standard automation properties
Application
Parent

Find
RegularExpression
Replace
DoReplace
ReplaceOnDisk
MatchWholeWord
MatchCase

SearchLocation
StartFolder
IncludeSubfolders
SearchInProjectFilesDoExternal
FileExtension
AdvancedXMLSearch
XMLElementNames
XMLElementContents
XMLAttributeNames
XMLAttributeContents
XMLComments
XMLCData
XMLPI
XMLRest
ShowResult

Description
Use this object to configure the search (or replacement) for strings in files. The method FindInFiles expects a FindInFilesDlg as parameter.

29.3.2.17.1 AdvancedXMLSearch

Property: AdvancedXMLSearch as Boolean

Description
Specifies if the XML search properties (XMLElementNames, XMLElementContents, XMLAttributeNames, XMLAttributeContents, XMLComments, XMLCData, XMLPI and XMLRest) are considered. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.2 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.3 DoReplace

Property: DoReplace as Boolean

© 2016-2022 Altova GmbH

Description
Specifies if the matched string is replaced by the string defined in Replace. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.4 FileExtension

Property: FileExtension as String

Description
Specifies the file filter of the files that should be considered during the search. Multiple file filters must be delimited with a semicolon (eg: *.xml;*.dtd;a*.xsd). Use the wildcards * and ? to define the file filter.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.5 Find

Property: Find as String

Description
Specifies the string to search for.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.6 IncludeSubfolders

Property: IncludeSubfolders as Boolean

Description
Specifies if subfolders are searched too. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.7 MatchCase

Property: MatchCase as Boolean

Description
Specifies if the search is case sensitive. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.8 MatchWholeWord

Property: MatchWholeWord as Boolean

Description
Specifies whether the whole word or just a part of it must match. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.9 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.10 RegularExpression

Property: RegularExpression as Boolean

Description
Specifies if Find contains a regular expression. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.11 Replace

Property: Replace as String

Description
Specifies the replacement string. The matched string is only replaced if DoReplace is set true.

Errors
3500 The object is no longer valid.

© 2016-2022 Altova GmbH

3501 Invalid address for the return parameter was specified.

29.3.2.17.12 ReplaceOnDisk

Property: ReplaceOnDisk as Boolean

Description
Specifies if the replacement is done directly on disk. The modified file is not opened. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.13 SearchInProjectFilesDoExternal

Property: SearchInProjectFilesDoExternal as Boolean

Description
Specifies if the external folders in the open project are searched, when a project search is performed. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.14 SearchLocation

Property: SearchLocation as SPYFindInFilesSearchLocation

Description
Specifies the location of the search. The default is spyFindInFiles_Documents.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.15 ShowResult

Property: ShowResult as Boolean

Description
Specifies if the result is displayed in the Find in Files output window. The default is false.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.16 StartFolder

Property: StartFolder as String

Description
Specifies the folder where the disk search starts.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.17 XMLAttributeContents

Property: XMLAttributeContents as Boolean

Description
Specifies if attribute contents are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.18 XMLAttributeNames

Property: XMLAttributeNames as Boolean

Description
Specifies if attribute names are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.19 XMLCData

Property: XMLCData as Boolean

Description
Specifies if CData tags are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.17.20 XMLComments

Property: XMLComments as Boolean

Description
Specifies if comments are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.21 XMLElementContents

Property: XMLElementContents as Boolean

Description
Specifies if element contents are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.22 XMLElementNames

Property: XMLElementNames as Boolean

Description
Specifies if element names are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.23 XMLPI

Property: XMLPI as Boolean

Description
Specifies if XML processing instructions are searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.17.24 XMLRest

Property: XMLRest as Boolean

Description
Specifies if the rest of the XML (which is not covered by the other XML search properties) is searched when AdvancedXMLSearch is true. The default is true.

Errors
3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

29.3.2.18 FindInFilesResult

Properties and Methods

Standard automation properties
Application
Parent

Count
Item

Path
Document

Description
This object represents a file that matched the search criteria. It contains a list of FindInFilesResultMatch objects that describe the matching position.

29.3.2.18.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.18.2 Count

Property: Count as long (read-only)

Description

© 2016-2022 Altova GmbH

Count of elements in this collection.

29.3.2.18.3 Document

Property: Path as Document (read-only)

Description
This property returns the Document object if the matched file is already open in XMLSpy.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.18.4 Item

Method: Item(n as long) as FindInFilesResultMatch

Description
Gets the element with the index n from this collection. The first item has index 1.

29.3.2.18.5 Parent

Property: Parent as FindInFilesResults (read-only)

Description
Access the parent of the object.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.18.6 Path

Property: Path as String (read-only)

Description
Returns the path of the file that matched the search criteria.

Errors
3700 The object is no longer valid.
3701 Invalid address for the return parameter was specified.

29.3.2.19 FindInFilesResultMatch

Properties and Methods

Standard automation properties
Application
Parent

Line
Position
Length
LineText
Replaced

Description
Contains the exact position in the file of the matched string.

29.3.2.19.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.2 Length

Property: Length as Long (read-only)

Description
Returns the length of the matched string.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.3 Line

Property: Line as Long (read-only)

Description
Returns the line number of the match. The line numbering starts with 0.

© 2016-2022 Altova GmbH

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.4 LineText

Property: LineText as String (read-only)

Description
Returns the text of the line.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.5 Parent

Property: Parent as FindInFilesResult (read-only)

Description
Access the parent of the object.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.6 Position

Property: Position as Long (read-only)

Description
Returns the start position of the match in the line. The position numbering starts with 0.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.19.7 Replaced

Property: Replaced as Boolean (read-only)

Description

True if the matched string was replaced.

Errors
3800 The object is no longer valid.
3801 Invalid address for the return parameter was specified.

29.3.2.20 FindInFilesResults

Properties and Methods

Standard automation properties
Application
Parent

Count
Item

Description
This is the result of the FindInFiles method. It is a list of FindInFilesResult objects.

29.3.2.20.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
3600 The object is no longer valid.
3601 Invalid address for the return parameter was specified.

29.3.2.20.2 Count

Property: Count as long (read-only)

Description
Count of elements in this collection.

29.3.2.20.3 Item

Method: Item(n as long) as FindInFilesResult

Description
Gets the element with the index n from this collection. The first item has index 1.

© 2016-2022 Altova GmbH

29.3.2.20.4 Parent

Property: Parent as Application (read-only)

Description
Access the parent of the object.

Errors
3600 The object is no longer valid.
3601 Invalid address for the return parameter was specified.

29.3.2.21 GenerateSampleXMLDlg

Properties and Methods

Standard automation properties
Application
Parent

NonMandatoryAttributes
NonMandatoryElements
RepeatCount
FillAttributesWithSampleData
FillElementsWithSampleData
ContentOfNillableElementsIsNonMandatory
TryToUseNonAbstractTypes
SchemaOrDTDAssignment
LocalNameOfRootElement
NamespaceURIOfRootElement
OptionsDialogAction

Properties that are no longer supported
TakeFirstChoice - obsolete
FillWithSampleData - obsolete
Optimization - obsolete

Description
Used to set the parameters for the generation of sample XML instances based on a W3C schema or DTD.

29.3.2.21.1 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors

2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.2 ChoiceMode

Property: `ChoiceMode` as `SPYSampleXMLGenerationChoiceMode`

Description
Specifies which elements will be generated.

Errors

2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.3 ConsiderSampleValueHints

Property: `ConsiderSampleValueHints` as Boolean

Description
Selects whether to use `SampleValueHints` or not.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.4 ContentOfNillableElementsIsNonMandatory

Property: ContentOfNillableElementsIsNonMandatory as Boolean

Description
If true, the contents of elements that are nillable will not be treated as mandatory.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.21.5 FillAttributesWithSampleData

Property: FillAttributesWithSampleData as Boolean

Description
If true, attributes will have sample content.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.6 FillElementsWithSampleData

Property: FillElementsWithSampleData as Boolean

Description
If true, elements will have sample content.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.7 FillWithSampleData - obsolete

Property: FillWithSampleData as Boolean

Description
Do no longer access this property. Use FillAttributesWithSampleData and FillElementsWithSampleData, instead.

Errors
0001 The property is no longer accessible.

29.3.2.21.8 LocalNameOfRootElement

Property: LocalNameOfRootElement as String

Description
Specifies the local name of the root element for the generated sample XML.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.9 NamespaceURIOfRootElement

Property: NamespaceURIOfRootElement as String

Description
Specifies the namespace URI of the root element for the generated sample XML.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.10 NonMandatoryAttributes

Property: NonMandatoryAttributes as Boolean

Description
If true attributes which are not mandatory are created in the sample XML instance file.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.11 NonMandatoryElements

Property: NonMandatoryElements as Boolean

Description
If `true`, elements which are not mandatory are created in the sample XML instance file.

Errors
2200 The object is no longer valid.
2201 Invalid address was specified for the return parameter.

29.3.2.21.12 Optimization - obsolete

Property: Optimization as SPYSampleXMLGenerationOptimization

Description
Do not use this property any longer. Use ChoiceMode and NonMandatoryElements.

Errors
0001 The property is no longer accessible.

© 2016-2022 Altova GmbH

29.3.2.21.13 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
2200 The object is no longer valid.
2201 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.21.14 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.15 RepeatCount

Property: RepeatCount as long

Description
Number of elements to create for repeated types.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.16 SampleValueHints

Property: `SampleValueHints` as `SPYSampleXMLGenerationSampleValueHints`

Description
Specifies how to select data for the generated sample file.

Errors
2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

29.3.2.21.17 SchemaOrDTDAssignment

Property: SchemaOrDTDAssignment as SPYSampleXMLGenerationSchemaOrDTDAssignment

Description
Specifies in which way a reference to the related schema or DTD - which is this document - will be generated into the sample XML.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.21.18 TakeFirstChoice - obsolete

Property: TakeFirstChoice as Boolean

Description
Do no longer use this property.

Errors
0001 The property is no longer accessible.

29.3.2.21.19 TryToUseNonAbstractTypes

Property: TryToUseNonAbstractTypes as Boolean

Description
If true, tries to use a non-abstract type for xsi:type, if element has an abstract type.

Errors
2200 The object is no longer valid.
2201 Invalid address for the return parameter was specified.

29.3.2.22 GridView

Methods
Deselect
Select

SetFocus

Properties
CurrentFocus

© 2016-2022 Altova GmbH

IsVisible

Description
GridView Class

29.3.2.22.1 Events

29.3.2.22.1.1 OnBeforeDrag

Event: OnBeforeDrag() as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeDrag()
	' On_BeforeStartEditing = False  ' to prohibit dragging
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeDrag()
{
	// return false;  /* to prohibit dragging */
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (4, ...)   // nEventId = 4
```

Description
This event gets fired on an attempt to drag an XMLData element on the grid view. Return *false* to prevent dragging the data element to a different position.

29.3.2.22.1.2 OnBeforeDrop

Event: OnBeforeDrop(*objXMLData* as XMLData) as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeDrop(objXMLData)
	' On_BeforeStartEditing = False  ' to prohibit dropping
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeDrop(objXMLData)
{
	// return false;  /* to prohibit dropping */
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (5, ...)   // nEventId = 5
```

Description
This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.

29.3.2.22.1.3 OnBeforeStartEditing

Event: OnBeforeStartEditing(*objXMLData* as XMLData, *bEditingName* as Boolean)as Boolean

XMLSpy scripting environment - VBScript:
```
Function On_BeforeStartEditing(objXMLData, bEditingName)
        ' On_BeforeStartEditing = False  ' to prohibit editing the field
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeStartEditing(objXMLData, bEditingName)
{
        // return false;  /* to prohibit editing the field */
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (1, ...) // nEventId = 1

Description
This event gets fired before the editing mode for a grid cell gets entered. If the parameter *bEditingName* is true, the name part of the element will be edited, it its value is false, the value part will be edited.

29.3.2.22.1.4 OnEditingFinished

Event: OnEditingFinished(*objXMLData* as XMLData, *bEditingName* as Boolean)

XMLSpy scripting environment - VBScript:
```
Function On_EditingFinished(objXMLData, bEditingName)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_EditingFinished(objXMLData, bEditingName)
{
}
```

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (2, ...) // nEventId = 2

Description
This event gets fired when the editing mode of a grid cell gets left. The parameter *bEditingName* specifies if the name part of the element has been edited.

© 2016-2022 Altova GmbH

29.3.2.22.1.5 OnFocusChanged

Event: OnFocusChanged(*objXMLData* as XMLData, *bSetFocus* as Boolean, *bEditingName* as Boolean)

XMLSpy scripting environment - VBScript:
Function On_FocusChanged(*objXMLData*, *bSetFocus*, *bEditingName*)
End Function

XMLSpy scripting environment - JScript:
function On_FocusChanged(*objXMLData*, *bSetFocus*, *bEditingName*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent (3, ...) // nEventId = 3

Description
This event gets fired whenever a grid cell receives or looses the cursor focus. If the parameter *bEditingName* is *true*, focus of the name part of the grid element has changed. Otherwise, focus of the value part has changed.

29.3.2.22.2 CurrentFocus

Property: CurrentFocus as XMLData

Description
Holds the XML element with the current focus. This property is read-only.

29.3.2.22.3 Deselect

Method: Deselect(*pData* as XMLData)

Description
Deselects the element pData in the grid view.

29.3.2.22.4 IsVisible

Property: IsVisible as Boolean

Description
True if the grid view is the active view of the document. This property is read-only.

29.3.2.22.5 Select

Method: Select (*pData* as XMLData)

Description
Selects the XML element pData in the grid view.

29.3.2.22.6 SetFocus

Method: SetFocus (*pFocusData* as XMLData)

Description
Sets the focus to the element pFocusData in the grid view.

29.3.2.23 SchemaDocumentationDlg

Properties and Methods

Standard automation properties
Application
Parent

Interaction and visibility properties
OutputFile
OutputFileDialogAction
OptionsDialogAction
ShowProgressBar
ShowResult

Document generation options and methods
OutputFormat
UseFixedDesign
SPSFile
EmbedDiagrams
DiagramFormat
MultipleOutputFiles
EmbedCSSInHTML
CreateDiagramsFolder
GenerateRelativeLinks

IncludeAll
IncludeIndex
IncludeGlobalAttributes
IncludeGlobalElements
IncludeLocalAttributes
IncludeLocalElements
IncludeGroups
IncludeComplexTypes

© 2016-2022 Altova GmbH

IncludeSimpleTypes
IncludeAttributeGroups
IncludeRedefines
IncludeReferencedSchemas

AllDetails
ShowDiagram
ShowNamespace
ShowType
ShowChildren
ShowUsedBy
ShowProperties
ShowSingleFacets
ShowPatterns
ShowEnumerations
ShowAttributes
ShowIdentityConstraints
ShowAnnotations
ShowSourceCode

Description
This object combines all options for schema document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of schema documentation. However, before using the object you have to set the SetOutputFile property to a valid file path. Use OptionsDialogAction, OutputFileDialogAction and ShowProgressBar to specify the level of user interaction desired. You can use IncludeAll and AllDetails to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.23.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
2900 The object is no longer valid.

29.3.2.23.2 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is false.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.4 DiagramFormat

Property: DiagramFormat as SPYImageKind

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is PNG.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.23.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description
Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.7 GenerateRelativeLinks

Property: GenerateRelativeLinks as Boolean

Description
Set this property to true, to create relative paths to local files. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is false.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description
Use this method to mark or unmark all include options.

Errors
2900 The object is no longer valid.

29.3.2.23.9 IncludeAttributeGroups

Property: IncludeAttributeGroups as Boolean

Description

Set this property to true, to include attribute groups in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.10 IncludeComplexTypes

Property: IncludeComplexTypes as Boolean

Description
Set this property to true, to include complex types in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.11 IncludeGlobalAttributes

Property: IncludeGlobalAttributes as Boolean

Description
Set this property to true, to include global attributes in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.12 IncludeGlobalElements

Property: IncludeGlobalElements as Boolean

Description
Set this property to true, to include global elements in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.23.13 IncludeGroups

Property: IncludeGroups as Boolean

Description
Set this property to true, to include groups in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.14 IncludeIndex

Property: IncludeIndex as Boolean

Description
Set this property to true, to include an index in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.15 IncludeLocalAttributes

Property: IncludeLocalAttributes as Boolean

Description
Set this property to true, to include local attributes in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.16 IncludeLocalElements

Property: IncludeLocalElements as Boolean

Description

Set this property to true, to include local elements in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.17 IncludeRedefines

Property: IncludeRedefines as Boolean

Description
Set this property to true, to include redefines in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.18 IncludeReferencedSchemas

Property: IncludeReferencedSchemas as Boolean

Description
Set this property to true, to include referenced schemas in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.19 IncludeSimpleTypes

Property: IncludeSimpleTypes as Boolean

Description
Set this property to true, to include simple types in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.23.20 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description
Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is false.

Errors
2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.21 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.22 OutputFile

Property: OutputFile as String

Description
Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateSchemaDocumentation.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.23 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.24 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is HTML.

Errors

2900 The object is no longer valid.
2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.23.25 Parent

Property: Parent as Dialogs (read-only)

Description

Access the parent of the object.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.26 ShowAnnotations

Property: ShowAnnotations as Boolean

Description

Set this property to true, to show the annotations to a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.23.27 ShowAttributes

Property: ShowAttributes as Boolean

Description
Set this property to true, to show the type definitions attributes in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.28 ShowChildren

Property: ShowChildren as Boolean

Description
Set this property to true, to show the children of a type definition as links in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.29 ShowDiagram

Property: ShowDiagram as Boolean

Description
Set this property to true, to show type definitions as diagrams in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.30 ShowEnumerations

Property: ShowEnumerations as Boolean

Description

Set this property to true, to show the enumerations contained in a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.31 ShowIdentityConstraints

Property: ShowIdentityConstraints as Boolean

Description

Set this property to true, to show a type definitions identity constraints in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.32 ShowNamespace

Property: ShowNamespace as Boolean

Description

Set this property to true, to show the namespace of type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.33 ShowPatterns

Property: ShowPatterns as Boolean

Description

Set this property to true, to show the patterns of a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.

© 2016-2022 Altova GmbH

2901 Invalid address for the return parameter was specified.

29.3.2.23.34 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.35 ShowProperties

Property: ShowProperties as Boolean

Description
Set this property to true, to show the type definition properties in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.36 ShowResult

Property: ShowResult as Boolean

Description
Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.37 ShowSingleFacets

Property: ShowSingleFacets as Boolean

Description
Set this property to true, to show the facets of a type definition in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.38 ShowSourceCode

Property: ShowSourceCode as Boolean

Description
Set this property to true, to show the XML source code for type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.39 ShowType

Property: ShowType as Boolean

Description
Set this property to true, to show the type of type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.40 ShowUsedBy

Property: ShowUsedBy as Boolean

Description
Set this property to true, to show the used-by relation for type definitions in the schema documentation. The property is initialized with the value used during the last call to Document.GenerateSchemaDocumentation. The default for the first run is true.

Errors

© 2016-2022 Altova GmbH

2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.41 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.23.42 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

29.3.2.24 SpyProject

Methods
CloseProject
SaveProject
SaveProjectAs

Properties
RootItems
ProjectFile

Description
SpyProject Class

29.3.2.24.1 CloseProject

Declaration: CloseProject(*bDiscardChanges* as Boolean, *bCloseFiles* as Boolean, *bDialog* as Boolean)

Parameters
bDiscardChanges
Set bDiscardChanges to FALSE if you want to save the changes of the open project files and the project.

bCloseFiles
Set bCloseFiles to TRUE to close all open project files.

bDialog
Show dialogs for user input.

Description
CloseProject closes the current project.

29.3.2.24.2 ProjectFile

Declaration: ProjectFile as String

Description
Path and filename of the project.

29.3.2.24.3 RootItems

Declaration: RootItems as SpyProjectItems

Description
Root level of collection of project items.

29.3.2.24.4 SaveProject

Declaration: SaveProject

Description
SaveProject saves the current project.

29.3.2.24.5 SaveProjectAs

Declaration: SaveProjectAs (strPath as String, bDialog as Boolean)

Parameters
strPath
Full path with file name of new project file.

bDialog
If bDialog is TRUE, a file-dialog will be displayed.

Description
SaveProjectAs stores the project data into a new location.

© 2016-2022 Altova GmbH

29.3.2.25 SpyProjectItem

Methods
Open

Properties
ChildItems
ParentItem
FileExtensions
ItemType
Name
Path
ValidateWith
XMLForXSLTransformation
XSLForXMLTransformation
XSLTransformationFileExtension
XSLTransformationFolder

Description
SpyProjectItem Class

29.3.2.25.1 ChildItems

Declaration: ChildItems as SpyProjectItems

Description
If the item is a folder, ChildItems is the collection of the folder content.

29.3.2.25.2 FileExtensions

Declaration: FileExtensions as String

Description
Used to set the file extensions if the project item is a folder.

29.3.2.25.3 ItemType

Declaration: ItemType as SPYProjectItemTypes

Description
This property is read-only.

29.3.2.25.4 Name

Declaration: Name as String

Description
Name of the project item. This property is read-only.

29.3.2.25.5 Open

Declaration: Open as Document

Return Value
The project item opened as document.

Description
Opens the project item.

29.3.2.25.6 ParentItem

Declaration: ParentItem as SpyProjectItem

Description
Parent item of the current project item. Can be NULL (Nothing) if the project item is a top-level item.

29.3.2.25.7 Path

Declaration: Path as String

Description
Path of project item. This property is read-only.

29.3.2.25.8 ValidateWith

Declaration: ValidateWith as String

Description
Used to set the schema/DTD for validation.

29.3.2.25.9 XMLForXSLTransformation

Declaration: XMLForXSLTransformation as String

Description
Used to set the XML for XSL transformation.

© 2016-2022 Altova GmbH

29.3.2.25.10 XSLForXMLTransformation

Declaration: XSLForXMLTransformation as String

Description
Used to set the XSL for XML transformation.

29.3.2.25.11 XSLTransformationFileExtension

Declaration: XSLTransformationFileExtension as String

Description
Used to set the file extension for XSL transformation output files.

29.3.2.25.12 XSLTransformationFolder

Declaration: XSLTransformationFolder as String

Description
Used to set the destination folder for XSL transformation output files.

29.3.2.26 SpyProjectItems

Methods
AddFile
AddFolder
AddURL
RemoveItem

Properties
Count
Item

Description
SpyProjectItems Class

29.3.2.26.1 AddFile

Declaration: AddFile (*strPath* as String)

Parameters

strPath
Full path with file name of new project item

Description

The method adds a new file to the collection of project items.

29.3.2.26.2 AddFolder

Declaration: AddFolder (*strName* as String)

Parameters

strName
Name of the new folder.

Description
The method AddFolder adds a folder with the name strName to the collection of project items.

29.3.2.26.3 AddURL

Declaration: AddURL (*strURL* as String, *nURLType* as SPYURLTypes, *strUser* as String, *strPassword* as String, *bSave* as Boolean)

Description

strURL
URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentification. Can be empty.

bSave
Save user and password information.

Description
The method adds an URL item to the project collection.

29.3.2.26.4 Count

Declaration: Count as long

Description
This property gets the count of project items in the collection. The property is read-only.

© 2016-2022 Altova GmbH

29.3.2.26.5 Item

Declaration: Item (*n* as long) as SpyProjectItem

Description
Retrieves the n-th element of the collection of project items. The first item has index 1.

29.3.2.26.6 RemoveItem

Declaration: RemoveItem (*pItem* as SpyProjectItem)

Description
RemoveItem deletes the item pItem from the collection of project items.

29.3.2.27 TextImportExportSettings

Properties for import only
ImportFile

Properties for export only
DestinationFolder
FileExtension
CommentIncluded
RemoveDelimiter
RemoveNewline

Properties for import and export
HeaderRow
FieldDelimiter
EnclosingCharacter
Encoding
EncodingByteOrder

Description
TextImportExportSettings contains options common to text import and export functions.

29.3.2.27.1 CommentIncluded

Property: CommentIncluded as Boolean

Description
This property tells whether additional comments are added to the generated text file. Default is true. This property is used only when exporting to text files.

29.3.2.27.2 DestinationFolder

Property: DestinationFolder as String

Description
The property DestinationFolder sets the folder where the created files are saved during text export.

29.3.2.27.3 EnclosingCharacter

Property: EnclosingCharacter as SPYTextEnclosing

Description
This property defines the character that encloses all field values for import and export. Default is spyNoEnclosing.

29.3.2.27.4 Encoding

Property: Encoding as String

Description
The property Encoding sets the character encoding for the text files for importing and exporting.

29.3.2.27.5 EncodingByteOrder

Property: EncodingByteOrder as SPYEncodingByteOrder

Description
The property EncodingByteOrder sets the byte order for Unicode characters. Default is spyNONE.

29.3.2.27.6 FieldDelimiter

Property: FieldDelimiter as SPYTextDelimiters

Description
The property FieldDelimiter defines the delimiter between the fields during import and export. Default is spyTabulator.

29.3.2.27.7 FileExtension

Property: FileExtension as String

Description
This property sets the file extension for files created on text export.

© 2016-2022 Altova GmbH

29.3.2.27.8 HeaderRow

Property: HeaderRow as Boolean

Description
The property HeaderRow is used during import and export. Set HeaderRow true on import, if the first line of the text file contains the names of the columns. Set HeaderRow true on export, if the first line in the created text files should contain the name of the columns. Default value is true.

29.3.2.27.9 ImportFile

Property: ImportFile as String

Description
This property is used to set the text file for import. The string has to be a full qualified path. See also Import and Export.

29.3.2.27.10 RemoveDelimiter

Property: RemoveDelimiter as Boolean

Description
The property RemoveDelimiter defines whether characters in the text that are equal to the delimiter character are removed. Default is false. This property is used only when exporting to text files.

29.3.2.27.11 RemoveNewline

Property: RemoveNewline as Boolean

Description
The property RemoveNewline defines whether newline characters in the text are removed. Default is false. This property is used only when exporting to text files.

29.3.2.28 TextView

Properties and Methods

Application
Parent

LineFromPosition
PositionFromLine
LineLength
SelText
GetRangeText

ReplaceText
MoveCaret
GoToLineChar
SelectText
SelectionStart
SelectionEnd
Text
LineCount
Length

Description

29.3.2.28.1 Events

29.3.2.28.1.1 OnBeforeShowSuggestions

Event: OnBeforeShowSuggestions() as Boolean

Description
This event gets fired before a suggestion window is shown. The Document property Suggestions contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the Suggestions property. The best location for this is the OnDocumentOpened event. To prevent the suggestion window to show up return false and true to continue its display.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_BeforeShowSuggestions()
End Function
```

XMLSpy scripting environment - JScript:
```
function On_BeforeShowSuggestions()
{
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (33, ...) // nEventId = 33
```

29.3.2.28.1.2 OnChar

Event: OnChar(nChar as Long, bExistSuggestion as Boolean) as Boolean

Description

© 2016-2022 Altova GmbH

This event gets fired on each key stroke. The parameter nChar is the key that was pressed and bExistSuggestions tells whether a XMLSpy generated suggestions window is displayed after this key. The Document property Suggestions contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the Suggestions property. The best location for this is the OnDocumentOpened event. To prevent the suggestion window to show up return false and true to continue its display.
It is also possible to create a new suggestions window when none is provided by XMLSpy. Set the Document property Suggestions to a string array with your recommendations and return true.
This event is fired before the OnBeforeShowSuggestions event. If you prevent to show the suggestion window by returning false then OnBeforeShowSuggestions is not fired.

Examples
Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:
```
Function On_Char(nChar, bExistSuggestions)
End Function
```

XMLSpy scripting environment - JScript:
```
function On_Char(nChar, bExistSuggestions)
{
}
```

XMLSpy IDE Plugin:
```
IXMLSpyPlugIn.OnEvent (35, ...) // nEventId = 35
```

29.3.2.28.2 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.3 GetRangeText

Method: GetRangeText(nStart as Long, nEnd as Long) as String

Description
Returns the text in the specified range.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.4 GoToLineChar

Method: GoToLineChar(nLine as Long, nChar as Long)

Description
Moves the caret to the specified line and character position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.5 Length

Property: Length as Long

Description
Returns the character count of the document.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.6 LineCount

Property: LineCount as Long

Description
Returns the number of lines in the document.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.7 LineFromPosition

Method: LineFromPosition(nCharPos as Long) as Long

Description
Returns the line number of the character position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.28.8 LineLength

Method: LineLength(nLine as Long) as Long

Description
Returns the length of the line.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.9 MoveCaret

Method: MoveCaret(nDiff as Long)

Description
Moves the caret nDiff characters.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.10 Parent

Property: Parent as Document (read-only)

Description
Access the parent of the object.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.11 PositionFromLine

Method: PositionFromLine(nLine as Long) as Long

Description
Returns the start position of the line.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.12 ReplaceText

Method: ReplaceText(nPosFrom as Long, nPosTill as Long, sText as String)

Description
Replaces the text in the specified range.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.13 SelectionEnd

Property: SelectionEnd as Long

Description
Returns/sets the text selection end position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.14 SelectionStart

Property: SelectionStart as Long

Description
Returns/sets the text selection start position.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.15 SelectText

Method: SelectText(nPosFrom as Long, nPosTill as Long)

Description
Selects the text in the specified range.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.28.16 SelText

Property: SelText as String

Description
Returns/sets the selected text.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.28.17 Text

Property: Text as String

Description
Returns/sets the document text.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29 WSDLDocumentationDlg

Properties and Methods

Standard automation properties
Application
Parent

Interaction and visibility properties
GlobalElementsAndTypesOnly
OptionsDialogAction
OutputFile
OutputFileDialogAction
SeparateSchemaDocument
ShowProgressBar
ShowResult

Document generation options and methods
OutputFormat
UseFixedDesign
SPSFile
EmbedDiagrams
DiagramFormat
MultipleOutputFiles
EmbedCSSInHTML
CreateDiagramsFolder

IncludeAll

IncludeBinding
IncludeImportedWSDLFiles
IncludeMessages
IncludeOverview
IncludePortType
IncludeService
IncludeTypes

AllDetails
ShowBindingDiagram
ShowExtensibility
ShowMessageParts
ShowPort
ShowPortTypeDiagram
ShowPortTypeOperations
ShowServiceDiagram
ShowSourceCode
ShowTypesDiagram
ShowUsedBy

Description
This object combines all options for WSDL document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the OutputFile property to a valid file path. Use OptionsDialogAction, OutputFileDialogAction and ShowProgressBar to specify the level of user interaction desired. You can use IncludeAll and AllDetails to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.29.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
4300 The object is no longer valid.

29.3.2.29.2 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors

© 2016-2022 Altova GmbH

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is false.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.4 DiagramFormat

Property: DiagramFormat as SPYImageKind

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is PNG.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description
Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.7 GlobalElementsAndTypesOnly

Property: GlobalElementsAndTypesOnly as Boolean

Description
Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description
Use this method to mark or unmark all include options.

Errors
4300 The object is no longer valid.

29.3.2.29.9 IncludeBinding

Property: IncludeBinding as Boolean

Description
Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

© 2016-2022 Altova GmbH

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.10 IncludeImportedWSDLFiles

Property: IncludeImportedWSDLFiles as Boolean

Description
Set this property to true, to include imported WSDL finles in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.11 IncludeMessages

Property: IncludeMessages as Boolean

Description
Set this property to true, to include messages in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.12 IncludeOverview

Property: IncludeOverview as Boolean

Description
Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.13 IncludePortType

Property: IncludePortType as Boolean

Description

Set this property to true, to include port types in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.14 IncludeService

Property: IncludeService as Boolean

Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.15 IncludeTypes

Property: IncludeTypes as Boolean

Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.16 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is false.

Errors

3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.29.17 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.18 OutputFile

Property: OutputFile as String

Description
Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateWSDLDocumentation.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.19 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.20 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is HTML.

Errors

3900 The object is no longer valid.
3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.29.21 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.22 SeparateSchemaDocument

Property: SeparateSchemaDocument as Boolean

Description
Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.23 ShowBindingDiagram

Property: ShowBindingDiagram as Boolean

Description
Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.29.24 ShowExtensibility

Property: ShowExtensibility as Boolean

Description
Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.25 ShowMessageParts

Property: ShowMessageParts as Boolean

Description
Set this property to true, to show message parts of messges in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.26 ShowPort

Property: ShowPort as Boolean

Description
Set this property to true, to show service ports in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.27 ShowPortTypeDiagram

Property: ShowPortTypeDiagram as Boolean

Description

Set this property to true, to show port type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.28 ShowPortTypeOperations

Property: ShowPortTypeOperations as Boolean

Description
Set this property to true, to show port type operations in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.29 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.30 ShowResult

Property: ShowResult as Boolean

Description
Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.29.31 ShowServiceDiagram

Property: ShowServiceDiagram as Boolean

Description
Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.32 ShowSourceCode

Property: ShowSourceCode as Boolean

Description
Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.33 ShowTypesDiagram

Property: ShowTypesDiagram as Boolean

Description
Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.34 ShowUsedBy

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation. The default for the first run is true.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.35 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.29.36 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.30 WSDL20DocumentationDlg

Properties and Methods

Standard automation properties
Application
Parent

Interaction and visibility properties
GlobalElementsAndTypesOnly
OptionsDialogAction
OutputFile
OutputFileDialogAction
SeparateSchemaDocument
ShowProgressBar

© 2016-2022 Altova GmbH

ShowResult

Document generation options and methods
OutputFormat
UseFixedDesign
SPSFile
EmbedDiagrams
DiagramFormat
MultipleOutputFiles
EmbedCSSInHTML
CreateDiagramsFolder

IncludeAll
IncludeBinding
IncludeImportedWSDLFiles
IncludeInterface
IncludeOverview
IncludeService
IncludeTypes

AllDetails
ShowBindingDiagram
ShowExtensibility
ShowEndpoint
ShowFault
ShowInterfaceDiagram
ShowOperation
ShowServiceDiagram
ShowSourceCode
ShowTypesDiagram
ShowUsedBy

Description
This object combines all options for WSDL document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the OutputFile property to a valid file path. Use OptionsDialogAction, OutputFileDialogAction and ShowProgressBar to specify the level of user interaction desired. You can use IncludeAll and AllDetails to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.30.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
4300 The object is no longer valid.

29.3.2.30.2 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateWSD20LDocumentation. The default for the first run is false.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.4 DiagramFormat

Property: DiagramFormat as SPYImageKind

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is PNG.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description

© 2016-2022 Altova GmbH

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.7 GlobalElementsAndTypesOnly

Property: GlobalElementsAndTypesOnly as Boolean

Description

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.30.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

4300 The object is no longer valid.

29.3.2.30.9 IncludeBinding

Property: IncludeBinding as Boolean

Description
Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.10 IncludeImportedWSDLFiles

Property: IncludeImportedWSDLFiles as Boolean

Description
Set this property to true, to include imported WSDL finles in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.11 IncludeInterface

Property: IncludeInterface as Boolean

Description
Set this property to true, to include interfaces in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.12 IncludeOverview

Property: IncludeOverview as Boolean

Description
Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

© 2016-2022 Altova GmbH

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.13 IncludeService

Property: IncludeService as Boolean

Description
Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.14 IncludeTypes

Property: IncludeTypes as Boolean

Description
Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.15 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description
Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is false.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.16 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.17 OutputFile

Property: OutputFile as String

Description
Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateWSDL20Documentation.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.18 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.19 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat

Description
Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is HTML.

Errors

© 2016-2022 Altova GmbH

4300 The object is no longer valid.
4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.30.20 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.21 SeparateSchemaDocument

Property: SeparateSchemaDocument as Boolean

Description
Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors
3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

29.3.2.30.22 ShowBindingDiagram

Property: ShowBindingDiagram as Boolean

Description
Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.23 ShowEndpoint

Property: ShowEndpoint as Boolean

Description

Set this property to true, to show service endpoints in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.24 ShowExtensibility

Property: ShowExtensibility as Boolean

Description
Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.25 ShowFault

Property: ShowFault as Boolean

Description
Set this property to true, to show faults in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.26 ShowInterfaceDiagram

Property: ShowInterfaceDiagram as Boolean

Description
Set this property to true, to show interface diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.27 ShowOperation

Property: ShowOperation as Boolean

Description
Set this property to true, to show interface and binding operations in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.28 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.29 ShowResult

Property: ShowResult as Boolean

Description
Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.30 ShowServiceDiagram

Property: ShowServiceDiagram as Boolean

Description

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.31 ShowSourceCode

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.32 ShowTypesDiagram

Property: ShowTypesDiagram as Boolean

Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.33 ShowUsedBy

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.30.34 SPSFile

Property: SPSFile as String

Description
Full path and name of the SPS file that will be used to generate the documentation.

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.30.35 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
4300 The object is no longer valid.
4301 Invalid address for the return parameter was specified.

29.3.2.31 XBRLDocumentationDlg

Properties and Methods

Standard automation properties
Application
Parent

Interaction and visibility properties
OptionsDialogAction
OutputFile
OutputFileDialogAction
ShowProgressBar
ShowResult

Document generation options and methods
OutputFormat
UseFixedDesign
SPSFile
EmbedDiagrams
DiagramFormat

EmbedCSSInHTML
CreateDiagramsFolder

IncludeAll
IncludeOverview
IncludeNamespacePrefixes
IncludeGlobalElements
IncludeDefinitionLinkroles
IncludePresentationLinkroles
IncludeCalculationLinkroles

AllDetails
ShowDiagram
ShowSubstitutiongroup
ShowItemtype
ShowBalance
ShowPeriod
ShowAbstract
ShowNillable
ShowLabels
ShowReferences
ShowLinkbaseReferences

ShortQualifiedName
ShowImportedElements

Description
This object combines all options for XBRL document generation as they are available through user interface dialog boxes in XMLSpy. The document generation options are initialized with the values used during the last generation of XBRL documentation. However, before using the object you have to set the OutputFile property to a valid file path. Use OptionsDialogAction, OutputFileDialogAction and ShowProgressBar to specify the level of user interaction desired. You can use IncludeAll and AllDetails to set whole option groups at once or the individual properties to operate on a finer granularity.

29.3.2.31.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description
Use this method to turn all details options on or off.

Errors
4400 The object is no longer valid.

© 2016-2022 Altova GmbH

29.3.2.31.2 Application

Property: Application as Application (read-only)

Description
Access the XMLSpy application object.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description
Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is false.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.4 DiagramFormat

Property: DiagramFormat as SPYImageKind

Description
This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is PNG.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description
Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized

with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description
Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.7 IncludeAll

Method: IncludeAll (i_blnclude as Boolean)

Description
Use this method to mark or unmark all include options.

Errors
4400 The object is no longer valid.

29.3.2.31.8 IncludeCalculationLinkroles

Property: IncludeCalculationLinkroles as Boolean

Description
Set this property to true, to include calculation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.9 IncludeDefinitionLinkroles

Property: IncludeDefinitionLinkroles as Boolean

Description
Set this property to true, to include definition linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.10 IncludeGlobalElements

Property: IncludeGlobalElements as Boolean

Description
Set this property to true, to include global elements in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.11 IncludeNamespacePrefixes

Property: IncludeNamespacePrefixes as Boolean

Description
Set this property to true, to include namespace prefixes in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.12 IncludeOverview

Property: IncludeOverview as Boolean

Description
Set this property to true, to include an overview in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.13 IncludePresentationLinkroles

Property: IncludePresentationLinkroles as Boolean

Description
Set this property to true, to include presentation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.14 OptionsDialogAction

Property: OptionsDialogAction as SPYDialogAction

Description
To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4400 The object is no longer valid.
4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.31.15 OutputFile

Property: OutputFile as String

Description
Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Document.GenerateXBRLDocumentation.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.16 OutputFileDialogAction

Property: OutputFileDialogAction as SPYDialogAction

Description
To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in OutputFile should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors
4400 The object is no longer valid.
4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.31.17 OutputFormat

Property: OutputFormat as SPYSchemaDocumentationFormat

Description
Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is HTML.

Errors
4400 The object is no longer valid.
4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

29.3.2.31.18 Parent

Property: Parent as Dialogs (read-only)

Description
Access the parent of the object.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.19 ShortQualifiedName

Property: ShortQualifiedName as Boolean

Description

Set this property to true, to use short qualified names in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.20 ShowAbstract

Property: ShowAbstract as Boolean

Description
Set this property to true, to show abstracts in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.21 ShowBalance

Property: ShowBalance as Boolean

Description
Set this property to true, to show balances in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.22 ShowDiagram

Property: ShowDiagram as Boolean

Description
Set this property to true, to show diagrams in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.23 ShowImportedElements

Property: ShowImportedElements as Boolean

Description
Set this property to true, to show imported elements in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.24 ShowItemtype

Property: ShowItemtype as Boolean

Description
Set this property to true, to show item types in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.25 ShowLabels

Property: ShowLabels as Boolean

Description
Set this property to true, to show labels in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.26 ShowLinkbaseReferences

Property: ShowLinkbaseReferences as Boolean

Description

Set this property to true, to show linkbase references in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.27 ShowNillable

Property: ShowNillable as Boolean

Description
Set this property to true, to show nillable properties in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.28 ShowPeriod

Property: ShowPeriod as Boolean

Description
Set this property to true, to show periods in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.29 ShowProgressBar

Property: ShowProgressBar as Boolean

Description
Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

© 2016-2022 Altova GmbH

29.3.2.31.30 ShowReferences

Property: ShowReferences as Boolean

Description
Set this property to true, to show references in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.31 ShowResult

Property: ShowResult as Boolean

Description
Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in XMLSpy. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.32 ShowSubstitutiongroup

Property: ShowSubstitutiongroup as Boolean

Description
Set this property to true, to show substitution groups in the XBRL documentation. The property is initialized with the value used during the last call to Document.GenerateXBRLDocumentation. The default for the first run is true.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.33 SPSFile

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.31.34 UseFixedDesign

Property: UseFixedDesign as Boolean

Description
Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors
4400 The object is no longer valid.
4401 Invalid address for the return parameter was specified.

29.3.2.32 XMLData

Properties
Kind
Name
TextValue

HasChildren
MayHaveChildren
Parent

Methods
GetFirstChild
GetNextChild
GetCurrentChild

InsertChild
InsertChildAfter
InsertChildBefore
AppendChild

EraseAllChildren
EraseChild
EraseCurrentChild

IsSameNode

CountChildren
CountChildrenKind

© 2016-2022 Altova GmbH

GetChild
GetChildAttribute
GetChildElement
GetChildKind
GetNamespacePrefixForURI

HasChildrenKind
SetTextValueXMLEncoded

Description
The XMLData interface provides direct XML-level access to a document. You can read and directly modify the XML representation of the document. However, please, note the following restrictions:

- The XMLData representation is only valid when the document is shown in grid view or authentic view.
- When in authentic view, additional XMLData elements are automatically inserted as parents of each visible document element. Typically this is an XMLData of kind spyXMLDataElement with the Name property set to 'Text'.
- When you use the XMLData interface while in a different view mode you will not receive errors, but changes are not reflected to the view and might get lost during the next view switch.

Note also:

- Setting a new text value for an XML element is possible if the element does not have non-text children. A text value can be set even if the element has attributes.
- When setting a new text value for an XML element which has more than one text child, the latter will be deleted and replaced by one new text child.
- When reading the text value of an XML element which has more than one text child, only the value of the first text child will be returned.

Objects of this class represent the different atomic parts of an XML document. See the enumeration type SPYXMLDataKind for the available part types. Each part knows its children, thus forming a XMLData tree with Document.RootElement at its top. To get the top element of the document content - ignoring the XML header - use Document.DataRoot. For an examples on how to traverse the XMLData tree, see GetNextChild.

29.3.2.32.1 AppendChild

Declaration: AppendChild (*pNewData* as XMLData)

Description
AppendChild appends pNewData as last child to the XMLData object.

Errors

1500	The XMLData object is no longer valid.
1505	Invalid XMLData kind was specified.
1506	Invalid address for the return parameter was specified.
1507	Element cannot have Children
1512	Cyclic insertion - new data element is already part of document
1514	Invalid XMLData kind was specified for this position.
1900	Document must not be modified

Example

```
Dim objCurrentParent As XMLData
Dim objNewChild As XMLData

Set objNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)
Set objCurrentParent = objSpy.ActiveDocument.RootElement

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

29.3.2.32.2 CountChildren

Declaration: CountChildren as long

Description
CountChildren gets the number of children.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.

29.3.2.32.3 CountChildrenKind

Declaration: CountChildrenKind (*nKind* as SPYXMLDataKind) as long

Description
CountChildrenKind gets the number of children of the specific kind.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.

29.3.2.32.4 EraseAllChildren

Declaration: EraseAllChildren

Description
EraseAllChildren deletes all associated children of the XMLData object.

Errors

© 2016-2022 Altova GmbH

1500 The XMLData object is no longer valid.
1900 Document must not be modified

Example
The sample erases all elements of the active document.

```
Dim objCurrentParent As XMLData

Set objCurrentParent = objSpy.ActiveDocument.RootElement
objCurrentParent.EraseAllChildren
```

29.3.2.32.5 EraseChild

Method: EraseChild (Child as XMLData)

Description
Deletes the given child node.

Errors
1500 Invalid object.
1506 Invalid input xml
1510 Invalid parameter.

29.3.2.32.6 EraseCurrentChild

Declaration: EraseCurrentChild

Description
EraseCurrentChild deletes the current XMLData child object. Before you call EraseCurrentChild you must initialize an internal iterator with XMLData.GetFirstChild. After deleting the current child, EraseCurrentChild increments the internal iterator of the XMLData element. No error is returned when the last child gets erased and the iterator is moved past the end of the child list. The next call to EraseCurrentChild however, will return error 1503.

Errors
1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object, or the iterator points past the last child.
1900 Document must not be modified

Examples

```
// ------------------------------------
// XMLSpy scripting environment - JScript
// erase all children of XMLData
// ------------------------------------
// let's get an XMLData element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application.ActiveDocument.GridView.CurrentFocus;
```

```
// the following line would be shorter, of course
//        objList.EraseAllChildren ();

// but we want to demonstrate the usage of EraseCurrentChild
if ((objList != null) && (objList.HasChildren))
{
        try
        {
                objEle = objList.GetFirstChild(-1);
                while (objEle != null)
                        objList.EraseCurrentChild();
                        // no need to call GetNextChild
        }
        catch (err)
                // 1503 - we reached end of child list
                { if ((err.number & 0xffff) != 1503) throw (err); }
}
```

29.3.2.32.7 GetChild

Declaration: GetChild (*position* as long) as XMLData

Return Value
Returns an XML element as XMLData object.

Description
GetChild() returns a reference to the child at the given index (zero-based).

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.8 GetChildAttribute

Method: GetChildAttribute (strName as string) child as XMLData object (`NULL` on error)

Description
Retrieves the attribute having the given name.

Errors
1500 Invalid object.
1510 Invalid parameter.

© 2016-2022 Altova GmbH

29.3.2.32.9 GetChildElement

Method: GetChildElement (strName as string, nIndex as long) child as XMLData object (`NULL` on error)

Description
Retrieves the Nth child element with the given name.

Errors
1500 Invalid object.
1510 Invalid parameter.

29.3.2.32.10 GetChildKind

Declaration: GetChildKind (*position* as long, *nKind* as SPYXMLDataKind) as XMLData

Return Value
Returns an XML element as XMLData object.

Description
GetChildKind() returns a reference to a child of this kind at the given index (zero-based). The position parameter is relative to the number of children of the specified kind and not to all children of the object.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.11 GetCurrentChild

Declaration: GetCurrentChild as XMLData

Return Value
Returns an XML element as XMLData object.

Description
GetCurrentChild gets the current child. Before you call GetCurrentChild you must initialize an internal iterator with XMLData.GetFirstChild.

Errors
1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object.
1510 Invalid address for the return parameter was specified.

29.3.2.32.12 GetFirstChild

Declaration: GetFirstChild (*nKind* as SPYXMLDataKind) as XMLData

Return Value
Returns an XML element as XMLData object.

Description
GetFirstChild initializes a new iterator and returns the first child. Set nKind = -1 to get an iterator for all kinds of children.
REMARK: The iterator is stored inside the XMLData object and gets destroyed when the XMLData object gets destroyed. Be sure to keep a reference to this object as long as you want to use GetCurrentChild, GetNextChild or EraseCurrentChild.

Errors
1500 The XMLData object is no longer valid.
1501 Invalid XMLData kind was specified.
1504 Element has no children of specified kind.
1510 Invalid address for the return parameter was specified.

Example
See the example at XMLData.GetNextChild.

29.3.2.32.13 GetNamespacePrefixForURI

Method: GetNamespacePrefixForURI (strURI as string) strNS as string

Description
Returns the namespace prefix of the supplied URI.

Errors
1500 Invalid object.
1510 Invalid parameter.

29.3.2.32.14 GetNextChild

Declaration: GetNextChild as XMLData

Return Value
Returns an XML element as XMLData object.

Description
GetNextChild steps to the next child of this element. Before you call GetNextChild you must initialize an internal iterator with XMLData.GetFirstChild.

© 2016-2022 Altova GmbH

Check for the last child of the element as shown in the sample below.

Errors

1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object.
1510 Invalid address for the return parameter was specified.

Examples

```
' ---------------------------------------------
' VBA code snippet - iterate XMLData children
' ---------------------------------------------
On Error Resume Next
Set objParent = objSpy.ActiveDocument.RootElement

'get elements of all kinds
Set objCurrentChild = objParent.GetFirstChild(-1)

Do
        'do something useful with the child

        'step to next child
        Set objCurrentChild = objParent.GetNextChild
Loop Until (Err.Number - vbObjectError = 1503)
```

```
// ---------------------------------------
// XMLSpy scripting environment - JScript
// iterate through children of XMLData
// ---------------------------------------
try
{
        var objXMLData = ... // initialize somehow
        var objChild = objXMLData.GetFirstChild(-1);

        while (true)
        {
                // do something usefull with objChild

                objChild = objXMLData.GetNextChild();
        }
}
catch (err)
{
        if ((err.number & 0xffff) == 1504)
                ; // element has no children
        else if ((err.number & 0xffff) == 1503)
                ;       // last child reached
        else
                throw (err);
}
```

29.3.2.32.15 GetTextValueXMLDecoded

Method: GetTextValueXMLDecoded ()as string

Description
Gets the decoded text value of the XML.

Errors
1500 Invalid object.
1510 Invalid parameter.

29.3.2.32.16 HasChildren

Declaration: HasChildren as Boolean

Description
The property is true if the object is the parent of other XMLData objects. This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.17 HasChildrenKind

Declaration: HasChildrenKind (*nKind* as SPYXMLDataKind) as Boolean

Description
The method returns true if the object is the parent of other XMLData objects of the specific kind.

Available with TypeLibrary version 1.5

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.18 InsertChild

Declaration: InsertChild (*pNewData* as XMLData)

Description
InsertChild inserts the new child before the current child (see also XMLData.GetFirstChild, XMLData.GetNextChild to set the current child).

© 2016-2022 Altova GmbH

Errors

1500 The XMLData object is no longer valid.
1503 No iterator is initialized for this XMLData object.
1505 Invalid XMLData kind was specified.
1506 Invalid address for the return parameter was specified.
1507 Element cannot have Children
1512 Cyclic insertion - new data element is already part of document
1514 Invalid XMLData kind was specified for this position.
1900 Document must not be modified

29.3.2.32.19 InsertChildAfter

Method: InsertChildBefore (Node as XMLData, NewData as XMLData)

Description
Inserts a new XML node (supplied with the second parameter) after the specified node (first parameter).

Errors

1500 Invalid object.
1506 Invalid input xml
1507 No children allowed
1510 Invalid parameter.
1512 Child is already added
1514 Invalid kind at position

29.3.2.32.20 InsertChildBefore

Method: InsertChildBefore (Node as XMLData, NewData as XMLData)

Description
Inserts a new XML node (supplied with the second parameter) before the specified node (first parameter).

Errors

1500 Invalid object.
1506 Invalid input xml
1507 No children allowed
1510 Invalid parameter.
1512 Child is already added
1514 Invalid kind at position

29.3.2.32.21 IsSameNode

Declaration: IsSameNode (*pNodeToCompare* as XMLData) as Boolean

Description

Returns true if pNodeToCompare references the same node as the object itself.

Errors
1500 The XMLData object is no longer valid.
1506 Invalid address for the return parameter was specified.

29.3.2.32.22 Kind

Declaration: Kind as SPYXMLDataKind

Description
Kind of this XMLData object. This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.23 MayHaveChildren

Declaration: MayHaveChildren as Boolean

Description
Indicates whether it is allowed to add children to this XMLData object.
This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.24 Name

Declaration: Name as String

Description
Used to modify and to get the name of the XMLData object.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.25 Parent

Declaration: Parent as XMLData

Return value

© 2016-2022 Altova GmbH

Parent as XMLData object. Nothing (or NULL) if there is no parent element.

Description
Parent of this element. This property is read-only.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.2.32.26 SetTextValueXMLEncoded

Method: SetTextValueXMLEncoded (*strVal* as String)

Description
Sets the encoded text value of the XML.

Errors
1500 Invalid object.
1513 Modification not allowed.

29.3.2.32.27 TextValue

Declaration: TextValue as String

Description
Used to modify and to get the text value of this XMLData object.

Errors
1500 The XMLData object is no longer valid.
1510 Invalid address for the return parameter was specified.

29.3.3 Interfaces (obsolete)

Interfaces contained in this book are obsolete. It is recommended to migrate your applications to the new interfaces. See the different properties and methods in this book for migration hints.

29.3.3.1 AuthenticEvent (obsolete)

Superseded by AuthenticView and AuthenticRange

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.

DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage - except for AuthenticDataTransfer - is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interface. All functionality available up to now in DocEditView, DocEditSelection, DocEditEvent and DocEditDataTransfer is now available via AuthenticView, AuthenticRange and AuthenticDataTransfer. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Properties

altKey
altLeft
ctrlKey
ctrlLeft
shiftKey
shiftLeft

keyCode
repeat

button

clientX
clientY

dataTransfer

srcElement
fromElement

propertyName

cancelBubble
returnValue

type

Description
DocEditEvent interface.

© 2016-2022 Altova GmbH

29.3.3.1.1 altKey (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if (Application.ActiveDocument.DocEditView.event.altKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//       MsgBox ("alt key is down");
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
   if (i_nVirtualKeyStatus & spyAltKeyMask)
      MsgBox ("alt key is down");
}
```

Declaration: altKey as Boolean

Description
True if the right ALT key is pressed.

29.3.3.1.2 altLeft (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyDown ()
// {
//     if (Application.ActiveDocument.DocEditView.event.altKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("alt key is down");
// }
// use now:
function On_AuthenticView_KeyDown (SPYKeyEvent i_eKeyEvent, long i_nKeyCode, SPYVirtualKeyMask
i_nVirtualKeyStatus)
{
    if (i_nVirtualKeyStatus & spyAltKeyMask)
        MsgBox ("alt key is down");
}
```

Declaration: altLeft as Boolean

Description
True if the left ALT key is pressed.

29.3.3.1.3 button (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditButtonDown ()
// {
//     if (Application.ActiveDocument.DocEditView.event.button == 1)
//         MsgBox ("left mouse button down detected");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyLeftButtonDownMask)
        MsgBox ("left mouse button down detected");
}
```


© 2016-2022 Altova GmbH

Declaration: button as long

Description
Specifies which mouse button is pressed:

0	No button is pressed.
1	Left button is pressed.
2	Right button is pressed.
3	Left and right buttons are both pressed.
4	Middle button is pressed.
5	Left and middle buttons both are pressed.
6	Right and middle buttons are both pressed.
7	All three buttons are pressed.

29.3.3.1.4 cancelBubble (obsolete)

Superseded by the boolean return value of following event handler functions
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

Returning *true* from an event handler function signals that the event has beend handled and normal event handling should be aborted.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if (Application.ActiveDocument.DocEditView.event.keyCode == 0x20)
//     {
//        // cancel key processing, swallow spaces :-)
//        Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//     }
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if (i_nKeyCode == 0x20)
        return true;  // cancel key processing, swallow spaces :-)
}
```

Declaration: cancelBubble as Boolean

Description
Set cancelBubble to TRUE if the default event handler should not be called.

29.3.3.1.5 clientX (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnBeforeDrop (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     MsgBox ("moving over " + Application.ActiveDocument.DocEditView.event.clientX +
//             "/" + Application.ActiveDocument.DocEditView.event.clientY);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
   if (i_eMouseEvent & spyMouseMoveMask)
      MsgBox ("moving over " + i_nXPos + "/" + i_nYPos);
}
```

Declaration: clientX as long

Description
X value of the current mouse position in client coordinates.

29.3.3.1.6 clientY (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnBeforeDrop (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

© 2016-2022 Altova GmbH

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     MsgBox ("moving over " + Application.ActiveDocument.DocEditView.event.clientX +
//             "/" + Application.ActiveDocument.DocEditView.event.clientY);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyMouseMoveMask)
        MsgBox ("moving over " + i_nXPos + "/" + i_nYPos);
}
```

Declaration: clientY as long

Description
Y value of the current mouse position in client coordinates.

29.3.3.1.7 ctrlKey (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     if (Application.ActiveDocument.DocEditView.event.ctrlKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("control key is down");
// }
// use now:
```

```
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyCtrlKeyMask)
        MsgBox ("control key is down");
}
```

Declaration: ctrlKey as Boolean

Description
True if the right CTRL key is pressed.

29.3.3.1.8 ctrlLeft (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     if (Application.ActiveDocument.DocEditView.event.ctrlKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//       MsgBox ("control key is down");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyCtrlKeyMask)
        MsgBox ("control key is down");
}
```

Declaration: ctrlLeft as Boolean

Description
True if the left CTRL key is pressed.

© 2016-2022 Altova GmbH

29.3.3.1.9 dataTransfer (obsolete)

Superseded by parameters to
AuthenticView.OnBeforeDrop (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDrop ()
// {
//     if (Application.ActiveDocument.DocEditView.event.dataTransfer != null)
//         if (! Application.ActiveDocument.DocEditView.event.dataTransfer.ownDrag)
//         {
//           // cancel key processing, don't drop foreign objects :-)
//           Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//         }
// }
// use now:
function On_AuthenticView_BeforeDrop (long i_nXPos, long i_nYPos,
                                      IAuthenticRange *i_ipRange,
                                      IAuthenticDataTransfer *i_ipData)
{
    if (i_ipRange != null)
        if (! i_ipRange.ownDrag)
            return true;  // cancel key processing, don't drop foreign objects :-)

    return false;
}
```

Declaration: dataTransfer as Variant

Description
Property dataTransfer.

29.3.3.1.10 fromElement (obsolete)

Not supported

Declaration: fromElement as Variant (not supported)

Description
Currently no event sets this property.

29.3.3.1.11 keyCode (obsolete)

Superseded by a parameter to AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if (Application.ActiveDocument.DocEditView.event.keyCode == 0x20)
//     {
//        // cancel key processing, swallow spaces :-)
//        Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//     }
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
   if (i_nKeyCode == 0x20)
      return true;  // cancel key processing, swallow spaces :-)
}
```

Declaration: keyCode as long

Description
Keycode of the currently pressed key. This property is read-write.

29.3.3.1.12 propertyName (obsolete)

Not supported

Declaration: propertyName as String (not supported)

Description

© 2016-2022 Altova GmbH

Currently no event sets this property.

29.3.3.1.13 repeat (obsolete)

Not supported

Declaration: repeat as Boolean (not supported)

Description
True if the onkeydown event is repeated.

29.3.3.1.14 returnValue (obsolete)

No longer supported

Declaration: returnValue as Variant

Description
Use returnValue to set a return value for your event handler.

29.3.3.1.15 shiftKey (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDragOver ()
// {
//     if (Application.ActiveDocument.DocEditView.event.shiftKey ||
```

```
//        Application.ActiveDocument.DocEditView.event.shiftLeft)
//       MsgBox ("shift key is down");
// }
// use now:
function On_AuthenticView_DragOver (long i_nXPos, long i_nYPos,
                                    SPYMouseEvent i_eMouseEvent,
                                    IAuthenticRange *i_ipRange,
                                    IAuthenticDataTransfer *i_ipData)
{
   if (i_eMouseEvent & spyShiftKeyMask)
      MsgBox ("shift key is down");
}
```

Declaration: shiftKey as Boolean

Description
True if the right SHIFT key is pressed.

29.3.3.1.16 shiftLeft (obsolete)

Superseded by parameters to
AuthenticView.OnKeyboardEvent (On_AuthenticView_KeyPressed)
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDragOver ()
// {
//     if (Application.ActiveDocument.DocEditView.event.shiftKey ||
//         Application.ActiveDocument.DocEditView.event.shiftLeft)
//       MsgBox ("shift key is down");
// }
// use now:
function On_AuthenticView_DragOver (long i_nXPos, long i_nYPos,
                                    SPYMouseEvent i_eMouseEvent,
                                    IAuthenticRange *i_ipRange,
                                    IAuthenticDataTransfer *i_ipData)
{
   if (i_eMouseEvent & spyShiftKeyMask)
      MsgBox ("shift key is down");
}
```


© 2016-2022 Altova GmbH

Declaration: shiftLeft as Boolean

Description
True if the left SHIFT key is pressed.

29.3.3.1.17 srcElement (obsolete)

Superseded by parameters to
AuthenticView.OnMouseEvent (On_AuthenticView_MouseEvent)
AuthenticView.OnBeforeDrop (On_AuthenticView_BeforeDrop)
AuthenticView.OnDragOver (On_AuthenticView_DragOver)

The event object that holds the information of the last event is now replaced by parameters to the different event handler functions to simplify data access. The event object will be supported for a not yet defined period of time for compatibility reasons. No improvements are planned. It is highly recommended to migrate to the new event handler functions.

With the new event handler function, a range object selecting this element is provided instead of the XMLData element currently below the mouse cursor.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     var objEvent = Application.ActiveDocument.DocEditView.event;
//     if (objEvent.srcElement != null)
//         MsgBox ("moving over " + objEvent.srcElement.Parent.Name);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos, SPYMouseEvent i_eMouseEvent,
IAuthenticRange *i_ipRange)
{
   if ((i_eMouseEvent & spyMouseMoveMask) &&
       (i_ipRange != null))
       MsgBox ("moving over " + i_ipRange.FirstXMLData.Parent.Name);
}
```

Declaration: srcElement as Variant

Description
Element which fires the current event. This is usually an XMLData object.

29.3.3.1.18 type (obsolete)

Not supported

Declaration: type as String (not supported)

Description
Currently no event sets this property.

29.3.3.2 AuthenticSelection (obsolete)

Superseded by AuthenticRange

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage - except for AuthenticDataTransfer - is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interface. All functionality available up to now in DocEditView, DocEditSelection, DocEditEvent and DocEditDataTransfer is now available via AuthenticView, AuthenticRange and AuthenticDataTransfer. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Properties
Start
StartTextPosition
End
EndTextPosition

29.3.3.2.1 End (obsolete)

Superseded by AuthenticRange.LastXMLData

// ----- javascript sample -----
// instead of:

© 2016-2022 Altova GmbH

```
// var objXMLData = Application.ActiveDocument.DocEditView.CurrentSelection.End;
// use now:
var objXMLData = Application.ActiveDocument.AuthenticView.Selection.LastXMLData;
```

Declaration: End as XMLData

Description
XML element where the current selection ends.

29.3.3.2.2 EndTextPosition (obsolete)

Superseded by AuthenticRange.LastXMLDataOffset

```
// ----- javascript sample -----
// instead of:
// var nOffset = Application.ActiveDocument.DocEditView.CurrentSelection.EndTextPosition;
// use now:
var nOffset = Application.ActiveDocument.AuthenticView.Selection.LastXMLDataOffset;
```

Declaration: EndTextPosition as long

Description
Position in DocEditSelection.End.TextValue where the selection ends.

29.3.3.2.3 Start (obsolete)

Superseded by AuthenticRange.FirstXMLData

```
// ----- javascript sample -----
// instead of:
// var objXMLData = Application.ActiveDocument.DocEditView.CurrentSelection.Start;
// use now:
var objXMLData = Application.ActiveDocument.AuthenticView.Selection.FirstXMLData;
```

Declaration: Start as XMLData

Description
XML element where the current selection starts.

29.3.3.2.4 StartTextPosition (obsolete)

Superseded by AuthenticRange.FirstXMLDataOffset

```
// ----- javascript sample -----
// instead of:
// var nOffset = Application.ActiveDocument.DocEditView.CurrentSelection.StartTextPosition;
// use now:
var nOffset = Application.ActiveDocument.AuthenticView.Selection.FirstXMLDataOffset;
```

Declaration: StartTextPosition as long

Description
Position in DocEditSelection.Start.TextValue where the selection starts.

29.3.3.3 OldAuthenticView (obsolete)

Superseded by AuthenticView and AuthenticRange

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage - except for AuthenticDataTransfer - is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces. All functionality available up to now in DocEditView, DocEditSelection, DocEditEvent and DocEditDataTransfer is now available via AuthenticView, AuthenticRange and AuthenticDataTransfer. Many new features have been added.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Methods

LoadXML
SaveXML

EditClear
EditCopy
EditCut
EditPaste
EditRedo
EditSelectAll
EditUndo

© 2016-2022 Altova GmbH

RowAppend
RowDelete
RowDuplicate
RowInsert
RowMoveDown
RowMoveUp

ApplyTextState
IsTextStateApplied
IsTextStateEnabled

MarkUpView

SelectionSet
SelectionMoveTabOrder

GetNextVisible
GetPreviousVisible

GetAllowedElements

Properties
CurrentSelection

event

XMLRoot

IsEditClearEnabled
IsEditCopyEnabled
IsEditCutEnabled
IsEditPasteEnabled
IsEditRedoEnabled
IsEditUndoEnabled

IsRowAppendEnabled
IsRowDeleteEnabled
IsRowDuplicateEnabled
IsRowInsertEnabled
IsRowMoveDownEnabled
IsRowMoveUpEnabled

Description
Interface for Authentic View.

29.3.3.3.1 ApplyTextState (obsolete)

Superseded by AuthenticRange.PerformAction

Use spyAuthenticApply for the eAction parameter. The PerformAction method allows to apply text state attributes to any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.ApplyTextState ("bold");
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.PerformAction (spyAuthenticApply, "bold"))
    MsgBox ("Error: can't set current selection to bold");
```

Declaration: ApplyTextState (*elementName* as String)

Description
Applies or removes the text state defined by the parameter elementName. Common examples for the parameter elementName would be strong and italic.

In an XML document there are segments of data, which may contain sub-elements. For example consider the following HTML:

```
<b>fragment</b>
```

The HTML tag <b> will cause the word fragment to be bold. However, this only happens because the HTML parser knows that the tag <b> is bold. With XML there is much more flexibility. It is possible to define any XML tag to do anything you desire. The point is that it is possible to apply a Text state using XML. But the Text state that is applied must be part of the schema. For example in the OrgChart.xml, OrgChart.sps, OrgChart.xsd example the tag is the same as bold. And to apply bold the method ApplyTextState() is called. But like the row and edit operations it is necessary to test if it is possible to apply the text state.

See also IsTextStateEnabled and IsTextStateApplied.

29.3.3.3.2 CurrentSelection (obsolete)

Superseded by AuthenticView.Selection

The returned AuthenticRange object supports navigation via XMLData elements as well as navigation by document elements (e.g. characters, words, tags) or text cursor positions.

```
// ----- javascript sample -----
// instead of:
// var objDocEditSel = Application.ActiveDocument.DocEditView.CurrentSelection;
// use now:
```


© 2016-2022 Altova GmbH

```
var objRange = Application.ActiveDocument.AuthenticView.Selection;
```

Declaration: CurrentSelection as DocEditSelection

Description
The property provides access to the current selection in the Authentic View.

29.3.3.3.3 EditClear (obsolete)

Superseded by AuthenticRange.Delete

The Delete method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditClear();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Delete())
    MsgBox ("Error: can't delete current selection");
```

Declaration: EditClear

Description
Deletes the current selection.

29.3.3.3.4 EditCopy (obsolete)

Superseded by AuthenticRange.Copy

The Copy method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditCopy();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Copy())
    MsgBox ("Error: can't copy current selection");
```

Declaration: EditCopy

Description
Copies the current selection to the clipboard.

29.3.3.3.5 EditCut (obsolete)

Superseded by AuthenticRange.Cut

The Cut method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditCut();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Cut())
    MsgBox ("Error: can't cut out current selection");
```

Declaration: EditCut

Description
Cuts the current selection from the document and copies it to the clipboard.

29.3.3.3.6 EditPaste (obsolete)

Superseded by AuthenticRange.Paste

The Paste method of AuthenticRange allows to delete any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditPaste();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.Paste())
    MsgBox ("Error: can't paste to current selection");
```

Declaration: EditPaste

Description
Pastes the content from the clipboard into the document.

© 2016-2022 Altova GmbH

29.3.3.3.7 EditRedo (obsolete)

Superseded by AuthenticView.Redo

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditRedo();
// use now:
if (! Application.ActiveDocument.AuthenticView.Redo())
    MsgBox ("Error: no redo step available");
```

Declaration: EditRedo

Description
Redo the last undo step.

29.3.3.3.8 EditSelectAll (obsolete)

Superseded by AuthenticView.WholeDocument and **AuthenticRange.Select**

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditSelectAll();
// use now:
Application.ActiveDocument.AuthenticView.WholeDocument.Select();
```

Declaration: EditSelectAll

Description
The method selects the complete document.

29.3.3.3.9 EditUndo (obsolete)

Superseded by AuthenticView.Undo

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.EditUndo();
// use now:
if (! Application.ActiveDocument.AuthenticView.Undo())
    MsgBox ("Error: no undo step available");
```

Declaration: EditUndo

Description
Undo the last action.

29.3.3.3.10 event (obsolete)

Superseded by parameters to AuthenticView events.

Declaration: event as DocEditEvent

Description
The event property holds a DocEditEvent object which contains information about the current event.

29.3.3.3.11 GetAllowedElements (obsolete)

Superseded by AuthenticRange.CanPerformActionWith

AuthenticRange now supports all functionality of the 'elements' entry helper. Besides querying the elements that can be inserted, appended, etc., you can invoke the action as well. See AuthenticRange.PerformAction for more information.

```
// ----- javascript sample -----
// instead of:
// var arrElements = New Array();
// var objDocEditView = Application.ActiveDocument.DocEditView;
// var objStartElement = objDocEditView.CurrentSelection.Start;
// var objEndElement = objDocEditView.CurrentSelection.End;
// objDocEditView.GetAllowedElements(k_ActionInsertBefore, objStartElement, objEndElement,
arrElements);
// use now:
var arrElements = New Array();
Application.ActiveDocument.AuthenticView.Selection.CanPerformActionWith (spyAuthenticInsertBefore,
arrElements);
```


© 2016-2022 Altova GmbH

Declaration: GetAllowedElements (*nAction* as SpyAuthenticElementActions, *pStartElement* as XMLData, *pEndElement* as XMLData, *pElements* as Variant)

Description
GetAllowedElements() returns the allowed elements for the various actions specified by nAction.

JavaScript example:

```
Function GetAllowed()
{
	var objView = Application.ActiveDocument.DocEditView;

	var arrElements = New Array(1);

	var objStart = objView.CurrentSelection.Start;
	var objEnd = objView.CurrentSelection.End;

	var strText;
	strText = "valid elements at current selection:\n\n";

	For(var i = 1;i <= 4;i++)  {
		objPlugIn.GetAllowedElements(i,objStart,objEnd,arrElements);
		strText = strText + ListArray(arrElements) + "----------\n";
	}

	Return strText;
}

Function ListArray(arrIn)
{
	var strText = "";

	If(TypeOf(arrIn) == "object")	{
		For(var i = 0;i <= (arrIn.length - 1);i++)
			strText = strText + arrIn[i] + "\n";
	}

	Return strText;
}
```

VBScript example:

```
Sub DisplayAllowed
	Dim objView
	Set objView = Application.ActiveDocument.DocEditView

	Dim arrElements()

	Dim objStart
	Dim objEnd
	Set objStart = objView.CurrentSelection.Start
	Set objEnd = objView.CurrentSelection.End
```

```
        Dim strText
        strText = "valid elements at current selection:" & chr(13) & chr(13)

        Dim i

        For i = 1 To 4
                objView.GetAllowedElements i,objStart,objEnd,arrElements
                strText = strText & ListArray(arrElements) & "----------" & chr(13)
        Next

        msgbox strText
End Sub

Function ListArray(arrIn)
        Dim strText

        If IsArray(arrIn) Then
                Dim i

                For i = 0 To UBound(arrIn)
                        strText = strText & arrIn(i) & chr(13)
                Next
        End If

        ListArray = strText
End Function
```

29.3.3.3.12 GetNextVisible (obsolete)

Superseded by AuthenticRange.SelectNext

AuthenticRange now supports a wide range of element navigation methods based on document elements like characters, words, tags and many more. Selecting the text passage that represents the content of the next XML element is just one of them.

```
// ----- javascript sample -----
// instead of:
// var objCurrXMLData = ...
// var objXMLData = Application.ActiveDocument.DocEditView.GetNextVisible(objCurrXMLData);
// Application.ActiveDocument.DocEditView.SelectionSet (objXMLData, 0, objXMLData, -1);
// use now:
var objRange = ...
try
   { objRange.SelectNext (spyAuthenticTag).Select(); }
catch (err)
{
```


© 2016-2022 Altova GmbH

```
    if ((err.number & 0xffff) == 2003)
        MsgBox ("end of document reached");
    else
        throw (err);
}
```

Declaration: GetNextVisible (*pElement* as XMLData) as XMLData

Description
The method gets the next visible XML element in the document.

29.3.3.3.13 GetPreviousVisible (obsolete)

Superseded by AuthenticRange.SelectPrevious

AuthenticRange now supports a wide range of element navigation methods based on document elements like characters, words, tags and many more. Selecting the text passage that represents the content of the previous XML element is just one of them.

```
// ----- javascript sample -----
// instead of:
// var objCurrXMLData = ...
// var objXMLData = Application.ActiveDocument.DocEditView.GetPreviousVisible(objCurrXMLData);
// Application.ActiveDocument.DocEditView.SelectionSet (objXMLData, 0, objXMLData, -1);
// use now:
var objRange = ...
try
    { objRange.SelectPrevious (spyAuthenticTag).Select(); }
catch (err)
{
    if ((err.number & 0xffff) == 2004)
        MsgBox ("begin of document reached");
    else
        throw (err);
}
```

Declaration: GetPreviousVisible (*pElement* as XMLData) as XMLData

Description
The method gets the previous visible XML element in the document.

29.3.3.3.14 IsEditClearEnabled (obsolete)

Superseded by AuthenticRange.IsDeleteEnabled

The IsDeleteEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditClearEnabled)
//    Application.ActiveDocument.DocEditView.EditClear();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsDeleteEnabled)
   objCurrSelection.Delete();
```

Declaration: IsEditClearEnabled as Boolean

Description
True if EditClear is possible. See also Editing operations.

29.3.3.3.15 IsEditCopyEnabled (obsolete)

Superseded by AuthenticRange.IsCopyEnabled

The IsCopyEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditCopyEnabled)
//    Application.ActiveDocument.DocEditView.EditCopy();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsCopyEnabled)
   objCurrSelection.Copy();
```

Declaration: IsEditCopyEnabled as Boolean

Description
True if copy to clipboard is possible. See also EditCopy and Editing operations.

© 2016-2022 Altova GmbH

29.3.3.3.16 IsEditCutEnabled (obsolete)

Superseded by AuthenticRange.IsCutEnabled

The IsCutEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditCutEnabled)
//    Application.ActiveDocument.DocEditView.EditCut();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsCutEnabled)
   objCurrSelection.Cut();
```

Declaration: IsEditCutEnabled as Boolean

Description
True if EditCut is currently possible. See also Editing operations.

29.3.3.3.17 IsEditPasteEnabled (obsolete)

Superseded by AuthenticRange.IsPasteEnabled

The IsPasteEnabled property is now supported for any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditPasteEnabled)
//    Application.ActiveDocument.DocEditView.EditPaste();
// use now:
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;
if (objCurrSelection.IsPasteEnabled)
   objCurrSelection.Paste();
```

Declaration: IsEditPasteEnabled as Boolean

Description
True if EditPaste is possible. See also Editing operations.

29.3.3.3.18 IsEditRedoEnabled (obsolete)

Superseded by AuthenticView.IsRedoEnabled

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditRedoEnabled)
//    Application.ActiveDocument.DocEditView.EditRedo();
// use now:
if (Application.ActiveDocument.AuthenticView.IsRedoEnabled)
   Application.ActiveDocument.AuthenticView.Redo();
```

Declaration: IsEditRedoEnabled as Boolean

Description
True if EditRedo is currently possible. See also Editing operations.

29.3.3.3.19 IsEditUndoEnabled (obsolete)

Superseded by AuthenticView.IsUndoEnabled

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsEditUndoEnabled)
//    Application.ActiveDocument.DocEditView.EditUndo();
// use now:
if (Application.ActiveDocument.AuthenticView.IsUndoEnabled)
   Application.ActiveDocument.AuthenticView.Undo();
```

Declaration: IsEditUndoEnabled as Boolean

Description
True if EditUndo is possible. See also Editing operations.

29.3.3.3.20 IsRowAppendEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable

© 2016-2022 Altova GmbH

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowAppendEnabled)
//    Application.ActiveDocument.DocEditView.RowAppend();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
   Application.ActiveDocument.AuthenticView.Selection.AppendRow();
```

Declaration: IsRowAppendEnabled as Boolean

Description
True if RowAppend is possible. See also Row operations.

29.3.3.3.21 IsRowDeleteEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowDeleteEnabled)
//    Application.ActiveDocument.DocEditView.Rowdelete();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
   Application.ActiveDocument.AuthenticView.Selection.DeleteRow();
```

Declaration: IsRowDeleteEnabled as Boolean

Description
True if RowDelete is possible. See also Row operations.

29.3.3.3.22 IsRowDuplicateEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowDuplicateEnabled)
//    Application.ActiveDocument.DocEditView.RowDuplicate();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
   Application.ActiveDocument.AuthenticView.Selection.DuplicateRow();
```

Declaration: IsRowDuplicateEnabled as Boolean

Description
True if RowDuplicate is currently possible. See also Row operations.

29.3.3.3.23 IsRowInsertEnabled (obsolete)

Superseded by AuthenticRange.IsInDynamicTable

The operations 'insert', 'append', 'delete' and 'duplicate' row are available whenever the selection is inside a dynamic table.

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowInsertEnabled)
//    Application.ActiveDocument.DocEditView.RowInsert();
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable())
   Application.ActiveDocument.AuthenticView.Selection.InsertRow();
```

Declaration: IsRowInsertEnabled as Boolean

Description
True if RowInsert is possible. See also Row operations.

29.3.3.3.24 IsRowMoveDownEnabled (obsolete)

Superseded by AuthenticRange.IsLastRow

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.OldAuthenticView.IsRowMoveDownEnabled)
```


© 2016-2022 Altova GmbH

```
//    Application.ActiveDocument.DocEditView.RowMoveDown();
// use now:
if (!Application.ActiveDocument.AuthenticView.Selection.IsLastRow)
    Application.ActiveDocument.AuthenticView.Selection.MoveRowDown();
```

Declaration: IsRowMoveDownEnabled as Boolean

Description
True if RowMoveDown is currently possible. See also Row operations.

29.3.3.3.25 IsRowMoveUpEnabled (obsolete)

Superseded by AuthenticRange.IsFirstRow

```
// ----- javascript sample -----
// instead of:
// if (Application.ActiveDocument.DocEditView.IsRowMoveUpEnabled)
//    Application.ActiveDocument.DocEditView.RowMoveUp();
// use now:
if (!Application.ActiveDocument.AuthenticView.Selection.IsFirstRow)
    Application.ActiveDocument.AuthenticView.Selection.MoveRowUp();
```

Declaration: IsRowMoveUpEnabled as Boolean

Description
True if RowMoveUp is possible. See also Row operations.

29.3.3.3.26 IsTextStateApplied (obsolete)

Superseded by AuthenticRange.IsTextStateApplied

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.IsTextStateApplied ("bold");
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.IsTextStateApplied ("bold"))
    MsgBox ("bold on");
else
    MsgBox ("bold off");
```

Declaration: IsTextStateApplied (*elementName* as String) as Boolean

Description
Checks to see if the it the text state has already been applied. Common examples for the parameter elementName would be strong and italic.

29.3.3.3.27 IsTextStateEnabled (obsolete)

Superseded by AuthenticRange.CanPerformAction

Use spyAuthenticApply for the eAction parameter. The CanPerformAction method allows to operate on any range of the document, not only the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.IsTextStateEnabled ("bold");
// use now:
if (Application.ActiveDocument.AuthenticView.Selection.CanPerformAction (spyAuthenticApply, "bold"))
    ... // e.g. enable 'bold' button
```

Declaration: IsTextStateEnabled (*i_strElementName* as String) as Boolean

Description
Checks to see if it is possible to apply a text state. Common examples for the parameter elementName would be strong and italic.

29.3.3.3.28 LoadXML (obsolete)

Superseded by AuthenticView.AsXMLString

AuthenticView now supports the property AsXMLString that can be used to directly access and replace the document content as an XMLString.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.LoadXML (strDocAsXMLString);
// use now:
try
    { Application.ActiveDocument.AuthenticView.AsXMLString = strDocAsXMLString; }
catch (err)
    { MsgBox ("Error: invalid XML string"); }
```

Declaration: LoadXML (*xmlString* as String)

© 2016-2022 Altova GmbH

Description
Loads the current XML document with the XML string applied. The new content is displayed immediately. The *xmlString* parameter must begin with the XML declaration, e.g.,
objPlugIn.LoadXML("<?xml version='1.0' encoding='UTF-8'?><root></root>");

29.3.3.3.29 MarkUpView (obsolete)

Superseded by AuthenticView.MarkupVisibility

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.MarkuUpView = 2;
// use now:
Application.ActiveDocument.AuthenticView.MarkupVisibility = spyAuthenticMarkupLarge;
```

Declaration: MarkUpView (*kind* as long)

Description
By default the document displayed is using HTML techniques. But sometimes it is desirable to show the editing tags. Using this method it is possible to display three different types of markup tags:

0	hide the markup tags
2	show the large markup tags
3	show the mixed markup tags.

29.3.3.3.30 RowAppend (obsolete)

Superseded by AuthenticRange.AppendRow

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowAppend();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.AppendRow())
    MsgBox ("Error: can't append row");
```

Declaration: RowAppend

Description
Appends a row at the current position.

See also Row operations.

29.3.3.3.31 RowDelete (obsolete)

Superseded by AuthenticRange.DeleteRow

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowDelete();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.DeleteRow())
    MsgBox ("Error: can't delete row");
```

Declaration: RowDelete

Description
Deletes the currently selected row(s).

See also Row operations.

29.3.3.3.32 RowDuplicate (obsolete)

Superseded by AuthenticRange.DuplicateRow

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowDuplicate();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.DuplicateRow())
    MsgBox ("Error: can't duplicate row");
```

Declaration: RowDuplicate

© 2016-2022 Altova GmbH

Description
The method duplicates the currently selected rows.

See also Row operations.

29.3.3.3.33 RowInsert (obsolete)

Superseded by AuthenticRange.InsertRow

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowInsert();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.InsertRow())
    MsgBox ("Error: can't insert row");
```

Declaration: RowInsert

Description
Inserts a new row immediately above the current selection.

See also Row operations.

29.3.3.3.34 RowMoveDown (obsolete)

Superseded by AuthenticRange.MoveRowDown

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowMoveDown();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.MoveRowDown())
    MsgBox ("Error: can't move row down");
```

Declaration: RowMoveDown

Description
Moves the current row one position down.

See also Row operations.

29.3.3.3.35 RowMoveUp (obsolete)

Superseded by AuthenticRange.MoveRowUp

The table operations of AuthenticRange now allow to manipulate any table in the current document independent of the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.RowAppend();
// use now:
if (! Application.ActiveDocument.AuthenticView.Selection.MoveRowUp())
    MsgBox ("Error: can't move row up");
```

Declaration: RowMoveUp

Description
Moves the current row one position up.

See also Row operations.

29.3.3.3.36 SaveXML (obsolete)

Superseded by AuthenticView.AsXMLString

AuthenticView now supports the property XMLString that can be used to directly access and replace the document content as an XMLString.

```
// ----- javascript sample -----
// instead of:
// var strDocAsXMLString = Application.ActiveDocument.DocEditView.SaveXML();
// use now:
try
{
    var strDocAsXMLString = Application.ActiveDocument.AuthenticView.AsXMLString;
    ... // do something here
}
```


© 2016-2022 Altova GmbH

```
catch (err)
   { MsgBox ("Error: invalid XML string"); }
```

Declaration: SaveXML as String

Return Value
XML structure as string

Description
Saves the current XML data to a string that is returned to the caller.

29.3.3.3.37 SelectionMoveTabOrder (obsolete)

Superseded by AuthenticRange.SelectNext

AuthenticRange now supports a wide range of element navigation methods based on document elements like characters, words, tags and many more. Selecting the next paragraph is just one of them, and navigation is not necessarily bound to the current UI selection.

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.DocEditView.SelectionMoveTabOrder(true, true);
// use now:
Application.ActiveDocument.AuthenticView.Selection.SelectNext (spyAuthenticParagraph).Select();
// to append a row to a table use AuthenticRange.AppendRow
```

Declaration: SelectionMoveTabOrder (*bForward* as Boolean, *bTag* as Boolean)

Description
SelectionMoveTabOrder() moves the current selection forwards or backwards.

If bTag is false and the current selection is at the last cell of a table a new line will be added.

29.3.3.3.38 SelectionSet (obsolete)

Superseded by AuthenticRange.FirstXMLData and related properties

AuthenticRange supports navigation via XMLData elements as well as navigation by document elements (e.g. characters, words, tags) or text cursor positions.

```
// ----- javascript sample -----
// instead of:
```

```
// if (! Application.ActiveDocument.DocEditView.SelectionSet(varXMLData1, 0, varXMLData2, -1))
//    MsgBox ("Error: invalid data position");
// use now:
try
{
   var objSelection = Application.ActiveDocument.AuthenticView.Selection;
   objSelection.FirstXMLData = varXMLData1;
   objSelection.FirstXMLdataOffset = 0;
   objSelection.LastXMLData = varXMLData2;
   objSelection.LastXMLDataOffset = -1;
   objSelection.Select();
}
catch (err)
   { MsgBox ("Error: invalid data position"); }
// to select all text between varXMLData1 and varXMLdata2, inclusive
```

Declaration: SelectionSet (*pStartElement* as XMLData, *nStartPos* as long, *pEndElement* as XMLData, *nEndPos* as long) as Boolean

Description
Use SelectionSet() to set a new selection in the Authentic View. Its possible to set pEndElement to null (nothing) if the selection should be just over one (pStartElement) XML element.

29.3.3.3.39 XMLRoot (obsolete)

Superseded by AuthenticView.XMLDataRoot

```
// ----- javascript sample -----
// instead of:
// var objXMLData = Application.ActiveDocument.DocEditView.XMLRoot;
// use now:
var objXMLData = Application.ActiveDocument.AuthenticView.XMLDataRoot;
```

Declaration: XMLRoot as XMLData

Description
XMLRoot is the parent element of the currently displayed XML structure. Using the XMLData interface you have full access to the complete content of the file.

29.3.4 Enumerations

This is a list of all enumerations used by the XMLSpy API. If your scripting environment does not support enumerations use the number-values instead.

© 2016-2022 Altova GmbH

29.3.4.1 ENUMApplicationStatus

Description
Enumeration to specify the current Application status.

Possible values:

eApplicationRunning	= 0
eApplicationAfterLicenseCheck	= 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

29.3.4.2 SPYAttributeTypeDefinition

Description
Attribute type definition that can be selected for generation of Sample XML.
This type is used with the method GenerateDTDOrSchema and GenerateDTDOrSchemaEx.

Possible values:

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

29.3.4.3 SPYAuthenticActions

Description
Actions that can be performed on AuthenticRange objects.

Possible values:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

29.3.4.4 SPYAuthenticDocumentPosition

Description
Relative and absolute positions used for navigating with AuthenticRange objects.

Possible values:

spyAuthenticDocumentBegin	= 0

spyAuthenticDocumentEnd = 1
spyAuthenticRangeBegin = 2
spyAuthenticRangeEnd = 3

29.3.4.5 SPYAuthenticElementActions

Description
Actions that can be used with the obsolete object GetAllowedElements (superseded by AuthenticRange.CanPerformActionWith).

Possible values:
k_ActionInsertAt = 0
k_ActionApply = 1
k_ActionClearSurr = 2
k_ActionAppend = 3
k_ActionInsertBefore = 4
k_ActionRemove = 5

29.3.4.6 SPYAuthenticElementKind

Description
Enumeration of the different kinds of elements used for navigation and selection within the AuthenticRange and AuthenticView objects.

Possible values:
spyAuthenticChar = 0
spyAuthenticWord = 1
spyAuthenticLine = 3
spyAuthenticParagraph = 4
spyAuthenticTag = 6
spyAuthenticDocument = 8
spyAuthenticTable = 9
spyAuthenticTableRow = 10
spyAuthenticTableColumn = 11

29.3.4.7 SPYAuthenticMarkupVisibility

Description
Enumeration values to customize the visibility of markup with MarkupVisibility.

Possible values:
spyAuthenticMarkupHidden = 0
spyAuthenticMarkupSmall = 1
spyAuthenticMarkupLarge = 2
spyAuthenticMarkupMixed = 3

© 2016-2022 Altova GmbH

29.3.4.8 SPYAuthenticToolbarButtonState

Description
Authentic toolbar button states are given by the following enumeration:

Possible values:

authenticToolbarButtonDefault	= 0
authenticToolbarButtonEnabled	= 1
authenticToolbarButtonDisabled	= 2

29.3.4.9 SPYDatabaseKind

Description
Values to select different kinds of databases for import. See DatabaseConnection.DatabaseKind for its use.

Possible values:

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7
spyDB_PostgreSQL	= 8
spyDB_iSeries	= 9

29.3.4.10 SPYDialogAction

Description
Values to simulate different interactions on dialogs. See Dialogs for all dialogs available.

Possible values:

spyDialogOK	= 0	// simulate click on OK button
spyDialogCancel	= 1	// simulate click on Cancel button
spyDialogUserInput	= 2	// show dialog and allow user interaction

29.3.4.11 SPYDOMType

Description
Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

spyDOMType_msxml4	= 0	Obsolete
spyDOMType_xerces	= 1	
spyDOMType_xerces3	= 2	
spyDOMType_msxml6	= 3	

`spyDOMType_xerces` indicates Xerces 2.x usage; `spyDOMType_xerces3` indicates Xerces 3.x usage.

29.3.4.12 SPYDTDSchemaFormat

Description
Enumeration to identify the different schema formats.

Possible values:

spyDTD	= 0
spyW3C	= 1

29.3.4.13 SPYEncodingByteOrder

Description
Enumeration values to specify encoding byte ordering for text import and export.

Possible values:

spyNONE	= 0
spyLITTLE_ENDIAN	= 1
spyBIG_ENDIAN	= 2

29.3.4.14 SPYExportNamespace

Description
Enumeration type to configure handling of namespace identifiers during export.

Possible values:

spyNoNamespace	= 0
spyReplaceColonWithUnderscore	= 1

29.3.4.15 SPYFindInFilesSearchLocation

Description
The different locations where a search can be performed. This type is used with the FindInFilesDlg dialog.

Possible values:

spyFindInFiles_Documents	= 0
spyFindInFiles_Project	= 1
spyFindInFiles_Folder	= 2

© 2016-2022 Altova GmbH

29.3.4.16 SPYFrequentElements

Description
Enumeration value to parameterize schema generation.

Possible values:

spyGlobalElements	= 0
spyGlobalComplexType	= 1

29.3.4.17 SPYImageKind

Description
Enumeration values to parameterize image type of the generated documentation. These values are used in SchemaDocumentationDialog.DiagramFormat and WSDLDocumentationDlg.DiagramFormat.

Possible values:

spyImageType_PNG	= 0
spyImageType_EMF	= 1

29.3.4.18 SPYImportColumnsType

Description
Enumeration to specify different Import columns types.

Possible values:

spyImportColumns_Element	= 0
spyImportColumns_Attribute	= 1

29.3.4.19 SPYKeyEvent

Description
Enumeration type to identify the different key events. These events correspond with the equally named windows messages.

Possible values:

spyKeyDown	= 0
spyKeyUp	= 1
spyKeyPressed	= 2

29.3.4.20 SPYKeyStatus

Description
Enumeration type to identify the key status.

Possible values:

spyLeftShiftKeyMask	= 1
spyRightShiftKeyMask	= 2
spyLeftCtrlKeyMask	= 4
spyRightCtrlKeyMask	= 8
spyLeftAltKeyMask	= 16
spyRightAltKeyMask	= 32

29.3.4.21 SPYLibType

Description
Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

spyLibType_static	= 0
spyLibType_dll	= 1

29.3.4.22 SPYLoading

Description
Enumeration values to define loading behaviour of URL files.

Possible values:

spyUseCacheProxy	= 0
spyReload	= 1

29.3.4.23 SPYMouseEvent

Description
Enumeration type that defines the mouse status during a mouse event. Use the enumeration values as bitmasks rather then directly comparing with them.

Examples

```
' to check for ctrl-leftbutton-down in VB
If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or XMLSpyLib.spyCtrlKeyDownMask)) Then
        ' react on ctrl-leftbutton-down
End If

' to check for double-click with any button in VBScript
If (((i_eMouseEvent And spyDoubleClickMask) <> 0) Then
        ' react on double-click
End If
```

Possible values:

spyNoButtonMask	= 0

© 2016-2022 Altova GmbH

```
spyMouseMoveMask          = 1
spyLeftButtonMask         = 2
spyMiddleButtonMask       = 4
spyRightButtonMask        = 8
spyButtonUpMask           = 16
spyButtonDownMask         = 32
spyDoubleClickMask        = 64
spyShiftKeyDownMask       = 128
spyCtrlKeyDownMask        = 256
spyLeftButtonDownMask     = 34   // spyLeftButtonMask | spyButtonDownMask
spyMiddleButtonDownMask   = 36   // spyMiddleButtonMask | spyButtonDownMask
spyRightButtonDownMask    = 40   // spyRightButtonMask | spyButtonDownMask
spyLeftButtonUpMask       = 18   // spyLeftButtonMask | spyButtonUpMask
spyMiddleButtonUpMask     = 20   // spyMiddleButtonMask | spyButtonUpMask
spyRightButtonUpMask      = 24   // spyRightButtonMask | spyButtonUpMask
spyLeftDoubleClickMask    = 66   // spyRightButtonMask | spyButtonUpMask
spyMiddleDoubleClickMask  = 68   // spyMiddleButtonMask | spyDoubleClickMask
spyRightDoubleClickMask   = 72   // spyRightButtonMask | spyDoubleClickMask
```

29.3.4.24 SPYNumberDateTimeFormat

Description
Enumeration value to configure database connections.

Possible values:

```
spySystemLocale      = 0
spySchemaCompatible  = 1
```

29.3.4.25 SPYProgrammingLanguage

Description
Enumeration values to select the programming language for code generation from schema definitions.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Possible values:

```
spyUndefinedLanguage  = -1
spyJava               = 0
spyCpp                = 1
spyCSharp             = 2
```

29.3.4.26 SPYProjectItemTypes

Description
Enumeration values to identify the different elements in project item lists. See SpyProjectItem.ItemType.

Possible values:

spyUnknownItem = 0
spyFileItem = 1
spyFolderItem = 2
spyURLItem = 3

29.3.4.27 SPYProjectType

Description
Enumeration values to parameterize generation of C# from schema definitions.

Possible values:

spyVisualStudioProject	= 0	Obsolete
spyVisualStudio2003Project	= 1	Obsolete
spyBorlandProject	= 2	Obsolete
spyMonoMakefile	= 3	Obsolete
spyVisualStudio2005Project	= 4	Obsolete
spyVisualStudio2008Project	= 5	Obsolete
spyVisualStudio2010Project	= 6	For C++ code also
spyVisualStudio2013Project	= 7	For C++ code also
spyVisualStudio2015Project	= 8	For C++ code also
spyVisualStudio2017Project	= 9	For C++ code also
spyVisualStudio2019Project	=10	For C++ code also

29.3.4.28 SpySampleXMLGenerationChoiceMode

Description
This enumeration is used in `GenerateSampleXMLDlg.ChoiceMode`:

spySampleXMLGen_FirstBranch = 0
spySampleXMLGen_AllBranches = 1
spySampleXMLGen_ShortestBranch = 2

29.3.4.29 SPYSampleXMLGenerationOptimization (Obsolete)

This enumeration is OBSOLETE since v2014.

Description
Specify the elements that will be generated in the Sample XML.
This enumeration is used in GenerateSampleXMLDlg.

Possible values:

spySampleXMLGen_Optimized	= 0

© 2016-2022 Altova GmbH

spySampleXMLGen_NonMandatoryElements = 1
spySampleXMLGen_Everything = 2

29.3.4.30 SpySampleXMLGenerationSampleValueHints

Description
This enumeration is used in `GenerateSampleXMLDlg.SampleValueHints`

spySampleXMLGen_FirstFit = 0
spySampleXMLGen_RandomFit = 1
spySampleXMLGen_CycleThrough = 2

29.3.4.31 SPYSampleXMLGenerationSchemaOrDTDAssignment

Description
Specifies what kind of reference to the schema/DTD should be added to the generated Sample XML.
This enumeration is used in GenerateSampleXMLDlg.

Possible values:
spySampleXMLGen_AssignRelatively = 0
spySampleXMLGen_AssignAbsolutely = 1
spySampleXMLGen_DoNotAssign = 2

29.3.4.32 SPYSchemaDefKind

Description
Enumeration type to select schema diagram types.

Possible values:
spyKindElement = 0
spyKindComplexType = 1
spyKindSimpleType = 2
spyKindGroup = 3
spyKindModel = 4
spyKindAny = 5
spyKindAttr = 6
spyKindAttrGroup = 7
spyKindAttrAny = 8

spyKindIdentityUnique = 9
spyKindIdentityKey = 10
spyKindIdentityKeyRef = 11
spyKindIdentitySelector = 12
spyKindIdentityField = 13
spyKindNotation = 14
spyKindInclude = 15
spyKindImport = 16
spyKindRedefine = 17
spyKindFacet = 18
spyKindSchema = 19
spyKindCount = 20

29.3.4.33 SPYSchemaDocumentationFormat

Description
Enumeration values to parameterize generation of schema documentation. These values are used in SchemaDocumentationDialog.OutputFormat and WSDLDocumentationDlg.OutputFormat.

Possible values:
spySchemaDoc_HTML = 0
spySchemaDoc_MSWord = 1
spySchemaDoc_RTF = 2
spySchemaDoc_PDF = 3

29.3.4.34 SPYSchemaExtensionType

Description
Enumeration to specify different Schema Extension types.

Possible values:

spySchemaExtension_None = 0
spySchemaExtension_SQL_XML = 1
spySchemaExtension_MS_SQL_Server = 2
spySchemaExtension_Oracle = 3

29.3.4.35 SPYSchemaFormat

Description
Enumeration to specify different Schema Format types.

Possible values:

spySchemaFormat_Hierarchical = 0
spySchemaFormat_Flat = 1

© 2016-2022 Altova GmbH

29.3.4.36 SPYTextDelimiters

Description
Enumeration values to specify text delimiters for text export.

Possible values:

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

29.3.4.37 SPYTextEnclosing

Description
Enumeration value to specify text enclosing characters for text import and export.

Possible values:

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

29.3.4.38 SPYTypeDetection

Description
Enumeration to select how type detection works during `GenerateDTDOrSchema` and `GenerateDTDOrSchemaEx`.

Possible values:

spyBestPossible	= 0
spyNumbersOnly	= 1
spyNoDetection	= 2

29.3.4.39 SPYURLTypes

Description
Enumeration to specify different URL types.

Possible values:

spyURLTypeAuto	= -1
spyURLTypeXML	= 0
spyURLTypeDTD	= 1

29.3.4.40 SPYValidateXSDVersion

Description
Enumeration values that select what XSD version to use. The XSD version that is selected depends on both (i) the presence/absence—and, if present, the value—of the `/xs:schema/@vc:minVersion` attribute of the XSD document, and (ii) the value of this enumeration.

`spyValidateXSDVersion_1_0` selects `XSD 1.0` if `vc:minVersion` is absent, or is present with any value.
`spyValidateXSDVersion_1_1` selects `XSD 1.1` if `vc:minVersion` is absent, or is present with any value.
`spyValidateXSDVersion_AutoDetect` selects `XSD 1.1` if `vc:minVersion=1.1`. If the `vc:minVersion` attribute is absent, or is present with a value other than `1.1`, then `XSD 1.0` is selected.

Possible values

`spyValidateXSDVersion_AutoDetect`	`= 0`
`spyValidateXSDVersion_1_1`	`= 1`
`spyValidateXSDVersion_1_0`	`= 2`

29.3.4.41 SPYValidateErrorFormat

Description
Enumeration values that select the format of the error message.

Possible values

`spyValidateErrorFormat_Text`	`= 0`
`spyValidateErrorFormat_ShortXML`	`= 1`
`spyValidateErrorFormat_LongXML`	`= 2`

29.3.4.42 SPYViewModes

Description
Enumeration values that define the different view modes for XML documents. The mode *spyViewAuthentic(4)* identifies the mode that was intermediately called DocEdit mode and is now called Authentic mode. The mode *spyViewJsonSchema* identifies a mode which is mapped to the Schema Design View on the GUI but is distinguished internally.

Possible values:

spyViewGrid	= 0
spyViewText	= 1
spyViewBrowser	= 2
spyViewSchema	= 3

© 2016-2022 Altova GmbH

```
spyViewContent      = 4     // obsolete
spyViewAuthentic    = 4
spyViewWSDL         = 5
spyViewZIP          = 6
spyViewEditionInfo  = 7
spyViewXBRL         = 8
spyViewJsonSchema   = 9
```

29.3.4.43 SPYVirtualKeyMask

Description
Enumeration type for the most frequently used key masks that identify the status of the virtual keys. Use these values as bitmasks rather then directly comparing with them. When necessary, you can create further masks by using the 'logical or' operator.

Examples

```
'  VBScript sample: check if ctrl-key is pressed
If ((i_nVirtualKeyStatus And spyCtrlKeyMask) <> 0)) Then
        ' ctrl-key is pressed
End If

'  VBScript sample: check if ONLY ctrl-key is pressed
If (i_nVirtualKeyStatus == spyCtrlKeyMask) Then
        ' exactly ctrl-key is pressed
End If

// JScript sample: check if any of the right virtual keys is pressed
if ((i_nVirtualKeyStatus & (spyRightShiftKeyMask | spyRightCtrlKeyMask | spyRightAltKeyMask)) != 0)
{
        ; ' right virtual key is pressed
}
```

Possible values:

```
spyNoVirtualKeyMask     = 0
spyLeftShiftKeyMask     = 1
spyRightShiftKeyMask    = 2
spyLeftCtrlKeyMask      = 4
spyRightCtrlKeyMask     = 8
spyLeftAltKeyMask       = 16
spyRightAltKeyMask      = 32
spyShiftKeyMask         = 3     // spyLeftShiftKeyMask | spyRightShiftKeyMask
spyCtrlKeyMask          = 12    // spyLeftCtrlKeyMask | spyRightCtrlKeyMask
spyAltKeyMask           = 48    // spyLeftAltKeyMask | spyRightAltKeyMask
```

29.3.4.44 SPYXMLDataKind

Description
The different types of XMLData elements available for XML documents.

Possible values:

spyXMLDataXMLDocStruct	= 0
spyXMLDataXMLEntityDocStruct	= 1
spyXMLDataDTDDocStruct	= 2
spyXMLDataXML	= 3
spyXMLDataElement	= 4
spyXMLDataAttr	= 5
spyXMLDataText	= 6
spyXMLDataCData	= 7
spyXMLDataComment	= 8
spyXMLDataPI	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

29.3.5 Application API for Java (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

The Application API in Java has an interface built up of Java classes, each of which corresponds to an object in the Application API. Developers can use these Java classes to interact with the COM API. These classes are listed below and described in subsequent sections. For a description of the Application API objects themselves, see the Application API documentation. Bear in mind that some API features are only available in scripting environments; these have therefore not been ported to Java.

Java classes

SpyApplication
  SpyProject
    SpyProjectItems
      SpyProjectItem
  SpyDocuments
    SpyDoc
      SpyAuthenticView
        SpyAuthenticRange
      SpyDocEditView
        SpyDocEditSelection
      SpyGridView

© 2016-2022 Altova GmbH

SpyTextView
SpyXMLData
SpyDialogs
SpyCodeGeneratorDlg
SpyDTDSchemaGeneratorDlg
SpyFileSelectionDlg
SpyFindInFilesDlg
SpyGenerateSampleXMLDlg
SpySchemaDocumentationDlg
SpyWSDL20DocumentationDlg
SpyWSDLDocumentationDlg
SpyXBRLDocumentationDlg
SpyDatabaseConnection
SpyElementList
SpyElementListItem
SpyExportSettings
SpyFindInFilesResults
SpyFindInFilesResult
SpyFindInFilesMatch
SpyTextImportExportSettings

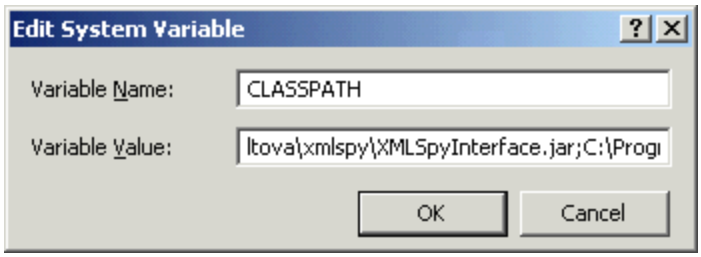
Implementation of COM properties in Java
Properties in Java have been defined to include both a **`set`** and **`get`** method (`set` if it is allowed by the COM implementation). For example, the COM class `Document` contains the **`GridView`** property. In Java the method is called `SpyDoc` and the property is defined as a **`GetGridView`** method.

If you encounter compiling problems, please check the following points:

- The `xmlspylib.dll` must be available in `..\windows\system32`.
- The `XMLSpyInterface.jar` file must be inserted in the ClassPath environment variable.

Setting the `ClassPath` variable in Windows XP

1. Click **Start | Settings | Control panel | System | Advanced | Environment Variables**. This opens the Environment Variables dialog box.
2. If a `ClassPath` entry already exists in the **System variables** group, select the `ClassPath` entry, and click the **Edit** button. Edit the path to: `"C:\Program Files\Altova\xmlspy\XMLSpyInterface.jar"`.



If a `ClassPath` entry does not exist in the System variables group, click the **New** button. The New System Variable dialog pops up. Enter `CLASSPATH` as the variable name, and `"C:\Program Files\Altova\xmlspy\XMLSpyInterface.jar"` as the `ClassPath` variable (alter the path to match your installation, if necessary).

29.3.5.1 Sample source code (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

The "`SpyDoc doc = app.GetDocuments().OpenFile(...)`" command parameter must be altered to suit your environment.

What the sample does:

- Starts a new XMLSpy instance
- Opens the Datasheet.xml file (alter the path here...)
- Switches to the Enhanced Grid view
- Appends a new child element called "NewChild" with the text value "NewValuE" element to the root element
- Checks if the document is valid and outputs a message to the Java console
- Quits and releases the XMLSpy application

```
import XMLSpyInterface.*;

public class TestSpyInterface
{
    public TestSpyInterface() {}

    public static void main(String[] args)
    {
        SpyApplication app = null;
        SpyDoc oDoc = null;
        SpyXMLData oData = null;
        SpyXMLData oNewChild = null;

        try
        {
            app = new SpyApplication();
            app.ShowApplication( true );

            oDoc = app.GetDocuments().OpenFile("C:\\FilePath\\OrgChart.xml", true );

            // OrgChart.xml is in the folder C:\Documents and Settings\<username>\My
            Documents\Altova\XMLSpy2023. The filepath should be in
            // the form: C:\\Documents and Settings\\Username\\Folder\\Filename.xml

            if ( oDoc != null )
            {
                oDoc.SwitchViewMode(SPYViewModes.spyViewGrid);
```


© 2016-2022 Altova GmbH

```
            oData = oDoc.GetRootElement();
            oNewChild = oDoc.CreateChild(SPYXMLDataKind.spyXMLDataElement);

            oNewChild.SetName( "NewChild" );
            oNewChild.SetTextValue("newVaLuE");
            oData.AppendChild(oNewChild);

            if ( oDoc.IsValid() == false )
            {
                // is to be expected after above insertion
                System.out.println( "!!!!!!validation error: " + oDoc.GetErrorString() );
                System.out.println( "!!!!!!validation error: " + oDoc.GetErrorPos() );
                System.out.println( "!!!!!!validation error: " + oDoc.GetBadData() );

            }
        }

        app.Quit();
    }
    finally
    {
        // Free any allocated resources by calling ReleaseInstance().
        if ( oNewChild != null )
            oNewChild.ReleaseInstance();

        if ( oData != null )
            oData.ReleaseInstance();

        if ( oDoc != null )
            oDoc.ReleaseInstance();

        if ( app != null )
            app.ReleaseInstance();
    }
  }
}
```

If you have difficulties compiling this sample, please try the following commands on the (**Start | Run | cmd**) command line. Please make sure you are currently in the folder that contains the sample java file.

compilation
javac -classpath c:*yourpathhere*\XMLSpyInterface.jar *testspyinterface.java*

Execution
java -classpath c:*yourpathhere*\XMLSpyInterface.jar *testspyinterface*

29.3.5.2 SpyApplication (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyApplication
{
  public void ReleaseInstance();
  public void ShowApplication( boolean bShow );
  public void Quit();
  public void AddMacroMenuItem( String sMacro, String sDisplayText );
  public void ClearMacroMenu();
  public SpyDoc GetActiveDocument();
  public SpyProject GetCurrentProject();
  public SpyDocuments GetDocuments();
  public SpyElementList GetDatabaseImportElementList( SpyDatabaseConnection oImportSettings
  );
  public SpyDatabaseConnection GetDatabaseSettings();
  public SpyElementList GetDatabaseTables( SpyDatabaseConnection oImportSettings );
  public SpyExportSettings GetExportSettings();
  public SpyElementList GetTextImportElementList( SpyTextImportExportSettings
  oImportSettings );
  public SpyTextImportExportSettings GetTextImportExportSettings();
  public SpyDoc ImportFromDatabase( SpyDatabaseConnection oImportSettings, SpyElementList
  oElementList );
  public SpyDoc ImportFromSchema( SpyDatabaseConnection oImportSettings, String strTable,
  SpyDoc oSchemaDoc );
  public SpyDoc ImportFromText( SpyTextImportExportSettings oImportSettings, SpyElementList
  oElementList );
  public SpyDoc ImportFromWord( String sFile );
  public void NewProject( String sPath, boolean bDiscardCurrent );
  public void OpenProject(String sPath , boolean bDiscardCurrent, boolean bDialog );
  public long ShowForm( String sName );
  public void URLDelete( String sURL, String sUser, String sPassword );
  public void URLMakeDirectory( String sURL, String sUser, String sPassword );
  public int GetWarningNumber();
  public String GetWarningText();

  // since Version 2004R4
  public SpyApplication GetApplication();
  public SpyApplication GetParent();
  public SpyDialogs GetDialogs();
  public boolean GetVisible();
  public void SetVisible( boolean i_bVisibility );
  public long GetWindowHandle();

  public void ReloadSettings();
  public SpyFindInFilesResults FindInFiles( SpyFindInFilesDlg dlgSettings );
  public boolean ShowFindInFiles( SpyFindInFilesDlg dlgSettings );
  public void Selection( String sVal );

  public long Status();
```


© 2016-2022 Altova GmbH

```
  public int MajorVersion();
  public int MinorVersion();
  public String Edition();
  public boolean IsAPISupported();
  public long ServicePackVersion();
  public void CreateXMLSchemaFromDBStructure( SpyDatabaseConnection oConnection,
SpyElementList oTables );
}
```

29.3.5.3 SpyCodeGeneratorDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

```
// since version 2004R4
public class SpyCodeGeneratorDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public SpyDialogs GetParent();
  public long GetProgrammingLanguage();
  public void SetProgrammingLanguage( long i_eVal );
  public String GetTemplateFileName();
  public void SetTemplateFileName( String i_strVal );
  public String GetOutputPath();
  public void SetOutputPath( String i_strVal );
  public long GetOutputPathDialogAction();
  public void SetOutputPathDialogAction( long i_eVal );
  public long GetPropertySheetDialogAction();
  public void SetPropertySheetDialogAction( long i_eVal );
  public long GetOutputResultDialogAction();
  public void SetOutputResultDialogAction( long i_eVal );
  public long GetCPPSettings_DOMType();
  public void SetCPPSettings_DOMType( long i_eVal );
  public long GetCPPSettings_LibraryType();
  public void SetCPPSettings_LibraryType( long i_eVal );
  public boolean GetCPPSettings_UseMFC();
  public void SetCPPSettings_UseMFC( boolean i_bVal );
  public long GetCSharpSettings_ProjectType();
  public void SetCSharpSettings_ProjectType( long i_eVal );
}
```

29.3.5.4 SpyDatabaseConnection (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDatabaseConnection
{
  public void ReleaseInstance();
  public String GetADOConnection();
  public void SetADOConnection( String sValue );
  public boolean GetAsAttributes();
  public void SetAsAttributes( boolean bValue );
  public boolean GetCreateMissingTables();
  public void SetCreateMissingTables( boolean bValue );
  public boolean GetCreateNew();
  public void SetCreateNew( boolean bValue );
  public boolean GetExcludeKeys();
  public void SetExcludeKeys( boolean bValue );
  public String GetFile();
  public void SetFile( String sValue );
  public boolean GetIncludeEmptyElements();
  public void SetIncludeEmptyElements( boolean bValue );
  public long GetNumberDateTimeFormat();
  public void SetNumberDateTimeFormat( long nValue );
  public String GetODBCConnection();
  public void SetODBCConnection( String sValue );
  public String GetSQLSelect();
  public void SetSQLSelect( String sValue );
  public long GetTextFieldLen();
  public void SetTextFieldLen( long nValue );

  // since version 2004R4
  public long GetDatabaseKind();
  public void SetDatabaseKind( long nValue );

  // since version 2008R2
  public boolean GetCommentIncluded();
  public void SetCommentIncluded( boolean bValue );
  public String GetNullReplacement();
  public void SetNullReplacement( String sValue );
  public String GetDatabaseSchema();
  public void SetDatabaseSchema( String sValue );
```


© 2016-2022 Altova GmbH

```
  // since version 2010r3
  public boolean GetPrimaryKeys()
  public void SetPrimaryKeys( boolean bValue )
  public boolean GetForeignKeys()
  public void SetForeignKeys( boolean bValue )
  public boolean GetUniqueKeys()
  public void SetUniqueKeys( boolean bValue )
  public long GetSchemaExtensionType()
  public void SetSchemaExtensionType( long nValue )
  public long GetSchemaFormat()
  public void SetSchemaFormat( long nValue )
  public long GetImportColumnsType()
  public void SetImportColumnsType( long nValue )
}
```

29.3.5.5 SpyDialogs (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R4
public class SpyDialogs
{
  public SpyApplication GetApplication();
  public SpyApplication GetParent();
  public SpyCodeGeneratorDlg GetCodeGeneratorDlg();
  public SpyFileSelectionDlg GetFileSelectionDlg();
  public SpySchemaDocumentationDlg GetSchemaDocumentationDlg();
  public SpyGenerateSampleXMLDlg GetGenerateSampleXMLDlg();
  public SpyDTDSchemaGeneratorDlg GetDTDSchemaGeneratorDlg();
  public SpyFindInFilesDlg GetFindInFilesDlg();
  public SpyWSDLDocumentationDlg GetWSDLDocumentationDlg();

  // Since version 2010
  public SpyWSDL20DocumentationDlg GetWSDL20DocumentationDlg();
  public SpyXBRLDocumentationDlg GetXBRLDocumentationDlg();
}
```

29.3.5.6 SpyDoc (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDoc
{
  public void ReleaseInstance();
  public void SetEncoding( String strEncoding );
  public void SetPathName( String strPath );
  public String GetPathName();
  public String GetTitle();
  public boolean IsModified();
  public void Save();
  public void Close( boolean bDiscardChanges );
  public void UpdateViews();
  public long GetCurrentViewMode();
  public boolean SwitchViewMode( long nMode );
  public SpyGridView GetGridView();
  public void SetActiveDocument();
  public void StartChanges();
  public void EndChanges();
  public void TransformXSL();
  public void AssignDTD( String sDTDFile, boolean bDialog );
  public void AssignSchema( String sSchemaFile, boolean bDialog );
  public void AssignXSL( String sXSLFile, boolean bDialog );
  public void ConvertDTDOrSchema( long nFormat, long nFrequentElements );
  public SpyXMLData CreateChild( long nKind );
  public void CreateSchemaDiagram( long nKind, String sName, String sFile );
  public SpyDocEditView GetDocEditView();
  public void ExportToDatabase( SpyXMLData oFromChild, SpyExportSettings oExportSettings,
  SpyDatabaseConnection oDatabaseConnection );
  public void ExportToText( SpyXMLData oFromChild, SpyExportSettings oExportSettings,
  SpyTextImportExportSettings oTextSettings );
  public void GenerateDTDOrSchema( long nFormat, int nValuesList, long nDetection, long
  nFrequentElements );
  public SpyElementList GetExportElementList( SpyXMLData oFromChild, SpyExportSettings
  oExportSettings );
  public SpyXMLData GetRootElement();
  public String SaveInString( SpyXMLData oData, boolean bMarked );
  public void SaveToURL( String sUrl, String sUser, String sPassword );
  public String GetErrorString(); // See IsValid() or IsWellFormed()
  public int GetErrorPos(); // See IsValid() or IsWellFormed()
  public SpyXMLData GetBadData(); // See IsValid() or IsWellFormed()
  public boolean IsValid();
```


© 2016-2022 Altova GmbH

```
  public boolean IsWellFormed( SpyXMLData oData, boolean bWithChildren );

  // Since version 2004R3
  public SpyAuthenticView GetAuthenticView()

  // Since version 2004R4
  public SpyApplication GetApplication();
  public SpyDocuments GetParent();
  public String GetFullName();
  public void SetFullName( String i_strName );
  public String GetName();
  public String GetPath();
  public boolean GetSaved();
  public void SaveAs( String i_strFileNameOrPath );
  public String GetEncoding();
  public SpyXMLData GetDataRoot();
  public void GenerateProgramCode( SpyCodeGeneratorDlg i_dlg );
  public void AssignXSLFO( String i_strFile, boolean i_bUseDialog );
  public void TransformXSLFO();
  public void GenerateSchemaDocumentation( SpySchemaDocumentationDlg i_dlg );

  public void ExecuteXQuery( String i_strXMLSourceFile );
  public void SetExternalIsValid( boolean bIsValid );
  public SpyDoc GenerateSampleXML( SpyGenerateSampleXMLDlg ipGenerateXMLDlg );
  public boolean UpdateXMLData();
  public String GetAsXMLString();
  public void SetAsXMLString( String newVal );
  public SpyDoc GenerateDTDOrSchemaEx( SpyDTDSchemaGeneratorDlg ipDTDSchemaGeneratorDlg );
  public SpyDoc ConvertDTDOrSchemaEx( long nFormat, long nFrequentElements, String
  sOutputPath, long nOutputPathDialogAction );
  public SpyTextView GetTextView();
  public String[] GetSuggestions();
  public void SetSuggestions( String[] aList );
  public void SetSelection( String sVal );

  // Since version 2009
  public void GenerateWSDLDocumentation( SpyWSDLDocumentationDlg ipWSDLDocumenationDlg );
  public void TransformXSLEx( long nDialogAction );

  // Since version 2010
  public void GenerateWSDL20Documentation( SpyWSDL20DocumentationDlg
  ipWSD20DocumenationDlg );
  public void GenerateXBRLDocumentation( SpyXBRLDocumentationDlg ipXBRLDocumentationDlg );
  public SpyDoc ConvertToWSDL20( String sFilePath, boolean bShowDialogs );

  // Since version 2010r3
  public String CreateDBStructureFromXMLSchema( SpyDatabaseConnection oConnection,
  SpyElementList oTables, boolean bDropTableWithExistingName );
  public SpyElementList GetDBStructureList( SpyDatabaseConnection oConnection );
}
```

29.3.5.7 SpyDocuments (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDocuments
{
  public void ReleaseInstance();
  public long Count();
  public SpyDoc GetItem( long nNo );
  public SpyDoc NewFile( String strFile, String strType );
  public SpyDoc NewFileFromText( String nSource, String strType );
  public SpyDoc OpenFile( String sPath, boolean bDialog );
  public SpyDoc OpenURL( String sUrl, long nURLType, long nLoading, String sUser, String
  sPassword );
  public SpyDoc OpenURLDialog(String sURL, long nURLType, long nLoading, String sUser,
  String sPassword );
  // Since version 2011r2
  public SpyDoc NewAuthenticFile( String strSPSPath, String strXMLPath );
  public SpyDoc OpenAuthenticFile( String strSPSPath, String strXMLPath );
}
```

29.3.5.8 SpyDTDSchemaGeneratorDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDTDSchemaGeneratorDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public long GetDTDSchemaFormat();
  public void SetDTDSchemaFormat( long newVal );
  public short GetValueList();
  public void SetValueList( short newVal );
  public long GetTypeDetection();
```


© 2016-2022 Altova GmbH

```
  public void SetTypeDetection( long newVal );
  public long GetFrequentElements();
  public void SetFrequentElements( long newVal );
  public boolean GetMergeAllEqualNamed();
  public void SetMergeAllEqualNamed( boolean newVal );
  public boolean GetResolveEntities();
  public void SetResolveEntities( boolean newVal );
  public long GetAttributeTypeDefinition();
  public void SetAttributeTypeDefinition( long newVal );
  public boolean GetGlobalAttributes();
  public void SetGlobalAttributes( boolean newVal );
  public boolean GetOnlyStringEnums();
  public void SetOnlyStringEnums( boolean newVal );
  public long GetMaxEnumLength();
  public void SetMaxEnumLength( long newVal );
  public String GetOutputPath();
  public void SetOutputPath( String newVal );
  public long GetOutputPathDialogAction();
  public void SetOutputPathDialogAction( long newVal );
}
```

29.3.5.9 SpyElementList (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyElementList
{
  public void ReleaseInstance();
  public long GetCount();
  public SpyElementListItem GetItem( long nIndex );
  public void RemoveElement( long nIndex );
}
```

29.3.5.10 SpyElementListItem (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyElementListItem
{
  public void ReleaseInstance();
  public long GetElementKind();
  public void SetElementKind( long nKind );
  public long GetFieldCount();
  public String GetName();
  public long GetRecordCount();
}
```

29.3.5.11 SpyExportSettings (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyExportSettings
{
  public void ReleaseInstance();
  public boolean GetCreateKeys();
  public void SetCreateKeys( boolean bValue );
  public SpyElementList GetElementList();
  public void SetElementList( SpyElementList obj );
  public boolean GetEntitiesToText ();
  public void SetEntitiesToText( boolean bValue );
  public boolean GetExportAllElements();
  public void SetExportAllElements( boolean bValue );
  public boolean GetFromAttributes();
  public void SetFromAttributes( boolean bValue );
  public boolean GetFromSingleSubElements();
  public void SetFromSingleSubElements( boolean bValue );
  public boolean GetFromTextValues();
  public void SetFromTextValues( boolean bValue );
  public boolean GetIndependentPrimaryKey();
  public void SetIndependentPrimaryKey( boolean bValue );
  public long GetNamespace();
  public void SetNamespace( long nValue );
  public int GetSubLevelLimit();
  public void SetSubLevelLimit( int nValue );
}
```


© 2016-2022 Altova GmbH

29.3.5.12 SpyFileSelectionDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R4
public class SpyFileSelectionDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public SpyDialogs GetParent();
  public String GetFullName();
  public void SetFullName( String i_strName );
  public long GetDialogAction();
  public void SetDialogAction( long i_eAction );
}
```

29.3.5.13 SpyFindInFilesDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyFindInFilesDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public String GetFind();
  public void SetFind( String sNewVal );
  public boolean GetRegularExpression();
  public void SetRegularExpression( boolean bNewVal );
  public String GetReplace();
  public void SetReplace( String sNewVal );
  public boolean GetReplaceOnDisk();
  public void SetReplaceOnDisk( boolean bNewVal );
  public boolean GetDoReplace();
  public void SetDoReplace( boolean bNewVal );
```

```
  public boolean GetMatchWholeWord();
  public void SetMatchWholeWord( boolean bNewVal );
  public boolean GetMatchCase();
  public void SetMatchCase( boolean bNewVal );
  public long GetSearchLocation();
  public void SetSearchLocation( long nPosition );
  public String GetStartFolder();
  public void SetStartFolder( String sNewVal );
  public boolean GetIncludeSubfolders();
  public void SetIncludeSubfolders( boolean bNewVal );
  public boolean GetSearchInProjectFilesDoExternal();
  public void SetSearchInProjectFilesDoExternal( boolean bNewVal );
  public String GetFileExtension();
  public void SetFileExtension( String sNewVal );
  public boolean GetAdvancedXMLSearch();
  public void SetAdvancedXMLSearch( boolean bNewVal );
  public boolean GetXMLElementNames();
  public void SetXMLElementNames( boolean bNewVal );
  public boolean GetXMLElementContents();
  public void SetXMLElementContents( boolean bNewVal );
  public boolean GetXMLAttributeNames();
  public void SetXMLAttributeNames( boolean bNewVal );
  public boolean GetXMLAttributeContents();
  public void SetXMLAttributeContents( boolean bNewVal );
  public boolean GetXMLComments();
  public void SetXMLComments( boolean bNewVal );
  public boolean GetXMLCData();
  public void SetXMLCData( boolean bNewVal );
  public boolean GetXMLPI();
  public void SetXMLPI( boolean bNewVal );
  public boolean GetXMLRest();
  public void SetXMLRest( boolean bNewVal );
  public boolean GetShowResult();
  public void SetShowResult( boolean bNewVal );
}
```

29.3.5.14 SpyFindInFilesMatch (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyFindInFilesMatch
{
  public void ReleaseInstance();
  public long Line();
```


© 2016-2022 Altova GmbH

```
  public long Position();
  public long Length();
  public String LineText();
  public boolean Replaced();
}
```

29.3.5.15 SpyFindInFilesResult (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyFindInFilesResult
{
  public void ReleaseInstance();
  public long Count();
  public SpyFindInFilesMatch GetItem( long nNo );
  public String GetPath();
  public SpyDoc GetDocument();
}
```

29.3.5.16 SpyFindInFilesResults (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyFindInFilesResults
{
  public void ReleaseInstance();
  public long Count();
  public SpyFindInFilesResult GetItem( long nNo );
}
```

29.3.5.17 SpyGenerateSampleXMLDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyGenerateSampleXMLDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public boolean GetNonMandatoryAttributes();
  public void SetNonMandatoryAttributes( boolean newVal );
  public boolean GetNonMandatoryElements();
  public void SetNonMandatoryElements( boolean newVal );
  public boolean GetTakeFirstChoice();
  public void SetTakeFirstChoice( boolean newVal );
  public long GetRepeatCount();
  public void SetRepeatCount( long newVal );
  public boolean GetFillWithSampleData();
  public void SetFillWithSampleData( boolean newVal );
  public boolean GetFillElementsWithSampleData();
  public void SetFillElementsWithSampleData( boolean newVal );
  public boolean GetFillAttributesWithSampleData();
  public void SetFillAttributesWithSampleData( boolean newVal );
  public boolean GetContentOfNillableElementsIsNonMandatory();
  public void SetContentOfNillableElementsIsNonMandatory( boolean newVal );
  public boolean GetTryToUseNonAbstractTypes();
  public void SetTryToUseNonAbstractTypes( boolean newVal );
  public long GetOptimization();
  public void SetOptimization( long newVal );
  public long GetSchemaOrDTDAssignment();
  public void SetSchemaOrDTDAssignment( long newVal );
  public String GetLocalNameOfRootElement();
  public void SetLocalNameOfRootElement( String newVal );
  public String GetNamespaceURIOfRootElement();
  public void SetNamespaceURIOfRootElement( String newVal );
  public long GetOptionsDialogAction();
  public void SetOptionsDialogAction( long newVal );
}
```


© 2016-2022 Altova GmbH

29.3.5.18 SpyGridView (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyGridView
{
  public void ReleaseInstance();
  public SpyXMLData GetCurrentFocus();
  public void Deselect( SpyXMLData oData );
  public boolean GetIsVisible();
  public void Select( SpyXMLData oData );
  public void SetFocus( SpyXMLData oData );
}
```

29.3.5.19 SpyProject (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyProject
{
  public void ReleaseInstance();
  public void CloseProject( boolean bDiscardChanges, boolean bCloseFiles, boolean
  bDialog );
  public String GetProjectFile();
  public void SetProjectFile( String sFile );
  public SpyProjectItems GetRootItems();
  public void SaveProject();
  public void SaveProjectAs( String sPath, boolean bDialog );
}
```

29.3.5.20 SpyProjectItem (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyProjectItem
{
  public void ReleaseInstance();
  public SpyProjectItems GetChildItems();
  public String GetFileExtensions();
  public void SetFileExtensions( String sExtensions );
  public long GetItemType();
  public String GetName();
  public SpyDoc Open();
  public SpyProjectItem GetParentItem();
  public String GetPath();
  public String GetValidateWith();
  public void SetValidateWith( String sVal );
  public String GetXMLForXSLTransformation();
  public void SetXMLForXSLTransformation( String sVal );
  public String GetXSLForXMLTransformation();
  public void SetXSLForXMLTransformation( String sVal );
  public String GetXSLTransformationFileExtension();
  public void SetXSLTransformationFileExtension( String sVal );
  public String GetXSLTransformationFolder();
  public void SetXSLTransformationFolder( String sVal );
}
```

29.3.5.21 SpyProjectItems (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyProjectItems
{
  public void ReleaseInstance();
```


© 2016-2022 Altova GmbH

```
  public void AddFile( String sPath );
  public void AddFolder( String sName );
  public void AddURL( String sURL, long nURLType, String sUser, String sPassword, boolean
  bSave );
  public long Count();
  public SpyProjectItem GetItem( long nNumber );
  public void RemoveItem( SpyProjectItem oItemToRemove );
}
```

29.3.5.22 SpySchemaDocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R4
public class SpySchemaDocumentationDlg
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public SpyDialogs GetParent();

  public String GetOutputFile();
  public void SetOutputFile( String i_strVal );
  public long GetOutputFormat();
  public void SetOutputFormat( long i_eVal );

  public boolean GetShowResult();
  public void SetShowResult( boolean i_bVal );
  public long GetOptionsDialogAction();
  public void SetOptionsDialogAction( long i_eVal );
  public long GetOutputFileDialogAction();
  public void SetOutputFileDialogAction( long i_eVal );
  public boolean GetShowProgressBar();
  public void SetShowProgressBar( boolean i_bVal );

  public void IncludeAll( boolean i_bInclude );
  public boolean GetIncludeIndex();
  public void SetIncludeIndex( boolean i_bVal );
  public boolean GetIncludeGlobalElements();
  public void SetIncludeGlobalElements( boolean i_bVal );
  public boolean GetIncludeLocalElements();
  public void SetIncludeLocalElements( boolean i_bVal );
  public boolean GetIncludeGroups();
```

```
public void SetIncludeGroups( boolean i_bVal );
public boolean GetIncludeComplexTypes();
public void SetIncludeComplexTypes( boolean i_bVal );
public boolean GetIncludeSimpleTypes();
public void SetIncludeSimpleTypes( boolean i_bVal );
public boolean GetIncludeAttributeGroups();
public void SetIncludeAttributeGroups( boolean i_bVal );
public boolean GetIncludeRedefines();
public void SetIncludeRedefines( boolean i_bVal );

public void AllDetails( boolean i_bDetailsOn );
public boolean GetShowDiagram();
public void SetShowDiagram( boolean i_bVal );
public boolean GetShowNamespace();
public void SetShowNamespace( boolean i_bVal );
public boolean GetShowType();
public void SetShowType( boolean i_bVal );
public boolean GetShowChildren();
public void SetShowChildren( boolean i_bVal );
public boolean GetShowUsedBy();
public void SetShowUsedBy( boolean i_bVal );
public boolean GetShowProperties();
public void SetShowProperties( boolean i_bVal );
public boolean GetShowSingleFacets();
public void SetShowSingleFacets( boolean i_bVal );
public boolean GetShowPatterns();
public void SetShowPatterns( boolean i_bVal );
public boolean GetShowEnumerations();
public void SetShowEnumerations( boolean i_bVal );
public boolean GetShowAttributes();
public void SetShowAttributes( boolean i_bVal );
public boolean GetShowIdentityConstraints();
public void SetShowIdentityConstraints( boolean i_bVal );
public boolean GetShowAnnotations();
public void SetShowAnnotations( boolean i_bVal );
public boolean GetShowSourceCode();
public void SetShowSourceCode( boolean i_bVal );

// Since version 2009
public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams( boolean i_bVal );
public long GetDiagramFormat();
public void SetDiagramFormat( long i_nVal );
public boolean GetIncludeGlobalAttributes();
public void SetIncludeGlobalAttributes( boolean i_bVal );
public boolean GetIncludeLocalAttributes();
public void SetIncludeLocalAttributes( boolean i_bVal );
public boolean GetIncludeReferencedSchemas();
public void SetIncludeReferencedSchemas( boolean i_bVal );
public boolean GetMultipleOutputFiles();
```


© 2016-2022 Altova GmbH

```
  public void SetMultipleOutputFiles( boolean i_bVal );

  // Since version 2010
  public boolean GetEmbedCSSInHTML();
  public void SetEmbedCSSInHTML( boolean i_bVal );
  public boolean GetCreateDiagramsFolder();
  public void SetCreateDiagramsFolder( boolean i_bVal );

  // Since version 2010r3
  public boolean GetGenerateRelativeLinks();
  public void SetGenerateRelativeLinks( boolean i_bVal );

  // Since version 2011r2
  public boolean GetUseFixedDesign();
  public void SetUseFixedDesign( boolean i_bVal );
  public String GetSPSFile();
  public void SetSPSFile( String i_strVal );
  }
```

29.3.5.23 SpyTextImportExportSettings (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyTextImportExportSettings
{
  public void ReleaseInstance();
  public String GetDestinationFolder();
  public void SetDestinationFolder( String sVal );
  public long GetEnclosingCharacter();
  public void SetEnclosingCharacter( long nEnclosing );
  public String GetEncoding();
  public void SetEncoding( String sVal );
  public long GetEncodingByteOrder();
  public void SetEncodingByteOrder( long nByteOrder );
  public long GetFieldDelimiter();
  public void SetFieldDelimiter( long nDelimiter );
  public String GetFileExtension ();
  public void SetFileExtension( String sVal );
  public boolean GetHeaderRow();
  public void SetHeaderRow( boolean bVal );
  public String GetImportFile();
```

```
  public void SetImportFile( String sVal );
}
```

29.3.5.24 SpyTextView (obsolete)

```
public class SpyTextView
{
  public void ReleaseInstance();
  public SpyApplication GetApplication();
  public SpyDoc GetParent();
  public long LineFromPosition( long nCharPos );
  public long PositionFromLine( long nLine );
  public long LineLength( long nLine );
  public String GetSelText();
  public void SetSelText( String sText );
  public String GetRangeText( long nPosFrom, long nPosTill );
  public void ReplaceText( long nPosFrom, long nPosTill, String sText );
  public void MoveCaret( long nDiff );
  public void GoToLineChar( long nLine, long nChar );
  public void SelectText( long nPosFrom, long nPosTill );
  public long GetSelectionStart();
  public void SetSelectionStart( long nNewVal );
  public long GetSelectionEnd();
  public void SetSelectionEnd( long nNewVal );
  public String GetText();
  public void SetText( String sText );
  public long LineCount();
  public long Length();
}
```

29.3.5.25 SpyWSDL20DocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2010
public class SpyWSDL20DocumentationDlg
{
	public void ReleaseInstance();
	public SpyApplication GetApplication();

	public long GetOptionsDialogAction();
```


© 2016-2022 Altova GmbH

```
public void SetOptionsDialogAction( long nNewVal );

public long GetOutputFileDialogAction();
public void SetOutputFileDialogAction( long nNewVal );

public boolean GetShowProgressBar();
public void SetShowProgressBar( boolean bNewVal );

public String GetOutputFile();
public void SetOutputFile( String sNewVal );

public long GetOutputFormat();
public void SetOutputFormat( long nNewVal );

public boolean GetMultipleOutputFiles();
public void SetMultipleOutputFiles( boolean bNewVal );

public boolean GetEmbedCSSInHTML();
public void SetEmbedCSSInHTML( boolean bNewVal );

public long GetDiagramFormat();
public void SetDiagramFormat( long nNewVal );

public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams( boolean bNewVal );

public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder( boolean bNewVal );

public boolean GetShowResult();
public void SetShowResult( boolean bNewVal );

public void IncludeAll( boolean bNewVal );
public void AllDetails( boolean bNewVal );

public boolean GetIncludeOverview();
public void SetIncludeOverview( boolean bNewVal );

public boolean GetIncludeService();
public void SetIncludeService( boolean bNewVal );

public boolean GetIncludeBinding();
public void SetIncludeBinding( boolean bNewVal );

public boolean GetIncludeInterface();
public void SetIncludeInterface( boolean bNewVal );

public boolean GetIncludeTypes();
public void SetIncludeTypes( boolean bNewVal );
```

```
        public boolean GetIncludeImportedWSDLFiles();
        public void SetIncludeImportedWSDLFiles( boolean bNewVal );

        public boolean GetShowServiceDiagram();
        public void SetShowServiceDiagram( boolean bNewVal );

        public boolean GetShowBindingDiagram();
        public void SetShowBindingDiagram( boolean bNewVal );

        public boolean GetShowInterfaceDiagram();
        public void SetShowInterfaceDiagram( boolean bNewVal );

        public boolean GetShowTypesDiagram();
        public void SetShowTypesDiagram( boolean bNewVal );

        public boolean GetShowEndpoint();
        public void SetShowEndpoint( boolean bNewVal );

        public boolean GetShowSourceCode();
        public void SetShowSourceCode( boolean bNewVal );

        public boolean GetShowExtensibility();
        public void SetShowExtensibility( boolean bNewVal );

        public boolean GetShowUsedBy();
        public void SetShowUsedBy( boolean bNewVal );

        public boolean GetShowOperation();
        public void SetShowOperation( boolean bNewVal );

        public boolean GetShowFault();
        public void SetShowFault( boolean bNewVal );

        // Since version 2011r2
        public boolean GetUseFixedDesign();
        public void SetUseFixedDesign( boolean i_bVal );

        public String GetSPSFile();
        public void SetSPSFile( String i_strVal );

}
```

29.3.5.26 SpyWSDLDocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

© 2016-2022 Altova GmbH

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2008r2sp1
public class SpyWSDLDocumentationDlg
{
	public void ReleaseInstance();
	public SpyApplication GetApplication();

	public String GetOutputFile();
	public void SetOutputFile( String sNewVal );

	public long GetOutputFileDialogAction();
	public void SetOutputFileDialogAction( long nNewVal );

	public long GetOptionsDialogAction();
	public void SetOptionsDialogAction( long nNewVal );

	public boolean GetShowProgressBar();
	public void SetShowProgressBar( boolean bNewVal );

	public boolean GetShowResult();
	public void SetShowResult( boolean bNewVal );

	public long GetOutputFormat();
	public void SetOutputFormat( long nNewVal );

	public boolean GetEmbedDiagrams();
	public void SetEmbedDiagrams( boolean bNewVal );

	public long GetDiagramFormat();
	public void SetDiagramFormat( long nNewVal );

	public boolean GetMultipleOutputFiles();
	public void SetMultipleOutputFiles( boolean bNewVal );

	public void IncludeAll( boolean bNewVal );

	public boolean GetIncludeBinding();
	public void SetIncludeBinding( boolean bNewVal );

	public boolean GetIncludeImportedWSDLFiles();
	public void SetIncludeImportedWSDLFiles( boolean bNewVal );

	public boolean GetIncludeMessages();
	public void SetIncludeMessages( boolean bNewVal );
```

```
public boolean GetIncludeOverview();
public void SetIncludeOverview( boolean bNewVal );

public boolean GetIncludePortType();
public void SetIncludePortType( boolean bNewVal );

public boolean GetIncludeService();
public void SetIncludeService( boolean bNewVal );

public boolean GetIncludeTypes();
public void SetIncludeTypes( boolean bNewVal );

public void AllDetails( boolean bNewVal );

public boolean GetShowBindingDiagram();
public void SetShowBindingDiagram( boolean bNewVal );

public boolean GetShowExtensibility();
public void SetShowExtensibility( boolean bNewVal );

public boolean GetShowMessageParts();
public void SetShowMessageParts( boolean bNewVal );

public boolean GetShowPort();
public void SetShowPort( boolean bNewVal );

public boolean GetShowPortTypeDiagram();
public void SetShowPortTypeDiagram( boolean bNewVal );

public boolean GetShowPortTypeOperations();
public void SetShowPortTypeOperations( boolean bNewVal );

public boolean GetShowServiceDiagram();
public void SetShowServiceDiagram( boolean bNewVal );

public boolean GetShowSourceCode();
public void SetShowSourceCode( boolean bNewVal );

public boolean GetShowTypesDiagram();
public void SetShowTypesDiagram( boolean bNewVal );

public boolean GetShowUsedBy();
public void SetShowUsedBy( boolean bNewVal );

// Since version 2010
public boolean GetEmbedCSSInHTML();
public void SetEmbedCSSInHTML( boolean i_bVal );

public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder( boolean i_bVal );
```


© 2016-2022 Altova GmbH

```
        // Since version 2011r2
        public boolean GetUseFixedDesign();
        public void SetUseFixedDesign( boolean i_bVal );

        public String GetSPSFile();
        public void SetSPSFile( String i_strVal );

    }
```

29.3.5.27 SpyXBRLDocumentationDlg (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2010
public class SpyXBRLDocumentationDlg
{
        public void ReleaseInstance();
        public SpyApplication GetApplication();

        public long GetOptionsDialogAction();
        public void SetOptionsDialogAction( long nNewVal );

        public long GetOutputDialogAction();
        public void SetOutputDialogAction( long nNewVal );

        public boolean GetShowProgressBar();
        public void SetShowProgressBar( boolean bNewVal );

        public String GetOutputFile();
        public void SetOutputFile( String sNewVal );

        public long GetOutputFormat();
        public void SetOutputFormat( long nNewVal );

        public boolean GetEmbedCSSInHTML();
        public void SetEmbedCSSInHTML( boolean bNewVal );

        public long GetDiagramFormat();
        public void SetDiagramFormat( long nNewVal );
```

```
public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams( boolean bNewVal );

public boolean GetCreateDiagramsFolder();
public void SetCreateDiagramsFolder( boolean bNewVal );

public boolean GetShowResult();
public void SetShowResult( boolean bNewVal );

public void IncludeAll( boolean bNewVal );
public void AllDetails( boolean bNewVal );

public boolean GetIncludeOverview();
public void SetIncludeOverview( boolean bNewVal );

public boolean GetIncludeNamespacePrefixes();
public void SetIncludeNamespacePrefixes( boolean bNewVal );

public boolean GetIncludeGlobalElements();
public void SetIncludeGlobalElements( boolean bNewVal );

public boolean GetIncludeDefinitionLinkroles();
public void SetIncludeDefinitionLinkroles( boolean bNewVal );

public boolean GetIncludePresentationLinkroles();
public void SetIncludePresentationLinkroles( boolean bNewVal );

public boolean GetIncludeCalculationLinkroles();
public void SetIncludeCalculationLinkroles( boolean bNewVal );

public boolean GetShowDiagram();
public void SetShowDiagram( boolean bNewVal );

public boolean GetShowSubstitutiongroup();
public void SetShowSubstitutiongroup( boolean bNewVal );

public boolean GetShowItemtype();
public void SetShowItemtype( boolean bNewVal );

public boolean GetShowBalance();
public void SetShowBalance( boolean bNewVal );

public boolean GetShowPeriod();
public void SetShowPeriod( boolean bNewVal );

public boolean GetShowAbstract();
public void SetShowAbstract( boolean bNewVal );

public boolean GetShowNillable();
public void SetShowNillable( boolean bNewVal );
```


© 2016-2022 Altova GmbH

```
        public boolean GetShowLabels();
        public void SetShowLabels( boolean bNewVal );

        public boolean GetShowReferences();
        public void SetShowReferences( boolean bNewVal );

        public boolean GetShowLinkbaseReferences();
        public void SetShowLinkbaseReferences( boolean bNewVal );

        public boolean GetShortQualifiedName();
        public void SetShortQualifiedName( boolean bNewVal );

        public boolean GetShowImportedElements();
        public void SetShowImportedElements( boolean bNewVal );

        // Since version 2011r2
        public boolean GetUseFixedDesign();
        public void SetUseFixedDesign( boolean i_bVal );

        public String GetSPSFile();
        public void SetSPSFile( String i_strVal );
};
```

29.3.5.28 SpyXMLData (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyXMLData
{
  public void ReleaseInstance();
  public void AppendChild( SpyXMLData oNewData );
  public void EraseAllChildren();
  public void EraseCurrentChild();
  public SpyXMLData GetCurrentChild();
  public SpyXMLData GetFirstChild( long nKind );
  public SpyXMLData GetNextChild();
  public boolean GetHasChildren();
  public void InsertChild( SpyXMLData oNewData );
  public boolean IsSameNode( SpyXMLData oToComp);
  public long GetKind();
  public boolean GetMayHaveChildren();
  public String GetName();
```

```
  public void SetName( String sValue );
  public SpyXMLData GetParent();
  public String GetTextValue();
  public void SetTextValue( String sValue );
}
```

29.3.5.29 Authentic (obsolete)

29.3.5.29.1 SpyAuthenticRange (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R3
public class SpyAuthenticRange
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public SpyAuthenticView GetParent();
    public SpyAuthenticRange GotoNext( long eKind );
    public SpyAuthenticRange GotoPrevious( long eKind );
    public void Select();
    public long GetFirstTextPosition();
    public void SetFirstTextPosition( long nTextPosition );
    public long GetLastTextPosition();
    public void SetLastTextPosition( long nTextPosition );
    public String GetText();
    public void SetText( String strText );
    public boolean PerformAction( long eAction, String strElementName );
    public boolean CanPerformAction( long eAction, String strElementName );
    public String[] CanPerformActionWith( long eAction );
    public SpyAuthenticRange GoTo( long eKind, long nCount, long nFrom );
    public SpyAuthenticRange SelectNext( long eKind );
    public SpyAuthenticRange SelectPrevious( long eKind );
    public SpyAuthenticRange MoveBegin( long eKind, long nCount );
    public SpyAuthenticRange MoveEnd( long eKind, long nCount );
    public SpyAuthenticRange ExpandTo( long eKind );
    public SpyAuthenticRange CollapsToBegin();
    public SpyAuthenticRange CollapsToEnd();
    public SpyAuthenticRange GotoNextCursorPosition();
    public SpyAuthenticRange GotoPreviousCursorPosition();
    public boolean IsEmpty();
    public boolean IsEqual( SpyAuthenticRange ipCmp );
```


© 2016-2022 Altova GmbH

```
    public SpyAuthenticRange Clone();
    public SpyAuthenticRange SetFromRange( SpyAuthenticRange ipSrc );
    public boolean Delete();
    public boolean Cut();
    public boolean Copy();
    public boolean Paste();
    public SpyXMLData GetFirstXMLData();
    public void SetFirstXMLData( SpyXMLData objXMLDataPtr );
    public long GetFirstXMLDataOffset();
    public void SetFirstXMLDataOffset( long nOffset );
    public SpyXMLData GetLastXMLData();
    public void SetLastXMLData( SpyXMLData objXMLDataPtr );
    public long GetLastXMLDataOffset();
    public void SetLastXMLDataOffset( long nOffset );
    public String[] GetElementHierarchy();
    public String[] GetElementAttributeNames( String strElementName );
    public boolean HasElementAttribute( String strElementName, String strAttributeName );
    public String GetElementAttributeValue( String strElementName, String
strAttributeName );
    public void SetElementAttributeValue( String strElementName, String strAttributeName,
String strNewValue );
    public String[] GetEntityNames();
    public void InsertEntity( String strEntityName );
    public boolean IsInDynamicTable();
    public boolean AppendRow();
    public boolean InsertRow();
    public boolean DuplicateRow();
    public boolean DeleteRow();
    public boolean MoveRowUp();
    public boolean MoveRowDown();

    // Since version 2004R4
    public boolean IsCopyEnabled();
    public boolean IsCutEnabled();
    public boolean IsPasteEnabled();
    public boolean IsDeleteEnabled();
    public boolean IsTextStateApplied( String i_strElementName );
    public boolean IsFirstRow();
    public boolean IsLastRow();
}
```

29.3.5.29.2 SpyAuthenticView (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
// Since version 2004R3
public class SpyAuthenticView
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public SpyDoc GetParent();
    public SpyAuthenticRange GetSelection();
    public void SetSelection( SpyAuthenticRange obj );
    public SpyAuthenticRange GetDocumentBegin();
    public SpyAuthenticRange GetDocumentEnd();
    public SpyAuthenticRange GetWholeDocument();
    public long GetMarkupVisibility();
    public void SetMarkupVisibility( long eSpyAuthenticMarkupVisibility );
    public SpyAuthenticRange GoTo( long eKind, long nCount, long nFrom );
    public void Print( boolean bWithPreview, boolean bPromptUser );
    public boolean Undo();
    public boolean Redo();
    public void UpdateXMLInstanceEntities();

    // Since version 2004R4
    public String GetAsXMLString();
    public void SetAsXMLString( String i_strXML );
    public SpyXMLData GetXMLDataRoot();
    public boolean IsUndoEnabled();
    public boolean IsRedoEnabled();
}
```

29.3.5.29.3 SpyDocEditSelection (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDocEditSelection
{
  public void ReleaseInstance();
  public SpyXMLData GetEnd();
  public long GetEndTextPosition();
  public SpyXMLData GetStart();
  public long GetStartTextPosition();
}
```


© 2016-2022 Altova GmbH

29.3.5.29.4 SpyDocEditView (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SpyDocEditView
{
  public void ReleaseInstance();
  public void ApplyTextState( String sElementName );
  public SpyDocEditSelection GetCurrentSelection();
  public void EditClear();
  public void EditCopy();
  public void EditCut();
  public void EditPaste();
  public void EditRedo();
  public void EditSelectAll();
  public void EditUndo();
  public SpyXMLData GetNextVisible( SpyXMLData oElement );
  public SpyXMLData GetPreviousVisible( SpyXMLData oElement );
  public boolean GetIsEditClearEnabled();
  public boolean GetIsEditCopyEnabled();
  public boolean GetIsEditCutEnabled();
  public boolean GetIsEditPasteEnabled();
  public boolean GetIsEditRedoEnabled();
  public boolean GetIsEditUndoEnabled();
  public boolean GetIsRowAppendEnabled();
  public boolean GetIsRowDeleteEnabled();
  public boolean GetIsRowDuplicateEnabled();
  public boolean GetIsRowInsertEnabled();
  public boolean GetIsRowMoveDownEnabled();
  public boolean GetIsRowMoveUpEnabled();
  public boolean IsTextStateApplied( String sElementName );
  public boolean IsTextStateEnabled( String sElementName );
  public void LoadXML( String sXML );
  public void MarkUpView( long nKind );
  public void RowAppend();
  public void RowDelete();
  public void RowDuplicate();
  public void RowInsert();
  public void RowMoveDown();
  public void RowMoveUp();
  public String SaveXML();
  public void SelectionMoveTabOrder( boolean bForward, boolean bTag );
```

```
  public boolean SelectionSet( SpyXMLData oStart, long nStartPos, SpyXMLData oEndElement,
  long nEndPos );
  public SpyXMLData GetXMLRoot();
  public String[] GetAllowedElements( long nAction, SpyXMLData oStartPtr, SpyXMLData
  oEndPtr );
}
```

29.3.5.30 Predefined constants (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

This section lists all classes that define the predefined constants used by the Java interface.

29.3.5.30.1 SPYApplicationStatus (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYApplicationStatus
  {
      public final static long spyApplicationStatus_Running                  = 0;
      public final static long spyApplicationStatus_AfterLicenseCheck        = 1;
      public final static long spyApplicationStatus_BeforeLicenseCheck       = 2;
      public final static long                                               = 3;
      spyApplicationStatus_ConcurrentLicenseCheckFailed
      public final static long spyApplicationStatus_ProcessingCommandLine    = 4;
  }
```

29.3.5.30.2 SPYAttributeTypeDefinition (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

 © 2016-2022 Altova GmbH

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYAttributeTypeDefinition
{
   public final static long              = 0;
   spyMergedGlobal
   public final static long              = 1;
   spyDistinctGlobal
   public final static long spyLocal     = 2;
}
```

29.3.5.30.3 SPYAuthenticActions (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYAuthenticActions
{
    public final static long spyAuthenticInsertAt   = 0;
    public final static long spyAuthenticApply      = 1;
    public final static long spyAuthenticClearSurr  = 2;
    public final static long spyAuthenticAppend     = 3;
    public final static long                        = 4;
    spyAuthenticInsertBefore
    public final static long spyAuthenticRemove     = 5;
}
```

29.3.5.30.4 SPYAuthenticDocumentPosition (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYAuthenticDocumentPosition
{
    public final static long                          = 0;
    spyAuthenticDocumentBegin
    public final static long                          = 1;
    spyAuthenticDocumentEnd
    public final static long spyAuthenticRangeBegin = 2;
    public final static long spyAuthenticRangeEnd   = 3;
}
```

29.3.5.30.5 SPYAuthenticElementKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYAuthenticElementKind
{
   public final static long              = 0;
   spyAuthenticChar
   public final static long              = 1;
   spyAuthenticWord
   public final static long              = 3;
   spyAuthenticLine
   public final static long              = 4;
   spyAuthenticParagraph
   public final static long              = 6;
   spyAuthenticTag
   public final static long              = 8;
   spyAuthenticDocument
   public final static long              = 9;
   spyAuthenticTable
   public final static long              = 10;
   spyAuthenticTableRow
   public final static long              = 11;
   spyAuthenticTableColumn
}
```


© 2016-2022 Altova GmbH

29.3.5.30.6 SPYAuthenticMarkupVisibility (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYAuthenticMarkupVisibility
{
    public final static long                          = 0;
    spyAuthenticMarkupHidden
    public final static long spyAuthenticMarkupSmall= 1;
    public final static long spyAuthenticMarkupLarge= 2;
    public final static long spyAuthenticMarkupMixed= 3;
}
```

29.3.5.30.7 SPYDatabaseKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYLoading
  {
    public final static long spyDB_Access   = 0;
    public final static long                = 1;
    spyDB_SQLServer
    public final static long spyDB_Oracle   = 2;
    public final static long spyDB_Sybase   = 3;
    public final static long spyDB_MySQL    = 4;
    public final static long spyDB_DB2      = 5;
    public final static long spyDB_Other    = 6;
    public final static long                = 7;
    spyDB_Unspecified
  }
```

29.3.5.30.8 SPYDialogAction (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYDialogAction
  {
     public final static long spyDialogOK    = 0;
     public final static long                 = 1;
     spyDialogCancel
     public final static long                 = 2;
     spyDialogUserInput
  }
```

29.3.5.30.9 SPYDOMType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYDOMType
  {
     public final static long               = 0;
     spyDOMType_msxml4
     public final static long               = 1;
     spyDOMType_xerces
  }
```

29.3.5.30.10 SPYDTDSchemaFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

© 2016-2022 Altova GmbH

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYDTDSchemaFormat
{
    public final static long spyDTD     = 0;
    public final static long spyDCD     = 1;
    public final static long spyXMLData = 2;
    public final static long spyBizTalk = 3;
    public final static long spyW3C     = 4;
}
```

29.3.5.30.11 SPYEncodingByteOrder (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYEncodingByteOrder
{
    public final static long spyNONE          = 0;
    public final static long spyLITTLE_ENDIAN = 1;
    public final static long spyBIG_ENDIAN    = 2;
}
```

29.3.5.30.12 SPYExportNamespace (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYExportNamespace
{
    public final static long spyNoNamespace                = 0;
    public final static long spyReplaceColonWithUnderscore = 1;
```

```
}
```

29.3.5.30.13 SPYFindInFilesSearchLocation (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYFindInFilesSearchLocation
{
   public final static long                    = 0;
   spyFindInFiles_Documents
   public final static long                    = 1;
   spyFindInFiles_Project
   public final static long                    = 2;
   spyFindInFiles_Folder
}
```

29.3.5.30.14 SPYFrequentElements (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYFrequentElements
{
   public final static long              = 0;
   spyGlobalElements
   public final static long              = 1;
   spyGlobalComplexType
}
```


© 2016-2022 Altova GmbH

29.3.5.30.15 SPYImageKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYImageKind
  {
     public final static long                = 0;
     spyImageType_PNG
     public final static long                = 1;
     spyImageType_EMF
  }
```

29.3.5.30.16 SPYImportColumnsType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

29.3.5.30.17 SPYLibType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYLibType
  {
     public final static long                = 0;
     spyLibType_static
     public final static long spyLibType_dll= 1;
```

```
}
```

29.3.5.30.18 SPYLoading (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYLoading
{
    public final static long               = 0;
    spyUseCacheProxy
    public final static long spyReload     = 1;
}
```

29.3.5.30.19 SPYNumberDateTimeFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class  SPYNumberDateTimeFormat
{
    public final static long spySystemLocale = 0;
    public final static long                 = 1;
    spySchemaCompatible
}
```

29.3.5.30.20 SPYProgrammingLanguage (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

© 2016-2022 Altova GmbH

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYLoading
  {
      public final static long spyUndefinedLanguage = -1;
      public final static long spyJava               = 0;
      public final static long spyCpp                = 1;
      public final static long spyCSharp             = 2;
  }
```

29.3.5.30.21 SPYProjectItemTypes (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYProjectItemTypes
{
    public final static long            = 0;
    spyUnknownItem
    public final static long spyFileItem= 1;
    public final static long            = 2;
    spyFolderItem
    public final static long spyURLItem = 3;
}
```

29.3.5.30.22 SPYProjectType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYProjectType
  {
```

```
    public final static long                    = 0;
    spyVisualStudioProject
    public final static long                    = 1;
    spyVisualStudio2003Project
    public final static long spyBorlandProject = 2;
    public final static long spyMonoMakefile   = 3;
 }
```

29.3.5.30.23 SPYSampleXMLGenerationOptimization (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYSampleXMLGenerationOptimization
{
   public final static long spySampleXMLGen_Optimized  = 0;
   public final static long                            = 1;
   spySampleXMLGen_NonMandatoryElements
   public final static long spySampleXMLGen_Everything = 2;
}
```

29.3.5.30.24 SPYSampleXMLGenerationSchemaOrDTDAssignment (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYSampleXMLGenerationOptimization
{
   public final static long                            = 0;
   spySampleXMLGen_AssignRelatively
   public final static long                            = 1;
   spySampleXMLGen_AssignAbsolutely
   public final static long spySampleXMLGen_DoNotAssign= 2;
}
```


© 2016-2022 Altova GmbH

29.3.5.30.25 SPYSchemaDefKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYSchemaDefKind
{
    public final static long spyKindElement       = 0;
    public final static long spyKindComplexType   = 1;
    public final static long spyKindSimpleType    = 2;
    public final static long spyKindGroup         = 3;
    public final static long spyKindModel         = 4;
    public final static long spyKindAny           = 5;
    public final static long spyKindAttr          = 6;
    public final static long spyKindAttrGroup     = 7;
    public final static long spyKindAttrAny       = 8;
    public final static long spyKindIdentityUnique= 9;
    public final static long spyKindIdentityKey   = 10;
    public final static long spyKindIdentityKeyRef= 11;
    public final static long                      = 12;
    spyKindIdentitySelector
    public final static long spyKindIdentityField = 13;
    public final static long spyKindNotation      = 14;
    public final static long spyKindInclude       = 15;
    public final static long spyKindImport        = 16;
    public final static long spyKindRedefine      = 17;
    public final static long spyKindFacet         = 18;
    public final static long spyKindSchema        = 19;
    public final static long spyKindCount         = 20;
}
```

29.3.5.30.26 SPYSchemaDocumentationFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYSchemaDocumentationFormat
```

```
  {
     public final static long                 = 0;
     spySchemaDoc_HTML
     public final static long                 = 1;
     spySchemaDoc_MSWord
     public final static long                 = 2;
            spySchemaDoc_RTF
     public final static long                 = 3;
            spySchemaDoc_PDF
  }
```

29.3.5.30.27 SPYSchemaExtensionType (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYSchemaExtensionType
  {
     public final static long spySchemaExtension_None    = 0;
     public final static long spySchemaExtension_SQL_XML = 1;
     public final static long                            = 2;
     spySchemaExtension_MS_SQL_Server
     public final static long spySchemaExtension_Oracle  = 3;
  }
```

29.3.5.30.28 SPYSchemaFormat (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYSchemaFormat
  {
     public final static long                       = 0;
     spySchemaFormat_Hierarchical
     public final static long spySchemaFormat_Flat  = 1;
  }
```


© 2016-2022 Altova GmbH

29.3.5.30.29 SPYTextDelimiters (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYTextDelimiters
{
    public final static long spyTabulator = 0;
    public final static long spySemicolon = 1;
    public final static long spyComma     = 2;
    public final static long spySpace     = 3;
}
```

29.3.5.30.30 SPYTextEnclosing (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYTextEnclosing
{
   public final static long           = 0;
   spyNoEnclosing
   public final static long           = 1;
   spySingleQuote
   public final static long           = 2;
   spyDoubleQuote
}
```

29.3.5.30.31 SPYTypeDetection (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYTypeDetection
{
    public final static long                = 0;
    spyBestPossible
    public final static long                = 1;
    spyNumbersOnly
    public final static long                = 2;
    spyNoDetection
}
```

29.3.5.30.32 SPYURLTypes (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYURLTypes
{
    public final static long              = (-1);
    spyURLTypeAuto
    public final static long              = 0;
    spyURLTypeXML
    public final static long              = 1;
    spyURLTypeDTD
}
```


© 2016-2022 Altova GmbH

29.3.5.30.33 SpyViewModes (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYViewModes
{
    public final static long spyViewGrid     = 0;
    public final static long spyViewText     = 1;
    public final static long spyViewBrowser  = 2;
    public final static long spyViewSchema   = 3;
    public final static long spyViewContent  = 4;
    public final static long spyViewAuthentic= 4;
    public final static long spyViewWSDL     = 5;
    public final static long spyViewZIP      = 6;
    public final static long                 = 7;
    spyViewEditionInfo
}
```

29.3.5.30.34 SPYWhitespaceComparison (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYWhitespaceComparison
  {
     public final static long spyCompareAsIs  = 0;
     public final static long                 = 1;
     spyCompareNormalized
     public final static long spyStripAll     = 2;
  }
```

29.3.5.30.35 SPYXMLDataKind (obsolete)

The objects described in this section (Application API for Java) are obsolete from v2012 onwards.

For information about how to access the Application API from Java code, see the section: Programming Languages | Java.

```
public class SPYXMLDataKind
{
   public final static long                  = 0;
   spyXMLDataXMLDocStruct
   public final static long                  = 1;
   spyXMLDataXMLEntityDocStruct
   public final static long                  = 2;
   spyXMLDataDTDDocStruct
   public final static long spyXMLDataXML    = 3;
   public final static long                  = 4;
   spyXMLDataElement
   public final static long spyXMLDataAttr   = 5;
   public final static long spyXMLDataText   = 6;
   public final static long spyXMLDataCData  = 7;
   public final static long                  = 8;
   spyXMLDataComment
   public final static long spyXMLDataPI     = 9;
   public final static long                  = 10;
   spyXMLDataDefDoctype
   public final static long                  = 11;
   spyXMLDataDefExternalID
   public final static long                  = 12;
   spyXMLDataDefElement
   public final static long                  = 13;
   spyXMLDataDefAttlist
   public final static long                  = 14;
   spyXMLDataDefEntity
   public final static long                  = 15;
   spyXMLDataDefNotation
   public final static long                  = 16;
   spyXMLDataKindsCount
}
```


© 2016-2022 Altova GmbH

29.4 ActiveX Integration

The XMLSpy user interface and the functionality described in this section can be integrated into custom applications that can consume ActiveX controls. ActiveX technology enables a wide variety of languages to be used for integration, such as C++, C#, VB.NET, HTML. (Note that ActiveX components integrated in HTML must be run with Microsoft Internet Explorer versions and platforms that support ActiveX). All components are full OLE Controls. Integration into Java is provided through wrapper classes.

To integrate the ActiveX controls into your custom code, the XMLSpy Integration Package must be installed (see https://www.altova.com/components/download). Ensure that you install XMLSpy first, and then the XMLSpy Integration Package. Other prerequisites apply, depending on language and platform (see Prerequisites).

You can flexibly choose between two different levels of integration: application level and document level.

Integration at application level means embedding the complete interface of XMLSpy (including its menus, toolbars, panes, etc) as an ActiveX control into your custom application. For example, in the most simple scenario, your custom application could consist of only one form that embeds the XMLSpy graphical user interface. This approach is easier to implement than integration at document level but may not be suitable if you need flexibility to configure the XMLSpy graphical user interface according to your custom requirements.

Integration at document level means embedding XMLSpy into your own application piece-by-piece. This includes implementing not only the main XMLSpy control but also the main document editor window, and, optionally, any additional windows. This approach provides greater flexibility to configure the GUI, but requires advanced interaction with ActiveX controls in your language of choice.

The sections Integration at the Application Level and Integration at Document Level describe the key steps at these respective levels. The ActiveX Integration Examples section provides examples in C#, HTML, and Java. Looking through these examples will help you to make the right decisions quickly. The Object Reference section describes all COM objects that can be used for integration, together with their properties and methods.

For information about using XMLSpy as a Visual Studio plug-in, see XMLSpy in Visual Studio.

29.4.1 Prerequisites

To integrate the XMLSpy ActiveX control into a custom application, the following must be installed on your computer:

- XMLSpy
- The XMLSpy Integration Package, available for download at https://www.altova.com/components/download

To integrate the 64-bit ActiveX control, install the 64-bit versions of XMLSpy and XMLSpy Integration Package. For applications developed under Microsoft .NET platform with Visual Studio, both the 32-bit and 64-bit versions of XMLSpy and XMLSpy Integration Package must be installed, as explained below.

Microsoft .NET (C#, VB.NET) with Visual Studio

To integrate the XMLSpy ActiveX control into a 32-bit application developed under Microsoft .NET, the following must be installed on your computer:

- Microsoft .NET Framework 4.0 or later
- Visual Studio 2012/2013/2015/2017/2019/2022
- XMLSpy 32-bit and XMLSpy Integration Package 32-bit
- The ActiveX controls must be added to the Visual Studio toolbox (see Adding the ActiveX Controls to the Toolbox).

If you want to integrate the 64-bit ActiveX control, the following prerequisites apply in addition to the ones above:

- XMLSpy 32-bit and XMLSpy Integration Package 32-bit must still be installed (this is required to provide the 32-bit ActiveX control to the Visual Studio designer, since Visual Studio runs on 32-bit)
- XMLSpy 64-bit and XMLSpy Integration Package 64-bit must be installed (provides the actual 64-bit ActiveX control to your custom application at runtime)
- In Visual Studio, create a 64-bit build configuration and build your application using this configuration. For an example, see Running the Sample C# Solution.

Java

To integrate the XMLSpy ActiveX control into Java application using the Eclipse development environment, the following must be installed on your computer:

- Java Runtime Environment (JRE) or Java Development Kit (JDK) 7 or later
- Eclipse
- XMLSpy and XMLSpy Integration Package

Note: To run the 64-bit version of the XMLSpy ActiveX control, use a 64-bit version of Eclipse, as well as the 64-bit version of XMLSpy and the XMLSpy Integration Package.

XMLSpy integration and deployment on client computers

If you create a .NET application and intend to distribute it to other clients, you will need to install the following on the client computer(s):

- XMLSpy
- The XMLSpy Integration Package
- The custom integration code or application.

29.4.2 Adding the ActiveX Controls to the Toolbox

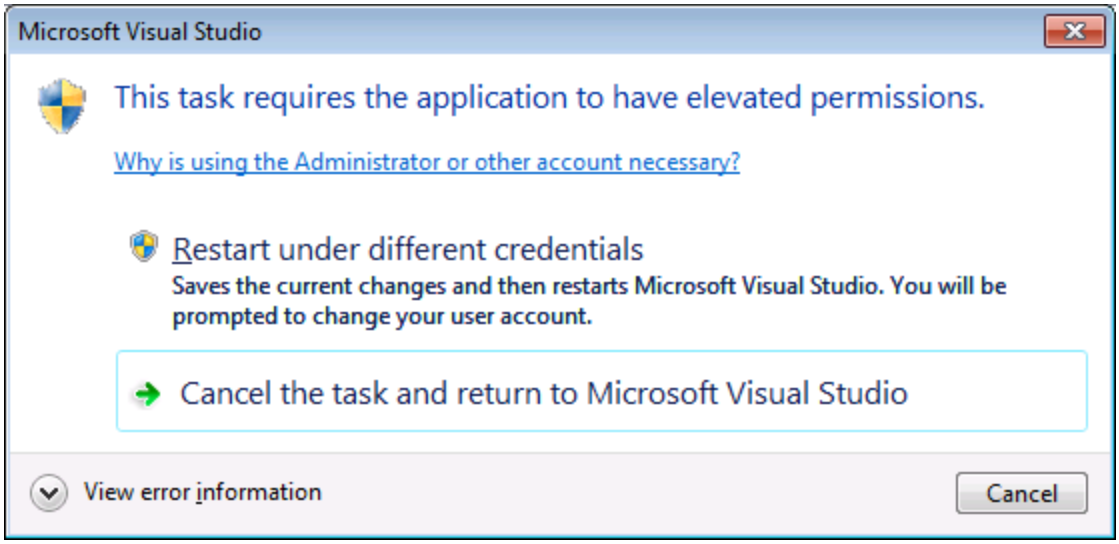
To use the XMLSpy ActiveX controls in an application developed with Visual Studio, the controls must first be added to the Visual Studio Toolbox, as follows:

1. On the **Tools** menu of Visual Studio, click **Choose Toolbox Items**.
2. On the **COM Components** tab, select the check boxes next to the XMLSpyControl, XMLSpyControl Document, and XMLSpyControl Placeholder.

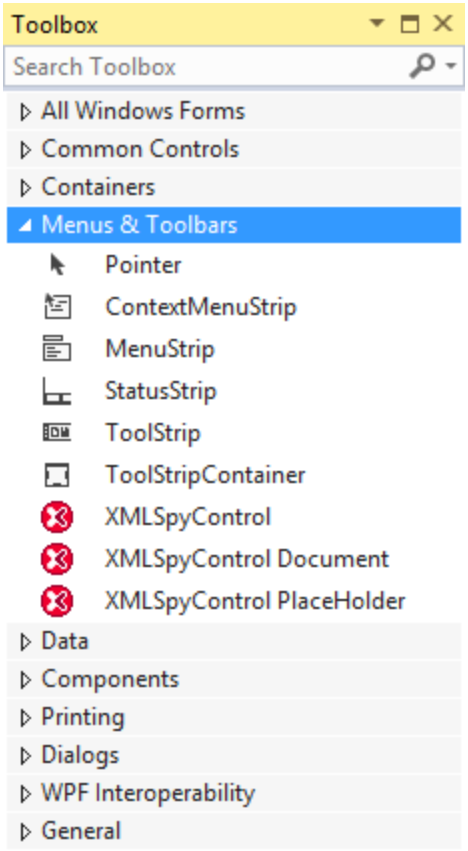
© 2016-2022 Altova GmbH

In case the controls above are not available, follow the steps below:

1. On the **COM Components** tab, click **Browse**, and select the **XMLSpyControl.ocx** file from the XMLSpy installation folder. Remember that the XMLSpy Integration Package must be installed; otherwise, this file is not available, see Prerequisites.
2. If prompted to restart Visual Studio with elevated permissions, click **Restart under different credentials**.



If the steps above were successful, the XMLSpy ActiveX controls become available in the Visual Studio Toolbox.



Note: For an application-level integration, only the **XMLSpyControl** ActiveX control is used (see Integration at Application Level). The **XMLSpyControl Document** and **XMLSpyControl Placeholder** controls are used for document-level integration (see Integration at Document Level).

29.4.3 Integration at Application Level

Integration at application level allows you to embed the complete interface of XMLSpy into a window of your application. With this type of integration, you get the whole user interface of XMLSpy, including all menus, toolbars, the status bar, document windows, and helper windows. Customization of the application's user interface is restricted to what XMLSpy provides. This includes rearrangement and resizing of helper windows and customization of menus and toolbars.

The only ActiveX control you need to integrate is `XMLSpyControl`. Do not instantiate or access `XMLSpyControlDocument` or `XMLSpyControlPlaceHolder` ActiveX controls when integrating at application-level.

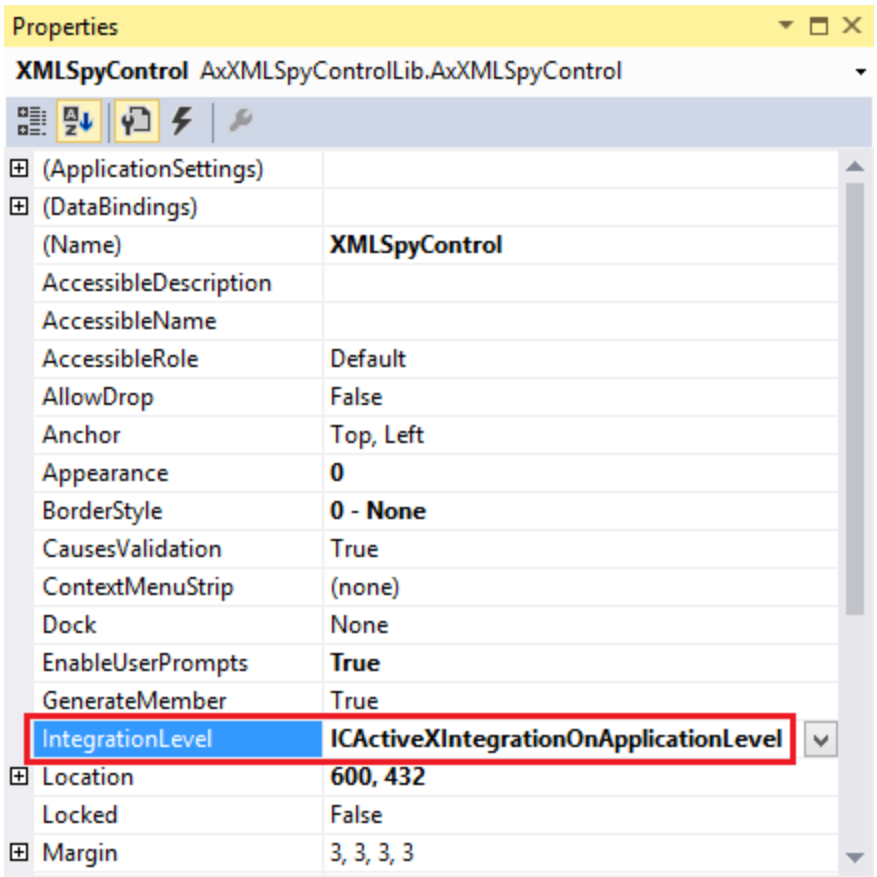
If you have any initialization to do or if you want to automate some behaviour of XMLSpy, use the properties, methods, and events described for `XMLSpyControl`. Consider using `XMLSpyControl.Application` for more complex access to XMLSpy functionality.

For an example that shows how the XMLSpy application can be embedded in an HTML page, see HTML Integration at Application Level.

© 2016-2022 Altova GmbH

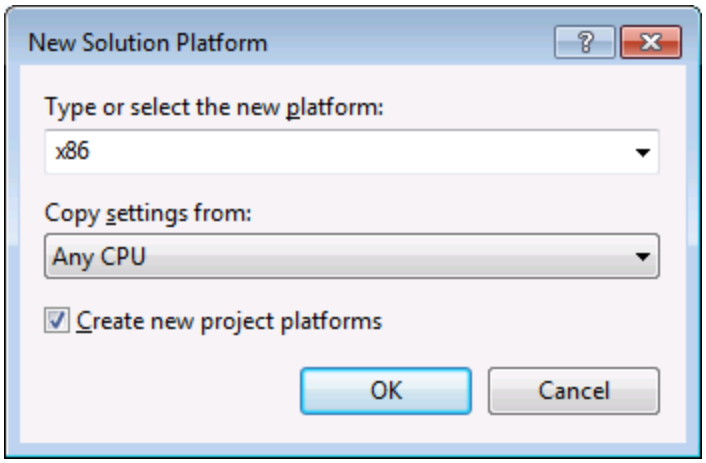
In C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the XMLSpy ActiveX controls at application level are as follows:

1. Check that all prerequisites are met (see Prerequisites).
2. Create a new Visual Studio Windows Forms project with a new empty form.
3. If you have not done that already, add the ActiveX controls to the toolbox (see Adding the ActiveX Controls to the Toolbox).
4. Drag the **XMLSpyControl** from the toolbox onto your new form.
5. Select the **XMLSpyControl** on the form, and, in the Properties window, set the **IntegrationLevel** property to **ICActiveXIntegrationOnApplicationLevel**.



6. Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:

   a. Right-click the solution in Visual Studio, and select **Configuration Manager**.
   b. Under **Active solution platform**, select **New...** and then select the x86 or x64 configuration (in this example, **x86**).



You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64).

29.4.4 Integration at Document Level

Compared to integration at application level, integration at document level is a more complex, yet more flexible way to embed XMLSpy functionality into your application by means of ActiveX controls. With this approach, your code can access selectively the following parts of the XMLSpy user interface:

- Document editing window
- Project window
- Entry helper windows
- Validator output window
- XPath profiler window
- XPath dialog window
- XSLT/XQuery debugger windows
- SOAP debugger window

As mentioned in Integration at Application Level, for an ActiveX integration at application level, only one control is required, namely the **XMLSpyControl**. However, for an ActiveX integration at document level, XMLSpy functionality is provided by the following ActiveX controls:

- XMLSpyControl
- XMLSpyControl Document
- XMLSpyControl Placeholder

These controls are supplied by the **XMLSpyControl.ocx** file available in the application installation folder of XMLSpy. When you develop the ActiveX integration with Visual Studio, you will need to add these controls to the Visual Studio toolbox (see Adding the ActiveX Controls to the Toolbox).

The basic steps to integrate the ActiveX controls at document level into your application are as follows:

1. First, instantiate **XMLSpyControl** in your application. Instantiating this control is mandatory; it enables support for the **XMLSpyControl Document** and **XMLSpyControl Placeholder** controls mentioned above. It is important to set the `IntegrationLevel` property to

 © 2016-2022 Altova GmbH

ICActiveXIntegrationOnDocumentLevel (or "1"). To hide the control from the user, set its **Visible** property to **False**.

Note: When integrating at document level, do not use the **Open** method of the **XMLSpyControl**; this might lead to unexpected results. Use the corresponding open methods of **XMLSpyControl Document** and **XMLSpyControl PlaceHolder** instead.

2. Create at least one instance of **XMLSpyControl Document** in your application. This control supplies the document editing window of XMLSpy to your application and can be instantiated multiple times if necessary.

   Use the method **Open** to load any existing file. To access document-related functionality, use the **Path** and **Save** or methods and properties accessible via the property **Document**.

Note: The control does not support a read-only mode. The value of the property **ReadOnly** is ignored.

3. Optionally, add to your application the **XMLSpyControl Placeholder** control for each additional window (other than the document window) that must be available to your application.

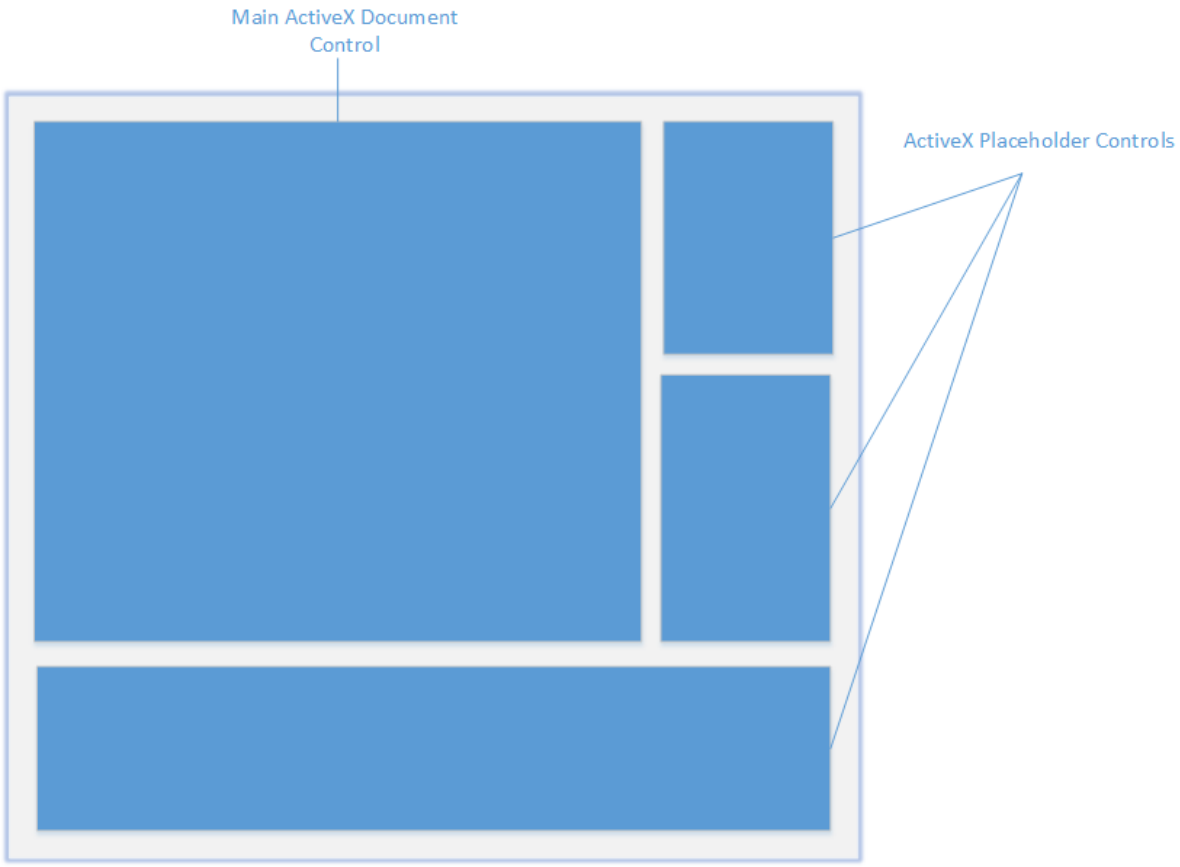
   Instances of **XMLSpyControl PlaceHolder** allow you to selectively embed additional windows of XMLSpy into your application. The window kind (for example, Project window) is defined by the property **PlaceholderWindowID**. Therefore, to set the window kind, set the property **PlaceholderWindowID**. For valid window identifiers, see XMLSpyControlPlaceholderWindow.

Note: Use only one **XMLSpyControl PlaceHolder** for each window identifier.

   For placeholder controls that select the XMLSpy project window, additional methods are available. Use **OpenProject** to load a XMLSpy project. Use the property Project and the methods and properties from the XMLSpy automation interface to perform any other project related operations.

For example, in C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the XMLSpy ActiveX controls at document level could be similar to those listed below. Note that your application may be more complex if necessary; however, the instructions below are important to understand the minimum requirements for an ActiveX integration at document level.

1. Create a new Visual Studio Windows Forms project with a new empty form.
2. If you have not done that already, add the ActiveX controls to the toolbox (see Adding the ActiveX Controls to the Toolbox).
3. Drag the **XMLSpyControl** from the toolbox onto your new form.
4. Set the **IntegrationLevel** property of the **XMLSpyControl** to **ICActiveXIntegrationOnDocumentLevel**, and the **Visible** property to **False**. You can do this either from code or from the **Properties** window.
5. Drag the **XMLSpyControl Document** from the toolbox onto the form. This control provides the main document window of XMLSpy to your application, so you may need to resize it to a reasonable size for a document.
6. Optionally, add one or more **XMLSpyControl Placeholder** controls to the form (one for each additional window type that your application needs, for example, the **Project** window). You will typically want to place such additional placeholder controls either below or to the right or left of the main document control, for example:



7. Set the **PlaceholderWindowID** property of each **XMLSpyControl Placeholder** control to a valid window identifier. For the list of valid values, see XMLSpyControlPlaceholderWindow.
8. Add commands to your application (at minimum, you will need to open, save and close documents), as shown below.

Querying XMLSpy Commands

When you integrate at document level, no XMLSpy menu or toolbar is available to your application. Instead, you can retrieve the required commands, view their status, and execute them programmatically, as follows:

- To retrieve all available commands, use the `CommandsList` property of the **XMLSpyControl**.
- To retrieve commands organized according to their menu structure, use the `MainMenu` property.
- To retrieve commands organized by the toolbar in which they appear, use the `Toolbars` property.
- To send commands to XMLSpy, use the `Exec` method.
- To query if a command is currently enabled or disabled, use the `QueryStatus` method.

This enables you to flexibly integrate XMLSpy commands into your application's menus and toolbars.

Your installation of XMLSpy also provides you with command label images used within XMLSpy. See the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your XMLSpy installation for icons in GIF format. The file names correspond to the command names as they are listed in the Command Reference section.

© 2016-2022 Altova GmbH

General considerations

To automate the behaviour of XMLSpy, use the properties, methods, and events described for the **XMLSpyControl**, **XMLSpyControl Document**, and **XMLSpyControl Placeholder**.

For more complex access to XMLSpy functionality, consider using the following properties:

- `XMLSpyControl.Application`
- `XMLSpyControlDocument.Document`
- `XMLSpyControlPlaceHolder.Project`

These properties give you access to the XMLSpy automation interface (XMLSpyAPI)

Note: To open a document, always use `XMLSpyControlDocument.Open` or `XMLSpyControlDocument.New` on the appropriate document control. To open a project, always use `XMLSpyControlPlaceHolder.OpenProject` on a placeholder control embedding a XMLSpy project window.

For examples that show how to instantiate and access the necessary controls in different programming environments, see ActiveX Integration Examples.

29.4.5 ActiveX Integration Examples

This section contains examples of XMLSpy document-level integration using different container environments and programming languages. Source code for all examples is available in the folder `<ApplicationFolder>\Examples\ActiveX` of your XMLSpy installation.

29.4.5.1 C#

A basic ActiveX integration example solution for C# and Visual Studio is available in the folder `<ApplicationFolder>\Examples\ActiveX\C#`. Before you compile the source code and run the sample, make sure that all prerequisites are met (see Running the Sample C# Solution).

29.4.5.1.1 Running the Sample C# Solution

The sample Visual Studio solution available in the folder **<ApplicationFolder>\Examples\ActiveX\C#** illustrates how to consume the XMLSpy ActiveX controls. Before attempting to build and run this solution, note the following steps:

Step 1: Check the prerequisites

Visual Studio 2010 or later is required to open the sample solution. For the complete list of prerequisites, see Prerequisites.

Step 2: Copy the sample to a directory where you have write permissions

To avoid running Visual Studio as an Administrator, copy the source code to a directory where you have write permissions, instead of running it from the default location.

Step 3: Check and set all required control properties

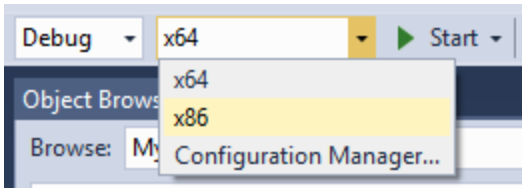
The sample application contains one instance of XMLSpyControlDocument and one instance of XMLSpyControlPlaceHolder controls. Double-check that the following properties of these controls are set as shown in the table below:

Control name	Property	Property value
XMLSpyControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel
XPathDialog	PlaceholderWindowID	16

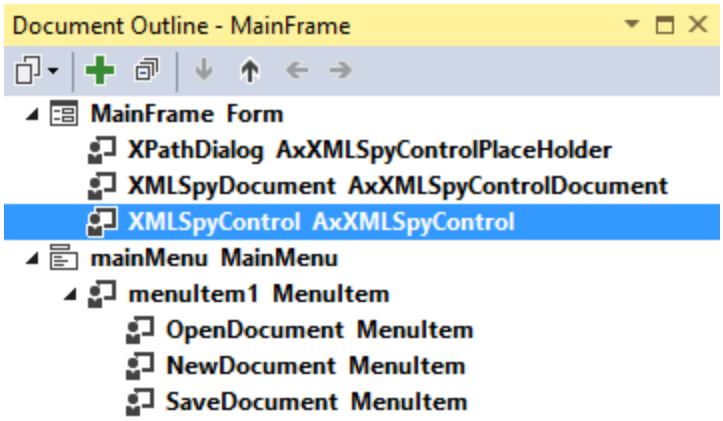
Here is how you can view or set the properties of an ActiveX control:

1. Open the **MDIMain.cs** form in the designer window.

Note: On 64-bit Windows, it may be necessary to change the build configuration of the Visual Studio solution to "x86" **before** opening the designer window. If you need to build the sample as a 64-bit application, see Prerequisites.

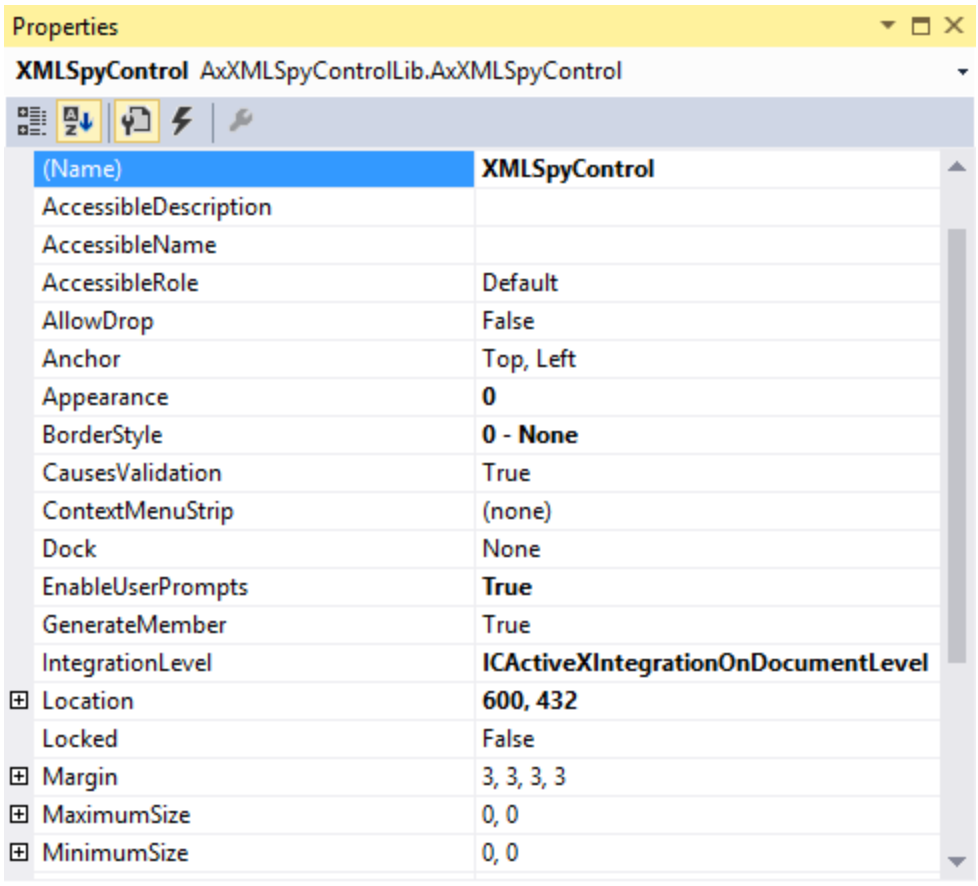


2. Open the **Document Outline** window of Visual Studio (On the **View** menu, click **Other Windows | Document Outline**).



3. Click an ActiveX control in the **Document Outline** window, and edit its required property in the **Properties** window, for example:

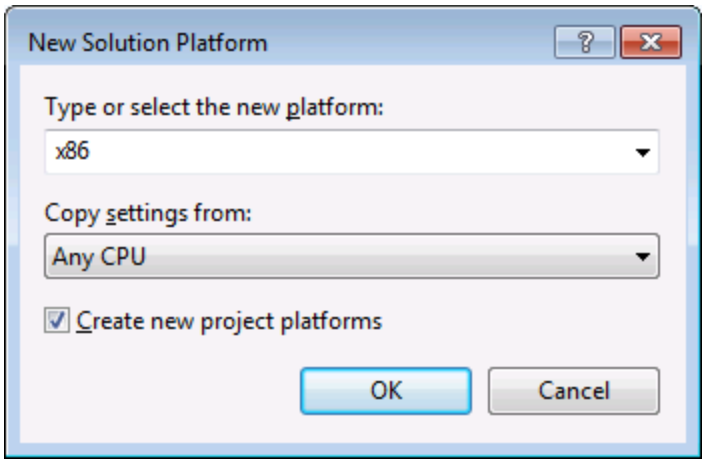
 © 2016-2022 Altova GmbH



Step 4: Set the build platform

- Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:

  a. Right-click the solution in Visual Studio, and select **Configuration Manager**.
  b. Under **Active solution platform**, select **New...** and then select the x86 or x64 configuration (in this example, **x86**).



You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64); otherwise, runtime errors might occur.

On running the sample, the main MDI Frame window is created and contains an editing window with an empty XML document and a XPath Dialog window of XMLSpy at the bottom. Use **File | Open** to open any XML file from the XMLSpy examples folder. The file is loaded and displayed. After you load the document, you can start using the XPath dialog. Note that you may need to slightly drag the lower-right corner of the form to cause the dialog to redraw itself and display its contents.

29.4.5.2 HTML

The code listings in this section show how to integrate the XMLSpyControl at application level and document level. Source code for all examples is available in the folder `<ApplicationFolder>\Examples\ActiveX\HTML` of your XMLSpy installation.

Note: ActiveX controls in an HTML page are supported only by Internet Explorer when it runs as a 32-bit application. When Internet Explorer 10 or 11 runs in 64-bit mode, it does not load ActiveX controls. The default browser security settings will normally block ActiveX, so you will need to explicitly allow blocked content to run on the page when prompted by Internet Explorer.

29.4.5.2.1 HTML Integration at Application Level

This example shows a simple integration of the XMLSpy control at application-level into a HTML page. The integration is described in the following sections:

- Instantiate a XMLSpyControl in HTML code.
- Implement buttons to load documents and automate code-generation tasks.
- Define actions for some application events.

The code for this example is available at the following location in your XMLSpy installation: `<ApplicationFolder>\Examples\ActiveX\HTML\XMLSpyActiveX_ApplicationLevel.htm`.

29.4.5.2.1.1 Instantiate the Control

The HTML `Object` tag is used to create an instance of the XMLSpyControl. The `Classid` is that of XMLSpyControl. Width and height specify the window size. No additional parameters are necessary, since application-level is the default.

```
<OBJECT id="objXMLSpyControl"
        Classid="clsid:a258bba2-3835-4c16-8590-72b44f52c471"
        width="1000"
        height="700"
        VIEWASTEXT>
</OBJECT>
```


© 2016-2022 Altova GmbH

29.4.5.2.1.2 Add Button to Open Default Document

As a simple example of how to automate some tasks, we add a button to the page:

```
<input type="button" value="Open OrgChart.pxf" onclick="BtnOpenMEFile()">
```

When clicked, a predefined document will be opened in the XMLSpyControl. The `MakeAbsolutePath` method creates an absolute path using the location of the script as a base path.

```
function BtnOpenMEFile()
{
   var strPath = MakeAbsolutePath("OrgChart.pxf");
   var objDoc = objXMLSpyControl.Open(strPath );

   if (objDoc == null)
      alert("Unable to locate Orgchart.pxf at: " + objXMLSpyControl.BaseHref);
}
```

29.4.5.2.1.3 Add Buttons for Code Generation

For direct access, we want to have a button that will validate the current document. The method is similar to that used in the previous section.

First comes the button:

```
<input type="button" value="Validate" onclick="BtnValidate()">
```

Then we provide the script that will validate the current document.

```
// ------------------------------------------------------------------------
// check validity of current document.
// if validation fails, show validation result in alert box .
function BtnValidate()
{
   // get top-level object of automation interface
   var objApp = objXMLSpyControl.Application;

   // get the active document
   var objDocument = objApp.ActiveDocument;

   if ( objDocument == null )
      alert( "no active document found" );
   else
   {
      // define as arrays to support their usage as return parameters
      var errorText = new Array(1);
      var errorPos = new Array(1);
```

```
      var badData = new Array(1);

      var valid = objDocument.IsValid(errorText, errorPos, badData);

      if (! valid)
      {
            // compose the error description
            var text = errorText;

            // access that XMLData object only if filled in
            if (badData[0] != null)
               text += "(" + badData[0].Name + "/" + badData[0].TextValue + ")";


            alert("Validation error[" + errorPos + "]: " + text);
      }
      else
         alert("Document is valid");
   }
}
```

29.4.5.2.1.4 Connect to Custom Events

The example implements two event callbacks for XMLSpyControl custom events to show the principle:

```
<!-- ------------------------------------------------------------ -->
<!--  custom event 'OnDocumentOpened" of XMLSpyControl object  -->
<SCRIPT FOR="objXMLSpyControl" event="OnDocumentOpened( objDocument )"
LANGUAGE="javascript">
   // alert("Document '" + objDocument.Name + "' opend!");
</SCRIPT>

<!-- ------------------------------------------------------------ -->
<!--  custom event 'OnDocumentClosed" of XMLSpyControl object  -->
<SCRIPT FOR="objXMLSpyControl" event="OnDocumentClosed( objDocument )"
LANGUAGE="javascript">
   // alert("Document '" + objDocument.Name + "' closed!");
</SCRIPT>
```

29.4.5.2.2 HTML Integration at Document Level

This example shows an integration of the XMLSpy control at document-level into a HTML page. The following topics are covered:

- Instantiate a XMLSpyControl ActiveX control object in HTML code
- Instantiate a XMLSpyControlDocument ActiveX control to allow editing a XMLSpy file
- Instantiate one XMLSpyControlPlaceHolder to alternatively host one of the XMLSpy helper windows
- Instantiate one XMLSpyControlPlaceHolder ActiveX control to show the XMLSpy validation output window

© 2016-2022 Altova GmbH

- Create a simple custom toolbar for some heavy-used XMLSpy commands
- Add some more buttons that use the COM automation interface of XMLSpy
- Use event handlers to update command buttons

This example is available in its entirety in the file `XMLSpyActiveX_DocumentLevel.htm` within the `<ApplicationFolder>\Examples\ActiveX\HTML\` folder of your XMLSpy installation.

29.4.5.2.2.1 Instantiate the XMLSpyControl

The HTML `OBJECT` tag is used to create an instance of the XMLSpyControl. The Classid is that of XMLSpyControl. Width and height are set to 0 since we use this control as manager control without use for its user interface. The integration level is specified as a parameter within the `OBJECT` tag.

```
<object id="objXMLSpyX" classid="clsid:a258bba2-3835-4c16-8590-72b44f52c471" width="0"
height="0" VIEWASTEXT>
   <param name="IntegrationLevel" value="1">
</object>
```

29.4.5.2.2.2 Create Editor Window

The HTML `OBJECT` tag is used to embed an editing window. The additional custom parameter specifies that the control is to be initialized with a new empty document.

```
<object id="objDoc1" classid="clsid:52A552E6-2AB8-4e3e-B545-BE998233DDA0" width="600"
height="500" VIEWASTEXT>
   <param name="NewDocument">
</object>
```

29.4.5.2.2.3 Create Placeholder for Helper Windows

The **XMLSpyControlPlaceHolder** control is required to host an application helper window, see also Integration at Document Level. In the code listing below, the HTML `object` tag is used to instantiate a control that will host the **Output** window by default (`PlaceholderWindowID` is 3).

```
<!-- create helper window placeholder control.           -->
<!-- the editor with focus will automatically direct its -->
<!-- output to the appropriate helper window.            -->
<object id="objOutputWindow" classid="clsid:135DEEF4-6DF0-47c2-8F8C-F145F5F3F672"
width="200" height="200" VIEWASTEXT>
   <param name="PlaceholderWindowID" value="3">
</object>
```

The HTML page also includes an example object that can host any entry helper window:

```
<!-- create helper window placeholder control.           -->
<!-- the editor with focus will automatically direct its -->
<!-- output to the appropriate helper window.            -->
<object id="objEHWindow" classid="clsid:135DEEF4-6DF0-47c2-8F8C-F145F5F3F672" width="200"
 height="200" VIEWASTEXT>
   <param name="PlaceholderWindowID" value="0">
</object>
```

It is possible to repopulate the object above with a different helper window by clicking the following buttons:

```
<!-- create buttons to switch between helper windows shown -->
<!-- in this placeholde window.                            -->
<tr border="0">
   <td align="center">
      <input type="button" value="EH - Elements" onclick="BtnHelperWindow(0)">
   </td>
</tr>
<tr align="center">
   <td align="center">
      <input type="button" value="EH - Attributes" onclick="BtnHelperWindow(1)">
   </td>
</tr>
<tr align="center">
   <td align="center">
      <input type="button" value="EH - Entities" onclick="BtnHelperWindow(2)">
   </td>
</tr>
```

When clicked, these buttons call the `BtnHelperWindow` method, which reassigns the `PlaceholderWindowID` of the control, and thus causes the ActiveX object to display a different helper window.

```
// specify which of the helper windows shall be shown in the placeholder control.
function BtnHelperWindow(i_ePlaceholderWindowID)
{
   objPlaceholderWindow.PlaceholderWindowID = i_ePlaceholderWindowID;
}
```

For the list of possible values of `PlaceholderWindowID`, see XMLSpyControlPlaceholderWindow.

29.4.5.2.2.4 Create a Custom Toolbar

The example HTML page also includes a custom toolbar (intended as a replica of the XMLSpy menu). The custom toolbar consists of buttons with images of XMLSpy commands, for example:

```
<button id="btnWellFormed" title="Check Well-formedness" onclick="BtnDoCommand(34049)">
   <img src="..\Images\IDC_CHECK_WELL_FORM.gif" width="16" height="16" />
</button>
<button id="btnValidate" title="Validate" onclick="BtnDoCommand(34174)">
```


© 2016-2022 Altova GmbH

```
   <img src="..\Images\IDC_VALIDATE.gif" width="16" height="16" />
</button>
```

The names of button images correspond to the command ID numbers, see Command Reference. On clicking the button, the corresponding command ID is sent to the main control and executed:

```
// perform any command specified by cmdID.
// command routing includes application, active document and view.
function BtnDoCommand(cmdID)
{
   objXMLSpyX.Exec( cmdID );
   msgtext.innerText = "Command " + cmdID + " performed.";
}
```

29.4.5.2.2.5 Create More Buttons

In the example, we add some more buttons to show some automation code.

```
<!-- add some buttons associated with above editor.               -->
<!-- generation of code is now implemented using the XMLSpy automation  -->
<!-- interface to select a target folder without prompting the user.      -->
<p>
   <input type="button" value="New File" onclick="BtnNewFile(objDoc1)">
   <input type="button" value="Save File" onclick="BtnSaveFile(objDoc1)">
   <input type="text" title="Path" id="strPath" width="150">
   <input type="button" value="Open OrgChart.pxf" onclick="BtnOpenFile(objDoc1,
'OrgChart.pxf')">
</p>
```

The corresponding JavaScript looks like this:

```
// ------------------------------------------------------------------
// open a new empty document in the specified document control window.
function BtnNewFile(objDocCtrl)
{
   objDocCtrl.Open("");
   objDocCtrl.setActive();
}

// ------------------------------------------------------------------
// Saves the current file in the specified document control window.
function BtnSaveFile(objDocCtrl)
{
   if(objDocCtrl.Path.length > 0)
objDocCtrl.Save();
   else
   {
if(strPath.value.length > 0)
```

```
{
   objDocCtrl.Path = strPath.value;
   objDocCtrl.Save();
}
else
{
   alert("Please set path for the document first!");
   strPath.focus();
}
   }

   objDocCtrl.setActive();
}

// ------------------------------------------------------------
// open a document in the specified document control window.
function BtnOpenFile(objDocCtrl, strFileName)
{
   // do not use XMLSpyX.Application.OpenDocument(...) to open a document,
   // since then XMLSpyControl wouldn't know a control window to show
   // the document in. Instead:

   var strPath = MakeAbsolutePath(strFileName);
   DoOpenFile(objDocCtrl, strPath);
}
```

29.4.5.2.2.6 Create Event Handler to Update Button Status

Availability of a command may vary with every mouse click or keystroke. The custom event `OnUpdateCmdUI` of XMLSpyControl gives us an opportunity to update the enabled/disabled state of buttons associated with XMLSpy commands. The method `XMLSpyControl.QueryStatus` is used to query whether a command is enabled or not.

```
<!--  custom event 'OnUpdateCmdUI" of XMLSpyControl object  -->
function objXMLSpyX::OnUpdateCmdUI()
{
   if ( document.readyState == "complete" )// 'complete'
   {
      // update status of buttons
      btnWellFormed.disabled =  ! (objDoc1.QueryStatus(34049) & 0x02);   // not enabled
      btnValidate.disabled =  ! (objDoc1.QueryStatus(34174) & 0x02);   // not enabled
   }
}
```

29.4.5.3 Java

XMLSpy ActiveX components can be accessed from Java code. Java integration is provided by the libraries listed below. These libraries are available in the folder `<ApplicationFolder>\Examples\JavaAPI` of your

© 2016-2022 Altova GmbH

XMLSpy installation, after you have installed both XMLSpy and the XMLSpy Integration Package (see also Prerequisites).

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (in case of the 32-bit installation of XMLSpy)
- `AltovaAutomation_x64.dll`: a JNI wrapper for Altova automation servers (in case of the 64-bit installation of XMLSpy)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `XMLSpyActiveX.jar`: Java classes that wrap the XMLSpy ActiveX interface
- `XMLSpyActiveX_JavaDoc.zip`: a Javadoc file containing help documentation for the Java interface

Note: In order to use the Java ActiveX integration, the .dll and .jar files must be included in the Java class search path.

Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details, see Example Java Project.

Rules for mapping the ActiveX Control names to Java

For the documentation of ActiveX controls, see Object Reference. Note that the object naming conventions are slightly different in Java compared to other languages. Namely, the rules for mapping between the ActiveX controls and the Java wrapper are as follows:

- ***Classes and class names***
  For every component of the XMLSpy ActiveX interface a Java class exists with the name of the component.

- ***Method names***
  Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with get and set can be used. If a property does not support write-access, no setter method is available. Example: For the `IntegrationLevel` property of the XMLSpyControl, the Java methods `getIntegrationLevel` and `setIntegrationLevel` are available.

- ***Enumerations***
  For every enumeration defined in the ActiveX interface, a Java enumeration is defined with the same name and values.

- ***Events and event handlers***
  For every interface in the automation interface that supports events, a Java interface with the same name plus 'Event' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus '`DefaultHandler`'. For example:
  `XMLSpyControl`: Java class to access the application
  `XMLSpyControlEvents`: Events interface for the XMLSpyControl
  `XMLSpyControlEventsDefaultHandler`: Default handler for `XMLSpyControlEvents`

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
XMLSpyControlDocument, method New	newDocument
Document, method SetEncoding	setFileEncoding
AuthenticView, method Goto	gotoElement
AuthenticRange, method Goto	gotoElement
AuthenticRange, method Clone	cloneRange

This section

This section shows how some basic XMLSpy ActiveX functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project
- Creating the ActiveX Controls
- Loading Data in the Controls
- Basic Event Handling
- Menus
- UI Update Event Handling
- Creating an XML Tree

29.4.5.3.1 Example Java Project

The XMLSpy installation package contains an example Java project, located in the ActiveX Examples folder of the application folder: <ApplicationFolder>\Examples\ActiveX\Java\.

The Java example shows how to integrate the XMLSpyControl in a common desktop application created with Java. You can test it directly from the command line using the batch file BuildAndRun.bat, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

File list

The Java examples folder contains all the files required to run the example project. These files are listed below:

.classpath	Eclipse project helper file
.project	Eclipse project file
AltovaAutomation.dll	Java-COM bridge: DLL part (for the 32-bit installation)
AltovaAutomation_x64.dll	Java-COM bridge: DLL part (for the 64-bit installation)
AltovaAutomation.jar	Java-COM bridge: Java library part
BuildAndRun.bat	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
XMLSpyActiveX.jar	Java classes of the XMLSpy ActiveX control

© 2016-2022 Altova GmbH

XMLSpyActiveX_JavaDoc.zip	Javadoc file containing help documentation for the Java API
XMLSpyContainer.java	Java example source code
XMLSpyContainerEventHandler.java	Java example source code
XMLTreeDialog.java	Java example source code

What the example does

The example places one XMLSpy document editor window, the XMLSpy project window, the XMLSpy XPath window and an XMLSpy entry helper in an AWT frame window. It reads out the File menu defined for XMLSpy and creates an AWT menu with the same structure. You can use this menu or the project window to open and work with files in the document editor.

You can modify the example in any way you like.

The following specific features are described in code listings:

- Creating the ActiveX Controls: Starts XMLSpy, which is registered as an automation server, or activates XMLSpy if it is already running.
- Loading Data in the Controls: Locates one of the example documents installed with XMLSpy and opens it.
- Basic Event Handling: Changes the view of all open documents to Text View. The code also shows how to iterate through open documents.
- Menus: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- UI Update Event Handling: Shows how to handle XMLSpy events.
- Creating an XML Tree: Shows how to create an XML tree and prepare it for modal activation.

Updating the path to the Examples folder

Before running the provided sample, you may need to edit the **XMLSpyContainer.java** file. Namely, check that the following path refers to the actual folder where the XMLSpy example files are stored on your operating system:

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\Documents\\Altova\
\XMLSpy2023\\XMLSpyExamples\\";
```

Running the example from the command line

To run the example from the command line:

1. Check that all prerequisites are met (see Prerequisites).
2. Open a command prompt window, change the current directory to the sample Java project folder, and type:

   ```
   buildAndRun.bat "<Path-to-the-Java-bin-folder>"
   ```

3. Press **Enter**.

The Java source in `XMLSpyContainer.java` will be compiled and then executed.

Compiling and running the example in Eclipse

To import the sample Java project into Eclipse:

1. Check that all prerequisites are met (see Prerequisites).
2. On the **File** menu, click **Import**.
3. Select **Existing Projects into Workspace**, and browse for the Eclipse project file located at `<ApplicationFolder>\Examples\ActiveX\Java\`. Since you may not have write-access in this folder, it is recommended to select the **Copy projects into workspace** check box on the Import dialog box.

To run the example application, right-click the project in Package Explorer and select the command **Run as | Java Application**.

Help for Java API classes is available through comments in code as well as the Javadoc view of Eclipse. To enable the Javadoc view in Eclipse, select the menu command **Window | Show View | JavaDoc**.

29.4.5.3.2 Creating the ActiveX Controls

The code listing below show how ActiveX controls can be created. The constructors will create the Java wrapper objects. Adding these Canvas-derived objects to a panel or to a frame will trigger the creation of the wrapped ActiveX object.

```
01  /**
02  * XMLSpy manager control - always needed
03  */
04    public static XMLSpyControl         xmlSpyControl = null;
05
06  /**
07  * XMLSpy document editing control
08  */
09    public static XMLSpyControlDocument     xmlSpyDocument = null;
10
11  /**
12  * Tool windows - XMLSpy place-holder controls
13  */
14    private static XMLSpyControlPlaceHolder   xmlSpyProjectToolWindow = null;
15    private static XMLSpyControlPlaceHolder   xmlSpyXpathToolWindow = null;
16    private static XMLSpyControlPlaceHolder   xmlSpyEHAttributeToolWindow = null;
17
18  // Create the XMLSpy ActiveX control; the parameter determines that we want
19  // to place document controls and place-holder controls individually.
20  // It gives us full control over the menu, as well.
21     xmlSpyControl = new XMLSpyControl(
       ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue() );
22     xmlSpyDocument = new XMLSpyControlDocument();
23     xmlSpyDocument.setPreferredSize( new Dimension ( 640, 480 ) );
24
25  // Create a project window and open the sample project in it
26     xmlSpyProjectToolWindow = new XMLSpyControlPlaceHolder(
```


© 2016-2022 Altova GmbH

```
27      XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWindowToolWnd.getValue() );
28      xmlSpyProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );
29      xmlSpyXpathToolWindow = new XMLSpyControlPlaceHolder(
        XMLSpyControlPlaceholderWindow.XMLSpyControlXPathDialogToolWnd.getValue() );
30      xmlSpyEHAttributeToolWindow = new XMLSpyControlPlaceHolder(
        XMLSpyControlPlaceholderWindow.XMLSpyControlEntryHelperTopToolWnd.getValue() );
31
32      frame.add( xmlSpyControl, BorderLayout.NORTH );
33      frame.add( xmlSpyDocument, BorderLayout.CENTER );
34      southPanel.add( xmlSpyProjectToolWindow );
35      southPanel.add( xmlSpyXpathToolWindow );
36      southPanel.add( xmlSpyEHAttributeToolWindow );
```

29.4.5.3.3 Loading Data in the Controls

The code listing below show how data can be loaded in the ActiveX controls.

```
1     // Locate samples installed with the product.
2        final String strExamplesFolder = System.getenv( "USERPROFILE" ) +
         "\\Documents\\Altova\\XMLSpy2022\\Examples\\";
3        xmlSpyProjectToolWindow.openProject( strExamplesFolder  + "Examples.spp" );
```

29.4.5.3.4 Basic Event Handling

The code listing below shows how basic events can be handled. When calling the XMLSpyControl's `open` method, or when trying to open a file via the menu or Project tree, the `onOpenedOrFocused` event is sent to the attached event handler. The basic handling for this event is opening the file by calling the XMLSpyDocumentControl's `open` method.

```
01        // Open the PXF file when button is pressed
02        btnOpenPxf.addActionListener( new ActionListener() {
03          public void actionPerformed(ActionEvent e) {
04            try {
05              xmlSpyControl.open( strExamplesFolder + "OrgChart.pxf" );
06            } catch (AutomationException e1) {
07              e1.printStackTrace();
08            }
09          }
10        } );
11        public void onOpenedOrFocused( String i_strFileName, boolean
i_bOpenWithThisControl, boolean i_bFileAlreadyOpened ) throws AutomationException
12    {
13      // Handle the New/Open events coming from the Project tree or from the menus
14      if ( !i_bFileAlreadyOpened )
15      {
16        // This is basically an SDI interface, so open the file in the already existing
document control
17        try {
18          XMLSpyContainer.xmlSpyDocument.open( i_strFileName );
19          XMLSpyContainer.xmlSpyDocument.requestFocusInWindow();
20        } catch (Exception e) {
21          e.printStackTrace();
22        }
```

```
23      }
24    }
```

29.4.5.3.5 Menus

The code listing below shows how menu items can be created. Each `XMLSpyCommand` object gets a corresponding `MenuItem` object, with the `ActionCommand` set to the ID of the command. The actions generated by all menu items are handled by the same function, which can perform specific handlings (like reinterpreting the closing mechanism) or can delegate the execution to the `XMLSpyControl` object by calling its `exec` method. The `menuMap` object that is filled during menu creation is used later (see section UI Update Event Handling).

```
01  // Load the file menu when the button is pressed
02        btnMenu.addActionListener( new ActionListener() {
03          public void actionPerformed(ActionEvent e) {
04            try {
05              // Create the menubar that will be attached to the frame
06              MenuBar mb = new MenuBar();
07              // Load the main menu's first item - the File menu
08              XMLSpyCommand xmlSpyMenu =
xmlSpyControl.getMainMenu().getSubCommands().getItem( 0 );
09              // Create Java menu items from the Commands objects
10              Menu fileMenu = new Menu();
11              handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
12              fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
13              mb.add( fileMenu );
14              frame.setMenuBar( mb );
15              frame.validate();
16            } catch (AutomationException e1) {
17              e1.printStackTrace();
18            }
19            // Disable the button when the action has been performed
20            ((AbstractButton) e.getSource()).setEnabled( false );
21          }
22        } ) ;
23 /** * Populates a menu with the commands and submenus contained in an XMLSpyCommands
object  */
24        public void fillMenu(Menu newMenu, XMLSpyCommands xmlSpyMenu) throws
AutomationException
25    {
26      // For each command/submenu in the xmlSpyMenu
27      for ( int i = 0 ; i < xmlSpyMenu.getCount() ; ++i  )
28      {
29        XMLSpyCommand xmlSpyCommand = xmlSpyMenu.getItem( i );
30        if ( xmlSpyCommand.getIsSeparator() )
31          newMenu.addSeparator();
32        else
33        {
34          XMLSpyCommands subCommands = xmlSpyCommand.getSubCommands();
35          // Is it a command (leaf), or a submenu?
36          if ( subCommands.isNull() || subCommands.getCount() == 0 )
37          {
38            // Command -> add it to the menu, set its ActionCommand to its ID and store it
in the menuMap
39            MenuItem mi = new MenuItem( xmlSpyCommand.getLabel().replace( "&", "" ) );
```

 © 2016-2022 Altova GmbH

```
40            mi.setActionCommand( "" + xmlSpyCommand.getID() );
41            mi.addActionListener( this );
42            newMenu.add( mi );
43            menuMap.put( xmlSpyCommand.getID(), mi );
44          }
45          else
46          {
47            // Submenu -> create submenu and repeat recursively
48            Menu newSubMenu = new Menu();
49            fillMenu( newSubMenu, subCommands );
50            newSubMenu.setLabel( xmlSpyCommand.getLabel().replace( "&", "" ) );
51            newMenu.add( newSubMenu );
52          }
53        }
54      }
55    }
56
57     /**
58     * Action handler for the menu items
59     * Called when the user selects a menu item; the item's action command corresponds to
the command table for XMLSpy
60     */
61    public void actionPerformed( ActionEvent e )
62    {
63      try
64      {
65        int iCmd = Integer.parseInt( e.getActionCommand() );
66        // Handle explicitly the Close commands
67        switch ( iCmd )
68        {
69          case 57602:       // Close
70          case 34050:       // Close All
71            XMLSpyContainer.initXmlSpyDocument();
72            break;
73          default:
74            XMLSpyContainer.xmlSpyControl.exec( iCmd );
75            break;
76        }
77      }
78      catch ( Exception ex )
79      {
80        ex.printStackTrace();
81      }
82
83    }
```

29.4.5.3.6 UI Update Event Handling

The code listing below shows how a UI-Update event handler can be created.

```
01 /**
02     * Call-back from the XMLSpyControl.
03     * Called to enable/disable commands
04     */
05    @Override
06    public void onUpdateCmdUI() throws AutomationException
```

```
07    {
08      // A command should be enabled if the result of queryStatus contains the Supported
(1) and Enabled (2) flags
09      for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )
10
       pair.getValue().setEnabled( XMLSpyContainer.xmlSpyControl.queryStatus( pair.getKey()
) > 2 );
11    }
12     /**
13     * Call-back from the XMLSpyControl.
14     * Usually called while enabling/disabling commands due to UI updates
15     */
16    @Override
17    public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
18    {
19      try {
20        return
XMLSpyContainer.xmlSpyDocument.getDocument().getFullName().equalsIgnoreCase( i_strFilePath
);
21      } catch ( Exception e ) {
22        return false;
23      }
24    }
```

29.4.5.3.7 Creating an XML Tree

The listing below loads an XML data object as nodes in a tree.

```
01 // access required XMLSpy Java-COM classes
02 import com.altova.automation.XMLSpy.XMLData;
03
04 // access AWT and Swing components
05 import java.awt.*;
06 import javax.swing.*;
07 import javax.swing.tree.*;
08
09 /**
10  * A simple example of a tree control loading the structure from an XMLData object.
11  * The class receives an XMLData object, loads its nodes in a JTree, and prepares
12  * for modal activation.
13  *
14  * Feel free to modify and extend this sample.
15  *
16  * @author Altova GmbH
17  */
18 class XMLTreeDialog extends JDialog
19 {
20   /**
21    * The tree control
22    */
23   private JTree myTree;
24
25   /**
26    * Root node of the tree control
27    */
```


© 2016-2022 Altova GmbH

```
28   private DefaultMutableTreeNode top ;
29
30   /**
31    * Constructor that prepares the modal dialog containing the filled tree control
32    * @param xml   The data to be displayed in the tree
33    * @param parent  Parent frame
34    */
35   public XMLTreeDialog( XMLData xml, Frame parent )
36   {
37     // Construct the modal dialog
38     super( parent, "XML tree", true );
39     // Arrange controls in the dialog
40     top = new DefaultMutableTreeNode("root");
41     myTree = new JTree(top);
42     setContentPane( new JScrollPane( myTree ) );
43     // Build up the tree
44     fillTree( top, xml );
45     myTree.expandRow( 0 );
46   }
47
48   /**
49    * Loads the nodes of an XML element under a given tree node
50    * @param node  Target tree node
51    * @param elem  Source XML element
52    */
53   private void fillTree( DefaultMutableTreeNode node, XMLData elem)
54   {
55     try
56     {
57       // There are several ways to iterate through child elements: either using the
getFirstChild/getNextChild,
58       // or by incrementing an index up to countChildren and calling getChild [as shown
below].
59       // If you only want to get children of one kind, you should use
countChildrenKind/getChildKind,
60       // or provide a kind to the getFirstChild before iterating with the getNextChild.
61       int nSize = elem.countChildren() ;
62       for ( int i = 0 ; i < nSize ; ++i)
63       {
64         // Create a new tree node for each child element, and continue recursively
65         XMLData newElem = elem.getChild(i) ;
66         DefaultMutableTreeNode newNode = new DefaultMutableTreeNode( newElem.getName() )
;
67         node.add( newNode ) ;
68         fillTree( newNode, newElem ) ;
69       }
70     }
71     catch (Exception e)
72     {
73       e.printStackTrace();
74     }
75   }
76
77 }
```

29.4.6 Command Reference

This section lists the names and identifiers of all menu commands that are available within XMLSpy. Every sub-section lists the commands from the corresponding top-level menu of XMLSpy. The command tables are organized as follows:

- The "Menu Item" column shows the command's menu text as it appears in XMLSpy, to make it easier for you to identify the functionality behind the command.
- The "Command Name" column specifies the string that can be used to get an icon with the same name from **ActiveX\Images** folder of the XMLSpy installation directory.
- The "ID" column shows the numeric identifier of the column that must be supplied as argument to methods which execute or query this command.

To execute a command, use the `XMLSpyControl.Exec` or the `XMLSpyControlDocument.Exec` methods. To query the status of a command, use the `XMLSpyControl.QueryStatus` or `XMLSpyControlDocument.QueryStatus` methods.

Depending on the edition of XMLSpy you have installed, some of these commands might not be supported.

29.4.6.1 "File" Menu

The "File" menu has the following commands:

Menu item	Command name	ID
New...	ID_FILE_NEW	57600
Open...	ID_FILE_OPEN	57601
Reload	IDC_FILE_RELOAD	34065
Encoding...	IDC_ENCODING	34061
Close	ID_FILE_CLOSE	57602
Close All	IDC_CLOSE_ALL	34050
Close All But Active	IDC_CLOSE_OTHERS	34271
Save	ID_FILE_SAVE	57603
Save As...	ID_FILE_SAVE_AS	57604
Save All	ID_FILE_SAVE_ALL	34208
Send by Mail...	ID_FILE_SEND_MAIL	57612
Print...	ID_FILE_PRINT	57607
Print Preview	IDC_PRINT_PREVIEW	34104
Print Setup...	ID_FILE_PRINT_SETUP	57606

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Recent File	ID_FILE_MRU_FILE1	57616
Exit	ID_APP_EXIT	57665

29.4.6.2 "Edit" Menu

The "Edit" menu has the following commands:

Menu item	Command name	ID
Undo	ID_EDIT_UNDO	57643
Redo	ID_EDIT_REDO	57644
Cut	ID_EDIT_CUT	57635
Copy	ID_EDIT_COPY	57634
Paste	ID_EDIT_PASTE	57637
Delete	ID_EDIT_CLEAR	57632
Copy as XML Text	IDC_COPY_AS_XML_TEXT	33443
Copy as Structured Text	IDC_COPY_AS_STRUCTURED_TEXT	33442
Copy XPath	IDC_COPY_XPATH	33444
Copy XPointer	IDC_COPY_XPOINTER	33445
File Path...	IDC_EDIT_INSERT_PATH_STRING	34013
XInclude...	IDC_EDIT_INSERT_XINCLUDE_STRING	34017
Encoded External File...	IDC_EDIT_INSERT_ENCODED_BINARY_STRING	34273
Pretty-Print	IDC_PRETTY_PRINT	34101
Strip Whitespaces	IDC_STRIP_WHITESPACES	34296
Select All	ID_EDIT_SELECT_ALL	57642
Find...	ID_EDIT_FIND	57636
Find Next	ID_EDIT_REPEAT	57640
Replace...	ID_EDIT_REPLACE	57641
Find in Files...	IDC_FIND_IN_FILES	34000
Insert/Remove Bookmark	IDC_TOGGLE_BOOKMARK	34162

Menu item	Command name	ID
Remove All Bookmarks	IDC_REMOVEALLBOOKMARKS	34132
Go to Next Bookmark	IDC_GOTONEXTBOOKMARK	34070
Go to Previous Bookmark	IDC_GOTOPREVBOOKMARK	34071
Comment In/Out	IDC_TOGGLE_XML_COMMENT	34029

29.4.6.3 "Project" Menu

The "Project" menu has the following commands:

Menu item	Command name	ID
New Project	IDC_ICPROJECTGUI_NEW	37200
Open Project...	IDC_ICPROJECTGUI_OPEN	37201
Reload Project	IDC_ICPROJECTGUI_RELOAD	37202
Close Project	IDC_ICPROJECTGUI_CLOSE	37203
Save Project	IDC_ICPROJECTGUI_SAVE	37204
Save Project As...	IDC_ICPROJECTGUI_SAVE_AS	37207
Enable Source Control	ID_SCC_ENABLE	38602
Add Files to Project...	IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT	37205
Add Global Resource to Project...	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT	37239
Add URL to Project...	IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT	37206
Add Active File to Project	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT	37208
Add Active and Related Files to Project	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT	37209
Add Project Folder to Project...	IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT	37210
Add External Folder to Project...	IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT	37211
Add External Web Folder to Project...	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT	37212

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Script settings...	IDC_PROJECT_SCRIPT_SETTINGS	34136
Properties...	IDC_ICPROJECTGUI_PROJECT_PROPERTIES	37223
Recent Project	IDC_ICPROJECTGUI_RECENT	37224

29.4.6.4 "XML" Menu

The "XML" menu has the following commands:

Menu item	Command name	ID
Attribute	IDC_INSERT_ATTRIBUTE	33449
Element	IDC_INSERT_STRUCT	33459
Text	IDC_INSERT_TEXT	33460
CDATA	IDC_INSERT_CDATA	33450
Comment	IDC_INSERT_COMMENT	33451
XML	IDC_INSERT_XML	33461
Processing Instruction	IDC_INSERT_PI	33458
XInclude...	IDC_INSERT_XINCLUDE	34019
DOCTYPE	IDC_INSERT_DEF_DOCTYPE	33453
ExternalID	IDC_INSERT_DEF_EXTERNAL_ID	33456
ELEMENT	IDC_INSERT_DEF_ELEMENT	33454
ATTLIST	IDC_INSERT_DEF_ATTLIST	33452
ENTITY	IDC_INSERT_DEF_ENTITY	33455
NOTATION	IDC_INSERT_DEF_NOTATION	33457
Encoded External File...	IDC_INSERT_ENCODED_BINARY	34274
Attribute	IDC_APPEND_ATTRIBUTE	33415
Element	IDC_APPEND_STRUCT	33425
Text	IDC_APPEND_TEXT	33426
CDATA	IDC_APPEND_CDATA	33416
Comment	IDC_APPEND_COMMENT	33417

Menu item	Command name	ID
XML	IDC_APPEND_XML	33427
Processing Instruction	IDC_APPEND_PI	33424
XInclude...	IDC_APPEND_XINCLUDE	34026
DOCTYPE	IDC_APPEND_DEF_DOCTYPE	33419
ExternalID	IDC_APPEND_DEF_EXTERNAL_ID	33422
ELEMENT	IDC_APPEND_DEF_ELEMENT	33420
ATTLIST	IDC_APPEND_DEF_ATTLIST	33418
ENTITY	IDC_APPEND_DEF_ENTITY	33421
NOTATION	IDC_APPEND_DEF_NOTATION	33423
Encoded External File...	IDC_APPEND_ENCODED_BINARY	34276
Attribute	IDC_ADD_CHILD_ATTRIBUTE	33402
Element	IDC_ADD_CHILD_STRUCT	33412
Text	IDC_ADD_CHILD_TEXT	33413
CDATA	IDC_ADD_CHILD_CDATA	33403
Comment	IDC_ADD_CHILD_COMMENT	33404
XML	IDC_ADD_CHILD_XML	33414
Processing Instruction	IDC_ADD_CHILD_PI	33411
XInclude...	IDC_ADD_CHILD_XINCLUDE	34027
DOCTYPE	IDC_ADD_CHILD_DEF_DOCTYPE	33406
ExternalID	IDC_ADD_CHILD_DEF_EXTERNAL_ID	33409
ELEMENT	IDC_ADD_CHILD_DEF_ELEMENT	33407
ATTLIST	IDC_ADD_CHILD_DEF_ATTLIST	33405
ENTITY	IDC_ADD_CHILD_DEF_ENTITY	33408
NOTATION	IDC_ADD_CHILD_DEF_NOTATION	33410
Encoded External File...	IDC_ADD_CHILD_ENCODED_BINARY	34277
Attribute	IDC_CONVERT_TO_ATTRIBUTE	33429
Element	IDC_CONVERT_TO_STRUCT	33439
Text	IDC_CONVERT_TO_TEXT	33440

© 2016-2022 Altova GmbH

Menu item	Command name	ID
CDATA	IDC_CONVERT_TO_CDATA	33430
Comment	IDC_CONVERT_TO_COMMENT	33431
XML	IDC_CONVERT_TO_XML	33441
Processing Instruction	IDC_CONVERT_TO_PI	33438
DOCTYPE	IDC_CONVERT_TO_DEF_DOCTYPE	33433
ExternalID	IDC_CONVERT_TO_DEF_EXTERNAL_ID	33436
ELEMENT	IDC_CONVERT_TO_DEF_ELEMENT	33434
ATTLIST	IDC_CONVERT_TO_DEF_ATTLIST	33432
ENTITY	IDC_CONVERT_TO_DEF_ENTITY	33435
NOTATION	IDC_CONVERT_TO_DEF_NOTATION	33437
Display as Table	IDC_GRID_VIEW_AS_TABLE	34075
Insert Row	IDC_TABLE_INSERT_ROW	34158
Append Row	IDC_TABLE_APPEND_ROW	34157
Ascending Sort	IDC_TABLE_SORT_ASC	33464
Descending Sort	IDC_TABLE_SORT_DESC	33465
Move Left	IDC_MOVE_LEFT	34091
Move Right	IDC_MOVE_RIGHT	34092
Enclose in Element	IDC_ENCLOSE_IN_ELEMENT	33446
Evaluate XPath...	IDC_EVALUATE_XPATH	34007
Check Well-Formedness	IDC_CHECK_WELL_FORM	34049
Validate XML	IDC_VALIDATE	32954
Validate XML on Server (high-performance)	IDC_VALIDATE_RAPTOR	34309
Update Entry Helpers	IDC_UPDATE_ELEMENT_CHOICE	34173
Namespace Prefix...	IDC_NAMESPACE	33462
Create XML Signature...	IDC_XML_SIGNATURE_CREATE	34280
Verify XML Signature...	IDC_XML_SIGNATURE_VERIFY	34281

29.4.6.5 "DTD/Schema" Menu

The "DTD/Schema" menu has the following commands:

Menu item	Command name	ID
Assign DTD...	IDC_ASSIGN_DTD	34032
Assign Schema...	IDC_ASSIGN_SCHEMA	34033
Include Another DTD...	IDC_INCLUDE_DTD	34084
Go to DTD	IDC_GOTO_DTD	34072
Go to Schema	IDC_GOTO_SCHEMA	34074
Go to Definition	IDC_GOTO_DEFINITION	33447
Generate DTD/Schema...	IDC_GENERATE_DTD_SCHEMA	34068
Flatten DTD...	IDC_FLATTEN_DTD	34301
Convert DTD To Schema...	IDC_CONVERT_DTD_TO_SCHEMA	34299
Flatten Schema...	IDC_FLATTEN_SCHEMA	34302
Convert Schema To DTD...	IDC_CONVERT_SCHEMA_TO_DTD	34300
Convert to UML...	IDC_CONVERT_SCHEMA_TO_UML	34008
Generate XML from DB, Excel, EDI with MapForce...	IDC_DTD_OPENIN_MAPFORCE	34056
Design HTML/PDF/Word Output with StyleVision...	IDC_DTD_OPENIN_STYLEVISION	34057
Generate Sample XML/JSON File...	IDC_GENERATE_XML_FROM_SCHEMA	34069
Generate Program Code...	IDC_GENERATE_CODE_FROM_SCHEMA	34067
Flush Memory Cache	IDC_FLUSH_CACHED_FILES	34066

29.4.6.6 "Schema design" Menu

The "Schema design" menu has the following commands:

Menu item	Command name	ID
Schema Settings...	IDC_SCHEMA_NAMESPACES	33571
Save Diagram...	IDC_SCHEMA_SAVE_DIAGRAM	33581

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Generate Documentation...	IDC_SCHEMA_DOCUMENTATION	34146
Configure View...	IDC_SCHEMA_VIEW_CONFIG	33593
Zoom...	IDC_SCHEMA_ZOOM	34150
Display All Globals	IDC_SCHEMA_MODE_GLOBALS	34147
Display Diagram	IDC_SCHEMA_MODE_DIAGRAM	33570
Enable Oracle Schema Extensions	IDC_SCHEMA_ORACLE_EXTENSIONS	33577
Oracle Schema Settings...	IDC_SCHEMA_ORACLE_SCHEMA_SETTINGS	33578
Enable Microsoft SQL Server Schema Extensions	IDC_SCHEMA_SQLSERVER_EXTENSIONS	33588
Named Schema Relationships...	IDC_SCHEMA_SQLSERVER_GLOBAL_RELATIONSHIPS	33589
Unnamed Element Relationships...	IDC_SCHEMA_SQLSERVER_LOCAL_RELATIONSHIPS	33590
Connect to SchemaAgent Server...	IDC_SCHEMA_SCHEMAAGENT_SERVER_CONNECT	33582
Disconnect from SchemaAgent Server	IDC_SCHEMA_SCHEMAAGENT_SERVER_DISCONNECT	33583
File Only	IDC_SCHEMAAGENT_SHOW_FILE_ONLY	33504
File and All Directly Referenced Schema Files	IDC_SCHEMAAGENT_SHOW_WITH_DIRECTLY_REFERENCED_SCHEMAS	33608
File and All Directly Referencing Schema Files	IDC_SCHEMAAGENT_SHOW_WITH_DIRECTLY_REFERENCING_SCHEMAS	33602
File and All Directly Related Schema Files	IDC_SCHEMAAGENT_SHOW_WITH_DIRECTLY_RELATED_SCHEMAS	33613
SchemaAgent Validation...	IDC_SCHEMA_EXTVALID_MENU	33539
Create Schema Subset...	IDC_SCHEMA_CREATE_SUBSET	33650
Flatten Schema...	IDC_SCHEMA_FLATTEN	33651

29.4.6.7 "XSL/XQuery" Menu

The "XSL/XQuery" menu has the following commands:

Menu item	Command name	ID
XSL Transformation	IDC_TRANSFORM_XSL	33006
XSL Speed Optimizer	IDC_TRANSFORM_XSLPBO	34306
XSL-FO Transformation	IDC_TRANSFORM_XSLFO	33007
XSL Parameters / XQuery Variables...	IDC_TRANSFORM_XSL_PARAMS	33008
XQuery/Update Execution	IDC_TRANSFORM_XQUERY	34170
Enable Back Mapping	IDC_ENABLE_BACKMAPPING	34364
Enable XSLT/ XQuery Profiling....	IDC_PROFILING_OPTIONS	34105
Assign XSL...	IDC_ASSIGN_XSL	33001
Assign XSL-FO...	IDC_ASSIGN_XSLFO	33002
Assign Sample XML File...	IDC_ASSIGN_SAMPLE_XML	33000
Go to XSL	IDC_GOTO_XSL	33004
Start Debugger / Go	ID_PROCESS_XSL	34212
Stop Debugger	ID_XSLT_DEBUGGER_STOP	33017
Restart Debugger	ID_XSLT_DEBUGGER_RESTART	33013
End Debugger Session	ID_XSLT_DEBUGGER_END_SESSION	33011
Step Into	ID_XSLT_DEBUGGER_STEP	33014
Step Out	ID_XSLT_DEBUGGER_STEP_OUT	33015
Step Over	ID_XSLT_DEBUGGER_STEP_OVER	33016
Show Current Execution Node	ID_XSLT_DEBUGGER_GO_TO_CURRENT_EXECUTION_NODES	33012
Insert/Remove Breakpoint	IDC_TOGGLE_BREAKPOINT	34246
Insert/Remove Tracepoint	IDC_TOGGLE_TRACEPOINT	34248
Enable/Disable Breakpoint	IDC_ENABLE_BREAKPOINT	34245
Enable/Disable Tracepoint	IDC_ENABLE_TRACEPOINT	34247
Breakpoints/Tracepoints...	ID_XSLTDEBUGGER_BREAKPOINTS	33009
Call Stack	ID_XSL_DEBUGWINDOWS_CALLSTACK	34238
XPath-Watch	ID_XSL_DEBUGWINDOWS_WATCH	34244
Context	ID_XSL_DEBUGWINDOWS_CONTEXT	34239

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Variables	ID_XSL_DEBUGWINDOWS_VARIABLE	34243
Messages	ID_XSL_DEBUGWINDOWS_MESSAGES	34240
Templates	ID_XSL_DEBUGWINDOWS_TEMPLATES	34241
Info	ID_XSLXQUERY_DEBUGWINDOWS_INFO	34237
Trace	ID_XSL_DEBUGWINDOWS_TRACES	34242
Debug Settings...	ID_XSLTDEBUGGER_SETTINGS	33010

29.4.6.8 "Authentic" Menu

The "Authentic" menu has the following commands:

Menu item	Command name	ID
New Document...	IDC_AUTHENTIC_NEW_FILE	34036
Edit Database Data...	IDC_AUTHENTIC_EDIT_DB	34035
Assign a StyleVision Stylesheet...	IDC_ASSIGN_SPS	34034
Edit StyleVision Stylesheet	IDC_EDIT_SPS	34060
Select New Row with XML Data for Editing...	IDC_CHANGE_WORKING_DB_XML_CELL	32861
XML Signature...	IDC_AUTHENTICGUI_XMLSIGNATURE	32862
Define XML Entities...	IDC_DEFINE_ENTITIES	32805
Hide Markup	IDC_MARKUP_HIDE	32855
Show Small Markup	IDC_MARKUP_SMALL	32858
Show Large Markup	IDC_MARKUP_LARGE	32856
Show Mixed Markup	IDC_MARKUP_MIXED	32857
Toggle Bold	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEBOLD	32813
Toggle Italic	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEITALIC	32814
Toggle Underline	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEUNDERLINE	32815
Toggle Strikethrough	IDC_AUTHENTICGUI_RICHEDIT_TOGGLESTRIKETHROUGH	32816

Menu item	Command name	ID
Foreground Color	IDC_AUTHENTICGUI_RICHEDIT_COLOR_FOREGROUND	32824
Background Color	IDC_AUTHENTICGUI_RICHEDIT_COLOR_BACKGROUND	32830
Align Left	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_LEFT	32818
Center	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_CENTER	32819
Align Right	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_RIGHT	32820
Append Row	IDC_ROW_APPEND	32806
Insert Row	IDC_ROW_INSERT	32809
Duplicate Row	IDC_ROW_DUPLICATE	32808
Move Row Up	IDC_ROW_MOVE_UP	32811
Move Row Down	IDC_ROW_MOVE_DOWN	32810
Delete Row	IDC_ROW_DELETE	32807
Generate an HTML document	IDC_PXF_GENERATE_HTML	34283
Generate an RTF document	IDC_PXF_GENERATE_RTF	34284
Generate a PDF document	IDC_PXF_GENERATE_PDF	34285
Generate a Word 2007+ document	IDC_PXF_GENERATE_DOCX	34286
Trusted Locations...	IDC_TRUSTED_LOCATIONS	34288

29.4.6.9 "DB" Menu

The "DB" menu has the following commands:

Menu item	Command name	ID
Query Database	IDC_QUERYDATABASE	34012
Manage XML Schemas...	IDC_DB_MANAGESCHEMAS	34014
Assign XML Schema...	IDC_DB_CHOOSEVALIDATIONSCHEMA	34016
Manage XML Schemas...	IDC_DB_MANAGESCHEMAS	34014
Manage XML Schemas...	IDC_DB_MANAGESCHEMAS	34014
Browse Oracle XML Documents...	ID_CONVERT_ORACLEXMLDB_BROWSE	34205

© 2016-2022 Altova GmbH

29.4.6.10 "Convert" Menu

The "Convert" menu has the following commands:

Menu item	Command name	ID
Import Text File...	IDC_IMPORT_TEXT	34082
Import Database Data...	IDC_IMPORT_DATABASE	34080
Import Microsoft Word Document...	IDC_IMPORT_WORD	34083
Create XML Schema from DB Structure	IDC_CREATE_DB_SCHEMA	34054
DB Import Based on XML Schema	IDC_IMPORT_DB_SCHEMA	34081
Create DB Structure from XML Schema	IDC_CREATE_DB_BASED_ON_SCHEMA	34053
Export to Text Files...	IDC_EXPORT_TEXTFILE	34064
Export to a Database...	IDC_EXPORT_DB	34003
Convert XML Instance to/from JSON...	IDC_JSON_CONVERT_TOFROM_XML	34135
Convert XML Schema to/from JSON Schema...	IDC_JSON_CONVERT_TOFROM_XSD	34350

29.4.6.11 "View" Menu

The "View" menu has the following commands:

Menu item	Command name	ID
Text View	IDC_VIEW_TEXT	34180
Enhanced Grid View	IDC_VIEW_GRID	34178
Schema Design View	IDC_VIEW_SCHEMA	34179
WSDL Design View	IDC_VIEW_WSDL	34117
XBRL Taxonomy View	IDC_VIEW_XBRL	34118
Authentic View	IDC_VIEW_CONTENT	34177
Browser View	IDC_VIEW_BROWSER	34176
Expand +	IDC_SEL_EXPAND	34152
Collapse -	IDC_SEL_COLLAPSE	34151
Expand Fully	IDC_SEL_EXPAND_ALL	33463

Menu item	Command name	ID
Collapse Unselected	IDC_COLLAPSE_UNSELECTED	33428
Optimal Widths	IDC_OPTIMAL_WIDTHS	34099
Word Wrap	IDC_WORD_WRAP	34181
Go to Line/Character	IDC_GOTO_LINE	34073
Go to File	IDC_GOTO_FILE	33448
Text View Settings	IDC_TEXTVIEW_SETTINGS	34119

29.4.6.12 "Browser" Menu

The "Browser" menu has the following commands:

Menu item	Command name	ID
Back	IDC_STEP_BACK	32958
Forward	IDC_STEP_FORWARD	32957
Stop	IDC_BROWSER_STOP	34047
Refresh	IDC_BROWSER_REFRESH	34046
Largest	IDC_BROWSER_FONT_LARGEST	34041
Larger	IDC_BROWSER_FONT_LARGE	34040
Medium	IDC_BROWSER_FONT_MEDIUM	34042
Smaller	IDC_BROWSER_FONT_SMALL	34043
Smallest	IDC_BROWSER_FONT_SMALLEST	34044

29.4.6.13 "WSDL" Menu

The "WSDL" menu has the following commands:

Menu item	Command name	ID
Insert Message	ID_WSDL_MESSAGES_ADDNEWMESSAGE	33715
Delete Message	ID_WSDL_MESSAGES_DELETESELECTEDMESSAGE	33717
Add Message Part (Parameter)	ID_WSDL_MESSAGES_ADDMESSAGEPART	33714

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Delete Message Part (Parameter)	ID_WSDL_MESSAGES_DELETEMESSAGEPART	33716
request-response	IDC_WSDL_OPERATION_APPENDREQUESTRESPONSE	33734
solicit-response	IDC_WSDL_OPERATION_APPENDSOLICITRESPONSE	33737
one-way	IDC_WSDL_OPERATION_APPENDONEWAY	33735
notification	IDC_WSDL_OPERATION_APPENDNOTIFICATION	33736
Empty Operation	ID_WSDL_OPERATIONS_APPENDAOPERATIONTOTHISPORTTYPE	33722
Delete Operation	ID_WSDL_OPERATIONS_DELETEOPERATION	33724
Add Input Element	ID_WSDL_OPERATIONS_ADDINPUTFUNCTION	33719
Add Output Element	ID_WSDL_OPERATIONS_ADDOUTPUTFUNCTION	33721
Add Fault Element	ID_WSDL_OPERATIONS_ADDFAULTFUNCTION	33718
Delete Input/Output/Fault Element	ID_WSDL_OPERATIONS_DELETEINPUTOUTPUTFUNCTION	33723
Add New Message to Input/Output/Fault Element	ID_WSDL_OPERATIONS_ADDNEWMESSAGETOTHISELEMENT	33720
Insert Port Type	ID_WSDL_PORTTYPE_INSERTAPORTTYPE	33727
Delete Port Type	ID_WSDL_PORTTYPE_DELETETHISPORTTYPE	33726
Insert Binding	ID_WSDL_BINDING_NEWBINDING	33713
Delete Binding	ID_WSDL_BINDING_DELETEBINDING	33711
soap:body	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPBODY	33706
soap:header	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPHEADER	33708
soap:headerfault	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPHEADERFAULT	33709

Menu item	Command name	ID
soap:fault	ID_WSDL_BINDING_APPENDEXTENSIBILITY_SOAPFAULT	33707
mime:content	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMECONTENT	33702
mime:multipartrelated	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMEMULTIPARTRELATED	33704
mime:part	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMEPART	33705
mime:mimeXml	ID_WSDL_BINDING_APPENDEXTENSIBILITY_MIMEMIMEXML	33703
http:urlencoded	ID_WSDL_BINDING_APPENDEXTENSIBILITY_HTTPURLENCODED	33700
http:urlreplacement	ID_WSDL_BINDING_APPENDEXTENSIBILITY_HTTPURLREPLACEMENT	33701
Delete Extensibility Element	ID_WSDL_BINDING_DELETEEXTENSIBILITY	33712
Insert Service	ID_WSDL_SERVICE_INSERTSERVICE	33731
Delete Service	ID_WSDL_SERVICE_DELETETHISSERVICE	33729
Insert Port	ID_WSDL_SERVICE_INSERTNEWPORT	33730
Delete Port	ID_WSDL_SERVICE_DELETETHISPORT	33728
Add New Interface	IDC_WSDL20_ADDINTERFACE	33794
Delete Interface	IDC_WSDL20_DELETEINTERFACE	33795
Add New Fault	IDC_WSDL20_ADDINTERFACEFAULT	33796
Delete Fault	IDC_WSDL20_DELETEINTERFACEFAULT	33808
In-only	IDC_WSDL20_ADDINTERFACEOPERATION_INONLY	33797
Robust-in-only	IDC_WSDL20_ADDINTERFACEOPERATION_ROBUSTINONLY	33798
In-out	IDC_WSDL20_ADDINTERFACEOPERATION_INOUT	33801
In-opt-out	IDC_WSDL20_ADDINTERFACEOPERATION_INOPTOUT	33802
Out-in	IDC_WSDL20_ADDINTERFACEOPERATION_OUTIN	33803

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Out-opt-in	IDC_WSDL20_ADDINTERFACEOPERATION_OUTOPTIN	33804
Out-only	IDC_WSDL20_ADDINTERFACEOPERATION_OUTONLY	33800
Robust-out-only	IDC_WSDL20_ADDINTERFACEOPERATION_ROBUSTOUTONLY	33799
Empty Operation	IDC_WSDL20_ADDINTERFACEOPERATION_EMPTY	33805
Delete Operation	IDC_WSDL20_DELETEINTERFACEOPERATION	33809
Add New Binding	IDC_WSDL20_ADDBINDING	33820
Delete Binding	IDC_WSDL20_DELETEBINDING	33821
Add New Fault	IDC_WSDL20_ADDBINDINGFAULT	33822
Delete Fault	IDC_WSDL20_DELETEBINDINGFAULT	33826
Add New Operation	IDC_WSDL20_ADDBINDINGOPERATION	33823
Delete Operation	IDC_WSDL20_DELETEBINDINGOPERATION	33827
Add New Service	IDC_WSDL20_ADDSERVICE	33839
Delete Service	IDC_WSDL20_DELETESERVICE	33840
Add New Endpoint	IDC_WSDL20_ADDENDPOINT	33841
Delete Endpoint	IDC_WSDL20_DELETEENDPOINT	33842
New Schema	ID_WSDL_TYPES_NEWSCHEMA	33733
Embed Schema	ID_WSDL_TYPES_EMBEDSCHEMA	39456
Extract Schema(s)	ID_WSDL_TYPES_EXTRACTSCHEMAS	39459
Edit Schema(s) in Schema View	ID_WSDL_TYPES_EDITTHISSCHEMA	33732
Save Diagram...	IDC_WSDL_SAVE_DIAGRAM	39451
Generate Documentation...	ID_WSDL_GENERATEDOCUMENTATION	39452
Reparse WSDL Document	IDC_WSDL_REPARSE	33774
Convert to WSDL 2.0	IDC_WSDL_CONVERT_TO_WSDL20	39453
Generate WSDL Program Code with MapForce...	IDC_WSDL_GENERATE_CODE_MAPFORCE	34122

29.4.6.14 "SOAP" Menu

The "SOAP" menu has the following commands:

Menu item	Command name	ID
Create New SOAP Request...	ID_SOAP_GENERATESOAPMESSAGE	34224
Send Request to Server...	ID_SOAP_SENDREQUESTTOSERVER	34225
SOAP Request Settings...	ID_SOAP_SOAPREQUESTSETTINGS	34227
Soap Debugger Session	ID_SOAP_SOAPDEBUGGER	34226
Go	ID_SOAPDEBUGGER_BUTTONPLAY	34221
Single Step	ID_SOAPDEBUGGER_SINGLESTEP	34222
Break on Next Request	ID_SOAPDEBUGGER_BREAKONNEXTREQUEST	34219
Break on Next Response	ID_SOAPDEBUGGER_BREAKONNEXTRESPONSE	34220
Stop the Proxy Server	ID_SOAPDEBUGGER_STOPSERVER	34223
Soap Debugger Options	ID_SOAPDEBUGGEROPTIONS	34218

29.4.6.15 "XBRL" Menu

The "XBRL" menu has the following commands:

Menu item	Command name	ID
Arcroles...	IDC_XMLSPYXBRLEDITOR_ARCROLES	34114
Linkroles...	IDC_XMLSPYXBRLEDITOR_LINKROLES	34115
Namespace Prefixes...	IDC_XMLSPYXBRLEDITOR_NAMESPACES	34116
Set Target Namespace...	IDC_XMLSPYXBRLEDITOR_SET_TARGETNAMESPACE	34039
Parameter Values...	IDC_ICXBRLEDITOR_PARAMETER_VALUES	38913
Import/Reference...	IDC_XMLSPYXBRLEDITOR_IMPORT_REFERENCE	34137
Find Component By Id...	IDC_ICXBRLEDITOR_FIND_COMPONENT_BY_ID	38893

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Generate Documentation...	IDC_XMLSPYXBRLEDITOR_GENERATEDOCUMENTATION	34125
View Settings...	IDC_XMLSPYXBRLEDITOR_VIEWSETTINGS	34113
Generate XBRL from DB, Excel, CSV with MapForce...	IDC_XBRL_GENERATE_WITH_MAPFORCE	34045
Present XBRL as HTML/PDF/Word with StyleVision...	IDC_XBRL_PRESENT_WITH_STYLEVISION	34121
Execute Formula...	IDC_XBRL_EXECUTE_FORMULA	34305
Execute Formula on Server (high-performance)...	IDC_XBRL_EXECUTE_FORMULA_RAPTOR	34352
Generate Table...	IDC_XBRL_GENERATE_TABLE	34304
Generate Table on Server (high-performance)...	IDC_XBRL_GENERATE_TABLE_RAPTOR	34353
Transform Inline XBRL	IDC_IXBRL_TRANSFORM	34354

29.4.6.16 "Tools" Menu

The "Tools" menu has the following commands:

Menu item	Command name	ID
Spelling...	IDC_SPELL_CHECK	34154
Spelling Options...	IDC_SPELL_OPTIONS	34155
Scripting Editor...	ID_SCRIPTFORMEDITOR_EDIT_PROJECT	39666
none	ID_SCRIPTFORMEDITOR_EXECUTE_MACRO_MENU_UPPDATE	39600
Compare Open File With...	ID_XMLDIFF_CHOOSE_FILES	34235
Compare Directories...	ID_XMLDIFF_CHOOSE_DIRECTORIES	34234
Compare Options...	ID_XMLDIFF_SETTINGS	34236
	IDC_TOOLS_ENTRY	34292
Global Resources	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENUENTRY1	37408
Manage Raptor Servers ...	IDC_VALIDATE_RAPTOR_MANAGER	34311

Menu item	Command name	ID
none	IDC_VALIDATE_RAPTOR_NOCFG	34326
Customize...	IDC_APP_TOOLS_CUSTOMIZE	32959
Options...	IDC_SETTINGS	34133
	ID_SCRIPTING_MACROITEMS	34249

29.4.6.17 "Window" Menu

The "Window" menu has the following commands:

Menu item	Command name	ID
Cascade	ID_WINDOW_CASCADE	57650
Tile horizontally	ID_WINDOW_TILE_HORZ	57651
Tile vertically	ID_WINDOW_TILE_VERT	57652
Project window	IDC_PROJECT_WINDOW	34128
Info window	IDC_INFO_WINDOW	34085
Entry Helpers	IDC_ENTRY_HELPERS	34062
Output windows	IDC_OUTPUT_DIALOGBARS	34004
Project and Entry Helpers	IDC_PROJECT_ENTRYHELPERS	34006
All on/off	IDC_ALL_BARS	34031

29.4.6.18 "Help" Menu

The "Help" menu has the following commands:

Menu item	Command name	ID
Table of Contents...	IDC_HELP_CONTENTS	32966
Index...	IDC_HELP_INDEX	32967
Search...	IDC_HELP_SEARCH	32969
Keyboard Map...	IDC_HELP_KEYMAPDLG	32968
Software Activation...	IDC_ACTIVATION	32970
Order Form...	IDC_OPEN_ORDER_PAGE	32971

© 2016-2022 Altova GmbH

Menu item	Command name	ID
Registration...	IDC_REGISTRATION	32972
Check for Updates...	IDC_CHECK_FOR_UPDATES	32973
XMLSpy Product Comparison...	IDC_PRODUCT_COMPARISON	32955
Support Center...	IDC_OPEN_SUPPORT_PAGE	32961
FAQ on the Web...	IDC_OPEN_FAQ_PAGE	32962
Download Components and Free Tools...	IDC_OPEN_COMPONENTS_PAGE	32963
Authentic on the Internet..	IDC_OPEN_HOME_PAGE	32964
Authentic Training...	IDC_OPEN_TRAINING_PAGE	32965
About XMLSpy...	ID_APP_ABOUT	57664

29.4.7 Object Reference

Objects:
XMLSpyCommand
XMLSpyCommands
XMLSpyControl
XMLSpyControlDocument
XMLSpyControlPlaceHolder

To give access to standard XMLSpy functionality, objects of the **XMLSpy automation interface** can be accessed as well. See XMLSpyControl.Application, XMLSpyControlDocument.Document and XMLSpyControlPlaceHolder.Project for more information.

29.4.7.1 XMLSpyCommand

Properties:
`ID`
`Label`
`Name`
`IsSeparator`
`ToolTip`
`StatusText`
`Accelerator`
`SubCommands`

Description:
A command object can be one of the following: an executable command, a command container (for example, a menu, submenu, or toolbar), or a menu separator. To determine what kind of information is stored in the current `Command` object, query its `ID`, `IsSeparator`, and `SubCommands` properties, as follows.

The Command object is...	When...
An executable command	• `ID` is greater than zero • `IsSeparator` is false • `SubCommands` is empty
A command container	• `ID` is zero • `IsSeparator` is false • `SubCommands` contains a collection of `Command` objects.
Separator	• `ID` is zero • `IsSeparator` is true

29.4.7.1.1 Accelerator

Property: `Accelerator` as string

Description:
Returns the accelerator key defined for the command. If the command has no accelerator key assigned, this property returns the empty string. The string representation of the accelerator key has the following format:

```
[ALT+][CTRL+][SHIFT+]key
```

Where `key` is converted using the Windows Platform SDK function `GetKeyNameText`.

29.4.7.1.2 ID

Property: `ID` as long

Description:
This property gets the unique identifier of the command. A command's ID is required to execute the command (using `Exec`) or query its status (using `QueryStatus`). If the command is a container for other commands (for example, a top-level menu), or a separator, the ID is 0.

29.4.7.1.3 IsSeparator

Property: `IsSeparator` as boolean

Description:
The property returns `true` if the command object is a menu separator; `false` otherwise. See also Command.

29.4.7.1.4 Label

Property: `Label` as string

© 2016-2022 Altova GmbH

Description:
This property gets the text of the command as it is displayed in the graphical user interface of XMLSpy. If the command is a separator, "Label" is an empty string. This property may also return an empty string for some toolbar commands that do not have any GUI text associated with them.

29.4.7.1.5 Name

Property: `Name` as string

Description:
This property gets the unique name of the command. This value can be used to get the icon file of the command, where it is available. The available icon files can be found in the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your XMLSpy installation.

29.4.7.1.6 StatusText

Property: `Label` as string

Description:
The status text is the text shown in the status bar of XMLSpy when the command is selected. It applies only to command objects that are not separators or containers of other commands; otherwise, the property is an empty string.

29.4.7.1.7 SubCommands

Property: `SubCommands` as `Commands`

Description:
The `SubCommands` property gets the collection of `Command` objects that are sub-commands of the current command. The property is applicable only to commands that are containers for other commands (menus, submenus, or toolbars). Such container commands have the `ID` set to 0, and the `IsSeparator` property set to `false`.

29.4.7.1.8 ToolTip

Property: `ToolTip` as string

Description:
This property gets the text that is shown as a tool-tip for each command. If the command does not have a tooltip text, the property returns an empty string.

29.4.7.2 XMLSpyCommands

Properties:
`Count`

Item

Description:
Collection of Command objects to get access to command labels and IDs of the XMLSpyControl. Those commands can be executed with the Exec method and their status can be queried with QueryStatus.

29.4.7.2.1 Count

Property: Count as long

Description:
Number of Command objects on this level of the collection.

29.4.7.2.2 Item

Property: Item (n as long) as Command

Description:
Gets the command with the index n in this collection. Index is 1-based.

29.4.7.3 XMLSpyControl

Properties:
IntegrationLevel
Appearance
Application
BorderStyle
CommandsList
EnableUserPrompts
MainMenu
Toolbars

Methods:
Open
Exec
QueryStatus

Events:
OnUpdateCmdUI
OnOpenedOrFocused
OnCloseEditingWindow
OnFileChangedAlert
OnContextChanged
OnDocumentOpened
OnValidationWindowUpdated

This object is a complete ActiveX control and should only be visible if the XMLSpy library is used in the Application Level mode.

© 2016-2022 Altova GmbH

```
CLSID: a258bba2-3835-4c16-8590-72b44f52c471
ProgID: Altova.XMLSpyControl
```

29.4.7.3.1 Properties

The following properties are defined:

IntegrationLevel
EnableUserPrompts
Appearance
BorderStyle

Command related properties:
CommandsList
MainMenu
Toolbars

Access to XMLSpyAPI:
Application

29.4.7.3.1.1 Appearance

Property: `Appearance` as short

Dispatch Id: *-520*

Description:
A value not equal to `0` displays a client edge around the control. Default value is `0`.

29.4.7.3.1.2 Application

Property: `Application` as Application

Dispatch Id: *1*

Description:
The `Application` property gives access to the `Application` object of the complete XMLSpy automation server API. The property is read-only.

29.4.7.3.1.3 BorderStyle

Property: `BorderStyle` as short

Dispatch Id: *-504*

Description:

A value of `1` displays the control with a thin border. Default value is `0`.

29.4.7.3.1.4 CommandsList

Property: `CommandList` as `Commands` (read-only)

Dispatch Id: *1004*

Description:
This property returns a flat list of all commands defined available with XMLSpyControl. To get commands organized according to their menu structure, use `MainMenu`. To get toolbar commands, use `Toolbars`.

```
public void GetAllXmlSpyCommands()
{
    // Get all commands from the XMLSpy ActiveX control assigned to the current form
    XMLSpyControlLib.XMLSpyCommands commands = this.axXMLSpyControl1.CommandList;
    // Iterate through all commands
    for (int i = 0; i < commands.Count; i++)
    {
      // Get each command by index and output it to the console
      XMLSpyControlLib.XMLSpyCommand cmd = axXMLSpyControl1.CommandList[i];
      Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}
```

C# example

29.4.7.3.1.5 EnableUserPrompts

Property: `EnableUserPrompts` as boolean

Dispatch Id: *1006*

Description:
Setting this property to *false*, disables user prompts in the control. The default value is *true*.

29.4.7.3.1.6 IntegrationLevel

Property: `IntegrationLevel` as `ICActiveXIntegrationLevel`

Dispatch Id: *1000*

Description:
The `IntegrationLevel` property determines the operation mode of the control. See also Integration at Application Level and Integration at Document Level for more information.

© 2016-2022 Altova GmbH

Note: It is important to set this property immediately after the creation of the `XMLSpyControl` object.

29.4.7.3.1.7 MainMenu

Property: `MainMenu` as Command (read-only)

Dispatch Id: *1003*

Description:
This property provides information about the structure and commands available in the XMLSpyControl main menu, as a `Command` object. The `Command` object contains all available submenus of XMLSpy (for example "File", "Edit", "View" etc.). To access the submenu objects, use the `SubCommands` property of the `MainMenu` property. Each submenu is also a `Command` object. For each submenu, you can then further iterate through their `SubCommands` property in order to get their corresponding child commands and separators (this technique may be used, for example, to create the application menu programmatically). Note that some menu commands act as containers ("parents") for other menu commands, in which case they also have a `SubCommands` property. To get the structure of all menu commands programmatically, you will need a recursive function.

```
public void GetXmlSpyMenus()
{
    // Get the main menu from the XMLSpy ActiveX control assigned to the current form
    XMLSpyControlLib.XMLSpyCommand mainMenu = this.axXMLSpyControl1.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
      XMLSpyControlLib.XMLSpyCommand menu = mainMenu.SubCommands[i];
      Console.WriteLine("{0} menu has {1} children items (including separators)",
menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}
```

C# example

29.4.7.3.1.8 Toolbars

Property: `Toolbars` as Commands (read-only)

Dispatch Id: *1005*

Description:
This property provides information about the structure of XMLSpyControl toolbars, as a `Command` object. The `Command` object contains all available toolbars of XMLSpy. To access the toolbars, use the `SubCommands` property of the `Toolbars` property. Each toolbar is also a `Command` object. For each toolbar, you can then further iterate through their `SubCommands` property in order to get their commands (this technique may be used, for example, to create the application's toolbars programmatically).

```
public void GetXmlSpyToolbars()
{
    // Get the application toolbars from the StyleVision ActiveX control assigned to the
current form
    XMLSpyControlLib.XMLSpyCommands toolbars = this.axXMLSpyControl1.Toolbars;

    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)
    {
      XMLSpyControlLib.XMLSpyCommand toolbar = toolbars[i];
      Console.WriteLine();
      Console.WriteLine("The toolbar \"{0}\" has the following commands:",
toolbar.Label);

      // Iterate through all commands of this toolbar
      for (int j = 0; j < toolbar.SubCommands.Count; j++)
      {
         XMLSpyControlLib.XMLSpyCommand cmd = toolbar.SubCommands[j];
         // Output only command objects that are not separators
         if (!cmd.IsSeparator)
         {
            Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
""));
         }
      }
    }
}
```

C# example

29.4.7.3.2 Methods

The following methods are defined:

Open
Exec
QueryStatus

29.4.7.3.2.1 Exec

Method: `Exec` (`nCmdID` as long) as boolean

Dispatch Id: *6*

Description:
This method calls the XMLSpy command with the ID `nCmdID`. If the command can be executed, the method returns `true`. To get a list of all available commands, use `CommandsList`. To retrieve the status of any command, use `QueryStatus`.

© 2016-2022 Altova GmbH

29.4.7.3.2.2 Open

Method: `Open` (`strFilePath` as string) as boolean

Dispatch Id: *5*

Description:
The result of the method depends on the extension passed in the argument `strFilePath`. If the file extension is `.sps`, a new document is opened. If the file extension is `.svp`, the corresponding project is opened. If a different file extension is passed into the method, the control tries to load the file as a new component into the active document.

Do not use this method to load documents or projects when using the control in document-level integration mode. Instead, use `XMLSpyControlDocument.Open` and `XMLSpyControlPlaceHolder.OpenProject`.

29.4.7.3.2.3 QueryStatus

Method: `QueryStatus` (`nCmdID` as long) as long

Dispatch Id: *7*

Description:
`QueryStatus` returns the enabled/disabled and checked/unchecked status of the command specified by `nCmdID`. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if `QueryStatus` returns `0` the command ID is not recognized as a valid XMLSpy command. If `QueryStatus` returns a value of `1` or `5`, the command is disabled.

29.4.7.3.3 Events

The XMLSpyControl ActiveX control provides the following connection point events:

`OnUpdateCmdUI`
`OnOpenedOrFocused`
`OnCloseEditingWindow`
`OnFileChangedAlert`
`OnContextChanged`

`OnDocumentOpened`
`OnValidationWindowUpdated`

29.4.7.3.3.1 OnCloseEditingWindow

Event: `OnCloseEditingWindow` (i_strFilePath as String) as boolean

Dispatch Id: *1002*

Description:
This event is triggered when XMLSpy needs to close an already open document. As an answer to this event, clients should close the editor window associated with *i_strFilePath*. Returning *true* from this event indicates that the client has closed the document. Clients can return *false* if no specific handling is required and XMLSpyControl should try to close the editor and destroy the associated document control.

29.4.7.3.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` as String, `i_bActive` as bool) as bool

Dispatch Id: *1004*

Description:
This event is triggered when XMLSpy activates or de-actives one of the following operational contexts:

- XSLT Profiling - "`XSLTProfiling`" is passed as the context name
- XSLT / XQuery debugging - "`DebuggingXSLT`" is passed as the context name
- SOAP debugging - "`DebuggingSOAP`" is passed as the context name (Enterprise edition only)

29.4.7.3.3.3 OnDocumentOpened

Event: `OnDocumentOpened` (`objDocument` as Document)

Dispatch Id: *1*

Description:
This event is triggered whenever a document is opened. The argument `objDocument` is a `Document` object from the XMLSpy automation interface and can be used to query for more details about the document, or perform additional operations. When integrating on document-level, it is often better to use the event `XMLSpyControlDocument.OnDocumentOpened` instead.

29.4.7.3.3.4 OnFileChangedAlert

Event: `OnFileChangedAlert` (`i_strFilePath` as String) as bool

Dispatch Id: *1001*

Description:

© 2016-2022 Altova GmbH

This event is triggered when a file loaded with XMLSpyControl is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if XMLSpy should handle it in its customary way, i.e. prompting the user for reload.

29.4.7.3.3.5 OnLicenseProblem

Event: `OnLicenseProblem` (`i_strLicenseProblemText` as String)

Dispatch Id: *1005*

Description:
This event is triggered when XMLSpyControl detects that no valid license is available for this control. In case of restricted user licenses this can happen some time after the control has been initialized. Integrators should use this event to disable access to this control's functionality. After returning from this event, the control will block access to its functionality (e.g. show empty windows in its controls and return errors on requests).

29.4.7.3.3.6 OnOpenedOrFocused

Event: `OnOpenedOrFocused` (`i_strFilePath` as String, `i_bOpenWithThisControl` as bool)

Dispatch Id: *1000*

Description:
When integrating at application level, this event informs clients that a document has been opened, or made active by XMLSpy.

When integrating at document level, this event instructs the client to open the file `i_strFilePath` in a document window. If the file is already open, the corresponding document window should be made the active window.

if `i_bOpenWithThisControl` is `true`, the document must be opened with XMLSpyControl, since internal access is required. Otherwise, the file can be opened with different editors.

29.4.7.3.3.7 OnToolWindowUpdated

Event: `OnToolWindowUpdated(pToolWnd` as long `)`

Dispatch Id: *1006*

Description:
This event is triggered when the tool window is updated.

29.4.7.3.3.8 OnUpdateCmdUI

Event: `OnUpdateCmdUI ()`

Dispatch Id: *1003*

Description:
Called frequently to give integrators a good opportunity to check status of XMLSpy commands using `XMLSpyControl.QueryStatus`. Do not perform long operations in this callback.

29.4.7.3.3.9 OnValidationWindowUpdated

Event: `OnValidationWindowUpdated` ()

Dispatch Id: *3*

Description:
This event is triggered whenever the validation output window is updated with new information.

29.4.7.4 XMLSpyControlDocument

Properties:
`Appearance`
`BorderStyle`
`Document`
`IsModified`
`Path`
`ReadOnly`

Methods:
`Exec`
`New`
`Open`
`QueryStatus`
`Reload`
`Save`
`SaveAs`

Events:
`OnDocumentOpened`
`OnDocumentClosed`
`OnModifiedFlagChanged`
`OnContextChanged`
`OnFileChangedAlert`
`OnActivate`

If the XMLSpyControl is integrated in the Document Level mode each document is displayed in an own object of type `XMLSpyControlDocument`. The `XMLSpyControlDocument` contains only one document at the time but can be reused to display different files one after another.

This object is a complete ActiveX control.

```
CLSID: 52A552E6-2AB8-4e3e-B545-BE998233DDA0
ProgID: Altova.XMLSpyControlDocument
```


© 2016-2022 Altova GmbH

29.4.7.4.1 Properties

The following properties are defined:

ReadOnly
IsModified
Path
Appearance
BorderStyle

Access to XMLSpyAPI:
Document

29.4.7.4.1.1 Appearance

Property: Appearance as short

Dispatch Id: *-520*

Description:
A value not equal to 0 displays a client edge around the document control. Default value is 0.

29.4.7.4.1.2 BorderStyle

Property: BorderStyle as short

Dispatch Id: *-504*

Description:
A value of 1 displays the control with a thin border. Default value is 0.

29.4.7.4.1.3 Document

Property: Document as Document

Dispatch Id: *1*

Description:
The Document property gives access to the Document object of the XMLSpy automation server API. This interface provides additional functionality which can be used with the document loaded in the control. The property is read-only.

29.4.7.4.1.4 IsModified

Property: IsModified as boolean (read-only)

Dispatch Id: *1006*

Description:
`IsModified` is *true* if the document content has changed since the last open, reload or save operation. It is *false*, otherwise.

29.4.7.4.1.5 Path

Property: `Path` as string

Dispatch Id: *1005*

Description:
Sets or gets the full path name of the document loaded into the control.

29.4.7.4.1.6 ReadOnly

Property: `ReadOnly` as boolean

Dispatch Id: *1007*

Description:
Using this property you can turn on and off the read-only mode of the document. If `ReadOnly` is `true` it is not possible to do any modifications.

29.4.7.4.2 Methods

The following methods are defined:

Document handling:
`New`
`Open`
`Reload`
`Save`
`SaveAs`

Command Handling:
`Exec`
`QueryStatus`

29.4.7.4.2.1 Exec

Method: `Exec` (`nCmdID` as long) as boolean

Dispatch Id: *8*

Description:

© 2016-2022 Altova GmbH

`Exec` calls the XMLSpy command with the ID `nCmdID`. If the command can be executed, the method returns `true`. This method should be called only if there is currently an active document available in the application.

To get commands organized according to their menu structure, use the `MainMenu` property of XMLSpyControl. To get toolbar commands, use the `Toolbars` property of the XMLSpyControl.

29.4.7.4.2.2 New

Method: `New` () as boolean

Dispatch Id: *1000*

Description:
This method initializes a new document inside the control.

29.4.7.4.2.3 Open

Method: `Open` (`strFileName` as string) as boolean

Dispatch Id: *1001*

Description:
`Open` loads the file `strFileName` as the new document into the control.

29.4.7.4.2.4 QueryStatus

Method: `QueryStatus` (`nCmdID` as long) as long

Dispatch Id: *9*

Description:
`QueryStatus` returns the enabled/disabled and checked/unchecked status of the command specified by `nCmdID`. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if `QueryStatus` returns `0` the command ID is not recognized as a valid XMLSpy command. If `QueryStatus` returns a value of `1` or `5` the command is disabled. The client should call the `QueryStatus` method of the document control if there is currently an active document available in the application.

29.4.7.4.2.5 Reload

Method: `Reload` () as boolean

Dispatch Id: *1002*

Description:
`Reload` updates the document content from the file system.

29.4.7.4.2.6 Save

Method: `Save` () as boolean

Dispatch Id: *1003*

Description:
`Save` saves the current document at the location `Path`.

29.4.7.4.2.7 SaveAs

Method: `SaveAs` (`strFileName` as string) as boolean

Dispatch Id: *1004*

Description:
`SaveAs` sets `Path` to *strFileName* and then saves the document to this location.

29.4.7.4.3 Events

The XMLSpyControlDocument ActiveX control provides following connection point events:

`OnDocumentOpened`
`OnDocumentClosed`
`OnModifiedFlagChanged`
`OnContextChanged`
`OnFileChangedAlert`
`OnActivate`
`OnSetEditorTitle`

29.4.7.4.3.1 OnActivate

Event: `OnActivate` ()

Dispatch Id: *1005*

Description:

© 2016-2022 Altova GmbH

This event is triggered when the document control is activated, has the focus, and is ready for user input.

29.4.7.4.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` as String, `i_bActive` as bool) as bool

Dispatch Id: *1004*

Description:
This event is triggered when this document is shown in a different XMLSpy view. The following values are passed:

- Grid view - `"View_0"` is passed as the context name
- Text view - `"View_1"` is passed as the context name
- Browser view - `"View_2"` is passed as the context name
- Schema view - `"View_3"` is passed as the context name
- Authentic view - `"View_4"` is passed as the context name
- WSDL view - `"View_5"` is passed as the context name

29.4.7.4.3.3 OnDocumentClosed

Event: `OnDocumentClosed` (`objDocument` as Document)

Dispatch Id: *1001*

Description:
This event is triggered whenever the document loaded into this control is closed. The argument `objDocument` is a `Document` object from the XMLSpy automation interface and should be used with care.

29.4.7.4.3.4 OnDocumentOpened

Event: `OnDocumentOpened` (`objDocument` as Document)

Dispatch Id: *1000*

Description:
This event is triggered whenever a document is opened in this control. The argument `objDocument` is a `Document` object from the XMLSpy automation interface, and can be used to query for more details about the document, or perform additional operations.

29.4.7.4.3.5 OnDocumentSaveAs

Event: `OnContextDocumentSaveAs` (`i_strFileName` as String)

Dispatch Id: *1007*

Description:
This event is triggered when this document gets internally saved under a new name.

29.4.7.4.3.6 OnFileChangedAlert

Event: `OnFileChangedAlert` () as bool

Dispatch Id: *1003*

Description:
This event is triggered when the file loaded into this document control is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if XMLSpy should handle it in its customary way, i.e. prompting the user for reload.

29.4.7.4.3.7 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` as boolean)

Dispatch Id: *1002*

Description:
This event gets triggered whenever the document changes between modified and unmodified state. The parameter *i_bIsModifed* is *true* if the document contents differs from the original content, and *false*, otherwise.

29.4.7.4.3.8 OnSetEditorTitle

Event: OnSetEditorTitle ()

Dispatch Id: *1006*

Description:
This event is being raised when the contained document is being internally renamed.

29.4.7.5 XMLSpyControlPlaceHolder

Properties available for all kinds of placeholder windows:
`PlaceholderWindowID`

Properties for project placeholder window:
`Project`

Methods for project placeholder window:
`OpenProject`
`CloseProject`

© 2016-2022 Altova GmbH

The `XMLSpyControlPlaceHolder` control is used to show the additional XMLSpy windows like Overview, Library or Project window. It is used like any other ActiveX control and can be placed anywhere in the client application.

```
CLSID: 135DEEF4-6DF0-47c2-8F8C-F145F5F5F3F672
ProgID: Altova.XMLSpyControlPlaceHolder
```

29.4.7.5.1 Properties

The following properties are defined:

`PlaceholderWindowID`

Access to XMLSpyAPI:
`Project`

29.4.7.5.1.1 Label

Property: `Label` as String (read-only)

Dispatch Id: *1001*

Description:
This property gives access to the title of the placeholder. The property is read-only.

29.4.7.5.1.2 PlaceholderWindowID

Property: `PlaceholderWindowID as XMLSpyControlPlaceholderWindow`

Dispatch Id: *1*

Description:
This property specifies which XMLSpy window should be displayed in the client area of the control. The `PlaceholderWindowID` can be set at any time to any valid value of the `XMLSpyControlPlaceholderWindow` enumeration. The control changes its state immediately and shows the new XMLSpy window.

29.4.7.5.1.3 Project

Property: `Project` as Project (read-only)

Dispatch Id: *2*

Description:
The `Project` property gives access to the `Project` object of the XMLSpy automation server API. This interface provides additional functionality which can be used with the project loaded into the control. The property will return a valid project interface only if the placeholder window has `PlaceholderWindowID` with a value of `XMLSpyXProjectWindow (=3)`. The property is read-only.

29.4.7.5.2 Methods

The following method is defined:

OpenProject
CloseProject

29.4.7.5.2.1 OpenProject

Method: `OpenProject` (`strFileName` as `string`) as `boolean`
Dispatch Id: *3*
Description:
`OpenProject` loads the file `strFileName` as the new project into the control. The method will fail if the placeholder window has a **`PlaceholderWindowID`** different to `XMLSpyXProjectWindow (=3)`.

29.4.7.5.2.2 CloseProject

Method: `CloseProject ()`

Dispatch Id: *4*

Description:
`CloseProject` closes the project loaded by the control. The method will fail if the placeholder window has a `PlaceholderWindowID` different to `XMLSpyXProjectWindow (=3)`.

29.4.7.5.3 Events

The XMLSpyControlPlaceholder ActiveX control provides following connection point events:

OnModifiedFlagChanged

29.4.7.5.3.1 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` as boolean)

Dispatch Id: *1*

Description:
This event gets triggered only for placeholder controls with a `PlaceholderWindowID` of `XMLSpyXProjectWindow (=3)`. The event is fired whenever the project content changes between modified and unmodified state. The parameter *i_bIsModifed* is *true* if the project contents differs from the original content, and *false*, otherwise.

© 2016-2022 Altova GmbH

29.4.7.5.3.2 OnSetLabel

Event: `OnSetLabel(i_strNewLabel as string)`

Dispatch Id: *1000*

Description:
Raised when the title of the placeholder window is changed.

29.4.7.6 Enumerations

The following enumerations are defined:

ICActiveXIntegrationLevel
XMLSpyControlPlaceholderWindow

29.4.7.6.1 ICActiveXIntegrationLevel

Possible values for the IntegrationLevel property of the XMLSpyControl.

```
ICActiveXIntegrationOnApplicationLevel  = 0
ICActiveXIntegrationOnDocumentLevel     = 1
```

29.4.7.6.2 XMLSpyControlPlaceholderWindow

This enumeration contains the list of the supported additional XMLSpy windows.

```
XMLSpyControlNoToolWnd                          = -1
XMLSpyControlEntryHelperTopToolWnd       = 0
XMLSpyControlEntryHelperMiddleToolWnd           = 1
XMLSpyControlEntryHelperBottomToolWnd           = 2
XMLSpyControlValidatorOutputToolWnd             = 3
XMLSpyControlProjectWindowToolWnd               = 4
XMLSpyControlXSLTDebuggerContextToolWnd         = 5
XMLSpyControlXSLTDebuggerCallstackToolWnd= 6
XMLSpyControlXSLTDebuggerVariableToolWnd = 7
XMLSpyControlXSLTDebuggerWatchToolWnd           = 8
XMLSpyControlXSLTDebuggerTemplateToolWnd = 9
XMLSpyControlXSLTDebuggerInfoToolWnd            = 10
XMLSpyControlXSLTDebuggerMessageToolWnd         = 11
XMLSpyControlXSLTDebuggerTraceToolWnd           = 12
XMLSpyControlSOAPDebuggerToolWnd                = 13
XMLSpyControlXPathProfilerListToolWnd           = 14
XMLSpyControlXPathProfilerTreeToolWnd           = 15
XMLSpyControlXPathDialogToolWnd                 = 16
XMLSpyControlDBQueryManagerToolWnd       = 17
XMLSpyControlInfoToolWnd                        = 18
XMLSpyControlXSLOutlineToolWnd                  = 19
XMLSpyControlSchemaFindToolWnd                  = 20
```

```
XMLSpyControlXBRLFindToolWnd                = 21
XMLSpyControlChartsToolWnd                  = 22
```


© 2016-2022 Altova GmbH

30 Annexes

Ces annexes contiennent des informations techniques concernant XMLSpy et des informations importantes concernant les licences. Chaque annexe contient les sous-sections suivantes :

Information Moteur

- Informations moteur XSLT et XQuery
- Fonctions d'extension XSLT et XQuery

Conversions de types de données entre les BD et les Schémas XML

- BD en Schémas XML
- Schémas XML en BD

Données techniques

- SE et exigences de mémoire
- Moteurs XSLT et XQuery Altova
- Prise en charge Unicode
- Utilisation Internet

Informations de licence

- Distribution électronique de logiciel
- Activation de logiciel et metering de licence
- Contrat de licence de l'utilisateur final

30.1 Informations concernant le moteur XSLT et XQuery

Les moteurs XSLT et XQuery de XMLSpy suivent de près les spécifications W3C et sont donc plus strictes que les moteurs Altova précédents, comme dans les versions précédentes de XMLSpy. Ainsi, de petites erreurs qui étaient ignorées par les moteurs précédents sont maintenant marquées en tant qu'erreurs par XMLSpy.

Par exemple :

- Il s'agit d'une erreur de type (`err:XPTY0018`) si le résultat d'un opérateur de chemin contient aussi bien les nœuds que les non-nœuds.
- Il s'agit d'une erreur de type (`err:XPTY0019`) si `E1` dans une expression de chemin `E1/E2` n'évalue pas à une séquence de nœuds.

Si vous rencontrez ce type d'erreur, modifiez soit le document XSLT/XQuery, soit le document d'instance selon vos besoins.

Cette section décrit les fonctions spécifiques à la mise en place des moteurs, organisée par spécification :

- XSLT 1.0
- XSLT 2.0
- XSLT 3.0
- XQuery 1.0
- XQuery 3.1

30.1.1 XSLT 1.0

Le moteur XSLT 1.0 de XMLSpy est conforme aux Recommandations XSLT 1.0 du 16 novembre 1999 et aux Recommandations XPath 1.0 du 16 novembre 1999 du World Wide Web Consortium (W3C's). Veuillez noter les informations suivantes concernant l'implémentation.

Notes concernant l'implémentation

Lorsque l'attribut `method` de `xsl:output` est défini sur HTML, ou si la sortie HTML est sélectionnée par défaut, les caractères spéciaux dans le fichier XML ou XSLT sont insérés dans le document HTML en tant que références de caractère HTML dans la sortie. Par exemple, le caractère U+00A0 (la référence de caractère hexadécimale pour un espace insécable) est inséré dans le code HTML soit en tant que référence de caractère (` ` ou ` `) soit en tant que référence d'entité, ` ` .

30.1.2 XSLT 2.0

Cette section :

- Conformité du moteur
- Rétrocompatibilité
- Espaces de nom
- Compatibilité avec le schéma

 © 2016-2022 Altova GmbH

- Comportement spécifique à l'implémentation

Conformité

Le moteur XSLT 2.0 de XMLSpy est conforme aux Recommandations XSLT 2.0 du 23 janvier 2007 et aux Recommandations XPath 2.0 du 14 décembre 2010 du World Wide Web Consortium (W3C's).

Rétrocompatibilité

Le moteur XSLT 2.0 est rétrocompatible. Le seul moment où la compatibilité rétroactive du moteur XSLT 2.0 prend effet est si vous utilisez le moteur XSLT 2.0 pour traiter une feuille de style XSLT 1.0. Veuillez noter qu'il peut y avoir des différences dans les sorties produites par le moteur XSLT 1.0 et la rétrocompatibilité du moteur XSLT 2.0.

Espaces de nom

Votre feuille de style XSLT 2.0 devrait déclarer les espaces de noms suivants afin que vous puissiez utiliser les constructeurs de type et les fonctions disponibles dans XSLT 2.0. Les préfixes indiqués ci-dessous sont utilisés de manière conventionnelle ; vous pourriez utiliser les préfixes alternatifs si vous le souhaitez.

Nom d'espace de nom	Préfixe	Espace de nom URI
Types de schéma XML	`xs:`	`http://www.w3.org/2001/XMLSchema`
Fonctions XPath 2.0	`fn:`	`http://www.w3.org/2005/xpath-functions`

Généralement ces espaces de nom seront déclarés sur l'élément `xsl:stylesheet` ou `xsl:transform`, tel que montré dans la liste suivante :

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    ...
</xsl:stylesheet>
```

Veuillez noter les points suivants :

- Le moteur XSLT 2.0 utilise l'espace de nom XPath 2.0 et Fonctions XQuery 1.0 (recensées dans la table ci-dessus) en tant que son **espace de noms de fonctions par défaut**. Ainsi, vous pouvez utiliser les fonctions XPath 2.0 et XSLT 2.0 dans votre feuille de style sans aucun préfixe. Si vous déclarez l'espace de nom des fonctions XPath 2.0 dans votre feuille de style avec un préfixe, vous pourrez utiliser en plus le préfixe attribué dans la déclaration.

- Lors de l'utilisation des constructeurs de types et des types provenant de l'espace de nom du Schéma XML, le préfixe utilisé dans la déclaration d'espace de nom doit être utilisé lors de l'appel du constructeur de type (par exemple, `xs:date`).
- Certaines fonctions XPath 2.0 portent le même nom que les types de données du schéma XML. Par exemple, pour les fonctions XPath `fn:string` et `fn:boolean`, il existe des types de données du schéma XML portant le même nom local : `xs:string` et `xs:boolean`. Donc si vous décidez d'utiliser l'expression XPath `string('Hello')`, l'expression évalue en tant que `fn:string('Hello')` et non pas en tant que `xs:string('Hello')`.

Compatibilité avec le schéma

Le moteur XSLT 2.0 est compatible avec le schéma. Vous pouvez ainsi utiliser des types de schéma définis par l'utilisateur et l'instruction `xsl:validate`.

Comportement spécifique à l'implémentation

Ci-dessous, vous trouverez une description de la gestion du moteur XSLT 2.0 des aspects spécifiques à l'implémentation du comportement de certaines fonctions XSLT 2.0.

`xsl:result-document`
Les encodages pris en charge en supplément sont (les codes spécifiques à Altova) : `x-base16tobinary` et `x-base64tobinary`.

`function-available`
La fonction teste la disponibilité des fonctions in-scope (XSLT, XPath, et fonctions d'extension).

`unparsed-text`
L'attribut `href` accepte (i) les chemins relatifs pour les fichiers dans le dossier de base-uri, et (ii) les chemins absolus avec ou sans le protocole `file://`. De plus, les encodages pris en charge supplémentaires sont (les codes spécifiques Altova) : `x-binarytobase16` et `x-binarytobase64`.

`unparsed-text-available`
L'attribut `href` accepte (i) les chemins relatifs pour les fichiers dans le dossier de base-uri, et (ii) les chemins absolus avec ou sans le protocole `file://`. De plus, les encodages pris en charge supplémentaires sont (les codes spécifiques Altova) : `x-binarytobase16` et `x-binarytobase64`.

Note : les valeurs d'encodage suivantes, qui ont été mises en place dans les versions plus anciennes du produit prédécesseur de RaptorXML, AltovaXML, sont désormais obsolètes : `base16tobinary`, `base64tobinary`, `binarytobase16` et `binarytobase64`.

© 2016-2022 Altova GmbH

30.1.3 XSLT 3.0

Le moteur XSLT 3.0 de XMLSpy est conforme aux Recommandation XSLT 3.0 du 8 juin 2017 et aux Recommandations XQuery 3.1 du 21 mars 2017 du World Wide Web Consortium (W3C's).

Le moteur XSLT 3.0 présente les mêmes caractéristiques spécifiques à l'implémentation que le moteur XSLT 2.0. De plus, il contient une prise en charge pour un certain nombre de nouvelles fonctions XSLT 3.0 : les fonctions et opérateurs XPath/XQuery 3.1, et la spécification XPath 3.1.

Note : la fonction de streaming optionnelle n'est pas prise en charge actuellement. Le document complet sera chargé dans la mémoire quelle que soit la valeur de l'attribut `streamable`. S'il y a assez de mémoire disponible, alors : (i) le document entier sera traité - sans streaming, (ii) les constructions garanties streamables seront traitées correctement, comme si l'exécution utilisait le streaming, et (iii) les erreurs de streaming ne seront pas détectées. Dans les applis 64-bit, l'exécution non-streaming ne devrait pas représenter de problème. S'il y a tout de même un souci avec la mémoire, une solution serait d'ajouter plus de mémoire au système.

30.1.4 XQuery 1.0

Cette section :

- Conformité du moteur
- Compatibilité du schéma (Schema awareness)
- Encodage
- Espaces de nom
- XML source et validation
- Contrôle de type statique et dynamique
- Modules bibliothèque
- Modules externes
- Collations
- Précision des données numériques
- Prise en charge des instructions XQuery

Conformité

Le moteur XQuery 1.0 de XMLSpy est conforme à la Recommandation XQuery 1.0 du 14 décembre 2010 du World Wide Web Consortium (W3C's). Le standard XQuery accorde un pouvoir discrétionnaire concernant la mise en place de nombreuses fonctions. Ci-dessous, vous trouverez une liste expliquant comment le moteur XQuery 1.0 implémente ces fonctions.

Compatibilité avec le schéma

Le moteur XQuery 1.0 est **schema-aware**.

Encodage

Les encodages de caractères UTF-8 et UTF-16 sont pris en charge.

Espaces de nom

Les URI d'espace de nom suivant et leurs liaisons associées sont prédéfinies.

Nom d'espaces de nom	Préfixe	URI Espace de noms
Types de schéma XML	`xs:`	`http://www.w3.org/2001/XMLSchema`
Instance de schéma	`xsi:`	`http://www.w3.org/2001/XMLSchema-instance`
Fonctions intégrées	`fn:`	`http://www.w3.org/2005/xpath-functions`
Fonctions locales	`local:`	`http://www.w3.org/2005/xquery-local-functions`

Veuillez noter les points suivants :

- Le moteur XQuery 1.0 Engine reconnaît les préfixes recensés ci-dessus comme étant liés aux espaces de noms correspondants.
- Étant donné que l'espace de noms des fonctions intégrées recensé ci-dessus est l'espace de noms des fonctions par défaut dans XQuery, le préfixe `fn:` ne doit pas nécessairement être utilisé lorsque des fonctions intégrées sont invoquées (par exemple, `string("Hello")` appellera la fonction `fn:string`). Néanmoins, le préfixe `fn:` peut être utilisé pour appeler une fonction intégrée sans avoir à déclarer l'espace de noms dans le prologue query (par exemple : `fn:string("Hello")`).
- Vous pouvez changer l'espace de noms des fonctions par défaut en déclarant l'expression `default function namespace` dans le prologue de de requête.
- En cas d'utilisation des types depuis l'espace de noms du Schéma XML, le préfixe `xs:` peut être utilisé sans devoir déclarer explicitement les espace de noms et lier ces préfixes dans le prologue de requête. (Exemple : `xs:date` and `xs:yearMonthDuration`.) Si vous souhaitez utiliser d'autres préfixes pour l'espace de noms du schéma XML, cela doit être déclaré explicitement dans le prologue de requête. (Exemple: `declare namespace alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04")`.)
- Veuillez noter que les types de données `untypedAtomic`, `dayTimeDuration`, et `yearMonthDuration` ont été déplacés, avec les CR de 23 January 2007, depuis l'espace de noms des Types de données XPath vers l'espace de noms du schéma XML, donc : `xs:yearMonthDuration`.

Si des espaces de noms pour les fonctions, les constructeurs de type, les tests de nœud, etc. sont mal attribués, une erreur est rapportée. Veuillez noter, néanmoins, que certaines fonctions portent le même nom que les types de données de schéma, par ex. `fn:string` et `fn:boolean`. (Les deux `xs:string` et `xs:boolean` sont définis.) Le préfixe d'espace de noms détermine si la fonction ou le constructeur de type est utilisé.

© 2016-2022 Altova GmbH

XML document de source et validation

Les documents XML utilisés dans l'exécution d'un document XQuery avec le moteur XQuery 1.0 doit être bien formé. Néanmoins, ils ne doivent pas être valides conformément à un schéma XML. Si le fichier n'est pas valide, le fichier invalide est chargé sans information de schéma. Si le fichier XML est associé avec un schéma interne et est valide conformément à ce schéma, l'information de validation post-schéma sera générée pour les données XML et sera utilisée pour l'évaluation de requête.

Contrôle de type statique et dynamique

La phase d'analyse statique contrôle les aspects de la requête comme la syntaxe, si des références externes existent (par ex. pour les modules), si des fonctions et des variables invoquées sont définies, etc. Si une erreur est détectée dans la phase de l'analyse statique, elle sera rapportée et l'exécution sera stoppée.

Le contrôle de type dynamique est effectué lors de l'exécution, lorsque la requête est réellement exécutée. Si un type est incompatible avec les exigences d'une opération, une erreur sera rapportée. Par exemple, l'expression `xs:string("1") + 1` retourne une erreur parce que l'opération d'édition ne peut pas être effectuée sur un opérande de type `xs:string`.

Modules de bibliothèque

Les modules de Bibliothèque stockent les fonctions et les variables de manière à ce qu'elles puissent être réutilisées. Le moteur XQuery 1.0 prend en charge des modules qui sont stockés **dans un seul fichier XQuery externe**. Un tel fichier de module doit contenir une déclaration `module` dans son prologue, qui associe un espace de noms cible. Voici un module d'exemple :

```
module namespace libns="urn:module-library";
declare variable $libns:company := "Altova";
declare function libns:webaddress() { "http://www.altova.com" };
```

Toutes les fonctions et les variables déclarées dans le module font partie de l'espace de noms associé au module. Celui-ci est utilisé en l'important dans un fichier XQuery avec l'instruction `import module` se trouvant dans le prologue de requête. L'instruction `import module` importe uniquement les fonctions et les variables déclarées directement dans le fichier de module de bibliothèque. Comme suit :

```
import module namespace modlib = "urn:module-library" at "modulefilename.xq";
if         ($modlib:company = "Altova")
then        modlib:webaddress()
else        error("No match found.")
```

Fonctions externes

Les fonctions externes ne sont pas prises en charge, c.à.d. dans les expressions utilisant le mot-clé `external`, comme dans :

```
declare function hoo($param as xs:integer) as xs:string external;
```

Collations

La collation par défaut est la collation de point de code Unicode, qui compare les chaînes sur la base de leur point de code Unicode. Les autres collations prises en charge sont les collations ICU recensées ici. Pour utiliser une collation spécifique, fournir son URI tel que donné dans la liste des collations prises en charge. Toute comparaison de chaîne, y compris pour les fonctions `fn:max` et `fn:min` seront effectuées conformément à la collation spécifiée. Si l'option de collation n'est pas spécifiée, la collation de point de code Unicode par défaut est utilisée.

Précision des types numériques

- Le type de données `xs:integer` est une précision arbitraire, c.à.d. il peut représenter n'importe quel chiffre.
- Le type de données `xs:decimal` a une limite de 20 chiffres après la virgule.
- Les types de données `xs:float` et `xs:double` ont une précision limitée de 15 chiffres.

Prise en charge des instructions XQuery

L'instruction `Pragma` n'est pas prise en charge. Si elle survient, elle sera ignorée et l'expression de fallback sera évaluée.

30.1.5 XQuery 3.1

Le moteur XQuery 3.1 de XMLSpy est conforme à la Recommandation XQuery 3.1 du 21 mars 2017 du Consortium du World Wide Web (W3C's) et comprend la prise en charge des Fonctions 3.1 XPath et XQuery. La spécification XQuery 3.1 est un sur ensemble de la spécification 3.0. C'est pourquoi le moteur XQuery 3.1 prend en charge les fonctions XQuery 3.0.

Les caractéristiques spécifiques à l'implémentation sont les mêmes que pour XQuery 1.0.

© 2016-2022 Altova GmbH

30.2 Fonctions XSLT et XPath/XQuery

Cette section réunit les fonctions d'extension Altova et d'autres fonctions d'extension qui peuvent être utilisées dans les expressions XPath et/ou XQuery. Les fonctions d'extension Altova peuvent être utilisées avec les moteurs XSLT et XQuery d'Altova, et elles offrent des fonctions supplémentaires à celles disponibles dans les bibliothèques de fonctions définies dans les standards W3C.

Points généraux

Les points généraux suivants devraient être notés :

Normaltrue

- Les fonctions provenant des librairies des fonctions essentielles définies dans les spécifications W3C peuvent être appelées sans un préfixe. Cela est dû au fait que les moteurs XSLT et XQuery lisent les fonctions non préfixées comme faisant partie d'un espace de nom des fonctions par défaut qui est celui spécifié dans les spécifications des fonctions XPath/XQuery `http://www.w3.org/2005/xpath-functions`. Si cet espace de nom est déclaré explicitement dans un document XSLT ou XQuery, le préfixe utilisé dans la déclaration d'espace de nom peut aussi être utilisé en option sur les noms de fonction.
- En général, si une fonction escompte une séquence d'un item en tant qu'argument, et qu'une séquence de plus d'un item est soumise, une erreur sera retournée.
- Toutes les comparaisons de strings sont réalisées en utilisant la collation de point de code Unicode.
- Les résultats qui sont des QNames sont sérialisés sous la forme `[prefix:]localname`.

Précision de la décimale xs:

La précision se réfère au nombre de chiffres dans le nombre et la spécification requiert un minimum de 18 chiffres. Pour les opérations de division qui produisent un résultat de type `xs:decimal`, la précision est de 19 chiffres après le point décimal sans arrondissement.

Fuseau horaire implicite

Lorsque deux valeurs `date`, `time`, ou `dateTime` doivent être comparées, le fuseau horaire des valeurs comparées doit être connu. Si le fuseau n'est explicitement donné dans une telle valeur, le fuseau horaire implicite est utilisé. Le fuseau horaire implicite est prélevé de l'horloge du système et sa valeur peut être contrôlée avec la fonction `implicit-timezone()`.

Collations

La collation par défaut est la collation de point de code Unicode qui compare les chaînes sur la base de leur point de code Unicode. Le processeur utilise l'Unicode Collation Algorithm. D'autres collations prises en charge sont les collations ICU regroupées ci-dessous : pour utiliser une collation spécifique, il faut fournir son URI tel que spécifié dans la table ci-dessous. Toute comparaison de chaîne, y compris en ce qui concerne les fonctions `max` et `min`, sera effectuée conformément à la collation spécifiée. Si l'option de collation n'est pas spécifiée, la collation de point de code Unicode par défaut sera utilisée.

Langue	URI
`da`: Danois	`da_DK`
`de`: Allemand	`de_AT, de_BE, de_CH, de_DE, de_LI, de_LU`

en: Anglais	en_AS, en_AU, en_BB, en_BE, en_BM, en_BW, en_BZ, en_CA, en_GB, en_GU, en_HK, en_IE, en_IN, en_JM, en_MH, en_MP, en_MT, en_MU, en_NA, en_NZ, en_PH, en_PK, en_SG, en_TT, en_UM, en_US, en_VI, en_ZA, en_ZW
es: Espagnol	es_419, es_AR, es_BO, es_CL, es_CO, es_CR, es_DO, es_EC, es_ES, es_GQ, es_GT, es_HN, es_MX, es_NI, es_PA, es_PE, es_PR, es_PY, es_SV, es_US, es_UY, es_VE
fr: Français	fr_BE, fr_BF, fr_BI, fr_BJ, fr_BL, fr_CA, fr_CD, fr_CF, fr_CG, fr_CH, fr_CI, fr_CM, fr_DJ, fr_FR, fr_GA, fr_GN, fr_GP, fr_GQ, fr_KM, fr_LU, fr_MC, fr_MF, fr_MG, fr_ML, fr_MQ, fr_NE, fr_RE, fr_RW, fr_SN, fr_TD, fr_TG
it: Italien	it_CH, it_IT
ja: Japonais	ja_JP
nb: Norvégien Bokmål	nb_NO
nl: Néerlandais	nl_AW, nl_BE, nl_NL
nn: Nynorsk	nn_NO
pt: Portugais	pt_AO, pt_BR, pt_GW, pt_MZ, pt_PT, pt_ST
ru: Russe	ru_MD, ru_RU, ru_UA
sv: Suédois	sv_FI, sv_SE

Axe du nom d'espace
L'axe du nom d'espace est devenu obsolète dans XPath 2.0. Néanmoins, l'utilisation de l'axe du nom d'espace est prise en charge. Pour accéder aux information de l'espace de nom, avec des mécanismes XPath 2.0, utilisez les fonctions `in-scope-prefixes()`, `namespace-uri()` et `namespace-uri-for-prefix()`.

30.2.1 Fonctions d'extension Altova

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova, `http://www.altova.com/xslt-extensions`**, et sont indiquées dans cette section par le préfixe **`altova:`**, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Les fonctions définies dans les spécifications de Fonctions XPath/XQuery de W3C peuvent être utilisées dans : (i) les expressions XPath dans un contexte XSLT, et (ii) dans les expressions XQuery dans un document XQuery. Dans cette documentation, nous indiquons les fonctions à utiliser dans le contexte précédent (XPath dans XSLT) avec un symbole **XP** et les appelons fonctions XPath ; les fonctions qui peuvent être utilisées dans le contexte à venir (XQuery) sont indiquées avec un symbole **XQ** ; elles fonctionnent en tant que fonctions

© 2016-2022 Altova GmbH

XQuery. Les spécifications XSLT de W3C —pas les spécifications de Fonctions XPath/XQuery —définissent également les fonctions qui peuvent être utilisées dans des expressions XPath dans des documents XSLT. Ces fonctions sont marquées avec un symbole XSLT et sont appelées fonctions XSLT. Les versions XPath/XQuery et XSLT dans lesquelles une fonction peut être utilisée sont indiquées dans la description de la fonction (*voir symboles ci-dessous*). Les fonctions provenant des librairies de fonction XPath/XQuery et XSLT sont recensées dans un préfixe. Les fonctions d'extension provenant d'autres librairies, comme les fonctions d'extension Altova, sont regroupés avec un préfixe.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

Fonctions XSLT

Les fonctions XSLT peuvent uniquement être utilisées dans des expressions XPath dans un contexte XSLT (comme les fonctions `current-group()` ou `key()` de XSLT 2.0). Ces fonctions ne sont pas prévues pour, et ne fonctionneront pas dans un contexte non-XSLT (par exemple, dans un contexte XQuery). Veuillez noter que les fonctions XSLT pour XBRL peuvent être utilisées uniquement avec les éditions des produits Altova qui ont une prise en charge XBRL.

Fonctions XPath/XQuery

Les fonctions XPath/XQuery peuvent être utilisées tous les deux dans les expressions XPath dans les contextes XSLT et dans les expressions XQuery :

- Date/Heure
- Géolocalisation
- Liée à l'image
- Numérique
- Séquence
- Chaîne
- Divers

Fonctions Graphiques (Éditions Enterprise et Server uniquement seulement)

Fonctions d'extension Altova pour les graphiques sont prises en charge uniquement par les édition Enterprise et Server des produits Altova et permettent de générer des graphiques d'être généré à partir de données XML.

30.2.1.1 Fonctions XSLT

Les fonctions d'extension XSLT peuvent être utilisées dans les expressions XPath dans un contexte XSLT. Elles ne fonctionneront pas dans un contexte non-XSLT (par exemple dans un contexte XQuery).

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

Fonctions standard

distinct-nodes [altova:]

altova:distinct-nodes(*node()**) **asnode()*** XSLT1 XSLT2 XSLT3

Prend un ensemble d'un ou de plusieurs nœuds en tant que son entrée et retourne le même ensemble moins les nœuds avec des valeurs dupliquées. La comparaison s'effectue en utilisant la fonction XPath/XQuery `fn:deep-equal`.

Exemples

- **altova:distinct-nodes**(country) retourne tous les nœuds `country` enfant moins ceux possédant des valeurs dupliquées.

evaluate [altova:]

altova:evaluate(XPathExpression *as xs:string*[, **ValueOf$p1, ... ValueOf$pN]) XSLT1 XSLT2 XSLT3**

Prend une expression XPath, passée en tant que chaîne, en tant que son argument obligatoire. Elle retourne la sortie de l'expression évaluée. Par exemple : **altova:evaluate('//Name[1]')** retourne les contenues du premier élément `Name` dans le document. Veuillez noter que l'expression `//Name[1]` est passée en tant que chaîne en l'enfermant dans des guillemets simples.

La fonction `altova:evaluate` peut prendre des arguments supplémentaires en option. Ces arguments sont les valeurs des variables in-scope qui portent les noms `p1`, `p2`, `p3`... `pN`. Veuillez noter les points suivants concernant l'utilisation : (i) Les variables doivent être définies avec les noms de la formule `pX`, lorsque `X` est un entier ; (ii) les arguments de la fonction `altova:evaluate` (*voir signature ci-dessus*), à partir du deuxième argument, fournissent les valeurs de la variables, avec la séquence des arguments correspondant à la séquence des variables classées numériquement : `p1` à `pN`: le deuxième argument sera la valeur de la variable `p1`, le troisième argument celui de la variable `p2`, etc. ; (iii) Les valeurs de variable doivent être de type `item*`.

© 2016-2022 Altova GmbH

Exemple

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 'hi')" />
```

outputs `"hi 20 10"`

Dans les listes ci-dessus, veuillez noter les points suivants :

- Le deuxième argument de l'expression `altova:evaluate` est la valeur attribuée à la variable `$p1`, le troisième argument est celui attribué à la variable `$p2`, etc.
- Veuillez noter que le quatrième argument de la fonction est une valeur de chaîne, ce qui est indiqué par le fait qu'elle est contenue dans des guillemets.
- L'attribut `select` de l'élément `xs:variable` fournit l'expression XPath. Puisque cette expression doit être de type `xs:string`, elle est contenue dans des guillemets simples.

Exemples pour mieux illustrer l'utilisation des variables

- ```
  <xsl:variable name="xpath" select="'$p1'" />
  <xsl:value-of select="altova:evaluate($xpath, //Name[1])" />
  ```
  Sort la valeur du premier élément `Name`.

- ```
  <xsl:variable name="xpath" select="'$p1'" />
  <xsl:value-of select="altova:evaluate($xpath, '//Name[1]')" />
  ```
  Sort `"//Name[1]"`

La fonction d'extension `altova:evaluate()` est utile lorsqu'une expression XPath dans la feuille de style XSLT contient une ou plusieurs parties qui doivent être évaluées dynamiquement. Par exemple, prenez comme exemple une situation dans laquelle un utilisateur saisit sa requête pour le critère de tri et le critère est stocké dans l'attribut `UserReq/@sortkey`. Dans la feuille de style, vous pouvez ensuite avoir l'expression : `<xsl:sort select="altova:evaluate(../UserReq/@sortkey)" order="ascending"/>`. La fonction `altova:evaluate()` lit l'attribut `sortkey` de l'élément enfant `UserReq` du parent du nœud contextuel. Si, par exemple, la valeur de l'attribut `sortkey` est `Price`, alors `Price` est retourné par la fonction `altova:evaluate()` et devient la valeur de l'attribut `select` : `<xsl:sort select="Price" order="ascending"/>`. Si cette instruction `sort` apparaît dans le contexte d'un élément appelé `Order`, alors les éléments `Order` seront triés conformément aux valeurs de leurs enfants `Price`. En alternative, si la valeur de `@sortkey` était, par exemple, `Date`, alors les éléments `Order` seraient triés selon les valeurs de leurs enfants `Date`. Donc le critère de triage pour `Order` est choisi à partir de l'attribut `sortkey` lors de l'exécution. Cela n'aurait pas pu se réaliser avec une expression telle que : `<xsl:sort select="../UserReq/@sortkey" order="ascending"/>`. Dans le cas montré ci-dessus, le critère de tri aurait été l'attribut `sortkey` lui-même, et non pas `Price` ou `Date` (ou tout autre contenu actuel de `sortkey`).

Note : Le contexte statique inclut des espaces de nom, des types et des fonctions, mais pas des variables, depuis l'environnement d'appel. L'URI de base et l'espace de nom par défaut sont hérités.

Plus d'exemples

- Variables statiques : `<xsl:value-of select="$i3, $i2, $i1" />`
  Sort les valeurs des trois variables.

- Expression XPath dynamique avec des variables dynamiques :

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 30)" />
```
Sortie "30 20 10"

- Expression XPath dynamique sans variable dynamique :
```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath)" />
```
Sortie erreur : Aucune variable définie pour $p3.

encode-for-rtf [altova:]

altova:encode-for-rtf(input *as xs:string*, **preserveallwhitespace** *as xs:boolean*, **preservenewlines** *as xs:boolean*) **asxs:string** XSLT2 XSLT3

Convertit la chaîne d'entrée en tant que code pour RTF. Les espaces blancs et les nouvelles lignes seront préservés selon la valeur booléenne spécifiée pour leurs arguments respectifs.

[Haut]

Fonctions XBRL

Les fonctions XBRL Altova peuvent uniquement être utilisées avec des éditions des produits Altova qui présentent une prise en charge XBRL.

xbrl-footnotes [altova:]

altova:xbrl-footnotes(*node()***)** **asnode()*** XSLT2 XSLT3

Prend un nœud en tant que son argument d'entrée et retourne l'ensemble des nœuds de notes de pieds XBRL référencées par le nœud d'entrée.

xbrl-labels [altova:]

altova:xbrl-labels(*xs:QName*, *xs:string***)** **asnode()*** XSLT2 XSLT3

Prend deux arguments d'entrée : un nom de nœud et l'emplacement de fichier de taxonomie contenant le nœud. La fonction retourne les nœuds de libellés XBRL associés avec le nœud d'entrée.

[Haut]

30.2.1.2 Fonctions XPath/XQuery : Date et heure

Les fonctions d'extension date/heure d'Altova peuvent être utilisées dans les expressions XPath et XQuery et fournissent des fonctions supplémentaires pour le traitement des données contenues en tant que les types de données de date et d'heures variés de XML Schema. Les fonctions dans cette section peuvent être utilisées avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova. Ils sont disponibles dans les contextes XPath/XQuery.

© 2016-2022 Altova GmbH

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

▼ Regroupées selon les fonctionnalités

- Ajouter une durée à xs:dateTime et retourner xs:dateTime
- Ajouter une durée à xs:date et retourner xs:date
- Ajouter une durée à xs:time et retourner à xs:time
- Formater et récupérer des durées
- Supprimer des fuseaux horaires de fonctions qui génèrent des date/heures actuels
- Retourne un jour de la semaine en tant qu'un entier à partir de la date
- Retourne le numéro de la semaine en tant qu'un entier à partir de la date
- Construire le type de date, d'heure ou de durée à partir des composants lexicaux de chaque type
- Construire le type de date, dateHeure ou heure à partir de l'entrée de chaîne
- Fonctions liées à l'âge
- Fonctions Epoch time (heure Unix)

▼ Recensées par ordre alphabétique

altova:add-days-to-date
altova:add-days-to-dateTime
altova:add-hours-to-dateTime
altova:add-hours-to-time
altova:add-minutes-to-dateTime
altova:add-minutes-to-time
altova:add-months-to-date
altova:add-months-to-dateTime
altova:add-seconds-to-dateTime
altova:add-seconds-to-time
altova:add-years-to-date
altova:add-years-to-dateTime
altova:age
altova:age-details
altova:build-date
altova:build-duration
altova:build-time

altova:current-dateTime-no-TZ
altova:current-date-no-TZ
altova:current-time-no-TZ
altova:date-no-TZ
altova:dateTime-from-epoch
altova:dateTime-from-epoch-no-TZ
altova:dateTime-no-TZ
altova:days-in-month
altova:epoch-from-dateTime
altova:hours-from-dateTimeDuration-accumulated
altova:minutes-from-dateTimeDuration-accumulated
altova:seconds-from-dateTimeDuration-accumulated
altova:format-duration
altova:parse-date
altova:parse-dateTime
altova:parse-duration
altova:parse-time
altova:time-no-TZ
altova:weekday-from-date
altova:weekday-from-dateTime
altova:weeknumber-from-date
altova:weeknumber-from-dateTime

[Haut]

Ajouter une durée à xs:dateTime XP3.1 XQ3.1

Ces fonctions ajoutent une durée à **xs:dateTime** et retournent **xs:dateTime**. Le type xs:dateTime a un format de CCYY-MM-DDThh:mm:ss.sss. Il s'agit d'une concaténation des formats xs:date et xs:time séparés par la lettre T. Un suffixe de fuseau horaire +01:00 (par exemple) (**+01:00**, par exemple) est optionnel.

add-years-to-dateTime [altova:]

altova:add-years-to-dateTime(DateTime *as xs:dateTime*, **Years** *as xs:integer***)**
asxs:dateTime XP3.1 XQ3.1

Ajoute une durée en années à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre d'années à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

Exemples

- **altova:add-years-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) retourne 2024-01-15T14:00:00
- **altova:add-years-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -4) retourne 2010-01-15T14:00:00

add-months-to-dateTime [altova:]

altova:add-months-to-dateTime(DateTime *as xs:dateTime*, **Months** *as xs:integer***)**
asxs:dateTime XP3.1 XQ3.1

Ajoute une durée en mois à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre de mois à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

© 2016-2022 Altova GmbH

Exemples

- **altova:add-months-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) retourne 2014-11-15T14:00:00
- **altova:add-months-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -2) retourne 2013-11-15T14:00:00

add-days-to-dateTime [altova:]

altova:add-days-to-dateTime(DateTime *as xs:dateTime*, **Days** *as xs:integer***) asxs:dateTime** **XP3.1** **XQ3.1**

Ajoute une durée en jours à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre de jours à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

Exemples

- **altova:add-days-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) retourne 2014-01-25T14:00:00
- **altova:add-days-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -8) retourne 2014-01-07T14:00:00

add-hours-to-dateTime [altova:]

altova:add-hours-to-dateTime(DateTime *as xs:dateTime*, **Hours** *as xs:integer***) asxs:dateTime** **XP3.1** **XQ3.1**

Ajoute une durée en heures à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre d'heures à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

Exemples

- **altova:add-hours-to-dateTime**(xs:dateTime("2014-01-15T13:00:00"), 10) retourne 2014-01-15T23:00:00
- **altova:add-hours-to-dateTime**(xs:dateTime("2014-01-15T13:00:00"), -8) retourne 2014-01-15T05:00:00

add-minutes-to-dateTime [altova:]

altova:add-minutes-to-dateTime(DateTime *as xs:dateTime*, **Minutes** *as xs:integer***) asxs:dateTime** **XP3.1** **XQ3.1**

Ajoute une durée en minutes à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre of minutes à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

Exemples

- **altova:add-minutes-to-dateTime**(xs:dateTime("2014-01-15T14:10:00"), 45) retourne 2014-01-15T14:55:00
- **altova:add-minutes-to-dateTime**(xs:dateTime("2014-01-15T14:10:00"), -5) retourne 2014-01-15T14:05:00

add-seconds-to-dateTime [altova:]

altova:add-seconds-to-dateTime(DateTime *as xs:dateTime*, **Seconds** *as xs:integer*)
asxs:dateTime XP3.1 XQ3.1
Ajoute une durée en secondes à `xs:dateTime` (*voir exemples ci-dessous*). Le deuxième argument est le nombre de secondes à être ajouté à `xs:dateTime` fourni en tant que le premier argument. Le résultat est de type `xs:dateTime`.

Exemples

- **altova:add-seconds-to-dateTime**(xs:dateTime("2014-01-15T14:00:10"), 20) retourne 2014-01-15T14:00:30
- **altova:add-seconds-to-dateTime**(xs:dateTime("2014-01-15T14:00:10"), -5) retourne 2014-01-15T14:00:05

[Haut]

Ajouter une durée à xs:date XP3.1 XQ3.1

Ces fonctions ajoutent une durée à **xs:date** et retournent **xs:date**. Le type `xs:date` a un format `CCYY-MM-DD`.

add-years-to-date [altova:]

altova:add-years-to-date(Date *as xs:date*, **Years** *as xs:integer*) **asxs:date** XP3.1 XQ3.1
Ajoute une durée en années à une date. Le deuxième argument est le nombre d'années à être ajouté à `xs:date` fourni en tant que le premier argument. Le résultat est de type `xs:date`.

Exemples

- **altova:add-years-to-date**(xs:date("2014-01-15"), 10) retourne 2024-01-15
- **altova:add-years-to-date**(xs:date("2014-01-15"), -4) retourne 2010-01-15

add-months-to-date [altova:]

altova:add-months-to-date(Date *as xs:date*, **Months** *as xs:integer*) **asxs:date** XP3.1 XQ3.1
Ajoute une durée en mois à une date. Le deuxième argument est le nombre de mois à être ajouté à `xs:date` fourni en tant que le premier argument. Le résultat est de type `xs:date`.

Exemples

- **altova:add-months-to-date**(xs:date("2014-01-15"), 10) retourne 2014-11-15
- **altova:add-months-to-date**(xs:date("2014-01-15"), -2) retourne 2013-11-15

add-days-to-date [altova:]

altova:add-days-to-date(Date *as xs:date*, **Days** *as xs:integer*) **asxs:date** XP3.1 XQ3.1
Ajoute une durée en jours à une date. Le deuxième argument est le nombre de jours à être ajouté à `xs:date` fourni en tant que le premier argument. Le résultat est de type `xs:date`.

Exemples

- **altova:add-days-to-date**(xs:date("2014-01-15"), 10) retourne 2014-01-25
- **altova:add-days-to-date**(xs:date("2014-01-15"), -8) retourne 2014-01-07

© 2016-2022 Altova GmbH

[Haut]

Formater et récupérer des durées XP3.1 XQ3.1

Ces fonctions parsent une entrée **xs:duration** ou **xs:string** et retournent respectivement un **xs:string** ou **xs:duration**.

format-duration [altova:]

altova:format-duration(Duration *as xs:duration*, **Picture** *as xs:string*) **asxs:string** XP3.1 XQ3.1

Formate une durée qui est soumise en tant que le premier argument, selon une chaîne d'image soumise en tant que le second argument. La sortie est une chaîne de texte formatée conformément à la chaîne d'image.

Exemples

- **altova:format-duration**(xs:duration("P2DT2H53M11.7S"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") retourne "Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"
- **altova:format-duration**(xs:duration("P3M2DT2H53M11.7S"), "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") retourne "Months:03 Days:02 Hours:02 Minutes:53"

parse-duration [altova:]

altova:parse-duration(InputString *as xs:string*, **Picture** *as xs:string*) **asxs:duration** XP3.1 XQ3.1

Prend un patterned string en tant que le premier argument et une chaîne image en tant que le second argument. La chaîne d'entrée est parsée sur la base de la chaîne d'image et une `xs:duration` est retournée.

Exemples

- **altova:parse-duration**("Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") retourne "P2DT2H53M11.7S"
- **altova:parse-duration**("Months:03 Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7", "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") retourne "P3M2DT2H53M"

[Haut]

Ajouter une durée à xs:time XP3.1 XQ3.1

Ces fonctions ajoutent une durée à **xs:time** et retournent **xs:time**. Le type `xs:time` a une forme lexicale de `hh:mm:ss.sss`. Un fuseau horaire en option peut être suffixé. La lettre `Z` indique le Temps universel coordonné (UTC). Tous les autres fuseaux horaires sont représentés par leur différence de l'UTC dans le format `+hh:mm`, ou `-hh:mm`. Si aucune valeur de fuseau horaire n'est présente, elle est considérée inconnue ; elle n'est pas considérée être UTC.

add-hours-to-time [altova:]

altova:add-hours-to-time(Time *as xs:time*, **Hours** *as xs:integer*) **as xs:time** XP3.1 XQ3.1
Ajoute une durée en heures à une heure de temps. Le deuxième argument est le nombre d'heures à être ajouté à `xs:time` fourni en tant que le premier argument. Le résultat est de type `xs:time`.

Exemples

- **altova:add-hours-to-time**(xs:time("11:00:00"), 10) retourne 21:00:00
- **altova:add-hours-to-time**(xs:time("11:00:00"), -7) retourne 04:00:00

add-minutes-to-time [altova:]

altova:add-minutes-to-time(Time *as xs:time*, **Minutes** *as xs:integer*) **as xs:time** XP3.1 XQ3.1
Ajoute une durée en minutes à une heure. Le deuxième argument est le nombre de minutes à être ajouté à `xs:time` fourni en tant que le premier argument. Le résultat est de type `xs:time`.

Exemples

- **altova:add-minutes-to-time**(xs:time("14:10:00"), 45) retourne 14:55:00
- **altova:add-minutes-to-time**(xs:time("14:10:00"), -5) retourne 14:05:00

add-seconds-to-time [altova:]

altova:add-seconds-to-time(Time *as xs:time*, **Minutes** *as xs:integer*) **as xs:time** XP3.1 XQ3.1
Ajoute une durée en secondes à une heure. Le deuxième argument est le nombre de secondes à être ajouté à `xs:time` fourni en tant que le premier argument. Le résultat est de type `xs:time`. Le composant Secondes peut être contenu dans une plage de `0` à `59.999`.

Exemples

- **altova:add-seconds-to-time**(xs:time("14:00:00"), 20) retourne 14:00:20
- **altova:add-seconds-to-time**(xs:time("14:00:00"), 20.895) retourne 14:00:20.895

[Haut]

Supprimer la partie du fuseau horaire des types de données date/heures XP3.1 XQ3.1

Ces fonctions permettent de supprimer le fuseau horaire des valeurs **xs:dateTime**, **xs:date** ou **xs:time** actuelles, respectivement. Veuillez noter que la différence entre `xs:dateTime` et `xs:dateTimeStamp` est que dans le cas de ce dernier, la partie fuseau horaire est requise (alors qu'elle est optionnelle dans le premier des deux cas). Donc, le format d'une valeur `xs:dateTimeStamp` est : `CCYY-MM-DDThh:mm:ss.sss±hh:mm`. ou `CCYY-MM-DDThh:mm:ss.sssZ`. Si la date et l'heure sont lues depuis l'horloge du système, en tant que `xs:dateTimeStamp`, la fonction `current-dateTime-no-TZ()` peut être utilisée pour supprimer le fuseau horaire s'il est requis.

current-date-no-TZ [altova:]

altova:current-date-no-TZ() as xs:date XP3.1 XQ3.1
Cette fonction ne prend aucun argument. Elle supprime la partie fuseau horaire de `current-date()` (qui est la date actuelle selon l'horloge système) et retourne une valeur `xs:date`.

Exemples

Si la date actuelle est **2014-01-15+01:00**:

© 2016-2022 Altova GmbH

- **altova:current-date-no-TZ()** retourne 2014-01-15

current-dateTime-no-TZ [altova:]

altova:current-dateTime-no-TZ() asxs:dateTime XP3.1 XQ3.1
Cette fonction ne prend aucun argument. Elle supprime la partie fuseau horaire de current-dateTime() (qui est la date-heure actuelle selon l'horloge système) et retourne une valeurxs:dateTime.

Exemples

Si la date-heure actuelle est **2014-01-15T14:00:00+01:00**:

- **altova:current-dateTime-no-TZ()** retourne 2014-01-15T14:00:00

current-time-no-TZ [altova:]

altova:current-time-no-TZ() asxs:time XP3.1 XQ3.1
Cette fonction ne prend aucun argument. Elle supprime la partie fuseau horaire de current-time() (qui est l'heure actuelle selon l'horloge système) et retourne une valeur xs:time.

Exemples

Si l'heure actuelle est **14:00:00+01:00**:

- **altova:current-time-no-TZ()** retourne 14:00:00

date-no-TZ [altova:]

altova:date-no-TZ(InputDate *as xs:date*) **asxs:date XP3.1 XQ3.1**
Cette fonction prend un argument xs:date, en supprime la partie fuseau horaire et retourne une valeur xs:date. Veuillez noter que la date n'est pas modifiée.

Exemples

- **altova:date-no-TZ**(xs:date("2014-01-15+01:00")) retourne 2014-01-15

dateTime-no-TZ [altova:]

altova:dateTime-no-TZ(InputDateTime *as xs:dateTime*) **asxs:dateTime XP3.1 XQ3.1**
Cette fonction prend un argument xs:dateTime, en supprime la partie fuseau horaire, et retourne une valeur xs:dateTime. Veuillez noter que ni la date ni l'heure n'est modifiée.

Exemples

- **altova:dateTime-no-TZ**(xs:date("2014-01-15T14:00:00+01:00")) retourne 2014-01-15T14:00:00

time-no-TZ [altova:]

altova:time-no-TZ(InputTime *as xs:time*) **asxs:time XP3.1 XQ3.1**
Cette fonction prend un argument xs:time, en supprime la partie de fuseau horaire, et retourne une valeur xs:time. Veuillez noter que l'heure n'est pas modifiée.

Exemples

- **altova:time-no-TZ**(xs:time("14:00:00+01:00")) retourne 14:00:00

[Haut]

Retourne le nombre de jours, d'heures, de minutes, de secondes des durées XP3.1 XQ3.1

Ces fonctions retournent le nombre de jours dans un mois, et le nombre d'heures, de minutes et de secondes, respectivement depuis les durées.

days-in-month [altova:]

altova:days-in-month(Year *as xs:integer*, **Month** *as xs:integer*) **asxs:integer** XP3.1 XQ3.1
Retourne le nombre de jours dans le mois spécifié. Le mois est spécifié avec les arguments `Year` et `Month`.

Exemples

- **altova:days-in-month**(2018, 10) retourne 31
- **altova:days-in-month**(2018, 2) retourne 28
- **altova:days-in-month**(2020, 2) retourne 29

hours-from-dayTimeDuration-accumulated

altova:hours-from-dayTimeDuration-accumulated(DayAndTime *as xs:duration*) **asxs:integer** XP3.1 XQ3.1
Retourne le nombre total d'heures dans la durée soumise par l'argument `DayAndTime` (qui est de type `xs:duration`). Les heures dans les composants `Day` et `Time` sont additionnés pour donner un résultat qui est un entier. Le décompte d'une nouvelle heure dure uniquement 60 minutes complètes. Des durées négatives entraînent une valeur d'heures négative.

Exemples

- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5D")) retourne 120, qui est le nombre total d'heures dans 5 jours.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H")) retourne 122, qui est le nombre total d'heures dans 5 jours plus 2 heures.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H60M")) retourne 123, qui est le nombre total d'heures dans 5 jours plus 2 heures et 60 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H119M")) retourne 123, qui est le nombre total d'heures dans 5 jours plus 2 heures et 119 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H120M")) retourne 124,qui est le nombre total d'heures dans 5 jours plus 2 heures et 120 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("-P5DT2H")) retourne -122

minutes-from-dayTimeDuration-accumulated

altova:minutes-from-dayTimeDuration-accumulated(DayAndTime *as xs:duration*) **asxs:integer** XP3.1 XQ3.1
Retourne le nombre total de minutes dans la durée soumise par l'argument `DayAndTime` (qui est de type `xs:duration`). Les minutes dans les composants `Day` et `Time` sont additionnés pour donner un résultat qui est un entier. Des durées négatives entraînent une valeur de minute négative.

Exemples

© 2016-2022 Altova GmbH

- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT60M")) retourne 60
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT1H")) retourne 60, qui est le nombre total de minutes dans une heure.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT1H40M")) retourne 100
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("P1D")) retourne 1440, qui est le nombre total de minutes dans un jour.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("-P1DT60M")) retourne -1500

seconds-from-dayTimeDuration-accumulated

altova:seconds-from-dayTimeDuration-accumulated(DayAndTime *as xs:duration*) **asxs:integer** **XP3.1** **XQ3.1**

Retourne le nombre total de minutes dans la durée soumise par l'argument DayAndTime (qui est de type xs:duration). Les minutes dans les composants Day et Time sont additionnés pour donner un résultat qui est un entier. Des durées négatives entraînent une valeur de minute négative .

Exemples

- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1M")) retourne 60, qui est le nombre total de secondes dans une minute.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1H")) retourne 3600, qui est le nombre total de secondes dans une heure.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1H2M")) retourne 3720
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("P1D")) retourne 86400, qui est le nombre total de secondes dans un jour.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("-P1DT1M")) retourne -86460

Retourne le jour de la semaine à partir de xs:dateTime ou xs:date **XP3.1** **XQ3.1**

Ces fonctions retournent le jour de la semaine (en tant qu'entier) depuis xs:dateTime ou xs:date. Les jours de la semaine sont numérotés (format américain) de 1 à 7, avec Sunday=1. Dans le format européen, la semaine commence par Lundi (=1). Dans le format américain elle commence par Sunday=1. Configurer en utilisant l'entier 0 et où un entier est accepté pour indiquer le format.

weekday-from-dateTime [altova:]

altova:weekday-from-dateTime(DateTime *as xs:dateTime*) **asxs:integer** **XP3.1** **XQ3.1**

Prend une date-avec-heure en tant que son seul argument et retourne le jour de la semaine de cette date sous forme d'un entier. Les jours de la semaine sont numérotés en commençant avec Sunday=1. Si le format européen est requis (où Monday=1), utiliser l'autre signature de cette fonction (*voir signature suivante ci-dessous*).

Exemples

- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00")) retourne 2, ce qui indique un lundi.

altova:weekday-from-dateTime(DateTime *as xs:dateTime*, **Format** *as xs:integer***)**
asxs:integer XP3.1 XQ3.1
Prend une date-avec-heure en tant que son premier argument et retourne le jour de la semaine de cette date sous forme d'un entier. Si le second argument (entier) est `0`, les jours de la semaine sont numérotés de `1` à `7` en commençant avec `Sunday=1`. Si le second argument est un entier différent de `0`, alors `Monday=1`. S'il n'y a pas de second argument, la fonction est lue comme possédant l'autre signature de cette fonction (*voir signature précédente*).

Exemples

- `altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00"), 1)` retourne `1`, ce qui indique un lundi
- `altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00"), 4)` retourne `1`, ce qui indique un lundi
- `altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00"), 0)` retourne `2`, ce qui indique un lundi.

weekday-from-date [altova:]

altova:weekday-from-date(Date *as xs:date***) asxs:integer** XP3.1 XQ3.1
Prend une date en tant que son seul argument et retourne le jour de la semaine de cette date sous forme d'un entier. Les jours de la semaine sont numérotés en commençant avec `Sunday=1`. Si le format européen est requis (où `Monday=1`), utiliser l'autre signature de cette fonction (*voir signature suivante ci-dessous*).

Exemples

- `altova:weekday-from-date(xs:date("2014-02-03+01:00"))` retourne `2`, ce qui indique un lundi.

altova:weekday-from-date(Date *as xs:date*, **Format** *as xs:integer***) asxs:integer** XP3.1 XQ3.1
Prend une date en tant que son premier argument et retourne le jour de la semaine de cette date sous forme d'un entier. Si le second argument (`Format`) est `0`, les jours de la semaine sont numérotés de `1` à `7` en commençant avec `Sunday=1`. Si le second argument est un entier différent de `0`, alors `Monday=1`. S'il n'y a pas de second argument, la fonction est lue comme possédant l'autre signature de cette fonction (*voir signature précédente*).

Exemples

- `altova:weekday-from-date(xs:date("2014-02-03"), 1)` retourne `1`, ce qui indique un lundi
- `altova:weekday-from-date(xs:date("2014-02-03"), 4)` retourne `1`, ce qui indique un lundi
- `altova:weekday-from-date(xs:date("2014-02-03"), 0)` retourne `2`, ce qui indique un lundi.

[Haut]

Retourne le nombre de la semaine à partir de xs:dateTime ou xs:date XP2 XQ1 XP3.1 XQ3.1

Ces fonctions retournent le numéro de la semaine (en tant qu'un entier) depuis `xs:dateTime` or `xs:date`. La numérotation des semaines est disponible dans les formats de calendrier US, ISO/European et Islamiques. La numérotation des semaines est différente dans ces formats de calendrier parce que la semaine est considérée démarrer avec un jour différent selon le format (dimanche pour le format US, lundi pour le format ISO/Européen, et samedi dans le format islamique).

© 2016-2022 Altova GmbH

weeknumber-from-date [altova:]

altova:weeknumber-from-date(Date *as xs:date*, **Calendar** *as xs:integer*) **asxs:integer** **XP2** **XQ1** **XP3.1** **XQ3.1**

Retourne le numéro de la semaine de l'argument **Date** soumis en tant qu'entier. Le deuxième argument (**Calendar**) spécifie le système de calendrier à suivre.
Les valeurs de **Calendar** prises en charge sont :

- **0 = US calendar** (*semaine commence dimanche*)
- **1 = ISO standard, European calendar** (*semaine commence lundi*)
- **2 = Islamic calendar** (*semaine commence samedi*)

Le réglage par défaut est **0**.

Exemples

- **altova:weeknumber-from-date**(xs:date("2014-03-23"), 0) retourne 13
- **altova:weeknumber-from-date**(xs:date("2014-03-23"), 1) retourne 12
- **altova:weeknumber-from-date**(xs:date("2014-03-23"), 2) retourne 13
- **altova:weeknumber-from-date**(xs:date("2014-03-23")) retourne 13

Le jour de la date dans les exemples ci-dessus (2014-03-23) est dimanche. Les calendriers US et musulmans sont donc une semaine en avant par rapport au calendrier européen à ce jour.

weeknumber-from-dateTime [altova:]

altova:weeknumber-from-dateTime(DateTime *as xs:dateTime*, **Calendar** *as xs:integer*) **asxs:integer** **XP2** **XQ1** **XP3.1** **XQ3.1**

Retourne le numéro de la semaine de l'argument **DateTime** soumis en tant qu'entier. Le deuxième argument (**Calendar**) spécifie le système de calendrier à suivre.
Les valeurs de **Calendar** prises en charge sont :

- **0 = US calendar** (*semaine commence dimanche*)
- **1 = ISO standard, European calendar** (*semaine commence lundi*)
- **2 = Islamic calendar** (*semaine commence samedi*)

Le réglage par défaut est **0**.

Exemples

- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00"), 0) retourne 13
- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00"), 1) retourne 12
- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00"), 2) retourne 13
- **altova:weeknumber-from-dateTime**(xs:dateTime("2014-03-23T00:00:00")) retourne 13

Le jour du dateTime dans les exemples ci-dessus (2014-03-23T00:00:00) est dimanche. Les calendriers US et musulmans sont donc une semaine en avant par rapport au calendrier européen à ce jour.

[Haut]

Construire le type de date, d'heure ou de durée à partir de leurs composants lexicaux XP3.1 XQ3.1

Les fonctions prennent les composants lexicaux du type de données `xs:date`, `xs:time` ou `xs:duration` en tant qu'arguments d'entrée et les combinent pour construire le type de données respectif.

build-date [altova:]

altova:build-date(Year *as xs:integer*, **Month** *as xs:integer*, **Date** *as xs:integer*) **asxs:date** XP3.1 XQ3.1

Les premier, second et troisième arguments sont respectivement l'année, le mois et la date. Ils sont combinés pour construire une valeur de type `xs:date`. Les valeurs de l'entier doivent se situer dans le cadre de la plage correcte de cette partie de la date. Par exemple, le deuxième argument (pour la partie du mois) ne devrait pas être supérieur à `12`.

Exemples

- **altova:build-date**(2014, 2, 03) retourne 2014-02-03

build-time [altova:]

altova:build-time(Hours *as xs:integer*, **Minutes** *as xs:integer*, **Seconds** *as xs:integer*) **asxs:time** XP3.1 XQ3.1

Les premiers, seconds et troisièmes arguments sont, respectivement, les valeurs d'heure (`0` to `23`), de minutes (`0` to `59`) et de secondes (`0` to `59`). Ils sont combinés pour construire une valeur de type `xs:time`. Les valeurs des entiers doivent se trouver dans le cadre de la plage correcte de cette partir de temps en particulier. Par exemple, le deuxième argument (`Minutes`) ne devrait pas être supérieur à `59`. Pour ajouter une partie fuseau horaire à la valeur, utiliser l'autre signature de cette fonction (*voir signature suivante*).

Exemples

- **altova:build-time**(23, 4, 57) retourne 23:04:57

altova:build-time(Hours *as xs:integer*, **Minutes** *as xs:integer*, **Seconds** *as xs:integer*, **TimeZone** *as xs:string*) **asxs:time** XP3.1 XQ3.1

Les premiers, seconds et troisièmes arguments sont, respectivement, les valeurs d'heure (`0` to `23`), de minutes (`0` to `59`) et de secondes (`0` to `59`). Le quatrième argument est une chaîne qui fournit la partie fuseau horaire de la valeur. Les quatre arguments sont combinés pour construire une valeur de type `xs:time`. Les valeurs des entiers doivent se trouver dans le cadre de la plage correcte de cette partie de temps en particulier. Par exemple, le deuxième argument (`Minutes`) ne doit pas être supérieur à `59`.

Exemples

- **altova:build-time**(23, 4, 57, '+1') retourne 23:04:57+01:00

build-duration [altova:]

altova:build-duration(Years *as xs:integer*, **Months** *as xs:integer*) **asxs:yearMonthDuration** XP3.1 XQ3.1

Prend deux arguments pour construire une valeur de type `xs:yearMonthDuration`. Les premiers arguments fournissent la partie `Years` de la valeur de durée, alors que le deuxième argument fournie la partie `Months`. Si le deuxième argument (`Months`) est supérieur ou égale à `12`, alors l'entier est divisé par `12`; le quotient est ajouté au premier argument pour fournir la partie `Years` de la valeur de durée alors que le reste (de la division) fournit la partie `Months`. Pour construire une durée de type `xs:dayTimeDuration`.,

© 2016-2022 Altova GmbH

voir la signature suivante.

Exemples

- `altova:build-duration(2, 10)` retourne `P2Y10M`
- `altova:build-duration(14, 27)` retourne `P16Y3M`
- `altova:build-duration(2, 24)` retourne `P4Y`

altova:build-duration(Days *as xs:integer*, **Hours** *as xs:integer*, **Minutes** *as xs:integer*, **Seconds** *as xs:integer*) as **xs:dayTimeDuration** **XP3.1** **XQ3.1**

Prend quatre arguments et les combine pour construire une valeur de type `xs:dayTimeDuration`. Le premier argument fournit la partie `Days` de la valeur de durée, le deuxième, troisième et quatrième argument fournit respectivement les parties `Hours`, `Minutes` et `Seconds` de la valeur de durée. Chacun des trois arguments Time est converti en une valeur équivalente en termes de l'unité suivante plus élevée et le résultat est utilisé pour le calcul d'une valeur de durée totale. Par exemple, 72 secondes est converti en `1M+12S` (1 minute et 12 secondes), et cette valeur est utilisée pour le calcul de la valeur de durée totale. Pour construire une durée de type `xs:yearMonthDuration`., voir la signature précédente.

Exemples

- `altova:build-duration(2, 10, 3, 56)` retourne `P2DT10H3M56S`
- `altova:build-duration(1, 0, 100, 0)` retourne `P1DT1H40M`
- `altova:build-duration(1, 0, 0, 3600)` retourne `P1DT1H`

[Haut]

Construire le type de date, dateHeure ou heure à partir de l'entrée de chaîne XP2 XQ1 XP3.1 XQ3.1

Ces fonctions prennent des chaînes en tant qu'arguments et construisent des types de données `xs:date`, `xs:dateTime`, ou `xs:time`. La chaîne est analysée pour les composants du type de données basé sur un argument de modèle soumis.

parse-date [altova:]

altova:parse-date(Date *as xs:string*, **DatePattern** *as xs:string*) as**xs:date** **XP2** **XQ1** **XP3.1** **XQ3.1**

Retourne la chaîne d'entrée **Date** en tant qu'une valeur **xs:date**. Le deuxième argument **DatePattern** spécifie le modèle (séquence des composants) de la chaîne d'entrée. **DatePattern** est décrit avec les spécificateurs de composants regroupés ci-dessous et avec les séparateurs de composant qui peuvent être n'importe quel caractère. Voir les exemples ci-dessous.

D	Jour
M	Mois
Y	Année

Le modèle dans **DatePattern** doit correspondre au modèle dans **Date**. Puisque la sortie est de type **xs:date**, la sortie aura toujours le format lexical **YYYY-MM-DD**.

Exemples

- `altova:parse-date(xs:string("09-12-2014"), "[D]-[M]-[Y]")` retourne `2014-12-09`
- `altova:parse-date(xs:string("09-12-2014"), "[M]-[D]-[Y]")` retourne `2014-09-12`
- `altova:parse-date("06/03/2014", "[M]/[D]/[Y]")` retourne `2014-06-03`

- `altova:parse-date("06 03 2014", "[M] [D] [Y]")` retourne `2014-06-03`
- `altova:parse-date("6 3 2014", "[M] [D] [Y]")` retourne `2014-06-03`

parse-dateTime [altova:]

altova:parse-dateTime(DateTime *as xs:string*, **DateTimePattern** *as xs:string*)
asxs:dateTime **XP2** **XQ1** **XP3.1** **XQ3.1**

Retourne la chaîne d'entrée **DateTime** en tant que valeur **xs:dateTime**. Le deuxième argument **DateTimePattern** spécife le modèle (séquence des composants) de la chaîne d'entrée. **DateTimePattern** est décrit avec les spécificateurs de composants regroupés ci-dessous et avec les séparateurs de composant qui peuvent être n'importe quel caractère. Voir les exemples ci-dessous.

D	Date
M	Mois
Y	Année
H	Heure
m	Minutes
s	Secondes

Le modèle dans **DateTimePattern** doit correspondre au modèle dans **DateTime**. Puisque la sortie est de type **xs:dateTime**, la sortie aura toujours le format lexical **YYYY-MM-DDTHH:mm:ss**.

Exemples

- **altova:parse-dateTime**(xs:string("09-12-2014 13:56:24"), "[M]-[D]-[Y] [H]:[m]:[s]") retourne `2014-09-12T13:56:24`
- **altova:parse-dateTime**("time=13:56:24; date=09-12-2014", "time=[H]:[m]:[s]; date=[D]-[M]-[Y]") retourne `2014-12-09T13:56:24`

parse-time [altova:]

altova:parse-time(Time *as xs:string*, **TimePattern** *as xs:string*) **asxs:time** **XP2** **XQ1** **XP3.1** **XQ3.1**

Retourne la chaîne d'entrée **Time** en tant qu'une valeur **xs:time**.Le deuxième argument **TimePattern** spécifie le modèle (séquence des composants) de la chaîne d'entrée. **TimePattern** est décrit avec les spécificateurs de composants regroupés ci-dessous et avec les séparateurs de composant qui peuvent être n'importe quel caractère. Voir les exemples ci-dessous.

H	Heure
m	minutes
s	secondes

Le modèle dans **TimePattern** doit correspondre au modèle dans **Time**. Puisque la sortie est de type **xs:time**, la sortie aura toujours le format lexical **HH:mm:ss**.

Exemples

- **altova:parse-time**(xs:string("13:56:24"), "[H]:[m]:[s]") retourne `13:56:24`
- **altova:parse-time**("13-56-24", "[H]-[m]") retourne `13:56:00`

© 2016-2022 Altova GmbH

- **altova:parse-time**("time=13h56m24s", "time=[H]h[m]m[s]s") retourne 13:56:24
- **altova:parse-time**("time=24s56m13h", "time=[s]s[m]m[H]h") retourne 13:56:24

[Top]

Fonctions liées à l'âge XP3.1 XQ3.1

Ces fonctions retournent l'âge tel que calculé (i) entre une date d'argument d'entrée et la date actuelle, ou (ii) entre deux dates d'argument d'entrée. La fonction **altova:age** retourne l'âge en termes d'années, la fonction **altova:age-details** retourne l'âge en tant qu'une séquence de trois entiers indiquant les années, mois et jours de l'âge.

age [altova:]

altova:age(StartDate *as xs:date***) asxs:integer** XP3.1 XQ3.1
Retourne un entier représentant l'âge *en années* d'un objet, en comptant depuis une date de départ soumise en tant que l'argument et se terminant avec la date actuelle (prise depuis l'horloge système). Si l'argument d'entrée est une date supérieure ou égale à une année dans le futur, la valeur de retour sera négative.

Exemples

Si la date actuelle est **2014-01-15** :

- **altova:age**(xs:date("2013-01-15")) retourne 1
- **altova:age**(xs:date("2013-01-16")) retourne 0
- **altova:age**(xs:date("2015-01-15")) retourne -1
- **altova:age**(xs:date("2015-01-14")) retourne 0

altova:age(StartDate *as xs:date*, **EndDate** *as xs:date*) **asxs:integer** XP3.1 XQ3.1
Retourne un entier représentant l'âge *en années* d'un objet, en comptant depuis une date de départ soumise en tant que l'argument jusqu'à une date de fin qui est de deuxième argument. La valeur de retour sera négative si le premier argument est tardif d'une année ou plus que le deuxième argument.

Exemples

Si la date actuelle est **2014-01-15**:

- **altova:age**(xs:date("2000-01-15"), xs:date("2010-01-15")) retourne 10
- **altova:age**(xs:date("2000-01-15"), current-date()) retourne 14 si la date actuelle est 2014-01-15
- **altova:age**(xs:date("2014-01-15"), xs:date("2010-01-15")) retourne -4

age-details [altova:]

altova:age-details(InputDate *as xs:date***)** as (**xs:integer**)* XP3.1 XQ3.1
Retourne trois entiers qui sont respectivement les années, les mois et les jours entre la date soumise en tant que l'argument et la date actuelle (prise depuis l'horloge système). Le résultat de la somme de years+months+days donne le total de la différence de temps entre les deux dates (la date d'entrée et la date actuelle). La date d'entrée peut avoir une valeur précédant ou succédant à la date actuelle mais que la date d'entrée soit précédente ou succédant n'est pas indiqué par le signe des valeurs de retour ; les valeurs de retour sont toujours positives.

Exemples

Si la date actuelle est **2014-01-15**:

- **altova:age-details**(xs:date("2014-01-16")) retourne (0 0 1)
- **altova:age-details**(xs:date("2014-01-14")) retourne (0 0 1)
- **altova:age-details**(xs:date("2013-01-16")) retourne (1 0 1)
- **altova:age-details**(current-date()) retourne (0 0 0)

altova:age-details(Date-1 *as xs:date*, **Date-2** *as xs:date*) as (**xs:integer**)* **XP3.1** **XQ3.1**
Retourne trois entiers qui sont respectivement les années, les mois et les jours entre les deux dates d'argument. Le résultat de la somme de years+months+days donne le total de la différence de temps entre les deux dates d'entrée ; peu importe que la date soit la précédente ou la subséquente des deux dates, elle est soumise en tant que le premier argument. Les valeurs de retour n'indiquent pas si la date d'entrée se produit avant ou après la date actuelle. Les valeurs de retour sont toujours positives.

Exemples

- **altova:age-details**(xs:date("2014-01-16"), xs:date("2014-01-15")) retourne (0 0 1)
- **altova:age-details**(xs:date("2014-01-15"), xs:date("2014-01-16")) retourne (0 0 1)

[Top]

Fonctions Epoch time (heure Unix) XP3.1 XQ3.1

Epoch time est un système horaire utilisé dans les systèmes Unix. Il définit tout moment donné comme étant le nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. Ces fonctions Epoch time convertissent les valeurs **xs:dateTime** en valeurs Epoch time et vice versa.

dateTime-from-epoch [altova:]

altova:dateTime-from-epoch(Epoch *as xs:decimal* **as xs:dateTime** **XP3.1** **XQ3.1**
Epoch time est un système horaire utilisé sur les systèmes Unix. Il définit tout moment donné comme étant l nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. La fonction **dateTime-from-epoch** retourne l'équivalent **xs:dateTime** d'un Epoch time, l'ajuste pour son fuseau horaire local et inclut l'information du fuseau horaire dans le résultat.

La fonction prend un argument xs:decimal et retourne une valeur xs:dateTime qui inclut une partie (fuseau horaire) **TZ**. Le résultat est obtenu en calculant l'équivalent UTC **dateTime** de Epoch time, et en l'ajoutant à son fuseau horaire local (pris de l'horloge système). Par exemple, si la fonction est exécutée sur un appareil qui a été défini pour être dans un fuseau horaire +01:00 (relatif à UTC), après avoir calculé l'équivalent UTC **dateTime**, une heure sera ajoutée au résultat. L'information du fuseau horaire, qui est une partie lexicale optionnelle du résultat **xs:dateTime**, est également rapportée dans le résultat **dateTime**. Comparez ce résultat avec celui de **dateTime-from-epoch-no-TZ**, et consultez également la fonction **epoch-from-dateTime**.

Exemples

Les exemples ci-dessous supposent un fuseau horaire local UTC +01:00. En conséquence, l'équivalent UTC **dateTime** de l'Epoch time soumis sera incrémenté d'une heure. Le fuseau horaire est rapporté dans le résultat.

- **altova:dateTime-from-epoch**(34) retourne 1970-01-01T01:00:34+01:00
- **altova:dateTime-from-epoch**(62) retourne 1970-01-01T01:01:02+01:00

© 2016-2022 Altova GmbH

▼ dateTime-from-epoch [altova:]

altova:dateTime-from-epoch(Epoch *as xs:decimal* **as xs:dateTime** XP3.1 XQ3.1
Epoch time est un système horaire utilisé sur les systèmes Unix. Il définit tout moment donné comme étant l nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. La fonction **dateTime-from-epoch** retourne l'équivalent **xs:dateTime** d'un Epoch time, l'ajuste pour son fuseau horaire local et inclut l'information du fuseau horaire dans le résultat.

La fonction prend un argument `xs:decimal` et retourne une valeur `xs:dateTime` qui inclut une partie (fuseau horaire) **TZ**. Le résultat est obtenu en calculant l'équivalent UTC **dateTime** de Epoch time, et en l'ajoutant à son fuseau horaire local (pris de l'horloge système). Par exemple, si la fonction est exécutée sur un appareil qui a été défini pour être dans un fuseau horaire +01:00 (relatif à UTC), après avoir calculé l'équivalent UTC **dateTime**, une heure sera ajoutée au résultat. L'information du fuseau horaire, qui est une partie lexicale optionnelle du résultat **xs:dateTime**, est également rapportée dans le résultat **dateTime**. Comparez ce résultat avec celui de **dateTime-from-epoch-no-TZ**, et consultez également la fonction **epoch-from-dateTime**.

▣ *Exemples*

Les exemples ci-dessous supposent un fuseau horaire local UTC +01:00. En conséquence, l'équivalent UTC **dateTime** de l'Epoch time soumis sera incrémenté d'une heure. Le fuseau horaire est rapporté dans le résultat.

- **altova:dateTime-from-epoch**(34) retourne `1970-01-01T01:00:34+01:00`
- **altova:dateTime-from-epoch**(62) retourne `1970-01-01T01:01:02+01:00`

▼ dateTime-from-epoch-no-TZ [altova:]

altova:dateTime-from-epoch-no-TZ(Epoch *as xs:decimal* **as xs:dateTime** XP3.1 XQ3.1
Epoch time est un système horaire utilisé sur les systèmes Unix. Il définit tout moment donné comme étant l nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. La fonction **dateTime-from-epoch-no-TZ** retourne l'équivalent **xs:dateTime** d'un Epoch time, l'ajuste pour son fuseau horaire local, mais n'inclut pas l'information du fuseau horaire dans le résultat.

La fonction prend un `xs:decimal` argument et retourne une valeur `xs:dateTime` qui n'inclut pas de partie (fuseau horaire) **TZ**. Le résultat est obtenu en calculant l'équivalent UTC **dateTime** de Epoch time, et en l'ajoutant à son fuseau horaire local (pris de l'horloge système). Par exemple, si la fonction est exécutée sur un appareil qui a été défini pour être dans un fuseau horaire +01:00 (relatif à UTC), après avoir calculé l'équivalent, une heure sera ajoutée au résultat. L'information du fuseau horaire, qui est une partie lexicale optionnelle du résultat **xs:dateTime**, n'est pas rapportée dans le résultat **dateTime**. Comparez ce résultat avec celui de **dateTime-from-epoch**, et consultez également la fonction **epoch-from-dateTime**.

▣ *Exemples*

Les exemples ci-dessous supposent un fuseau horaire local UTC +01:00. En conséquence, l'équivalent UTC **dateTime** de l'Epoch time soumis sera incrémenté d'une heure. Le fuseau horaire n'est pas rapporté dans le résultat.

- **altova:dateTime-from-epoch**(34) returns `1970-01-01T01:00:34`

- `altova:dateTime-from-epoch(62)` returns `1970-01-01T01:01:02`

[Top]

30.2.1.3 Fonctions XPath/XQuery : Géolocalisation

Les fonctions d'extension de géolocalisation XPath/XQuery suivantes sont prises en charge dans la version actuelle de XMLSpy et peuvent être utilisées dans (i) des expressions XPath dans un contexte XSLT, ou (ii) des expressions XQuery dans un document XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

format-geolocation [altova:]

```
altova:format-geolocation(Latitude as xs:decimal, Longitude as xs:decimal,
GeolocationOutputStringFormat as xs:integer) as xs:string
```

XP3.1 XQ3.1

Prend la latitude et la longitude en tant que les deux premiers arguments, et sort la géolocalisation en tant que chaîne. Le troisième argument, `GeolocationOutputStringFormat`, est le format de la chaîne de sortie de géolocalisation ; il utilise des valeurs d'entier allant de `1` à `4` pour identifier le format de chaîne de sortie (*voir 'Formats de chaîne de sortie de géolocalisation' ci-dessous*). Les valeurs de latitude vont de `+90` à `-90` (`N` à `S`). Les valeurs de longitude vont de `+180` à `-180` (`E` à `O`).

Note : La fonction `image-exif-data` et les attributs de métadonnées Exif peuvent être utilisés pour fournie les chaînes d'entrée.

Exemples

- `altova:format-geolocation(33.33, -22.22, 4)` retourne `xs:string "33.33 -22.22"`

© 2016-2022 Altova GmbH

- **altova:format-geolocation**(33.33, -22.22, 2) retourne xs:string "33.33N 22.22W"
- **altova:format-geolocation**(-33.33, 22.22, 2) retourne xs:string "33.33S 22.22E"
- **altova:format-geolocation**(33.33, -22.22, 1) retourne xs:string "33°19'48.00"S 22° 13'12.00"E"

Formats de chaîne de sortie de géolocalisation:

La latitude et longitude fournies sont formatées dans un des formats de sortie indiqués ci-dessous. Le format désiré est défini par son ID d'entier (1 à 4). Les valeurs de latitude vont de +90 à -90 (N à S). Les valeurs de longitude vont de +180 à -180 (E à O).

1
Degrés, minutes, secondes décimales, avec orientation suffixée (N/S, E/O) **D°M'S.SS"N/S D°M'S.SS"E/W** *Exemple* : **33°55'11.11"N 22°44'66.66"W**

2
Degrés décimaux, avec orientation suffixée (N/S, E/O) **D.DDN/S D.DDE/W** *Exemple* : **33.33N 22.22W**

3
Degrés, minutes, secondes décimales, avec signe préfixé (+/-); le signe plus (**N/E**) est optionnel **+/-D°M'S.SS" +/-D°M'S.SS"** *Exemple* : **33°55'11.11" -22°44'66.66"**

4
Degrés décimaux, avec signe préfixé (+/-); le signe plus (**N/E**) est optionnel **+/-D.DD +/-D.DD** *Exemple* : **33.33 -22.22**

Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut **Geolocation** personnalisable depuis les onglets standard de métadonnées Exif. **Geolocation** est une concaténation de quatre onglets Exif : GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

parse-geolocation [altova:]

altova:parse-geolocation(GeolocationInputString *as xs:string*) **asxs:decimal+** **XP3.1** **XQ3.1**
Parse l'argument `GeolocationInputString` fourni et retourne la latitude et longitude de géolocalisation (dans cet ordre) en séquence deux items décimaux `xs:decimal`. Les formats dans lesquels la chaîne d'entrée de géolocalisation peut être fournie sont recensés ci-dessous.

Note : La fonction `image-exif-data` et l'attribut `@Geolocation` de métadonnées Exif peuvent être utilisés pour fournir la chaîne d'entrée de géolocalisation (*voir exemple ci-dessous*).

Exemples

- `altova:parse-geolocation("33.33 -22.22")` retourne la séquence de deux `xs:decimals` `(33.33, 22.22)`
- `altova:parse-geolocation("48°51'29.6""N 24°17'40.2""")` retourne la séquence de deux `xs:decimals (48.8582222222222, 24.2945)`
- `altova:parse-geolocation('48°51''29.6"N 24°17''40.2"')` retourne la séquence de deux `xs:decimals (48.8582222222222, 24.2945)`
- `altova:parse-geolocation( image-exif-data(//MyImages/Image20141130.01)/@Geolocation )` retourne une séquence de deux `xs:decimals`

Formats de string d'entrée de géolocalisation :

Le string d'entrée de géolocalisation doit contenir la latitude et la longitude (dans cet ordre) séparées par un espace. Les strings peuvent tous présenter les formats suivants. Les combinaisons sont permises. La latitude peut donc être dans un format et la longitude dans un autre. Les valeurs de latitude varient de `+90` à `-90` (`N` à `S`). Les valeurs de longitude varient de `+180` à `-180` (`E` à `W`).

Note : L'utilisation de guillemets simples ou doubles pour la délimitation des arguments de string entraînera une non-concordance avec l'utilisation de guillemets simples ou doubles pour indiquer, respectivement les valeurs de minutes et de secondes. Dans ces cas, les guillemets utilisés pour indiquer les minutes et les secondes doivent être échappés en les doublant. Dans les exemples présentés dans cette section, les guillemets utilisés pour délimiter les strings d'entrée sont marqués en jaune (`"`) alors que les indicateurs d'unité échappés sont marqués en bleu (`""`).

- Degrés, minutes, secondes décimales, avec orientation suffixée (`N/S`, `E/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Exemple : **`33°55'11.11"N 22°44'55.25"W`**

- Degrés, minutes, secondes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (**`N/E`**) est optionnel
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Exemple : **`33°55'11.11" -22°44'55.25"`**

- Degrés, minutes décimales, avec orientation suffixée (`N/S`, `E/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Exemple : **`33°55.55'N 22°44.44'W`**

- Degrés, minutes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (**`N/E`**) est optionnel
  `+/-D°M.MM' +/-D°M.MM'`

© 2016-2022 Altova GmbH

Exemple : `+33°55.55'  -22°44.44'`

- Degrés décimaux, avec orientation suffixée (`N/S`, `E/W`)
  `D.DDN/S  D.DDW/E`
  Exemple : `33.33N  22.22W`

- Degrés décimaux, avec signe préfixé (`+/-`) ; le signe plus pour (`N/S E/W`) est optionnel
  `+/-D.DD  +/-D.DD`
  Exemple : `33.33  -22.22`

Exemples de combinaisons de format :

```
33.33N  -22°44'55.25"
33.33  22°44'55.25"W
33.33  22.45
```

Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut **`Geolocation`** personnalisable depuis les onglets standard de métadonnées Exif. **`Geolocation`** est une concaténation de quatre onglets Exif : `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-distance-km [altova:]

`altova:geolocation-distance-km(GeolocationInputString-1` *as xs:string*, **`GeolocationInputString-2`** *as xs:string*`)` **`as xs:decimal`** **XP3.1** **XQ3.1**

Calcule la distance entre deux géolocalisations en kilomètres. Les formats dans lesquels une chaîne d'entrée de géolocalisation peut être fournie sont recensés ci-dessous. Les valeurs de latitude vont de `+90` à `-90` (`N` à `S`). Les valeurs de longitude vont de `+180` à `-180` (`E` à `O`).

Note : La fonction `image-exif-data` et l'attribut `@Geolocation` des métadonnées d'Exif peuvent être utilisés pour fournir les chaînes d'entrée de géolocalisation.

Exemples

- **`altova:geolocation-distance-km`**`("33.33  -22.22", "48°51'29.6""N  24°17'40.2""")`
  retourne `xs:decimal 4183.08132372392`

Formats de string d'entrée de géolocalisation :

Le string d'entrée de géolocalisation doit contenir la latitude et la longitude (dans cet ordre) séparées par un espace. Les strings peuvent tous présenter les formats suivants. Les combinaisons sont permises. La latitude peut donc être dans un format et la longitude dans un autre. Les valeurs de latitude varient de `+90` à `-90` (`N` à `S`). Les valeurs de longitude varient de `+180` à `-180` (`E` à `W`).

Note : L'utilisation de guillemets simples ou doubles pour la délimitation des arguments de string entraînera une non-concordance avec l'utilisation de guillemets simples ou doubles pour indiquer, respectivement les valeurs de minutes et de secondes. Dans ces cas, les guillemets utilisés pour indiquer les minutes et les secondes doivent être échappés en les doublant. Dans les exemples présentés dans cette section, les guillemets utilisés pour délimiter les strings d'entrée sont marqués en jaune (") alors que les indicateurs d'unité échappés sont marqués en bleu ("").

- Degrés, minutes, secondes décimales, avec orientation suffixée (N/S, E/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Exemple : **33°55'11.11"N 22°44'55.25"W**

- Degrés, minutes, secondes décimales, avec signe préfixé (+/-) ; le signe plus pour (**N/E**) est optionnel
  +/-D°M'S.SS" +/-D°M'S.SS"
  Exemple : **33°55'11.11" -22°44'55.25"**

- Degrés, minutes décimales, avec orientation suffixée (N/S, E/W)
  D°M.MM'N/S D°M.MM'W/E
  Exemple : **33°55.55'N 22°44.44'W**

- Degrés, minutes décimales, avec signe préfixé (+/-) ; le signe plus pour (**N/E**) est optionnel
  +/-D°M.MM' +/-D°M.MM'
  Exemple : **+33°55.55' -22°44.44'**

- Degrés décimaux, avec orientation suffixée (N/S, E/W)
  D.DDN/S D.DDW/E
  Exemple : **33.33N 22.22W**

- Degrés décimaux, avec signe préfixé (+/-) ; le signe plus pour (**N/S E/W**) est optionnel
  +/-D.DD +/-D.DD
  Exemple : **33.33 -22.22**

Exemples de combinaisons de format :
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45

Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut **Geolocation** personnalisable depuis les onglets standard de métadonnées Exif. **Geolocation** est une concaténation de quatre onglets Exif : GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

© 2016-2022 Altova GmbH

geolocation-distance-mi [altova:]

altova:geolocation-distance-mi(GeolocationInputString-1 *as xs:string*, **GeolocationInputString-2** *as xs:string*) **asxs:decimal** **XP3.1** **XQ3.1**

Calcule la distance entre deux géolocalisations en miles. Les formats dans lesquels une chaîne d'entrée de géolocalisation peut être fournie sont recensés ci-dessous. Les valeurs de latitude vont de `+90` à `-90` (`N` à `S`). Les valeurs de longitude vont de `+180` à `-180` (`E` à `O`).

Note : La fonction `image-exif-data` et l'attribut `@Geolocation` des métadonnées d'Exif peuvent être utilisés pour fournir les chaînes d'entrée de géolocalisation.

Exemples

- **altova:geolocation-distance-mi**(**"**33.33 -22.22**"**, **"**48°51'29.6**""**N 24°17'40.2**"""**) retourne `xs:decimal 2599.40652340653`

Formats de string d'entrée de géolocalisation :

Le string d'entrée de géolocalisation doit contenir la latitude et la longitude (dans cet ordre) séparées par un espace. Les strings peuvent tous présenter les formats suivants. Les combinaisons sont permises. La latitude peut donc être dans un format et la longitude dans un autre. Les valeurs de latitude varient de `+90` à `-90` (`N` à `S`). Les valeurs de longitude varient de `+180` à `-180` (`E` à `W`).

Note : L'utilisation de guillemets simples ou doubles pour la délimitation des arguments de string entraînera une non-concordance avec l'utilisation de guillemets simples ou doubles pour indiquer, respectivement les valeurs de minutes et de secondes. Dans ces cas, les guillemets utilisés pour indiquer les minutes et les secondes doivent être échappés en les doublant. Dans les exemples présentés dans cette section, les guillemets utilisés pour délimiter les strings d'entrée sont marqués en jaune (**"**) alors que les indicateurs d'unité échappés sont marqués en bleu (**""**).

- Degrés, minutes, secondes décimales, avec orientation suffixée (`N/S`, `E/W`)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Exemple : **33°55'11.11"N 22°44'55.25"W**

- Degrés, minutes, secondes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (**N/E**) est optionnel
  +/-D°M'S.SS" +/-D°M'S.SS"
  Exemple : **33°55'11.11" -22°44'55.25"**

- Degrés, minutes décimales, avec orientation suffixée (`N/S`, `E/W`)
  D°M.MM'N/S D°M.MM'W/E
  Exemple : **33°55.55'N 22°44.44'W**

- Degrés, minutes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (**N/E**) est optionnel
  +/-D°M.MM' +/-D°M.MM'
  Exemple : **+33°55.55' -22°44.44'**

- Degrés décimaux, avec orientation suffixée (`N/S`, `E/W`)
  D.DDN/S D.DDW/E
  Exemple : **33.33N 22.22W**

- Degrés décimaux, avec signe préfixé (+/-) ; le signe plus pour (N/S E/W) est optionnel
  +/-D.DD +/-D.DD
  Exemple : 33.33 -22.22

Exemples de combinaisons de format :
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45

Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut **Geolocation** personnalisable depuis les onglets standard de métadonnées Exif. **Geolocation** est une concaténation de quatre onglets Exif : GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-within-polygon [altova:]

altova:geolocation-within-polygon(Geolocation *as xs:string*, **((PolygonPoint** *as xs:string*)+)) **asxs:boolean** XP3.1 XQ3.1
Détermine si **Geolocation** (le premier argument) se trouve dans l'espace polygonal décrit par les arguments **PolygonPoint**. Si les arguments PolygonPoint ne forment pas une figure fermée (formée lorsque le premier point et le dernier point sont identiques), alors le premier point est implicitement ajouté en tant que le dernier point afin de pouvoir clore la figure. Tous les arguments (Geolocation et PolygonPoint+) sont donnés par chaînes d'entrées de géolocalisation (*formats recensés ci-dessous*). Si l'argument Geolocation se trouve dans l'espace polygonal, la fonction retourne true(); sinon, elle retourne false(). Les valeurs de latitude vont de +90 à -90 (N à S). Les valeurs de longitude vont de +180 à -180 (E à O).

Note : La fonction image-exif-data et l'attribut @Geolocation de métadonnées d'Exif peut être utilisée pour fournir les chaînes d'entrée de géolocalisation.

Exemples

- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24", "58 -32"))** retourne true()
- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24"))** retourne true()
- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48°51'29.6""N 24°17'40.2""))** retourne true()

Formats de string d'entrée de géolocalisation :

Le string d'entrée de géolocalisation doit contenir la latitude et la longitude (dans cet ordre) séparées par un espace. Les strings peuvent tous présenter les formats suivants. Les combinaisons sont permises. La latitude peut donc être dans un format et la longitude dans un autre. Les valeurs de latitude varient de `+90` à `-90` (`N` à `S`). Les valeurs de longitude varient de `+180` à `-180` (`E` à `W`).

Note : L'utilisation de guillemets simples ou doubles pour la délimitation des arguments de string entraînera une non-concordance avec l'utilisation de guillemets simples ou doubles pour indiquer, respectivement les valeurs de minutes et de secondes. Dans ces cas, les guillemets utilisés pour indiquer les minutes et les secondes doivent être échappés en les doublant. Dans les exemples présentés dans cette section, les guillemets utilisés pour délimiter les strings d'entrée sont marqués en jaune (`''`) alors que les indicateurs d'unité échappés sont marqués en bleu (`""`).

- Degrés, minutes, secondes décimales, avec orientation suffixée (`N/S`, `E/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Exemple : **`33°55'11.11"N 22°44'55.25"W`**

- Degrés, minutes, secondes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (**`N/E`**) est optionnel
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Exemple : **`33°55'11.11" -22°44'55.25"`**

- Degrés, minutes décimales, avec orientation suffixée (`N/S`, `E/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Exemple : **`33°55.55'N 22°44.44'W`**

- Degrés, minutes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (**`N/E`**) est optionnel
  `+/-D°M.MM' +/-D°M.MM'`
  Exemple : **`+33°55.55' -22°44.44'`**

- Degrés décimaux, avec orientation suffixée (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Exemple : **`33.33N 22.22W`**

- Degrés décimaux, avec signe préfixé (`+/-`) ; le signe plus pour (**`N/S E/W`**) est optionnel
  `+/-D.DD +/-D.DD`
  Exemple : **`33.33 -22.22`**

Exemples de combinaisons de format :

```
33.33N  -22°44'55.25"
33.33  22°44'55.25"W
33.33  22.45
```

Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut **`Geolocation`** personnalisable depuis les onglets standard de métadonnées Exif. **`Geolocation`** est une concaténation de quatre onglets Exif : `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation

33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-within-rectangle [altova:]

altova:geolocation-within-rectangle(Geolocation *as xs:string*, **RectCorner-1** *as xs:string*, **RectCorner-2** *as xs:string*) **asxs:boolean** **XP3.1** **XQ3.1**

Détermine si **Geolocation** (le premier argument) se trouve dans le rectangle défini par le second et le troisième argument, **RectCorner-1** et **RectCorner-2**, qui spécifient les coins opposés du rectangle. Tous les arguments (Geolocation, **RectCorner-1** et **RectCorner-2**) sont indiqués par des chaînes d'entrées de géolocalisation (*formats recensés ci-dessous*). Si l'argument Geolocation se trouve dans le rectangle, la fonction retourne true(); sinon, elle retourne false(). Les valeurs de latitude vont de +90 à -90 (N à S). Les valeurs de longitude vont de +180 à -180 (E à O).

Note : La fonction image-exif-data et l'attribut @Geolocation de métadonnées Exif peuvent être utilisés pour fournir les chaînes d'entrée de géolocalisation.

Exemples

- **altova:geolocation-within-rectangle**("33 -22", "58 -32", "-48 24") retourne true()
- **altova:geolocation-within-rectangle**("33 -22", "58 -32", "48 24") retourne false()
- **altova:geolocation-within-rectangle**("33 -22", "58 -32", "48°51'29.6""S 24° 17'40.2""") retourne true()

Formats de string d'entrée de géolocalisation :

Le string d'entrée de géolocalisation doit contenir la latitude et la longitude (dans cet ordre) séparées par un espace. Les strings peuvent tous présenter les formats suivants. Les combinaisons sont permises. La latitude peut donc être dans un format et la longitude dans un autre. Les valeurs de latitude varient de +90 à -90 (N à S). Les valeurs de longitude varient de +180 à -180 (E à W).

Note : L'utilisation de guillemets simples ou doubles pour la délimitation des arguments de string entraînera une non-concordance avec l'utilisation de guillemets simples ou doubles pour indiquer, respectivement les valeurs de minutes et de secondes. Dans ces cas, les guillemets utilisés pour indiquer les minutes et les secondes doivent être échappés en les doublant. Dans les exemples présentés dans cette section, les guillemets utilisés pour délimiter les strings d'entrée sont marqués en jaune (") alors que les indicateurs d'unité échappés sont marqués en bleu ("").

- Degrés, minutes, secondes décimales, avec orientation suffixée (N/S, E/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Exemple : **33°55'11.11"N 22°44'55.25"W**

- Degrés, minutes, secondes décimales, avec signe préfixé (+/-) ; le signe plus pour (**N/E**) est optionnel
  +/-D°M'S.SS" +/-D°M'S.SS"
  Exemple : **33°55'11.11" -22°44'55.25"**

- Degrés, minutes décimales, avec orientation suffixée (N/S, E/W)

© 2016-2022 Altova GmbH

`D°M.MM'N/S D°M.MM'W/E`
Exemple : `33°55.55'N 22°44.44'W`

- Degrés, minutes décimales, avec signe préfixé (`+/-`) ; le signe plus pour (`N/E`) est optionnel
  `+/-D°M.MM' +/-D°M.MM'`
  Exemple : `+33°55.55' -22°44.44'`

- Degrés décimaux, avec orientation suffixée (`N/S`, `E/W`)
  `D.DDN/S D.DDW/E`
  Exemple : `33.33N 22.22W`

- Degrés décimaux, avec signe préfixé (`+/-`) ; le signe plus pour (`N/S E/W`) est optionnel
  `+/-D.DD +/-D.DD`
  Exemple : `33.33 -22.22`

Exemples de combinaisons de format :

```
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45
```

- *Attribut Altova Exif : Géolocalisation*

  La machine Altova XPath/XQuery génère l'attribut `Geolocation` personnalisable depuis les onglets standard de métadonnées Exif. `Geolocation` est une concaténation de quatre onglets Exif : `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef`, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

[Top]

30.2.1.4 Fonctions XPath/XQuery : Relatives aux images

Les fonctions d'extension XPath/XQuery relatives à l'image suivantes sont prises en charge dans la version actuelle de XMLSpy et peuvent être utilisées dans (i) des expressions XPath dans un contexte XSLT, ou dans (ii) des expressions XQuery dans un document XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette

section par le préfixe **altova:**, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

suggested-image-file-extension [altova:]

altova:suggested-image-file-extension(Base64String *as string*) **asstring?** XP3.1 XQ3.1

Prend le code Base64 d'un fichier d'image en tant que son argument et retourne l'extension de fichier de l'image comme enregistré dans le codage Base64 de l'image. La valeur retournée est une suggestion basée sur l'information du type d'image disponible dans le codage. Si cette information n'est pas disponible, une chaîne vide est retournée. Cette fonction est utile si vous souhaitez enregistrer une image Base64 en tant que fichier et que vous souhaitez extraire dynamiquement une extension de fichier appropriée.

Exemples

- `altova:suggested-image-file-extension(/MyImages/MobilePhone/Image20141130.01)` retourne `'jpg'`
- `altova:suggested-image-file-extension($XML1/Staff/Person/@photo)` retourne `''`

Dans les exemples ci-dessus, les nœuds fournis en tant qu'arguments de la fonction sont assumés contenir une image codée Base64. Le premier exemple extrait `jpg` en tant que type et extension de fichier. Dans le second exemple, le codage Base64 soumis ne fournit pas une information de fichier d'extension utile.

image-exif-data [altova:]

altova:image-exif-data(Base64BinaryString *as string*) **aselement?** XP3.1 XQ3.1

Prend une image codée Base64 en tant que son argument et retourne un élément appelé **Exif** qui contient les métadonnées Exif de l'image. Celles-ci sont créées en tant que paires attribute-value pairs de l'élément `Exif`. Les noms d'attribut sont les onglets de données Exif trouvés dans le codage Base64. La liste des onglets des spécifications Exif est indiquée ci-dessous. Si un onglet spécifique à un distributeur est présent dans les données Exif, cet onglet et sa valeur seront aussi retournés en tant que paire attribute-value. Outre les onglets de métadonnées Exif standard (*voir la liste ci-dessous*), des paires attribute-value spécifiques à Altova sont également générées. Ces attributs Exif Altova sont recensés ci-dessous.

Exemples

- Pour accéder à n'importe quel attribut, utiliser la fonction comme suit :

© 2016-2022 Altova GmbH

```
image-exif-data(//MyImages/Image20141130.01)/@GPSLatitude
image-exif-data(//MyImages/Image20141130.01)/@Geolocation
```

- Pour accéder à tous les attributs, utiliser la fonction comme suit:
  ```
  image-exif-data(//MyImages/Image20141130.01)/@*
  ```
- Pour accéder au nom de tous les attributs, utiliser l'expression suivante :
  ```
  for $i in image-exif-data(//MyImages/Image20141130.01)/@* return name($i)
  ```
  Cela est utile pour trouver les noms des attributs retournés par la fonction.

Attribut Altova Exif : Géolocalisation

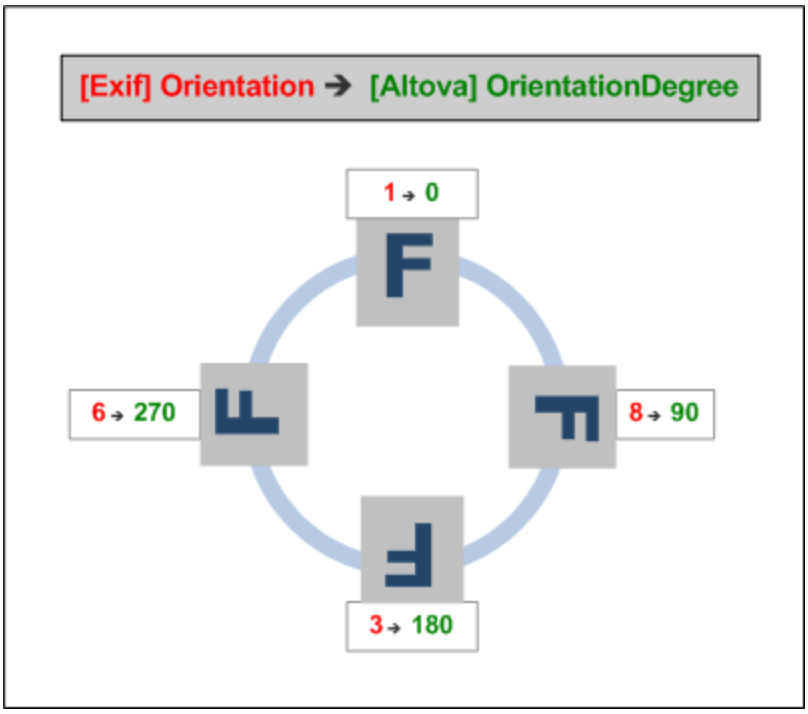
La machine Altova XPath/XQuery génère l'attribut **Geolocation** personnalisable depuis les onglets standard de métadonnées Exif. **Geolocation** est une concaténation de quatre onglets Exif : GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

Altova Exif Attribute: OrientationDegree

La machine Altova XPath/XQuery génère l'attribut personnalisé **OrientationDegree** à partir de l'onglet de métadonnées Exif **Orientation**.

OrientationDegree traduit l'onglet standard Exif Orientation à partir d'une valeur d'entier (1, 8, 3, ou 6) aux valeurs de degrés respectives de chacun (0, 90, 180, 270), tel que montré dans la figure ci-dessous. Veuillez noter qu'il n'y a pas de traductions de la valeur Orientation de 2, 4, 5, 7. (Ces orientations sont obtenus en basculant l'image 1 à travers son centre axial vertical pour obtenir l'image avec une valeur de 2, puis en pivotant cette image par sauts de 90° dans le sens des aiguilles d'une montre pour obtenir les valeurs de 7, 4, et 5, respectivement).



- *Listing of standard Exif meta tags*
  - ImageWidth
  - ImageLength
  - BitsPerSample
  - Compression
  - PhotometricInterpretation
  - Orientation
  - SamplesPerPixel
  - PlanarConfiguration
  - YCbCrSubSampling
  - YCbCrPositioning
  - XResolution
  - YResolution
  - ResolutionUnit
  - StripOffsets
  - RowsPerStrip
  - StripByteCounts
  - JPEGInterchangeFormat
  - JPEGInterchangeFormatLength
  - TransferFunction
  - WhitePoint
  - PrimaryChromaticities
  - YCbCrCoefficients
  - ReferenceBlackWhite
  - DateTime
  - ImageDescription
  - Make
  - Model

© 2016-2022 Altova GmbH

- Software
- Artist
- Copyright

- ExifVersion
- FlashpixVersion
- ColorSpace
- ComponentsConfiguration
- CompressedBitsPerPixel
- PixelXDimension
- PixelYDimension
- MakerNote
- UserComment
- RelatedSoundFile
- DateTimeOriginal
- DateTimeDigitized
- SubSecTime
- SubSecTimeOriginal
- SubSecTimeDigitized
- ExposureTime
- FNumber
- ExposureProgram
- SpectralSensitivity
- ISOSpeedRatings
- OECF
- ShutterSpeedValue
- ApertureValue
- BrightnessValue
- ExposureBiasValue
- MaxApertureValue
- SubjectDistance
- MeteringMode
- LightSource
- Flash
- FocalLength
- SubjectArea
- FlashEnergy
- SpatialFrequencyResponse
- FocalPlaneXResolution
- FocalPlaneYResolution
- FocalPlaneResolutionUnit
- SubjectLocation
- ExposureIndex
- SensingMethod
- FileSource
- SceneType
- CFAPattern
- CustomRendered
- ExposureMode
- WhiteBalance
- DigitalZoomRatio
- FocalLengthIn35mmFilm
- SceneCaptureType
- GainControl
- Contrast

- Saturation
- Sharpness
- DeviceSettingDescription
- SubjectDistanceRange
- ImageUniqueID

- GPSVersionID
- GPSLatitudeRef
- GPSLatitude
- GPSLongitudeRef
- GPSLongitude
- GPSAltitudeRef
- GPSAltitude
- GPSTimeStamp
- GPSSatellites
- GPSStatus
- GPSMeasureMode
- GPSDOP
- GPSSpeedRef
- GPSSpeed
- GPSTrackRef
- GPSTrack
- GPSImgDirectionRef
- GPSImgDirection
- GPSMapDatum
- GPSDestLatitudeRef
- GPSDestLatitude
- GPSDestLongitudeRef
- GPSDestLongitude
- GPSDestBearingRef
- GPSDestBearing
- GPSDestDistanceRef
- GPSDestDistance
- GPSProcessingMethod
- GPSAreaInformation
- GPSDateStamp
- GPSDifferential

[Top]

30.2.1.5 Fonctions XPath/XQuery : Numérique

Les fonctions d'extension numériques d'Altova peuvent être utilisées dans des expressions XPath et XQuery et proposent des fonctions supplémentaires pour le traitement des données. Les fonctions dans cette section peuvent être utilisée avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova. Ils sont disponibles dans des contextes XPath/XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des

© 2016-2022 Altova GmbH

fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

Fonctions de numérotation automatique

generate-auto-number [altova:]

altova:generate-auto-number(ID *as xs:string*, **StartsWith** *as xs:double*, **Increment** *as xs:double*, **ResetOnChange** *as xs:string*) as `xs:integer` XP1 XP2 XQ1 XP3.1 XQ3.1

Génère un numéro à chaque fois que la fonction est appelée. Le premier numéro, qui est généré la première fois que la fonction est appelée, est spécifié par l'argument `StartsWith`. Chaque appel subséquent vers la fonction génère un nouveau numéro, ce numéro est augmenté au-dessus du numéro précédemment généré par la valeur spécifiée dans l'argument `Increment`. En effet, la fonction `altova:generate-auto-number` crée un compteur comportant un nom spécifié par l'argument `ID`, et dont le compteur est augmenté à chaque fois que la fonction est appelée. Si la valeur de l'argument `ResetOnChange` change de celle de l'appel de fonction précédent, la valeur du numéro à générer est réinitialisée à la valeur `StartsWith`. La numérotation automatique peut être réinitialisée en utilisant la fonction `altova:reset-auto-number`.

Exemples

- `altova:generate-auto-number("ChapterNumber", 1, 1, "SomeString")` retournera un nombre à chaque fois que la fonction est appelée, en commençant avec `1`, et en augmentant de `1` avec chaque appel de la fonction. Tant que le quatrième argument demeure `"SomeString"` dans chaque appel subséquent, l'augmentation se poursuivra. Lorsque la valeur du quatrième argument change, le compteur (appelé `ChapterNumber`) sera réinitialisé à 1. La valeur de `ChapterNumber` peut aussi être réinitialisée par un appel de la fonction `altova:reset-auto-number` comme ceci : `altova:reset-auto-number("ChapterNumber")`.

reset-auto-number [altova:]

altova:reset-auto-number(ID *as xs:string*) XP1 XP2 XQ1 XP3.1 XQ3.1

Cette fonction réinitialise le numéro du compteur de numérotation automatique nommé dans l'argument `ID`. Le numéro est réinitialisé au numéro spécifié par l'argument `StartsWith` de la fonction `altova:generate-auto-number` qui a créé le compteur nommé dans l'argument `ID`.

Exemples

- `altova:reset-auto-number("ChapterNumber")` réinitialise le numéro du compteur de

numérotation automatique nommé `ChapterNumber` qui a été créé par la fonction `altova:generate-auto-number`. Le numéro est réinitialisé à la valeur de l'argument `StartsWith` de la fonction `altova:generate-auto-number` qui a créé `ChapterNumber`.

[Top]

Fonctions numériques

hex-string-to-integer [altova:]

altova:hex-string-to-integer(HexString *as xs:string*) **asxs:integer** XP3.1 XQ3.1

Prend un argument de chaîne qui est l'équivalent Base-16 d'un entier dans le système décimal (Base-10), et retourne l'entier décimal.

Exemples

- **altova:hex-string-to-integer**('1') retourne 1
- **altova:hex-string-to-integer**('9') retourne 9
- **altova:hex-string-to-integer**('A') retourne 10
- **altova:hex-string-to-integer**('B') retourne 11
- **altova:hex-string-to-integer**('F') retourne 15
- **altova:hex-string-to-integer**('G') retourne une erreur
- **altova:hex-string-to-integer**('10') retourne 16
- **altova:hex-string-to-integer**('01') retourne 1
- **altova:hex-string-to-integer**('20') retourne 32
- **altova:hex-string-to-integer**('21') retourne 33
- **altova:hex-string-to-integer**('5A') retourne 90
- **altova:hex-string-to-integer**('USA') retourne une erreur

integer-to-hex-string [altova:]

altova:integer-to-hex-string(Integer *as xs:integer*) **asxs:string** XP3.1 XQ3.1

Prend un argument d'entier et retourne son équivalent de Base-16 en tant que chaîne.

Exemples

- **altova:integer-to-hex-string**(1) retourne '1'
- **altova:integer-to-hex-string**(9) retourne '9'
- **altova:integer-to-hex-string**(10) retourne 'A'
- **altova:integer-to-hex-string**(11) retourne 'B'
- **altova:integer-to-hex-string**(15) retourne 'F'
- **altova:integer-to-hex-string**(16) retourne '10'
- **altova:integer-to-hex-string**(32) retourne '20'
- **altova:integer-to-hex-string**(33) retourne '21'
- **altova:integer-to-hex-string**(90) retourne '5A'

[Top]

© 2016-2022 Altova GmbH

Fonctions de formatage de numéro

[Top]

30.2.1.6 Fonctions XPath/XQuery : Schéma

Les fonctions d'extension Altova recensées ci-dessous retournent l'information concernant le schéma. Ci-dessous, vous trouverez les descriptions des fonctions, ainsi que des (i) exemples et (ii) une liste des composants de schéma et de leurs propriétés respectives. Elles peuvent être utilisées avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova et sont disponibles dans des contextes XPath/XQuery.

Information de schéma depuis les documents de schéma
La fonction **altova:schema** détient deux arguments : un avec zéro arguments et l'autre avec deux arguments. La fonction à zéro argument retourne l'ensemble du schéma. Ensuite, à partir de là, vous pouvez naviguer dans le schéma pour localiser les composants de schéma que vous souhaitez. La fonction à deux arguments retourne un type de composant spécifique qui est identifié par son QName. Dans les deux cas, la valeur de retour est un fonction. Pour naviguer dans le composant retourné, vous devez sélectionner une propriété de ce composant spécifique. Si la propriété est un item non atomique (c'est à dire, s'il s'agit d'un composant), vous pouvez aller plus loin en choisissant une propriété de ce composant. Si la propriété sélectionnée est un item atomique, la valeur de l'item est retournée et vous ne pouvez pas aller plus loin.

Note: Dans les expressions XQuery, le schéma doit être importé explicitement. Dans les expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple, dans XSLT avec l'instruction **xslt:import** .

Information de schéma depuis les nœuds XML
La fonction **altova:type** soumet le nœud à un document XML et retoune l'information de type du nœud depuis le PSVI.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova, http://www.altova.com/xslt-extensions**, et sont indiquées dans cette section par le préfixe **altova:**, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

▼ Schéma (zéro arguments)

altova:schema() as (function(xs:string) as item()*)? XP3.1 XQ3.1
Retourne le composant de **schema** en entier. Vous pouvez donc aller plus loin dans le composant de **schema** en sélectionnant une des propriétés du composant de **schema**.

- Si cette propriété est un composant, vous pouvez aller encore plus en loin en sélectionnant une des propriétés de ce composant. Vous pouvez renouveler cette étape en allant plus loin dans le schéma.
- Si le composant est une valeur atomique, celle-ci sera retournée et vous ne pourrez pas aller plus loin.

Les propriétés du composant de **schema** sont :

```
"type definitions"
"attribute declarations"
"element declarations"
"attribute group definitions"
"model group definitions"
"notation declarations"
"identity-constraint definitions"
```

Les propriétés de tous les types de composant (à part **schema**) sont regroupées ci-dessous.

Note: Dans des expressions XQuery, le schéma doit être importé explicitement. Dans des expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple dans XSLT avec l'instruction **xslt:import**.

▫ *Exemples*

- `import schema "" at "C:\Test\ExpReport.xsd"; for $typedef in altova:schema() ("type definitions") return $typedef ("name")` retourne les noms de tous les Types simples ou Types complexes dans le schéma

- `import schema "" at "C:\Test\ExpReport.xsd"; altova:schema() ("type definitions")[1]("name")` retourne le nom du premier de tous les Types simples ou Types complexes dans le schéma

Composants et leurs propriétés

▫ Assertion

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Assertion"
test	XPath Property Record	

© 2016-2022 Altova GmbH

Déclaration d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Declaration"
name	string	Local name of the attribute
target namespace	string	Namespace URI of the attribute
type definition	Type simple or Type complexe	
scope	A function with properties ("class":"Scope", "variety": "global" or "local", "parent": the containing Type complexe or Attribute Group)	
value constraint	If present, a function with properties ("class": "Value Constraint", "variety": "fixed" or "default", "value": atomic value, "lexical form": string. Note that the "value" property is not available for namespace-sensitive types	
inheritable	boolean	

Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
name	string	Local name of the attribute group
target namespace	string	Namespace URI of the attribute group
attribute uses	Sequence of (Attribute Use)	
attribute wildcard	Optional Attribute Wildcard	

Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
required	boolean	true if the attribute is required, false if optional
value constraint	See Attribute Declaration	
inheritable	boolean	

Caractère générique

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"

namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict"|"lax"|"skip")	

Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
base type definition	Complex Type Definition	
final	Sequence of strings ("restriction"|"extension")	
context	Empty sequence (not implemented)	
derivation method	string ("restriction"|"extension")	
abstract	boolean	
attribute uses	Sequence of Attribute Use	
attribute wildcard	Optional Attribute Wildcard	
content type	function with properties: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
prohibited substitutions	Sequence of strings ("restriction"|"extension")	
assertions	Sequence of Assertion	

Déclaration d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"

© 2016-2022 Altova GmbH

name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
type definition	Type simple or Type complexe	
type table	function with properties ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Type simple or Type complexe)	
scope	function with properties ("class":"Scope", "variety": ("global"|"local"), "parent": optional Type complexe)	
value constraint	see Attribute Declaration	
nillable	boolean	
identity-constraint definitions	Sequence of Identity Constraint	
substitution group affiliations	Sequence of Element Declaration	
substitution group exclusions	Sequence of strings ("restriction"|"extension")	
disallowed substitutions	Sequence of strings ("restriction"|"extension"|"substitution")	
abstract	boolean	

Caractère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict"|"lax"|"skip")	

Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	The name of the facet, for example "minLength" or "enumeration"

value	depends on facet	The value of the facet
fixed	boolean	
typed-value	For the enumeration facet only, array(xs:anyAtomicType*)	An array containing the enumeration values, each of which may in general be a sequence of atomic values. (Note: for the enumeration facet, the "value" property is a sequence of strings, regardless of the actual type)

Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
name	string	Local name of the constraint
target namespace	string	Namespace URI of the constraint
identity-constraint category	string ("key"|"unique"|"keyRef")	
selector	XPath Property Record	
fields	Sequence of XPath Property Record	
referenced key	(For keyRef only): Identity Constraint	The corresponding key constraint

Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Groupe de modèle"
compositor	string ("sequence"|"choice"|"all")	
particles	Séquence de particule	

Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de groupe de modèle"
name	string	Nom local du groupe de modèle
target namespace	string	URI d'espace du groupe de modèle
model group	Groupe de modèle	

Notation

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Déclaration de notation"
name	string	Nom local de la notation

© 2016-2022 Altova GmbH

target namespace	string	URI d'espace de nom de la notation
system identifier	anyURI	
public identifier	string	

- Particule

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Particule"
min occurs	entier	
max occurs	entier ou string("unbounded")	
term	Déclaration d'élément, Caractère générique d'élément ou ModelGroup	

- Type simple

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de type simple"
name	string	Nom local du type (vide si anonyme)
target namespace	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction"|"extension"|"list"|"union")	
context	composant contenant	
base type definition	Type simple	
facets	Séquence de Facette	
fundamental facets	Séquence vide (pas implémentée)	
variety	string ("atomic"|"list"|"union")	
primitive type definition	Type simple	
item type definition	uniquement pour les types de liste) Type simple	
member type definitions	(uniquement pour les types d'union) Séquence de Type simple	

- Alternative de type

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type Alternative"

test	XPath Property Record	
type definition	Type simple ou Type complexe	

XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
namespace bindings	Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

Schéma (deux arguments)

altova:schema(ComponentKind *as xs:string*, **Name** *as xs:QName*) **as (function(xs:string) as item()*)?** **XP3.1** **XQ3.1**

Retourne le type du composant qui est spécifié dans le premier argument qui a un nom identique à celui fourni dans le second argument. Vous pouvez donc aller plus loin en sélectionnant une des propriétés du composant.

- Si cette propriété est un composant, vous pouvez aller encore plus en profondeur en sélectionnant une des propriétés de ce composant. Vous pouvez renouveler cette étape en allant plus loin dans le schéma.
- Si le composant est une valeur atomique, celle-ci sera retournée et vous ne pourrez pas aller plus loin.

Note: Dans des expressions XQuery, le schéma doit être importé explicitement. Dans des expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple dans XSLT avec l'instruction **xslt:import**.

Exemples

- ```
  import schema "" at "C:\Test\ExpReport.xsd";
  altova:schema("element declaration", xs:QName("OrgChart"))("type definition")
  ("content type")("particles")[3]!.("term")("kind")
  ```
  retourne la propriété `kind` du terme du troisième composant de `particles`. Ce composant de particules est un descendant de la déclaration d'élément ayant un `QName` de `OrgChart`

- ```
  import schema "" at "C:\Test\ExpReport.xsd";
  let $typedef := altova:schema("type definition", xs:QName("emailType"))
  for $facet in $typedef ("facets")
  return [$facet ("kind"), $facet("value")]
  ```
  retourne, pour chaque **facet** de chaque composant **emailType**, un array contenant le type et la valeur de cette facette

© 2016-2022 Altova GmbH

Composants et leurs propriétés

- Assertion

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Assertion"
test	XPath Property Record	

- Déclaration d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Declaration"
name	string	Local name of the attribute
target namespace	string	Namespace URI of the attribute
type definition	Type simple or Type complexe	
scope	A function with properties ("class":"Scope", "variety": "global" or "local", "parent": the containing Type complexe or Attribute Group)	
value constraint	If present, a function with properties ("class": "Value Constraint", "variety": "fixed" or "default", "value": atomic value, "lexical form": string. Note that the "value" property is not available for namespace-sensitive types	
inheritable	boolean	

- Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
name	string	Local name of the attribute group
target namespace	string	Namespace URI of the attribute group
attribute uses	Sequence of (Attribute Use)	
attribute wildcard	Optional Attribute Wildcard	

- Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
required	boolean	true if the attribute is required, false if optional

value constraint	See Attribute Declaration	
inheritable	boolean	

Caractère générique

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict"|"lax"|"skip")	

Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
base type definition	Complex Type Definition	
final	Sequence of strings ("restriction"|"extension")	
context	Empty sequence (not implemented)	
derivation method	string ("restriction"|"extension")	
abstract	boolean	
attribute uses	Sequence of Attribute Use	
attribute wildcard	Optional Attribute Wildcard	
content type	function with properties: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
prohibited	Sequence of strings	

© 2016-2022 Altova GmbH

substitutions	("restriction"|"extension")	
assertions	Sequence of Assertion	

- Déclaration d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
type definition	Type simple or Type complexe	
type table	function with properties ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Type simple or Type complexe)	
scope	function with properties ("class":"Scope", "variety": ("global"|"local"), "parent": optional Type complexe)	
value constraint	see Attribute Declaration	
nillable	boolean	
identity-constraint definitions	Sequence of Identity Constraint	
substitution group affiliations	Sequence of Element Declaration	
substitution group exclusions	Sequence of strings ("restriction"|"extension")	
disallowed substitutions	Sequence of strings ("restriction"|"extension"|"substitution")	
abstract	boolean	

- Caractère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict"|"lax"|"skip")	

Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	The name of the facet, for example "minLength" or "enumeration"
value	depends on facet	The value of the facet
fixed	boolean	
typed-value	For the enumeration facet only, array(xs:anyAtomicType*)	An array containing the enumeration values, each of which may in general be a sequence of atomic values. (Note: for the enumeration facet, the "value" property is a sequence of strings, regardless of the actual type)

Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
name	string	Local name of the constraint
target namespace	string	Namespace URI of the constraint
identity-constraint category	string ("key"|"unique"|"keyRef")	
selector	XPath Property Record	
fields	Sequence of XPath Property Record	
referenced key	(For keyRef only): Identity Constraint	The corresponding key constraint

Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Groupe de modèle"
compositor	string ("sequence"|"choice"|"all")	
particles	Séquence de particule	

Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de groupe de modèle"
name	string	Nom local du groupe de modèle
target namespace	string	URI d'espace du groupe de modèle

© 2016-2022 Altova GmbH

model group	Groupe de modèle	

Notation

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Déclaration de notation"
name	string	Nom local de la notation
target namespace	string	URI d'espace de nom de la notation
system identifier	anyURI	
public identifier	string	

Particule

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Particule"
min occurs	entier	
max occurs	entier ou string("unbounded")	
term	Déclaration d'élément, Caractère générique d'élément ou ModelGroup	

Type simple

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de type simple"
name	string	Nom local du type (vide si anonyme)
target namespace	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction"|"extension"|"list"|"union")	
context	composant contenant	
base type definition	Type simple	
facets	Séquence de Facette	
fundamental facets	Séquence vide (pas implémentée)	
variety	string ("atomic"|"list"|"union")	
primitive type definition	Type simple	
item type definition	uniquement pour les types de liste) Type simple	

member type definitions	(uniquement pour les types d'union) Séquence de Type simple	

Alternative de type

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type Alternative"
test	XPath Property Record	
type definition	Type simple ou Type complexe	

XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
namespace bindings	Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

Type

`altova:type(Node` *as item?*`)` **`as (function(xs:string) as item()*)?`** **XP3.1** **XQ3.1**

La fonction **`altova:type`** soumet un nœud d'élément ou d'attribut d'un document XML et document et retourne l'information du type du nœud depuis le PSVI.

Note: Le document XML doit avoir une déclaration de schéma afin que ce schéma puisse être référencé.

Exemples

- ```
  for $element in //Email
  let $type := altova:type($element)
  return $type
  ```
  retourne une fonction qui contient l'information du type du nœud

- ```
  for $element in //Email
  let $type := altova:type($element)
  return $type ("kind")
  ```
  prend le composant du type du nœud (Type simple ou Type complexe) et retourne la valeur de la propriété **`kind`** du composant

Composants et leurs propriétés

Assertion

© 2016-2022 Altova GmbH

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Assertion"
test	XPath Property Record	

Déclaration d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Declaration"
name	string	Local name of the attribute
target namespace	string	Namespace URI of the attribute
type definition	Type simple or Type complexe	
scope	A function with properties ("class":"Scope", "variety": "global" or "local", "parent": the containing Type complexe or Attribute Group)	
value constraint	If present, a function with properties ("class": "Value Constraint", "variety": "fixed" or "default", "value": atomic value, "lexical form": string. Note that the "value" property is not available for namespace-sensitive types	
inheritable	boolean	

Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
name	string	Local name of the attribute group
target namespace	string	Namespace URI of the attribute group
attribute uses	Sequence of (Attribute Use)	
attribute wildcard	Optional Attribute Wildcard	

Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
required	boolean	true if the attribute is required, false if optional
value constraint	See Attribute Declaration	
inheritable	boolean	

Caractère générique

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict"|"lax"|"skip")	

Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
base type definition	Complex Type Definition	
final	Sequence of strings ("restriction"|"extension")	
context	Empty sequence (not implemented)	
derivation method	string ("restriction"|"extension")	
abstract	boolean	
attribute uses	Sequence of Attribute Use	
attribute wildcard	Optional Attribute Wildcard	
content type	function with properties: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
prohibited substitutions	Sequence of strings ("restriction"|"extension")	
assertions	Sequence of Assertion	

Déclaration d'élément

© 2016-2022 Altova GmbH

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
type definition	Type simple or Type complexe	
type table	function with properties ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Type simple or Type complexe)	
scope	function with properties ("class":"Scope", "variety": ("global"|"local"), "parent": optional Type complexe)	
value constraint	see Attribute Declaration	
nillable	boolean	
identity-constraint definitions	Sequence of Identity Constraint	
substitution group affiliations	Sequence of Element Declaration	
substitution group exclusions	Sequence of strings ("restriction"|"extension")	
disallowed substitutions	Sequence of strings ("restriction"|"extension"|"substitution")	
abstract	boolean	

- Caractère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict"|"lax"|"skip")	

- Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	The name of the facet, for

		example "minLength" or "enumeration"
value	depends on facet	The value of the facet
fixed	boolean	
typed-value	For the enumeration facet only, array(xs:anyAtomicType*)	An array containing the enumeration values, each of which may in general be a sequence of atomic values. (Note: for the enumeration facet, the "value" property is a sequence of strings, regardless of the actual type)

Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
name	string	Local name of the constraint
target namespace	string	Namespace URI of the constraint
identity-constraint category	string ("key"|"unique"|"keyRef")	
selector	XPath Property Record	
fields	Sequence of XPath Property Record	
referenced key	(For keyRef only): Identity Constraint	The corresponding key constraint

Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Groupe de modèle"
compositor	string ("sequence"|"choice"|"all")	
particles	Séquence de particule	

Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de groupe de modèle"
name	string	Nom local du groupe de modèle
target namespace	string	URI d'espace du groupe de modèle
model group	Groupe de modèle	

Notation

Nom de propriété	Type de propriété	Valeur de propriété

© 2016-2022 Altova GmbH

kind	string	"Déclaration de notation"
name	string	Nom local de la notation
target namespace	string	URI d'espace de nom de la notation
system identifier	anyURI	
public identifier	string	

Particule

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Particule"
min occurs	entier	
max occurs	entier ou string("unbounded")	
term	Déclaration d'élément, Caractère générique d'élément ou ModelGroup	

Type simple

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de type simple"
name	string	Nom local du type (vide si anonyme)
target namespace	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction"|"extension"|"list"|"union")	
context	composant contenant	
base type definition	Type simple	
facets	Séquence de Facette	
fundamental facets	Séquence vide (pas implémentée)	
variety	string ("atomic"|"list"|"union")	
primitive type definition	Type simple	
item type definition	uniquement pour les types de liste) Type simple	
member type definitions	(uniquement pour les types d'union) Séquence de Type simple	

Alternative de type

Nom de propriété	Type de propriété	Valeur de propriété

kind	string	"Type Alternative"
test	XPath Property Record	
type definition	Type simple ou Type complexe	

XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
namespace bindings	Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

30.2.1.7 Fonctions XPath/XQuery : Séquence

Les fonctions d'extension de la séquence d'Altova peuvent être utilisées dans les expressions XPath et XQuery et proposent des fonctions supplémentaires pour le traitement des données. Les fonctions dans cette section peuvent être utilisées avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova. Ils sont disponibles dans des contextes XPath/XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery	XQ1 XQ3.1

© 2016-2022 Altova GmbH

dans XQuery) :	

▼ attributes [altova:]

altova:attributes(AttributeName *as xs:string***) asattribute()* XP3.1 XQ3.1**
Retourne tous les attributs possédant un nom local qui est le même que le nom fourni dans l'argument d'entrée, `AttributeName`. La recherche est sensible à la casse et est conduite le long de l'axe `attribute::` . Cela signifie que le nœud contextuel doit être le nœud d'élément parent.

Exemples

- **altova:attributes**("MyAttribute") retourne `MyAttribute()*`

altova:attributes(AttributeName *as xs:string*, **SearchOptions** *as xs:string***) asattribute()* XP3.1 XQ3.1**
Retourne tous les attribut possédant un nom local qui est le même que le nom fourni dans l'argument d'entrée, `AttributeName`. La recherche est sensible à la casse et est conduite le long de l'axe `attribute::` . Le nœud contextuel doit être le nœud d'élément parent. Le deuxième argument est une chaîne contenant des flags optionnels. Les flags disponibles sont :
r = passe à une recherche d'expression régulière ; `AttributeName` doit alors être une chaîne de recherche d'expression régulière ;
f = si cette option est spécifiée, alors `AttributeName` fournit une concordance complète ; dans le cas contraire, `AttributeName` ne nécessite qu'une concordance partielle d'un nom d'attribut pour retourner cet attribut. Par exemple : si **f** n'est pas spécifié, `MyAtt` retournera `MyAttribute`;
i = passe à une recherche insensible à la casse ;
p = comprend le préfixe d'espace de nom dans la recherche ; `AttributeName` devrait ensuite contenir le préfixe d'espace de nom, par exemple : `altova:MyAttribute`.
Les flags peuvent être écrits dans n'importe quel ordre. Les flags invalides généreront des erreurs. Un ou plusieurs flags peuvent être omis. La chaîne vide est permise et produire le même effet que la fonction n'ayant qu'un seul argument (*signature précédente*). Néanmoins, une séquence vide n'est pas permise en tant que le deuxième argument.

Exemples

- **altova:attributes**("MyAttribute", "rfip") retourne `MyAttribute()*`
- **altova:attributes**("MyAttribute", "pri") retourne `MyAttribute()*`
- **altova:attributes**("MyAtt", "rip") retourne `MyAttribute()*`
- **altova:attributes**("MyAttributes", "rfip") ne retourne aucune correspondance.
- **altova:attributes**("MyAttribute", "") retourne `MyAttribute()*`
- **altova:attributes**("MyAttribute", "Rip") retourne une erreur de flag non reconnu.
- **altova:attributes**("MyAttribute",) retourne une erreur d'argument manquant.

▼ elements [altova:]

altova:elements(ElementName *as xs:string***) aselement()* XP3.1 XQ3.1**
Retourne tous les éléments qui ont un nom local identique au nom fourni dans l'argument d'entrée, `ElementName`. La recherche est sensible à la casse et est conduite le long de l'axe `child::`. Le nœud contextuel doit être le nœud parent de/s l'élément/s recherché.

Exemples

- **altova:elements**("MyElement") retourne `MyElement()*`

altova:elements(ElementName *as xs:string*, **SearchOptions** *as xs:string*) **aselement()*** **XP3.1** **XQ3.1**

Retourne tous les éléments qui ont un nom local identique au nom fourni dans l'argument d'entrée, `ElementName`. La recherche est sensible à la casse et est conduite le long de l'axe `child::`. Le nœud contextuel doit être le nœud parent de/s l'élément/s recherché. Le second argument est une chaîne contenant des flags optionnels. Les flags disponibles sont :
r = passe à une recherche d'expression régulière ; `ElementName` doit alors être une chaîne de recherche d'expression régulière ;
f = si cette option est spécifiée, alors `ElementName` fournit une concordance complète ; dans le cas contraire, `ElementName` ne nécessite qu'une concordance partielle d'un nom d'élément pour retourner cet élément. Par exemple : si **f** n'est pas spécifié, `MyElem` retournera `MyElement` ;
i = passe à une recherche insensible à la casse ;
p = comprend le préfixe d'espace de nom dans la recherche ; `ElementName` devrait ensuite contenir le préfixe d'espace de nom, par exemple : `altova:MyElement`.
Les flags peuvent être écrits dans n'importe quel ordre. Les flags invalides généreront des erreurs. Un ou plusieurs flags peuvent être omis. La chaîne vide est autorisée et produira le même effet que la fonction n'ayant qu'un argument (*signature précédente*). Néanmoins, une séquence vide n'est pas autorisée.

Exemples

- **altova:elements**("MyElement", "rip") retourne MyElement()*
- **altova:elements**("MyElement", "pri") retourne MyElement()*
- **altova:elements**("MyElement", "") retourne MyElement()*
- **altova:elements**("MyElem", "rip") retourne MyElement()*
- **altova:elements**("MyElements", "rfip") retourne aucune correspondance
- **altova:elements**("MyElement", "Rip") retourne une erreur flag-non reconnu.
- **altova:elements**("MyElement",) retourne une erreur second-argument-manquant.

find-first [altova:]

altova:find-first((Sequence *as item()**), **(Condition(Sequence-Item** *as xs:boolean*))
asitem()? **XP3.1** **XQ3.1**

Cette fonction prend deux arguments. Le premier argument est une séquence d'un ou de plusieurs items de tout type de données. Le second argument, `Condition`, est une référence à une fonction XPath qui prend un argument (possède une arité de **1**) et retourne un booléen. Chaque item de **Sequence** est soumis à son tour à la fonction référencée dans `Condition`. (*Rappel :* cette fonction prend un seul argument.) Le premier item **Sequence** qui cause la fonction dans **Condition** à évaluer à **true()** est retourné en tant que le résultat de **altova:find-first**, et l'itération s'arrête.

Exemples

- **altova:find-first**(5 to 10, function($a) {$a mod 2 = 0}) retourne xs:integer 6
  L'argument **Condition** référence la fonction en ligne XPath 3.0, **function()**, qui déclare une fonction en ligne nommée **$a** puis la définit. Chaque item dans l'argument **Sequence** de **altova:find-first** est passé à son tour à **$a** en tant que sa valeur d'entrée. La valeur d'entrée est testée sur la condition dans la définition de la fonction (`$a mod 2 = 0`). La première valeur d'entrée pour satisfaire cette condition est retournée en tant que le résultat de **altova:find-first** (dans ce cas **6**).

- **altova:find-first**((1 to 10), (function($a) {$a+3=7})) retourne xs:integer 4

© 2016-2022 Altova GmbH

Autres exemples
Si le fichier **C:\Temp\Customers.xml** existe :
- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) retourne xs:string C:\Temp\Customers.xml

Si le fichier **C:\Temp\Customers.xml** n'existe pas et que **http://www.altova.com/index.html** existe :
- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) retourne xs:string http://www.altova.com/index.html

Si le fichier **C:\Temp\Customers.xml** n'existe pas, et que **http://www.altova.com/index.html** n'existe pas non plus :
- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) ne retourne aucun résultat

Notes à propos des exemples indiqués ci-dessus

- La fonction XPath 3.0, doc-available, prend un seul argument de chaîne, qui est utilisé en tant qu'URI, et retourne true si un nœud de document est trouvé à l'URI soumis. (Le document à l'URI soumis doit donc être un document XML.)
- La fonction doc-available peut être utilisée pour **Condition**, le second argument de altova:find-first, parce qu'il ne prend qu'un seul argument (arité=1), parce qu'il prend un item() en tant qu'entrée (une chaîne qui est utilisée en tant qu'URI), et retourne une valeur booléenne.
- Veuillez noter que la fonction doc-available est uniquement référencée, elle n'est pas appelée. Le suffixe #1 qui y est attaché indique une fonction avec une arité de 1. Sous sa forme complète, doc-available#1 signifie simplement : *Utiliser la fonction doc-availabe() à l'arité=1, en l'y passant en tant que son seul argument, chacun à son tour, chacun des items dans la première séquence*. En résultat, chacun des deux chaînes sera passée à **doc-available()**, qui utilise la chaîne en tant qu'URI et teste si un nœud de document existe à l'URI. S'il en existe un, **doc-available()** évalue à true() et cette chaîne est retournée en tant que le résultat de la fonction **altova:find-first**. *Note à propos de la fonction doc-available() : les chemins relatifs sont résolus relativement à l'URI de base actuel, qui est par défaut l'URI du document XML à partir duquel la fonction est chargée.*

find-first-combination [altova:]

altova:find-first-combination((Seq-01 *as item()**)**,* **(Seq-02** *as item()**)**,*
(Condition(Seq-01-Item, Seq-02-Item *as xs:boolean***)) asitem()*** **XP3.1** **XQ3.1**
Cette fonction prend trois arguments :

- Les deux premiers arguments, **Seq-01** et **Seq-02**, sont des séquences d'un ou de plusieurs items de tout type de données.
- Le troisième argument, **Condition**, est une référence à une fonction XPath qui prend deux arguments (a une arité de 2) et retourne un booléen.

Les items de **Seq-01** et **Seq-02** sont passés dans des paires ordonnées (un item de chaque séquence faisant une paire) en tant que les arguments de la fonction dans **Condition**. Les paires sont classées comme suit :

If Seq-01 = X1, X2, X3 ... Xn
And Seq-02 = Y1, Y2, Y3 ... Yn

```
Then (X1 Y1), (X1 Y2), (X1 Y3) ... (X1 Yn), (X2 Y1), (X2 Y2) ... (Xn Yn)
```

La première paire ordonnée qui entraîne la fonction **Condition** à évaluer à **true()** est retournée en tant que le résultat de **altova:find-first-combination**. Veuillez noter que : (i) si la fonction **Condition** itère par le biais des paires d'argument soumises et n'évalue pas une fois à **true()**, alors **altova:find-first-combination** retournera *Aucun résultat* ; (ii) Le résultat de **altova:find-first-combination** sera toujours une paire d'items (de tout type de données) ou aucun item.

Exemples

- **altova:find-first-combination**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32}) retourne la séquence de xs:integers (11, 21)
- **altova:find-first-combination**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33}) retourne la séquence de f xs:integers (11, 22)
- **altova:find-first-combination**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 34}) retourne la séquence de xs:integers (11, 23)

find-first-pair [altova:]

altova:find-first-pair((Seq-01 *as item()**)**,* **(Seq-02** *as item()**)**,* **(Condition(Seq-01-Item, Seq-02-Item** *as xs:boolean*)) **asitem()*** **XP3.1** **XQ3.1**

Cette fonction prend trois arguments :

- Les deux premiers arguments, **Seq-01** et **Seq-02**, sont des séquences d'un ou de plusieurs items de tout type de données.
- Le troisième argument, **Condition**, est une référence à une fonction XPath qui prend deux arguments (a une arité de 2) et retourne un booléen.

Les items de **Seq-01** et **Seq-02** sont passés dans des paires ordonnées en tant que les arguments de la fonction dans **Condition**. Les paires sont classées comme suit :

```
If   Seq-01 = X1, X2, X3 ... Xn
And  Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)
```

La première paire ordonnée qui cause la fonction **Condition** à évaluer à **true()** est retournée en tant que le résultat de **altova:find-first-pair**. Veuillez noter que : (i) Si la fonction **Condition** itère par le biais des paires d'arguments soumis et n'évalue pas une seule fois à **true()**, alors **altova:find-first-pair** retournera *Aucun résultat*; (ii) Le résultat de **altova:find-first-pair** sera toujours une paire d'items (de tout type de données) ou aucun item.

Exemples

- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32}) retourne la séquence de xs:integers (11, 21)
- **altova:find-first-pair**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33}) retourne *Aucun résultat*

Veuillez noter à partir des deux exemples ci-dessus que l'ordonnance des paires est : (11, 21) (12, 22) (13, 23)...(20, 30). C'est pourquoi le second exemple retourne *Aucun résultat* (parce qu'aucune paire ordonnée de donne une somme de 33).

© 2016-2022 Altova GmbH

find-first-pair-pos [altova:]

altova:find-first-pair-pos((Seq-01 *as item()**), **(Seq-02** *as item()**), **(Condition(Seq-01-Item, Seq-02-Item** *as xs:boolean*)) **asxs:integer** XP3.1 XQ3.1

Cette fonction prend trois arguments :

- Les deux premiers arguments, **Seq-01** and **Seq-02**, sont des séquences d'un ou de plusieurs items de tout type de données.
- Le troisième argument, **Condition**, est une référence à une fonction XPath qui prend deux arguments (a une arité de 2) et retourne un booléen.

Les items de **Seq-01** et **Seq-02** sont passés dans des paires ordonnées en tant que les arguments de la fonction dans **Condition**. Les paires sont classées comme suit :

```
If   Seq-01 = X1, X2, X3 ... Xn
And  Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)
```

La position d'index de la première paire ordonnée qui entraîne la fonction **Condition** à évaluer à **true()** est retournée en tant que le résultat de **altova:find-first-pair-pos**. Veuillez noter que si la fonction **Condition** itère par le biais des paires d'arguments soumises et n'évalue pas une seule fois à **true()**, alors **altova:find-first-pair-pos** retournera *Aucun résultat*.

Exemples

- **altova:find-first-pair-pos**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32}) retourne 1
- **altova:find-first-pair-pos**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33}) retourne *Aucun résultat*

Veuillez noter à partir des deux exemples ci-dessus que l'ordonnance des paires est : (11, 21) (12, 22) (13, 23)...(20, 30). dans le premier exemple, la première paire entraîne la fonction **Condition** à évaluer à **true()**, et donc sa position d'index dans la séquence, **1**, est retournée. Le second exemple retourne *Aucun résultat* parce qu'aucune paire ne totalise pas une somme de 33.

find-first-pos [altova:]

altova:find-first-pos((Sequence *as item()**), **(Condition(Sequence-Item** *as xs:boolean*)) **asxs:integer** XP3.1 XQ3.1

Cette fonction prend deux arguments. Le premier argument est une séquence d'un ou de plusieurs items de tout type de données. Le second argument, Condition, est une référence à une fonction XPath qui prend un argument (a une arité de **1**) et retourne une booléenne. Chaque item de **Sequence** est soumis à son tour à la fonction référencée dans Condition. (*Rappel :* cette fonction prend un seul argument.) Le premier item **Sequence** qui cause la fonction dans **Condition** à évaluer à **true()** voit sa position index dans **Sequence** retournée en tant que résultat de **altova:find-first-pos**, et l'itération stoppe.

Exemples

- **altova:find-first-pos**(5 to 10, function($a) {$a mod 2 = 0}) retourne xs:integer 2

L'argument **Condition** référence la fonction en ligne XPath 3.0, **function()**, qui déclare une fonction en ligne nommée **$a** et puis la définit. Chaque item dans l'argument de **Sequence** de **altova:find-first-pos** est passé à son tour à **$a** en tant que sa valeur d'entrée. La valeur d'entrée est testée à la condition dans la définition de la fonction ($a mod 2 = 0). La position d'index dans la séquence de

la première valeur d'entrée pour satisfaire à cette condition est retournée en tant que le résultat de **altova:find-first-pos** (dans ce cas 2, puisque 6, la première valeur (dans la séquence) afin de satisfaire à la condition, est à la position d'index 2 dans la séquence).

- **altova:find-first-pos**((2 to 10), (function($a) {$a+3=7})) retourne xs:integer 3

Autres exemples
Si le fichier **C:\Temp\Customers.xml** existe :
- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) retourne 1

Si le fichier **C:\Temp\Customers.xml** n'existe pas, et que **http://www.altova.com/index.html** existe :
- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) retourne 2

Si le fichier **C:\Temp\Customers.xml** n'existe pas, et que **http://www.altova.com/index.html** n'existe pas non plus :
- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) ne retourne aucun résultat

Notes à propos des exemples donnés ci-dessus

- La fonction XPath 3.0, doc-available, prend un seul argument de chaîne qui est utilisé en tant qu'un URI, et retourne true si un nœud de document est trouvé à l'URI soumis. (Le document à l'URI soumis doit donc être un document XML.)
- La fonction doc-available peut être utilisée pour **Condition**, le second argument de altova:find-first-pos, parce qu'il ne prend qu'un seul argument (arité=1), parce qu'il prend un item() en tant qu'entrée (une chaîne qui est utilisée en tant qu'un URI), et retourne une valeur booléenne.
- Veuillez noter que la fonction doc-available est uniquement référencée, elle n'est pas appelée. Le suffixe #1 qui y est attaché indique une fonction avec une arité de 1. dans sa totalité, doc-available#1 signifie simplement : *Utiliser la fonction doc-availabe() qui a arité=1, y passant, en tant que son argument simple, chacun à son tour, chaque item dans la première séquence*. En tant que résultat. chacune des deux chaînes sera passée à **doc-available()**, qui utilise la chaîne en tant qu'un URI et teste si un nœud de document existe à l'URI. Si c'est le cas, la fonction **doc-available()** évalue à true() et la position de l'index de cette chaîne dans la séquence est retournée en tant que le résultat de la fonction **altova:find-first-pos**. *Note à propos de la fonction doc-available() : les chemins relatifs sont résolus relativement à l'URI de base actuel, qui par défaut est l'URI du document XML à partir duquel la fonction est chargée.*

for-each-attribute-pair [altova:]

altova:for-each-attribute-pair(Seq1 *as element()?*, **Seq2** *as element()?*, **Function** *as function()*) **asitem()* XP3.1 XQ3.1**

Les premiers deux arguments identifient deux éléments, dont les attributs sont utilisés pour générer des paires d'attribut, dans laquelle un attribut d'une paire est obtenu depuis le premier élément et l'autre attribut est obtenu depuis le second élément. Les paires d'attribut sont sélectionnées sur le fait qu'ils présentent le même nom, et les paires sont classées par ordre alphabétique (sur leur nom) dans un

© 2016-2022 Altova GmbH

ensemble. Si, pour un attribut, aucun attribut correspondant n'existe dans l'autre élément, la paire sera "disjointe", ce qui signifie qu'elle consiste en un seul membre. L'item de fonction (troisième argument `Function`) est appliqué séparément à chaque paire dans la séquence des paires (joints et disjointe), résultant en une sortie qui est une séquence d'items.

Exemples

- **altova:for-each-attribute-pair**(/Example/Test-A, /Example/Test-B, function($a, $b) {$a+b}) retourne ...

  (2, 4, 6) si
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />

  (2, 4, 6) si
  <Test-A att2="2" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />

  (2, 6) si
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />

  Note : Le résultat (2 6) est obtenu par le biais de l'action suivante : (1+1 ()+2 3+3 4+()). Si un des opérandes est la séquence vide, comme c'est le cas des items 2 et 4, le résultat de l'addition est une séquence vide.

- **altova:for-each-attribute-pair**(/Example/Test-A, /Example/Test-B, concat#2) retourne ...

  (11 22 33) si
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />

  (11 2 33 4) si
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />

for-each-combination [altova:]

altova:for-each-combination(FirstSequence *as item()**, **SecondSequence** *as item()**, **Function($i,$j){$i || $j}) asitem()* XP3.1 XQ3.1**

Les items des deux séquences dans les deux premiers arguments sont combinés de manière à ce que chaque item de la première séquence est combiné, dans l'ordre, une fois avec chaque item de la seconde séquence. La fonction donnée en tant que le troisième argument est appliquée à chaque combinaison dans la séquence résultante, entraînant une sortie qui est une séquence d'items (*voir exemple*).

Exemples

- **altova:for-each-combination**(('a', 'b', 'c'), ('1', '2', '3'), function($i, $j) {$i || $j}) retourne ('a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3')

for-each-matching-attribute-pair [altova:]

altova:for-each-matching-attribute-pair(Seq1 *as element()?*, **Seq2** *as element()?*,

Function *as function()*) **asitem()*** **XP3.1** **XQ3.1**

Les premiers deux arguments identifient deux éléments, dont les attributs sont utilisés pour générer des paires d'attribut, dans laquelle un attribut d'une paire est obtenu depuis le premier élément et l'autre attribut est obtenu depuis le second élément. Les paires d'attribut sont sélectionnées sur le fait qu'ils présentent le même nom, et les paires sont classées par ordre alphabétique (sur leur nom) dans un ensemble. Si, pour un attribut, aucun attribut correspondant n'existe dans l'autre élément, aucune paire ne sera générée. L'item de fonction (troisième argument `Function`) est appliqué séparément à chaque paire dans la séquence des paires, résultant en une sortie qui est une séquence d'items.

Exemples

- **altova:for-each-matching-attribute-pair**(/Example/Test-A, /Example/Test-B, function($a, $b){$a+b}) retourne ...

  (2, 4, 6) si
  ```
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />
  ```

  (2, 4, 6) si
  ```
  <Test-A att2="2" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />
  ```

  (2, 6) si
  ```
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att3="1" />
  ```

- **altova:for-each-matching-attribute-pair**(/Example/Test-A, /Example/Test-B, concat#2) retourne ...

  (11, 22, 33) si
  ```
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />
  ```

  (11, 33) si
  ```
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />
  ```

substitute-empty [altova:]

altova:substitute-empty(FirstSequence *as item()**, **SecondSequence** *as item()*) **asitem()*** **XP3.1** **XQ3.1**

Si `FirstSequence` est vide, retourne `SecondSequence`. Si `FirstSequence` n'est pas vide, retourne `FirstSequence`.

Exemples

- **altova:substitute-empty**((1,2,3), (4,5,6)) retourne (1,2,3)
- **altova:substitute-empty**((), (4,5,6)) retourne (4,5,6)

© 2016-2022 Altova GmbH

30.2.1.8 Fonctions XPath/XQuery : Chaîne

Les fonctions d'extension de chaîne d'Altova peuvent être utilisées dans les expressions XPath et XQuery et proposent des fonctions supplémentaires pour le traitement des données. Les fonctions dans cette section peuvent être utilisées avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova. Ils sont disponibles dans des contextes XPath/XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

camel-case [altova:]

altova:camel-case(InputString *as xs:string*) **asxs:string** XP3.1 XQ3.1

Retourne la chaîne d'entrée **InputString** en CamelCase. La chaîne est analysée en utilisant l'expression régulière `'\s'` (qui est un raccourci pour le caractère d'espace blanc). Le premier caractère non-espace blanc après un espace blanc ou une séquence de plusieurs espaces blancs est mis en majuscule. Le premier caractère dans la chaîne de sortie est mis en majuscule.

Exemples

- `altova:camel-case("max")` retourne `Max`
- `altova:camel-case("max max")` retourne `Max Max`
- `altova:camel-case("file01.xml")` retourne `File01.xml`
- `altova:camel-case("file01.xml file02.xml")` retourne `File01.xml File02.xml`
- `altova:camel-case("file01.xml  file02.xml")` retourne `File01.xml  File02.xml`
- `altova:camel-case("file01.xml -file02.xml")` retourne `File01.xml -file02.xml`

altova:camel-case(InputString *as xs:string*, **SplitChars** *as xs:string*, **IsRegex** *as xs:boolean*) **asxs:string** XP3.1 XQ3.1

Convertit la chaîne d'entrée **InputString** en camel case en utilisant **SplitChars** pour déterminer le/s caractère/s qui déclenche/nt la prochaine mise en majuscule. **SplitChars** est utilisé en tant qu'expression régulière quand **IsRegex = true()**, ou en tant que caractères normaux quand **IsRegex = false()**. Le premier caractère dans la chaîne de sortie est mis en majuscule.

Exemples

- `altova:camel-case("setname getname", "set|get", true())` retourne `setName getName`
- `altova:camel-case("altova\documents\testcases", "\", false())` retourne `Altova\Documents\Testcases`

char [altova:]

`altova:char(Position` *`as xs:integer`*`)` `asxs:string` `XP3.1` `XQ3.1`
Retourne une chaîne contenant le caractère à la position spécifiée par l'argument `Position`, dans la chaîne obtenue en convertissant la valeur de l'item contextuel en `xs:string`. La chaîne de résultat sera vide si aucun caractère n'existe à l'index soumis par l'argument `Position`.

Exemples

Si l'item contextuel est **`1234ABCD`**:

- `altova:char(2)` retourne `2`
- `altova:char(5)` retourne `A`
- `altova:char(9)` retourne la chaîne vide.
- `altova:char(-2)` retourne la chaîne vide.

`altova:char(InputString` *`as xs:string`*`, `**`Position`** *`as xs:integer`*`)` `asxs:string` `XP3.1` `XQ3.1`
Retourne une chaîne contenant la caractère à la position spécifiée par l'argument `Position`, dans la chaîne soumise en tant que l'argument `InputString`. La chaîne de résultat sera vide si aucun caractère n'existe à l'index soumis par l'argument `Position`.

Exemples

- `altova:char("2014-01-15", 5)` retourne `-`
- `altova:char("USA", 1)` retourne `U`
- `altova:char("USA", 10)` retourne la chaîne vide.
- `altova:char("USA", -2)` retourne la chaîne vide.

create-hash-from-string[altova:]

`altova:create-hash-from-string(InputString` *`as xs:string`*`)` `asxs:string` `XP2` `XQ1` `XP3.1` `XQ3.1`
`altova:create-hash-from-string(InputString` *`as xs:string`*`, `**`HashAlgo`** *`as xs:string`*`)`
`asxs:string` `XP2` `XQ1` `XP3.1` `XQ3.1`
Génère un string de hashage depuis `InputString` en utilisant l'algorithme de hashage spécifié par l'argument `HashAlgo`. Les algorithmes de hashage suivants peuvent être spécifiés (en majuscule ou en minuscule) : **`MD5`**, **`SHA-1`**, **`SHA-224`**, **`SHA-256`**, **`SHA-384`**, **`SHA-512`**. Si le deuxième argument n'est pas spécifié (*voir la première signature ci-dessus*), l'algorithme de hashage **`SHA-256`** sera utilisé.

Exemples

- `altova:create-hash-from-string('abc')` retourne un string de hashage généré en utilisant l'algorithme de hachage **`SHA-256`**.
- `altova:create-hash-from-string('abc', 'md5')` retourne un string de hashage généré en utilisant l'algorithme de hachage **`MD5`**.
- `altova:create-hash-from-string('abc', 'MD5')` retourne un string de hashage généré en utilisant l'algorithme de hachage **`MD5`**.

first-chars [altova:]

© 2016-2022 Altova GmbH

altova:first-chars(X-Number *as xs:integer*) **asxs:string** XP3.1 XQ3.1
Retourne une chaîne contenant le premier `X-Number` des caractères de la chaîne obtenue en convertissant la valeur de l'item de contexte en `xs:string`.

Exemples

Si l'item de contexte est **1234ABCD** :

- **altova:first-chars**(2) retourne 12
- **altova:first-chars**(5) retourne 1234A
- **altova:first-chars**(9) retourne 1234ABCD

altova:first-chars(InputString *as xs:string*, **X-Number** *as xs:integer*) **asxs:string** XP3.1 XQ3.1
Retourne une chaîne contenant le premier `X-Number` des caractères de la chaîne soumise en tant que l'argument `InputString`.

Exemples

- **altova:first-chars**("2014-01-15", 5) retourne 2014-
- **altova:first-chars**("USA", 1) retourne U

format-string [altova:]

altova:format-string(InputString *as xs:string*, **FormatSequence** *as item()**) **asxs:string** XP3.1 XQ3.1
Le string d'entrée (premier argument) contient des paramètres de position (`%1`, `%2`, etc). Chaque paramètre est remplacé par l'item de string qui est situé dans la position correspondante dans la séquence de format (soumise en tant que le second argument). Donc le premier item dans la séquence de format remplace de paramètre de positionnement `%1`, le second item remplace `%2`, etc. La fonction retourne ce string formaté qui contient les remplacements. Si aucun string n'existe pour un paramètre de positionnement, alors le paramètre de positionnement lui-même est retourné. Cela se produit lorsque l'index d'un paramètre de positionnement est supérieur au nombre d'items dans la séquence de format.

Exemples

- **altova:format-string**('Hello %1, %2, %3', ('Jane','John','Joe')) retourne "Hello Jane, John, Joe"
- **altova:format-string**('Hello %1, %2, %3', ('Jane','John','Joe', 'Tom')) retourne "Hello Jane, John, Joe"
- **altova:format-string**('Hello %1, %2, %4', ('Jane','John','Joe', 'Tom')) retourne "Hello Jane, John, Tom"
- **altova:format-string**('Hello %1, %2, %4', ('Jane','John','Joe')) retourne "Hello Jane, John, %4"

last-chars [altova:]

altova:last-chars(X-Number *as xs:integer*) **asxs:string** XP3.1 XQ3.1
Retourne une chaîne contenant le dernier `X-Number` de caractères de la chaîne obtenue en convertissant la valeur de l'item contextuel en `xs:string`.

Exemples

Si l'item contextuel est **1234ABCD** :

- **altova:last-chars**(2) retourne CD
- **altova:last-chars**(5) retourne 4ABCD

- **altova:last-chars**(9) retourne 1234ABCD

altova:last-chars(InputString *as xs:string*, **X-Number** *as xs:integer*) **asxs:string** **XP3.1** **XQ3.1**
Retourne une chaîne contenant le dernier X-Number de caractères de la chaîne soumise en tant que l'argument InputString.

Exemples

- **altova:last-chars**("2014-01-15", 5) retourne 01-15
- **altova:last-chars**("USA", 10) retourne USA

pad-string-left [altova:]

altova:pad-string-left(StringToPad *as xs:string*, **StringLength** *as xs:integer*, **PadCharacter** *as xs:string*) **asxs:string** **XP3.1** **XQ3.1**
L'argument PadCharacter est un caractère unique. Il est bourré à la gauche de la chaîne pour augmenter le nombre de caractères dans StringToPad de manière à ce que ce nombre soit équivalent à la valeur d'entier de l'argument StringLength. L'argument StringLength peut avoir toute valeur d'entier (positive ou négative), mais le padding n'aura lieu que si la valeur de StringLength est supérieure au nombre de caractères dans StringToPad. Si StringToPad. comporte plus de caractères que la valeur de StringLength, alors StringToPad ne sera pas modifié.

Exemples

- **altova:pad-string-left**('AP', 1, 'Z') retourne 'AP'
- **altova:pad-string-left**('AP', 2, 'Z') retourne 'AP'
- **altova:pad-string-left**('AP', 3, 'Z') retourne 'ZAP'
- **altova:pad-string-left**('AP', 4, 'Z') retourne 'ZZAP'
- **altova:pad-string-left**('AP', -3, 'Z') retourne 'AP'
- **altova:pad-string-left**('AP', 3, 'YZ') retourne une erreur pad-character-too-long

pad-string-right [altova:]

altova:pad-string-right(StringToPad *as xs:string*, **StringLength** *as xs:integer*, **PadCharacter** *as xs:string*) **asxs:string** **XP3.1** **XQ3.1**
L'argument PadCharacter est un caractère unique. Il est bourré à la droite de la chaîne pour augmenter le nombre de caractères dans StringToPad de manière à ce que ce nombre soit équivalent à la valeur d'entier de l'argument StringLength. L'argument StringLength peut avoir toute valeur d'entier (positive ou négative), mais le padding n'aura lieu que si la valeur de StringLength est supérieure au nombre de caractères dans StringToPad. Si StringToPad. comporte plus de caractères que la valeur de StringLength, alors StringToPad ne sera pas modifié.

Exemples

- **altova:pad-string-right**('AP', 1, 'Z') retourne 'AP'
- **altova:pad-string-right**('AP', 2, 'Z') retourne 'AP'
- **altova:pad-string-right**('AP', 3, 'Z') retourne 'APZ'
- **altova:pad-string-right**('AP', 4, 'Z') retourne 'APZZ'
- **altova:pad-string-right**('AP', -3, 'Z') retourne 'AP'
- **altova:pad-string-right**('AP', 3, 'YZ') retourne une erreur pad-character-too-long

© 2016-2022 Altova GmbH

repeat-string [altova:]

altova:repeat-string(InputString *as xs:string*, **Repeats** *as xs:integer*) **asxs:string** **XP2** **XQ1** **XP3.1** **XQ3.1**

Génère une chaîne qui est composée du premier argument `InputString` répété `Repeats` nombre de fois.

Exemples

- **altova:repeat-string**("Altova #", 3) retourne "Altova #Altova #Altova #"

substring-after-last [altova:]

altova:substring-after-last(MainString *as xs:string*, **CheckString** *as xs:string*)
asxs:string **XP3.1** **XQ3.1**

Si `CheckString` est trouvé dans `MainString`, alors la sous-chaîne qui se produit après `CheckString` dans `MainString` est retournée. Si `CheckString` n'est pas trouvé dans `MainString`, la chaîne vide est retournée. Si `CheckString` est une chaîne vide, alors `MainString` est retourné dans sa totalité. S'il y a plus d'une survenance de `CheckString` dans `MainString`, alors la sous-chaîne après la dernière survenance de `CheckString` est retournée.

Exemples

- **altova:substring-after-last**('ABCDEFGH', 'B') retourne 'CDEFGH'
- **altova:substring-after-last**('ABCDEFGH', 'BC') retourne 'DEFGH'
- **altova:substring-after-last**('ABCDEFGH', 'BD') retourne ''
- **altova:substring-after-last**('ABCDEFGH', 'Z') retourne ''
- **altova:substring-after-last**('ABCDEFGH', '') retourne 'ABCDEFGH'
- **altova:substring-after-last**('ABCD-ABCD', 'B') retourne 'CD'
- **altova:substring-after-last**('ABCD-ABCD-ABCD', 'BCD') retourne ''

substring-before-last [altova:]

altova:substring-before-last(MainString *as xs:string*, **CheckString** *as xs:string*)
asxs:string **XP3.1** **XQ3.1**

Si `CheckString` est trouvé dans `MainString`, alors la sous-chaîne qui se produit avant `CheckString` dans `MainString` est retournée. Si `CheckString` n'est pas trouvé dans `MainString`, ou si `CheckString` est une chaîne vide, la chaîne vide est retournée. S'il y a plus d'une survenance de `CheckString` dans `MainString`, alors la sous-chaîne avant la dernière survenance de `CheckString` est retournée.

Exemples

- **altova:substring-before-last**('ABCDEFGH', 'B') retourne 'A'
- **altova:substring-before-last**('ABCDEFGH', 'BC') retourne 'A'
- **altova:substring-before-last**('ABCDEFGH', 'BD') retourne ''
- **altova:substring-before-last**('ABCDEFGH', 'Z') retourne ''
- **altova:substring-before-last**('ABCDEFGH', '') retourne ''
- **altova:substring-before-last**('ABCD-ABCD', 'B') retourne 'ABCD-A'
- **altova:substring-before-last**('ABCD-ABCD-ABCD', 'ABCD') retourne 'ABCD-ABCD-'

substring-pos [altova:]

altova:substring-pos(StringToCheck *as xs:string*, **StringToFind** *as xs:string*)
asxs:integer **XP3.1** **XQ3.1**

Retourne la position de caractère de la première occurrence de `StringToFind` dans la chaîne

StringToCheck. La position du caractère est retournée en tant qu'un entier. Le premier caractère de StringToCheck a la position 1. Si StringToFind ne se produit pas dans le cadre de StringToCheck, l'entier 0 est retourné. Pour contrôler la deuxième occurrence ou une occurrence ultérieure de StringToCheck, utiliser la signature suivante de cette fonction.

Exemples

- **altova:substring-pos**('Altova', 'to') retourne 3
- **altova:substring-pos**('Altova', 'tov') retourne 3
- **altova:substring-pos**('Altova', 'tv') retourne 0
- **altova:substring-pos**('AltovaAltova', 'to') retourne 3

altova:substring-pos(StringToCheck *as xs:string*, **StringToFind** *as xs:string*, **Integer** *as xs:integer*) **asxs:integer** XP3.1 XQ3.1

Retourne la position de caractère de StringToFind dans la chaîne, StringToCheck. La recherche de StringToFind commence à partir de la position de caractère indiquée par l'argument Integer ; la sous-chaîne du caractère avant cette position n'est pas recherchée. Néanmoins, l'entier retourné, est la position de la chaîne trouvée dans le cadre de la chaîne *entière*, StringToCheck. Cette signature est utile pour trouver la deuxième position ou une position ultérieure d'une chaîne qui se produit plusieurs fois avec StringToCheck. Si StringToFind ne se produit pas dans le cadre de StringToCheck, l'entier 0 est retourné.

Exemples

- **altova:substring-pos**('Altova', 'to', 1) retourne 3
- **altova:substring-pos**('Altova', 'to', 3) retourne 3
- **altova:substring-pos**('Altova', 'to', 4) retourne 0
- **altova:substring-pos**('Altova-Altova', 'to', 0) retourne 3
- **altova:substring-pos**('Altova-Altova', 'to', 4) retourne 10

trim-string [altova:]

altova:trim-string(InputString *as xs:string*) **asxs:string** XP3.1 XQ3.1

Cette fonction prend un argument xs:string, supprime tout espace blanc de tête et de fin et retourne une xs:string « nettoyée ».

Exemples

- **altova:trim-string**(" Hello World ") retourne "Hello World"
- **altova:trim-string**("Hello World ") retourne "Hello World"
- **altova:trim-string**(" Hello World") retourne "Hello World"
- **altova:trim-string**("Hello World") retourne "Hello World"
- **altova:trim-string**("Hello World") retourne "Hello World"

trim-string-left [altova:]

altova:trim-string-left(InputString *as xs:string*) **asxs:string** XP3.1 XQ3.1

Cette fonction prend un argument xs:string, supprime tout espace blanc de tête, et retourne une xs:string nettoyée à gauche.

Exemples

- **altova:trim-string-left**(" Hello World ") retourne "Hello World "
- **altova:trim-string-left**("Hello World ") retourne "Hello World "

© 2016-2022 Altova GmbH

- `altova:trim-string-left("  Hello World")` retourne `"Hello World"`
- `altova:trim-string-left("Hello World")` retourne `"Hello World"`
- `altova:trim-string-left("Hello  World")` retourne `"Hello World"`

trim-string-right [altova:]

`altova:trim-string-right(InputString` *`as xs:string`***`) asxs:string`** XP3.1 XQ3.1

Cette fonction prend un argument `xs:string`, supprime tout espace blanc de fin de ligne, et retourne une `xs:string` nettoyée à droite.

Exemples

- `altova:trim-string-right("  Hello World  ")` retourne `"  Hello World"`
- `altova:trim-string-right("Hello World  ")` retourne `"Hello World"`
- `altova:trim-string-right("  Hello World")` retourne `"  Hello World"`
- `altova:trim-string-right("Hello World")` retourne `"Hello World"`
- `altova:trim-string-right("Hello  World")` retourne `"Hello  World"`

30.2.1.9 Fonctions XPath/XQuery : Divers

L'objectif général suivant des fonctions d'extension XPath/XQuery sont prises en charge dans la version actuelle de XMLSpy et celles-ci peuvent être utilisées dans (i) des expressions XPath dans un contexte XSLT, ou dans (ii) des expressions XQuery dans un document XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova,** `http://www.altova.com/xslt-extensions`, et sont indiquées dans cette section par le préfixe `altova:`, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

decode-string [altova:]

`altova:decode-string(Input` *`as xs:base64Binary`***`) as xs:string`** XP3.1 XQ3.1

altova:decode-string(Input *as xs:base64Binary*, **Encoding** *as xs:string*) **as xs:string** XP3.1 XQ3.1

Décode l'entrée base64Binary soumise à un string en utilisant l'encodage spécifié. Si aucun codage n'est spécifié, l'encodage UTF-8 est utilisé. Les encodages suivants sont pris en charge: `US-ASCII`, `ISO-8859-1`, `UTF-16`, `UTF-16LE`, `UTF-16BE`, `ISO-10646-UCS2`, `UTF-32`, `UTF-32LE`, `UTF-32BE`, `ISO-10646-UCS4`

Exemples

- **altova:decode-string**(`$XML1/MailData/Meta/b64B`) retourne l'entrée base64Binary en tant qu'un string encodé en UTF-8
- **altova:decode-string**(`$XML1/MailData/Meta/b64B, "UTF-8"`) retourne l'entrée base64Binary en tant qu'un string encodé en UTF-8
- **altova:decode-string**(`$XML1/MailData/Meta/b64B, "ISO-8859-1"`) retourne l'entrée base64Binary en tant qu'un string encodé en ISO-8859-1

.

get-temp-folder [altova:]

altova:get-temp-folder() as `xs:string` XP2 XQ1 XP3.1 XQ3.1

Cette fonction ne prend aucun argument. Elle retourne le chemin vers le dossier temporaire de l'utilisateur actuel.

Exemples

- **altova:get-temp-folder**() retournerait, sur une machine Windows, quelque chose du genre `C:\Users\<UserName>\AppData\Local\Temp\` en tant que `xs:string`.

generate-guid [altova:]

altova:generate-guid() asxs:string XP2 XQ1 XP3.1 XQ3.1

Génère un string GUID unique.

Exemples

- **altova:generate-guid**() retourne (par exemple) `85F971DA-17F3-4E4E-994E-99137873ACCD`

high-res-timer [altova:]

altova:high-res-timer() asxs:double XP3.1 XQ3.1

Retourne une valeur de minuteur haute résolution en secondes. Un minuteur de haute résolution, lorsqu'il est présent dans un système, permet des mesures temporelles de haute précision lorsque celles-ci sont requises (par exemples pour des animations et pour déterminer précisément l'heure d'exécution du code). Cette fonction fournit la résolution du minuteur haute résolution du système.

Exemples

- **altova:high-res-timer**() retourne quelque chose comme `'1.16766146154566E6'`

parse-html [altova:]

altova:parse-html(HTMLText *as xs:string*) **asnode()** XP3.1 XQ3.1

L'argument `HTMLText` est un string qui contient le texte d'un document HTML. La fonction crée une

arborescence HTML depuis le string. Le string soumis peut contenir l'élément HTML ou pas. Dans tous les cas, l'élément racine de l'arborescence est un élément nommé **HTML**. Il est préférable de s'assurer que le code HTML dans le string soumis est un HTML valide.

Exemples

- `altova:parse-html("<html><head/><body><h1>Header</h1></body></html>")` crée une arborescence HTML depuis le string soumis

sleep[altova:]

altova:sleep(Millisecs *as xs:integer*) **asempty-sequence() XP2 XQ1 XP3.1 XQ3.1**
Suspend l'exécution de l'opération actuelle pour le nombre de millisecondes donné par l'argument `Millisecs`.

Exemples

- `altova:sleep(1000)` suspend l'exécution de l'opération actuelle pour 1000 millisecondes.

[Top]

30.2.1.10 Fonctions de graphique

Les fonctions de graphique recensées ci-dessous vous permettent de créer, de générer et d'enregistrer des graphiques en tant qu'images. Elles sont supportées dans la version actuelle de votre produit Altova de la manière décrite ci-dessous. Néanmoins, veuillez noter que dans des versions futures de votre produit, le support de l'une ou de plusieurs de ces fonctions peut être interrompu, en outre, le comportement de certaines fonctions peut changer. Veuillez consulter la documentation des publications futures pour obtenir des informations concernant le support des fonctions d'extension Altova dans cette publication.

Les fonctions de graphique sont des **fonctions XPath** (pas des fonctions XSLT), et sont organisées dans deux groupes :

- Fonctions pour générer et enregistrer des graphiques
- Fonctions pour créer des graphiques

Note : Les fonctions de graphique sont supportées uniquement dans les **produits de serveur d'Altova** et les **produits d'édition Enterprise de Altova**.

Note : Les formats d'image pris en charge pour les graphiques dans les éditions de serveur sont `jpg`, `png` et `bmp`. La meilleure option est png parce que ce format est comprimé sans pertes. Dans les éditions Enterprise, les formats pris en charge sont `jpg`. `png`, `bmp` et `gif`.

Fonctions de génération et d'enregistrement des graphiques

Ces fonctions prennent l'objet de graphique (obtenu avec les fonctions de création de graphique) et soit génèrent une image soit enregistrent une image dans le fichier

altova:generate-chart-image ($chart, $width, $height, $encoding) en tant que atomic

alors que

- `$chart` est l'item d'extension de graphique obtenu avec la fonction `altova:create-chart`
- `$width` et `$height` doivent être spécifiés avec une unité de longueur
- `$encoding` peut être `binarytobase64` ou `binarytobase16`

La fonction retourne l'image de graphique dans le codage spécifié.

`altova:generate-chart-image` `($chart, $width, $height, $encoding, $imagetype) en tant que atomic`

alors que

- `$chart` est l'item d'extension de graphique obtenu avec la fonction `altova:create-chart`
- `$width` et `$height` doivent être spécifiés avec une unité de longueur
- `$encoding` peut être `base64Binary` ou `hexBinary`
- `$imagetype` peut être un des formats d'image suivants : `png`, `gif`, `bmp`, `jpg`, `jpeg`. Veuillez noter que `gif` n'est pas pris en charge pour les produits de serveur. *Voir également la note en haut de la page*.

La fonction retourne l'image de graphique dans le format d'encodage et d'image spécifié.

`altova:save-chart-image` `($chart, $filename, $width, $height) en tant que empty()` ***(Windows uniquement)***

alors que

- `$chart` est l'item d'extension de graphique obtenu avec la fonction `altova:create-chart`
- `$filename` est le nom du fichier et le chemin sous lequel l'image de graphique doit être enregistré
- `$width` et `$height` doit être spécifié avec une unité de longueur

La fonction enregistre l'image de graphique sous le fichier spécifié dans `$filename`.

`altova:save-chart-image` `($chart, $filename, $width, $height, $imagetype) en tant que empty()` ***(Windows uniquement)***

alors que

- `$chart` est l'item d'extension de graphique obtenu avec la fonction `altova:create-chart`
- `$filename` est le nom du fichier et le chemin sous lequel l'image de graphique doit être enregistré
- `$width` et `$height` doivent être spécifiés avec une unité de longueur
- `$imagetype` peut être un des formats d'image suivants : `png`, `gif`, `bmp`, `jpg`, `jpeg`. Veuillez noter que `gif` n'est pas pris en charge pour les produits de serveur. *Voir également la note en haut de la page*.

La fonction enregistre l'image de graphique sous le fichier spécifié dans `$filename` dans le format d'image spécifié.

© 2016-2022 Altova GmbH

Fonctions de création de graphiques

Les fonctions suivantes sont utilisées pour créer des graphiques.

altova:create-chart($chart-config, $chart-data-series*) en tant que chart extension item

alors que

- $chart-config est l'item d'extension chart-config obtenu avec la fonction altova:create-chart-config ou via la fonction altova:create-chart-config-from-xml
- $chart-data-series est l'item d'extension chart-data-series obtenu avec la fonction altova:create-chart-data-series ou la fonction altova:create-chart-data-series-from-rows

La fonction retourne un item d'extension de graphique qui est créé à partir des données fournies par le biais des arguments.

altova:chart($chart-config, $chart-data-series*) en tant que chart extension item

où

- $chart-config est l'item d'extension chart-config. Il s'agit d'une série non ordonnée de quatre clés : paires de valeur, où les quatre clés sont "width", "height", "title", et "kind". Les valeurs de width et height sont des entiers et spécifient la largeur et la hauteur du graphique en pixels. La valeur de kind est une de : Pie, Pie3d, BarChart, BarChart3d, BarChart3dGrouped, LineChart, ValueLineChart, RoundGauge, BarGauge.
- $chart-data-series forme un array de taille 3, où chaque array définit une série de données de graphique. Chaque array est composé du : (i) nom de la série de données, (ii) des valeurs de l'axe X, (iii) des valeurs de l'axe Y. Plusieurs séries de données peuvent être soumises ; dans l'exemple ci-dessous les deux arrays donnent respectivement des données pour les températures mensuelles minimum et maximum.

La fonction retourne un item d'extension chart, qui est créé depuis les données fournies par le biais des arguments.

Exemple:
```
altova:chart( map{'width':800, 'height':600, "kind":"LineChart", "title":"Monthly Temperatures"}, (['Min', $temps/Month, $temps/Month/@min], ['Max', $temps/Month, $temps/Month/@max]) )
```

altova:create-chart-config($type-name, $title) en tant qu'item d'extension chart-config

alors que

- $type-name spécifie le type de graphique à créer : Pie, Pie3d, BarChart, BarChart3d, BarChart3dGrouped, LineChart, ValueLineChart, RoundGauge, BarGauge
- $title est le nom du graphique

La fonction retourne un item d'extension chart-config contenant les informations de configuration du graphique.

altova:create-chart-config-from-xml($xml-struct) en tant qu'item d'extension chart-config

alors que

- $xml-struct est la structure XML contenant les informations de configuration du graphique

La fonction retourne un item d'extension chart-config contenant les informations de configuration du graphique. Cette information est fournie dans un fragment de données XML.

altova:create-chart-data-series($series-name?, $x-values*, $y-values*) en tant qu'item d'extension chart-data-series

alors que

- $series-name spécifie le nom de la série
- $x-values donne la liste des valeurs de l'axe X
- $y-values donne la liste des valeurs de l'axe Y

La fonction retourne un item d'extension chart-data-series contenant les données pour la construction du graphique : c'est à dire, les noms de la série et les données des Axes.

altova:create-chart-data-row(x, y1, y2, y3, ...) en tant qu'item d'extension chart-data-x-*N*y-row

alors que

- x est la valeur de la colonne de l'axe X de la rangée des données de graphique
- yN sont les valeurs des colonnes de l'axe Y

La fonction retourne un item d'extension chart-data-x-Ny-row qui contient les données pour la colonne de l'axe X et les colonnes de l'axe Y d'une seule série.

altova:create-chart-data-series-from-rows($series-names as xs:string*, $row*) en tant qu'item d'extension chart-data-series

alors que

- $series-name est le nom de la série à créer
- $row est l'item d'extension chart-data-x-Ny-row qui doit être créé en tant qu'une série

La fonction retourne un item d'extension chart-data-series qui contient les données pour l'axe X et l'axe Y de la série.

altova:create-chart-layer($chart-config, $chart-data-series*) en tant qu'item d'extension chart-layer

alors que

- $chart-config est l'item d'extension chart-config obtenu avec la fonction altova:create-chart-config ou via la fonction altova:create-chart-config-from-xml

© 2016-2022 Altova GmbH

- `$chart-data-series` est l'item d'extension chart-data-series obtenu avec la fonction `altova:create-chart-data-series` ou la fonction `altova:create-chart-data-series-from-rows`

La fonction retourne un item d'extension chart-layer qui contient des données chart-layer.

`altova:create-multi-layer-chart``($chart-config, $chart-data-series*, $chart-layer*)`

alors que

- `$chart-config` est l'item d'extension chart-config obtenu avec la fonction `altova:create-chart-config` ou via la fonction `altova:create-chart-config-from-xml`
- `$chart-data-series` est l'item d'extension chart-data-series obtenu avec la fonction `altova:create-chart-data-series` ou la fonction `altova:create-chart-data-series-from-rows`
- `$chart-layer` est l'item d'extension chart-layer obtenu avec la fonction `altova:create-chart-layer`

La fonction retourne un item multi-layer-chart.

`altova:create-multi-layer-chart``($chart-config, $chart-data-series*, $chart-layer*, xs:boolean $mergecategoryvalues)`

alors que

- `$chart-config` est l'item d'extension chart-config obtenu avec la fonction `altova:create-chart-config` ou via la fonction `altova:create-chart-config-from-xml`
- `$chart-data-series` est l'item d'extension chart-data-series obtenu avec la fonction `altova:create-chart-data-series` ou la fonction `altova:create-chart-data-series-from-rows`
- `$chart-layer` est l'item d'extension chart-layer obtenu avec la fonction `altova:create-chart-layer`
- `$mergecategoryvalues` fusionne les valeurs de plusieurs séries de données si `true`, ne fusionne pas si `false`

La fonction retourne un item multi-layer-chart.

30.2.1.10.1 Structure XML des données de graphique

Ci-dessous, vous trouverez la structure XML des données de graphique, telle qu'elle pourrait s'afficher pour les Fonctions d'extension Altova pour les graphiques. Cela affecte l'apparence du graphique spécifique. Tous les éléments ne sont pas utilisés pour tous les types de graphiques, par ex. l'élément `<Pie>` est ignoré pour les graphiques à barres.

Note : Les fonctions de graphique sont uniquement prises en charge dans les **Édition Enterprise et Server** des produits Altova.

```
<chart-config>
```

<General
SettingsVersion="1" *must be provided*
ChartKind="BarChart" *Pie, Pie3d, BarChart, StackedBarChart, BarChart3d, BarChart3dGrouped, LineChart, ValueLineChart, AreaChart, StackedAreaChart, RoundGauge, BarGauge, CandleStick*
BKColor="#ffffff" *Color*
BKColorGradientEnd="#ffffff" *Color. In case of a gradient, BKColor and BKColorGradientEnd define the gradient's colors*
BKMode="#ffffff" *Solid, HorzGradient, VertGradient*
BKFile="Path+Filename" *String. If file exists, its content is drawn over the background.*
BKFileMode="Stretch" *Stretch, ZoomToFit, Center, Tile*
ShowBorder="1" *Bool*
PlotBorderColor="#000000" *Color*
PlotBKColor="#ffffff" *Color*
Title="" *String*
ShowLegend="1" *Bool*
OutsideMargin="3.%" *PercentOrPixel*
TitleToPlotMargin="3.%" *PercentOrPixel*
LegendToPlotMargin="3.%" *PercentOrPixel*
Orientation="vert" *Enumeration: possible values are: vert, horz*
>

```
<TitleFont
   Color="#000000"  Color
   Name="Tahoma"  String
   Bold="1"  Bool
   Italic="0"  Bool
   Underline="0"  Bool
   MinFontHeight="10.pt"  FontSize (only pt values)
   Size="8.%"  FontSize />
<LegendFont
   Color="#000000"
   Name="Tahoma"
   Bold="0"
   Italic="0"
   Underline="0"
   MinFontHeight="10.pt"
   Size="3.5%" />
<AxisLabelFont
   Color="#000000"
   Name="Tahoma"
   Bold="1"
   Italic="0"
   Underline="0"
   MinFontHeight="10.pt"
   Size="5.%" />
```

</General>

<Line
ConnectionShapeSize="1.%" *PercentOrPixel*
DrawFilledConnectionShapes="1" *Bool*
DrawOutlineConnectionShapes="0" *Bool*
DrawSlashConnectionShapes="0" *Bool*
DrawBackslashConnectionShapes="0" *Bool*
/>

© 2016-2022 Altova GmbH

<Bar
ShowShadow="1" *Bool*
ShadowColor="#a0a0a0" *Color*
OutlineColor="#000000" *Color*
ShowOutline="1" *Bool*
/>

<Area
Transparency="0" *UINT (0-255) 255 is fully transparent, 0 is opaque*
OutlineColor="#000000" *Color*
ShowOutline="1" *Bool*
/>

<CandleStick
FillHighClose="0" *Bool. If 0, the body is left empty. If 1, FillColorHighClose is used for the candle body*
FillColorHighClose="#ffffff" *Color. For the candle body when close > open*
FillHighOpenWithSeriesColor="1" *Bool. If true, the series color is used to fill the candlebody when open > close*
FillColorHighOpen="#000000" *Color. For the candle body when open > close and* `FillHighOpenWithSeriesColor` *is false*
/>

<Colors *User-defined color scheme: By default this element is empty except for the style and has no Color attributes*
UseSubsequentColors ="1" *Boolean. If* `0`*, then color in overlay is used. If* `1`*, then subsequent colors from previous chart layer is used*
Style="User" *Possible values are: "Default", "Grayscale", "Colorful", "Pastel", "User"*
Colors="#52aca0" *Color: only added for user defined color set*
Colors1="#d3c15d" *Color: only added for user defined color set*
Colors2="#8971d8" *Color: only added for user defined color set*
...
ColorsN="" *Up to ten colors are allowed in a set: from Colors to Colors9*
</Colors>

<Pie
ShowLabels="1" *Bool*
OutlineColor="#404040" *Color*
ShowOutline="1" *Bool*
StartAngle="0." *Double*
Clockwise="1" *Bool*
Draw2dHighlights="1" *Bool*
Transparency="0" *Int (0 to 255: 0 is opaque, 255 is fully transparent)*
DropShadowColor="#c0c0c0" *Color*
DropShadowSize="5.%" *PercentOrPixel*
PieHeight="10.%" *PercentOrPixel. Pixel values might be different in the result because of 3d tilting*
Tilt="40.0" *Double (10 to 90: The 3d tilt in degrees of a 3d pie)*
ShowDropShadow="1" *Bool*
ChartToLabelMargin="10.%" *PercentOrPixel*
AddValueToLabel="0" *Bool*
AddPercentToLabel="0" *Bool*
AddPercentToLabels_DecimalDigits="0" *UINT (0 – 2)*

```
        >
        <LabelFont
               Color="#000000"
               Name="Arial"
               Bold="0"
               Italic="0"
               Underline="0"
               MinFontHeight="10.pt"
               Size="4.%"    />
</Pie>

<XY>
        <XAxis  Axis
               AutoRange="1"  Bool
               AutoRangeIncludesZero="1"  Bool
               RangeFrom="0."  Double: manual range
               RangeTill="1."  Double : manual range
               LabelToAxisMargin="3.%"  PercentOrPixel
               AxisLabel=""  String
               AxisColor="#000000"  Color
               AxisGridColor="#e6e6e6"  Color
               ShowGrid="1"  Bool
               UseAutoTick="1"  Bool
               ManualTickInterval="1."  Double
               AxisToChartMargin="0.px"  PercentOrPixel
               TickSize="3.px"  PercentOrPixel
               ShowTicks="1"  Bool
               ShowValues="1"  Bool
               AxisPosition="LeftOrBottom"  Enums: "LeftOrBottom", "RightOrTop", "AtValue"
               AxisPositionAtValue = "0"  Double
               >
               <ValueFont
                      Color="#000000"
                      Name="Tahoma"
                      Bold="0"
                      Italic="0"
                      Underline="0"
                      MinFontHeight="10.pt"
                      Size="3.%"    />
        </XAxis>
        <YAxis  Axis (same as for XAxis)
               AutoRange="1"
               AutoRangeIncludesZero="1"
               RangeFrom="0."
               RangeTill="1."
               LabelToAxisMargin="3.%"
               AxisLabel=""
               AxisColor="#000000"
               AxisGridColor="#e6e6e6"
               ShowGrid="1"
               UseAutoTick="1"
               ManualTickInterval="1."
               AxisToChartMargin="0.px"
               TickSize="3.px"
               ShowTicks="1"  Bool
```


© 2016-2022 Altova GmbH

```
                ShowValues="1"  Bool
                AxisPosition="LeftOrBottom"  Enums: "LeftOrBottom", "RightOrTop", "AtValue"
                AxisPositionAtValue = "0"  Double
                >
                <ValueFont
                        Color="#000000"
                        Name="Tahoma"
                        Bold="0"
                        Italic="0"
                        Underline="0"
                        MinFontHeight="10.pt"
                        Size="3.%"/>
        </YAxis>
</XY>

<XY3d
        AxisAutoSize="1"  Bool: If false, XSize and YSize define the aspect ration of x and y axis.
If true, aspect ratio is equal to chart window
        XSize="100.%"  PercentOrPixel. Pixel values might be different in the result because of 3d
tilting and zooming to fit chart
        YSize="100.%"  PercentOrPixel. Pixel values might be different in the result because of 3d
tilting and zooming to fit chart
        SeriesMargin="30.%"  PercentOrPixel. Pixel values might be different in the result because
of 3d tilting and zooming to fit chart
        Tilt="20."  Double. -90 to +90 degrees
        Rot="20."  Double. -359 to +359 degrees
        FoV="50.">  Double. Field of view: 1-120 degree
        >
        <ZAxis
                AutoRange="1"
                AutoRangeIncludesZero="1"
                RangeFrom="0."
                RangeTill="1."
                LabelToAxisMargin="3.%"
                AxisLabel=""
                AxisColor="#000000"
                AxisGridColor="#e6e6e6"
                ShowGrid="1"
                UseAutoTick="1"
                ManualTickInterval="1."
                AxisToChartMargin="0.px"
                TickSize="3.px" >
                <ValueFont
                        Color="#000000"
                        Name="Tahoma"
                        Bold="0"
                        Italic="0"
                        Underline="0"
                        MinFontHeight="10.pt"
                        Size="3.%"/>
        </ZAxis>
</XY3d>

<Gauge
        MinVal="0."  Double
```

```
            MaxVal="100."   Double
            MinAngle="225"   UINT: -359-359
            SweepAngle="270"   UINT: 1-359
            BorderToTick="1.%"   PercentOrPixel
            MajorTickWidth="3.px"   PercentOrPixel
            MajorTickLength="4.%"   PercentOrPixel
            MinorTickWidth="1.px"   PercentOrPixel
            MinorTickLength="3.%"   PercentOrPixel
            BorderColor="#a0a0a0"   Color
            FillColor="#303535"   Color
            MajorTickColor="#a0c0b0"   Color
            MinorTickColor="#a0c0b0"   Color
            BorderWidth="2.%"   PercentOrPixel
            NeedleBaseWidth="1.5%"   PercentOrPixel
            NeedleBaseRadius="5.%"   PercentOrPixel
            NeedleColor="#f00000"   Color
            NeedleBaseColor="#141414"   Color
            TickToTickValueMargin="5.%"   PercentOrPixel
            MajorTickStep="10."   Double
            MinorTickStep="5."   Double
            RoundGaugeBorderToColorRange="0.%"   PercentOrPixel
            RoundGaugeColorRangeWidth ="6.%"   PercentOrPixel
            BarGaugeRadius="5.%"   PercentOrPixel
            BarGaugeMaxHeight="20.%"   PercentOrPixel
            RoundGaugeNeedleLength="45.%"   PercentOrPixel
            BarGaugeNeedleLength="3.%"   PercentOrPixel
            >
            <TicksFont
                  Color="#a0c0b0"
                  Name="Tahoma"
                  Bold="0"
                  Italic="0"
                  Underline="0"
                  MinFontHeight="10.pt"
                  Size="4.%"
            />
            <ColorRanges>   User-defined color ranges. By default empty with no child element entries
                  <Entry
                        From="50. "   Double
                        FillWithColor="1"   Bool
                        Color="#00ff00"   Color
                  />
                  <Entry
                        From="50.0"
                        FillWithColor="1"
                        Color="#ff0000"
                  />
                  ...
            </ColorRanges>
      </Gauge>
</chart-config>
```


© 2016-2022 Altova GmbH

30.2.1.10.2 Exemple : fonctions de graphique

Le document XSLT présenté ci-dessous à titre d'exemple montre comment les fonctions d'extension Altova pour les graphiques peuvent être utilisées. Ci-dessous, vous trouverez un document XML et une capture d'écran de l'image de sortie générée lorsque le document XML est traité avec le document XSLT en utilisant XSLT 2.0 ou 3.0 Engine.

Note: Les fonctions de graphique sont uniquement prises en charge dans les **Éditions Enterprise et Server** des produits Altova.

Note: Pour plus d'informations concernant la création des tables de données de graphique, voir la documentation des produits Altova XMLSpy et StyleVision.

Document XSLT

Ce document XSLT (*liste ci-dessous*) utilise fonctions d'extension de graphique Altova pour générer un camembert. Elles peuvent être utilisées pour traiter le document XML recensé ci-dessous.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:altovaext="http://www.altova.com/xslt-extensions"
   exclude-result-prefixes="#all">
   <xsl:output version="4.0" method="html" indent="yes" encoding="UTF-8"/>
   <xsl:template match="/">
      <html>
         <head>
            <title>
               <xsl:text>HTML Page with Embedded Chart</xsl:text>
            </title>
         </head>
         <body>
            <xsl:for-each select="/Data/Region[1]">
               <xsl:variable name="extChartConfig" as="item()*">
                  <xsl:variable name="ext-chart-settings" as="item()*">
                     <chart-config>
                        <General
                           SettingsVersion="1"
                           ChartKind="Pie3d"
                           BKColor="#ffffff"
                           ShowBorder="1"
                           PlotBorderColor="#000000"
                           PlotBKColor="#ffffff"
                           Title="{@id}"
                           ShowLegend="1"
                           OutsideMargin="3.2%"
                           TitleToPlotMargin="3.%"
                           LegendToPlotMargin="6.%"
                           >
```

```
                                <TitleFont
                                    Color="#023d7d"
                                    Name="Tahoma"
                                    Bold="1"
                                    Italic="0"
                                    Underline="0"
                                    MinFontHeight="10.pt"
                                    Size="8.%" />
                                </General>
                            </chart-config>
                        </xsl:variable>
                        <xsl:sequence select="altovaext:create-chart-config-from-xml( $ext-
chart-settings )"/>
                    </xsl:variable>
                    <xsl:variable name="chartDataSeries" as="item()*">
                        <xsl:variable name="chartDataRows" as="item()*">
                            <xsl:for-each select="(Year)">
                                <xsl:sequence select="altovaext:create-chart-data-row( (@id),
( .) )"/>
                            </xsl:for-each>
                        </xsl:variable>
                        <xsl:variable name="chartDataSeriesNames" as="xs:string*"
select=" ( (&amp;quot;Series 1&amp;quot;), &amp;apos;&amp;apos; )[1]"/>
                        <xsl:sequence
                            select="altovaext:create-chart-data-series-from-
rows( $chartDataSeriesNames, $chartDataRows)"/>
                    </xsl:variable>
                    <xsl:variable name="ChartObj" select="altovaext:create-
chart( $extChartConfig, ( $chartDataSeries), false() )"/>
                    <xsl:variable name="sChartFileName" select="'mychart1.png'"/>
                    <img src="{$sChartFileName, altovaext:save-chart-image( $ChartObj,
$sChartFileName, 400, 400 ) }"/>
                </xsl:for-each>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

Document XML

Ce document XML peut être traité avec le document XSLT ci-dessus. Les données se trouvant dans le document XML sont utilisées pour générer le camembert affiché dans la capture d'écran ci-dessous.

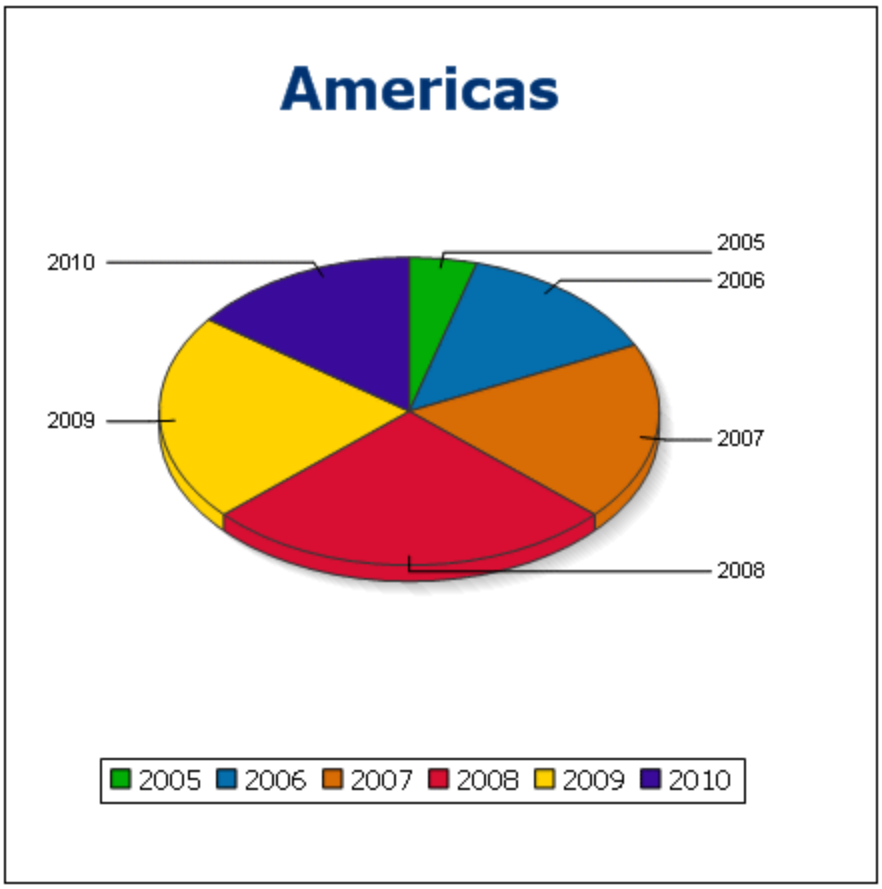
```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
    <ChartType>Pie Chart 2D</ChartType>
    <Region id="Americas">
        <Year id="2005">30000</Year>
        <Year id="2006">90000</Year>
        <Year id="2007">120000</Year>
        <Year id="2008">180000</Year>
```


© 2016-2022 Altova GmbH

```
        <Year id="2009">140000</Year>
        <Year id="2010">100000</Year>
    </Region>
    <Region id="Europe">
        <Year id="2005">50000</Year>
        <Year id="2006">60000</Year>
        <Year id="2007">80000</Year>
        <Year id="2008">100000</Year>
        <Year id="2009">95000</Year>
        <Year id="2010">80000</Year>
    </Region>
    <Region id="Asia">
        <Year id="2005">10000</Year>
        <Year id="2006">25000</Year>
        <Year id="2007">70000</Year>
        <Year id="2008">110000</Year>
        <Year id="2009">125000</Year>
        <Year id="2010">150000</Year>
    </Region>
</Data>
```

Image de sortie

Le camembert affiché ci-dessous est généré lorsque le document XML recensé ci-dessus est traité avec le document XSLT.



30.2.2 Fonctions d'extension diverses

Il existe plusieurs types de fonctions prêtes à l'utilisation dans les langages de programmation comme Java et C# qui ne sont pas disponibles en tant que fonctions XQuery/XPath ou en tant que fonctions XSLT. Un bon exemple sont les fonctions mathématiques disponibles en Java, comme `sin()` et `cos()`. Si ces fonctions étaient disponibles aux designers des feuilles de style XSLT et des requêtes XQuery, cela augmenterait le champ d'application des feuilles de style et des requêtes et faciliterait considérablement les tâches des créateurs de feuilles de style. Les moteurs XSLT et XQuery utilisés dans un grand nombre de produits Altova prennent en charge l'utilisation des fonctions d'extension dans Java et .NET, et pour les scripts MSXSL pour XSLT. Cette section décrit comment utiliser les fonctions d'extension et les scripts MSXSL dans vos feuilles de scripts XSLT et les documents XQuery. Les fonctions d'extension disponibles sont organisées dans les sections suivantes :

- Fonctions d'extension Java
- Fonctions d'extension .NET
- Scripts MSXSL pour XSLT

Les deux problèmes principaux considérés pour les descriptions sont : (i) comment sont appelées les fonctions dans les librairies respectives ; et (ii) quelles sont les règles à suivre pour la conversion d'arguments

© 2016-2022 Altova GmbH

dans un appel de fonction pour obtenir le format d'entrée requis de la fonction, et quelles sont les règles à suivre pour la conversion de retour (résultat de la fonction à l'objet de données XSLT/XQuery).

Exigences

Pour une prise en charge des fonctions d'extension, un Java Runtime Environment (pour l'accès aux fonctions Java) et le cadre de travail .NET Framework 2.0 (minimum, pour l'accès aux fonctions .NET) doit être installé sur la machine qui effectue la transformation XSLT ou l'exécution XQuery, ou doit être accessible pour les transformations.

30.2.2.1 Scripts MSXSL pour XSLT

L'élément `<msxsl:script>` contient des fonctions définies par l'utilisateur et des variables qui peuvent être appelées depuis des expressions XPath dans la feuille de style XSLT. Le `<msxsl:script>` et un élément de niveau supérieur, c'est à dire, qu'il doit être un élément enfant de `<xsl:stylesheet>` ou `<xsl:transform>`.

L'élément `<msxsl:script>` doit être dans l'espace de nom `urn:schemas-microsoft-com:xslt` (*voir exemple ci-dessus*).

Langage de script et espace de nom

Le langage de script utilisé dans le bloc est spécifié dans l'attribut `language` de l'élément `<msxsl:script>` et l'espace de nom à utiliser pour les appels de fonction depuis les expressions XPath est identifié avec l'attribut `implements-prefix` (*voir ci-dessous*).

```
<msxsl:script language="scripting-language" implements-prefix="user-namespace-prefix">

   function-1 or variable-1
   ...
   function-n or variable-n

</msxsl:script>
```

L'élément `<msxsl:script>` interagit avec le Windows Scripting Runtime, donc seuls des langages installés sur votre machine peuvent être utilisés dans l'élément `<msxsl:script>`. **La plate-forme .NET Framework 2.0 ou plus récente doit être installée pour pouvoir utiliser les scripts MSXSL.** Par conséquent, les langages de script .NET peuvent être utilisés dans l'élément `<msxsl:script>`.

L'attribut de langage doit accepter les mêmes valeurs que l'attribut `language` dans l'élément HTML `<script>`. Si l'attribut de langage n'est pas spécifié, alors Microsoft JScript est assumé par défaut.

L'attribut `implements-prefix` prend une valeur qui est un préfixe d'un nom d'espace in-scope déclaré. Cet espace de nom sera généralement un espace de nom d'utilisateur qui a été réservé pour une librairie de fonction. Toutes les fonctions et les variables définies dans l'élément `<msxsl:script>` se trouveront dans l'espace de nom identifié par le préfixe spécifié dans l'attribut `implements-prefix`. Lorsqu'une fonction est appelée depuis une expression XPath, le nom de la fonction entièrement qualifié doit se trouver dans le même espace de nom que la définition de la fonction.

Exemple

Voici un exemple d'une feuille de stype XSLT complète qui utilise une fonction définie dans un élément `<msxsl:script>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions"
   xmlns:msxsl="urn:schemas-microsoft-com:xslt"
   xmlns:user="http://mycompany.com/mynamespace">

  <msxsl:script language="VBScript" implements-prefix="user">
    <![CDATA[
    ' Input: A currency value: the wholesale price
    ' Returns: The retail price: the input value plus 20% margin,
    ' rounded to the nearest cent
    dim a as integer  = 13
    Function AddMargin(WholesalePrice) as integer
      AddMargin = WholesalePrice * 1.2 + a
    End Function
  ]]>
  </msxsl:script>

  <xsl:template match="/">
    <html>
      <body>
        <p>
          <b>Total Retail Price =
            $<xsl:value-of select="user:AddMargin(50)"/>
          </b>
          <br/>
          <b>Total Wholesale Price =
            $<xsl:value-of select="50"/>
          </b>
        </p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Types de données

Les valeurs des paramètres passés dans et hors du bloc de script sont limitées aux types de données XPath. Cette restriction ne s'applique pas aux données passées parmi les fonctions et les variables dans le bloc du script.

Assemblages

Un assemblage peut être importé dans le script en utilisant l'élément **`msxsl:assembly`**. L'assemblage est identifié par le biais d'un nom ou d'un URI. L'assemblage est importé lors de la compilation de la feuille de style. Voici une simple représentation de la manière d'utiliser l'élément `msxsl:assembly`.

© 2016-2022 Altova GmbH

```
<msxsl:script>
       <msxsl:assembly name="myAssembly.assemblyName" />
       <msxsl:assembly href="pathToAssembly" />

       ...

</msxsl:script>
```

Le nom d'assemblage peut être un nom complet comme :

```
"system.Math, Version=3.1.4500.1 Culture=neutral PublicKeyToken=a46b3f648229c514"
```

ou un nom court comme `"myAssembly.Draw"`.

Espaces de nom

Les espaces de nom peuvent être déclarés avec l'élément **`msxsl:using`**. Cela permet aux classes d'assemblage d'être écrits dans le script sans leurs espaces de nom, ce qui vous épargne un gros travail de saisie. Voici comment l'élément `msxsl:using` est utilisé de manière à déclarer des espaces de nom.

```
<msxsl:script>
       <msxsl:using namespace="myAssemblyNS.NamespaceName" />

       ...

</msxsl:script>
```

La valeur de l'attribut `namespace` est le nom de l'espace de nom.

30.3 Types de données dans les schémas XML générés par BD

Lorsqu'un Schéma XML est généré depuis une base de données (BD), les types de données spécifiques à cette BD sont convertis en des types de données de Schéma XML. Les mappages des types de données de BD vers des types de données de Schéma XML pour les BD les plus communément utilisées sont indiqués dans cet Appendice. Sélectionner depuis la liste ci-dessous.

- ADO
- MS Access
- Serveur MS SQL
- MySQL
- ODBC
- Oracle
- Sybase

30.3.1 ADO

Lorsqu'un schéma XML est généré depuis une base de données ADO (BD), les types de données ADO BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données ADO	Type de données Schéma XML
adGUID	xs:ID
adChar	xs:string
adWChar	xs:string
adVarChar	xs:string
adWVarChar	xs:string
adLongVarChar	xs:string
adWLongVarChar	xs:string
adVarWChar	xs:string
adBoolean	xs:boolean
adSingle	xs:float
adDouble	xs:double
adNumeric	xs:decimal
adCurrency	xs:decimal
adDBTimeStamp	xs:dateTime
adDate	xs:date
adBinary	xs:base64Binary

© 2016-2022 Altova GmbH

adVarBinary	xs:base64Binary
adLongVarBinary	xs:base64Binary
adInteger	xs:Integer
adUnsignedInt	xs:unsignedInt
adSmallInt	xs:short
adUnsignedSmallInt	xs:unsignedShort
adBigInt	xs:long
adUnsignedBigInt	xs:unsignedLong
adTinyInt	xs:byte
adUnsignedTinyInt	xs:unsignedByte

30.3.2 MS Access

Lorsqu'un schéma XML est généré depuis une base de données MS Access (BD), les types de données MS Access BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données MS Access	Type de données Schéma XML
GUID	xs:ID
char	xs:string
varchar	xs:string
memo	xs:string
bit	xs:boolean
Number(single)	xs:float
Number(double)	xs:double
Decimal	xs:decimal
Currency	xs:decimal
Date/Time	xs:dateTime
Number(Long Integer)	xs:integer
Number(Integer)	xs:short
Number(Byte)	xs:byte
OLE Object	xs:base64Binary

30.3.3 MS SQL Server

Lorsqu'un schéma XML est généré depuis une base de données Serveur MS SQL (BD), les types de données Serveur MS SQL BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données MS SQL Server	Type de données Schéma XML
uniqueidentifier	xs:ID
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
ntext	xs:string
sysname	xs:string
bit	xs:boolean
real	xs:float
float	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
binary	xs:base64Binary
varbinary	xs:base64Binary
image	xs:base64Binary
integer	xs:integer
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

© 2016-2022 Altova GmbH

30.3.4 MySQL

Lorsqu'un schéma XML est généré depuis une base de données MySQL (BD), les types de données MySQL BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données MySQL	Type de données Schéma XML
char	xs:string
varchar	xs:string
text	xs:string
tinytext	xs:string
mediumtext	xs:string
longtext	xs:string
tinyint(1)	xs:boolean
float	xs:float
double	xs:double
decimal	xs:decimal
datetime	xs:dateTime
blob	xs:base64Binary
tinyblob	xs:base64Binary
mediumblob	xs:base64Binary
longblob	xs:base64Binary
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

30.3.5 ODBC

Lorsqu'un schéma XML est généré depuis une base de données ODBC (BD), les types de données ODBC BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données ODBC	Type de données Schéma XML
SQL_GUID	xs:ID
SQL_CHAR	xs:string
SQL_VARCHAR	xs:string

SQL_LONGVARCHAR	xs:string
SQL_BIT	xs:boolean
SQL_REAL	xs:float
SQL_DOUBLE	xs:double
SQL_DECIMAL	xs:decimal
SQL_TIMESTAMP	xs:dateTime
SQL_DATE	xs:date
SQL_BINARY	xs:base64Binary
SQL_VARBINARY	xs:base64Binary
SQL_LONGVARBINARY	xs:base64Binary
SQL_INTEGER	xs:integer
SQL_SMALLINT	xs:short
SQL_BIGINT	xs:long
SQL_TINYINT	xs:byte

30.3.6 Oracle

Lorsqu'un schéma XML est généré depuis une base de données Oracle (BD), les types de données Oracle BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données Oracle	Type de données Schéma XML
ROWID	xs:ID
CHAR	xs:string
NCHAR	xs:string
VARCHAR2	xs:string
NVARCHAR2	xs:string
CLOB	xs:string
NCLOB	xs:string
NUMBER (with check constraint applied)*	xs:boolean
NUMBER	xs:decimal
FLOAT	xs:double
DATE	xs:dateTime

© 2016-2022 Altova GmbH

INTERVAL YEAR TO MONTH	xs:gYearMonth
BLOB	xs:base64Binary

* Si une contrainte de contrôle est appliquée à une colonne de type de données NUMBER, et que la contrainte de contrôle vérifie les valeurs 0 ou 1, alors le type de données NUMBER pour cette colonne sera convertie en un type de données de Schéma XML de xs:boolean. Ce mécanisme est utile pour générer un type de données xs:boolean datatype dans le Schéma XML généré.

30.3.7 Sybase

Lorsqu'un schéma XML est généré depuis une base de données Sybase (BD), les types de données Sybase BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données Sybase	Type de données Schéma XML
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
sysname-varchar(30)	xs:string
bit	xs:boolean
real	xs:float
float	xs:float
double	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
timestamp	xs:dateTime
binary<=255	xs:base64Binary
varbinary<=255	xs:base64Binary
image	xs:base64Binary
integer	xs:integer

smallint	xs:short
tinyint	xs:byte

© 2016-2022 Altova GmbH

30.4 Types de données dans les BD générés depuis des schémas XML

Lorsqu'une structure de BD est créée depuis un Schéma XML, les types de données spécifiques à cette BD sont générés depuis les types de données de Schéma XML. Les mappages des types de données de Schéma XML vers des types de données de BD pour les BD les plus communément utilisées sont indiqués dans cet Appendice. Sélectionner depuis la liste ci-dessous.

- MS Access
- Serveur MS SQL
- MySQL
- Oracle

30.4.1 MS Access

Lorsqu'une base de données MS Access (BD) est créée depuis un Schéma XML, les types de données de Schéma XML sont convertis en des types de données de MS Access tels que recensés dans la table ci-dessous.

Type de données Schéma XML	Type de données MS Access
xs:ID	GUID
xs:string	If no facets varchar (255)
	Size = either length or maxLength
	If Size <= 255 varchar (n)
	else memo
xs:normalizedString	Same as xs:string
xs:token	Same as xs:string
xs:Name	Same as xs:string
xs:NCName	Same as xs:string
xs:anyURI	Same as xs:string
xs:QName	Same as xs:string
xs:NOTATION	Same as xs:string
xs:boolean	bit
xs:float	Number (single)
xs:double	Number (double)
xs:decimal	Decimal

xs:duration	Date/Time
xs:dateTime	Date/Time
xs:time	Date/Time
xs:date	Date/Time
xs:gYearMonth	Date/Time
xs:gYear	Date/Time
xs:gMonthDay	Date/Time
xs:gDay	Date/Time
xs:gMonth	Date/Time
xs:hexBinary	If no facets varbinary (255)
	Size = either length or maxLength
	If Size <= 8000 varbinary
	else image (OLE Object)
xs:base64Binary	Same as xs:hexBinary
xs:integer	Number (Long Integer)
xs:int	Number (Long Integer)
xs:negativeInteger	Number (Long Integer); value constraint
xs:positiveInteger	Number (Long Integer); value constraint
xs:nonNegativeInteger	Number (Long Integer); value constraint
xs:nonPositiveInteger	Number (Long Integer); value constraint
xs:unsignedInt	Number (Long Integer)
xs:short	-- no equivalent --
xs:unsignedShort	-- no equivalent --
xs:long	-- no equivalent --
xs:unsignedLong	-- no equivalent --
xs:byte	Number (Byte)
xs:unsignedByte	Number (Byte)

30.4.2 MS SQL Server

Lorsqu'un schéma XML est généré depuis une base de données Serveur MS SQL (BD), les types de données Serveur MS SQL BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

© 2016-2022 Altova GmbH

Type de données Schéma XML	Type de données Serveur MS SQL
ID	uniqueidentifier
xs:string	If no facets
	{ if UNICODE nvarchar (255)
	else varchar (255) }
	else
	{ if UNICODE
	(Size = either length or maxLength)
	If Size <= 4000
	if FacetLengthIsSet then nChar
	else nVarChar
	if Size <= 1073741823 then nText }
	else
	{ if NON-UNICODE
	(Size = either length or maxLength)
	If Size <= 8000
	if FacetLengthIsSet then char
	else varchar
	if Size <= 2147483647 then text }
xs:normalizedString	Comme xs:string
xs:token	Comme xs:string
xs:Name	Comme xs:string
xs:NCName	Comme xs:string
xs:anyURi	Comme xs:string
xs:QName	Comme xs:string
xs:NOTATION	Comme xs:string
xs:boolean	bit
xs:float	real
xs:double	float
xs:decimal	decimal
xs:duration	datetime

xs:dateTime	datetime
xs:time	datetime
xs:date	datetime
xs:gYearMonth	datetime
xs:gYear	datetime
xs:gMonthDay	datetime
xs:gDay	datetime
xs:gMonth	datetime
xs:hexBinary	If no facets varbinary (255)
	(Size = either length or maxLength
	If Size <= 8000
	if FacetLengthIsSet then binary
	else varbinary
	if Size <= 2147483647 then image
xs:base64Binary	Same as xs:hexBinary
xs:integer	int
xs:int	int
xs:negativeInteger	Int (limité à {...,-2,-1})
xs:positiveInteger	Int (limité à {1,2,...})
xs:nonNegativeInteger	int (limité à {0,1,2,...})
xs:nonPositiveInteger	int (limité à {...,-2,-1,0})
xs:unsignedInt	int (contraintes supplémentaires)
xs:short	smallint
xs:unsignedShort	smallint (contraintes supplémentaires)
xs:long	bigint
xs:unsignedLong	bigint (contraintes supplémentaires)
xs:byte	tinyint
xs:unsignedByte	tinyint (contraintes supplémentaires)

30.4.3 MySQL

Lorsqu'un schéma XML est généré depuis une base de données MySQL (BD), les types de données MySQL BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

© 2016-2022 Altova GmbH

Type de données Schéma XML	Type de données MySQL
xs:ID	varchar(255)
xs:string	If no facets then varchar (255)
	else if facet length is set and <= 255 then char
	else if facet maxLength set and <= 255 then varchar
	else if maxLength is set and <= 65545 then text
	else if maxlength is set and <= 16777215 then mediumtext
	else if maxlength is set and <= 429496295 then longtext
xs:normalizedString	*Comme* xs:string
xs:token	*Comme* xs:string
xs:Name	*Comme* xs:string
xs:NCName	*Comme* xs:string
xs:anyURI	*Comme* xs:string
xs:QName	*Comme* xs:string
xs:NOTATION	*Comme* xs:string
xs:boolean	tinyint(1)
xs:float	float
xs:double	double
xs:decimal	decimal
xs:duration	timestamp
xs:dateTime	datetime
xs:time	time
xs:date	date
xs:gYearMonth	timestamp(4)
xs:gYear	year(4)
xs:gMonthDay	timestamp(8); constraints to check month, day
xs:gDay	timestamp(8); constraints to check day
xs:gMonth	timestamp(8); constraints to check month

xs:hexBinary	If no facets then blob (255)
	else if facet length is set and <= 255 then blob
	else if facet maxLength is set and <= 255 then tinyblob
	else if maxlength is set and <= 65545 then blob
	else if maxlength is set and <= 16777215 then mediumblob
	else if maxlength is set and <= 429496295 then longblob
xs:base64Binary	*Same as* xs:hexBinary
xs:integer	Integer
xs:int	int
xs:negativeInteger	Integer (limité à {...,-2,-1})
xs:positiveInteger	Integer (limité à {1,2,...})
xs:nonNegativeInteger	Integer (limité à {0,1,2,...})
xs:nonPositiveInteger	Integer (limité à {...,-2,-1,0})
xs:unsignedInt	Int (contraintes supplémentaires)
xs:short	Smallint
xs:unsignedShort	Smallint (contraintes supplémentaires)
xs:long	Bigint
xs:unsignedLong	Bigint (contraintes supplémentaires)
xs:byte	Tinyint
xs:unsignedByte	Tinyint (additional constraints)

30.4.4 Oracle

Lorsqu'un schéma XML est généré depuis une base de données Oracle (BD), les types de données Oracle BD sont convertis en des types de données de Schéma XML tels que recensés dans la table ci-dessous.

Type de données Schéma XML	Type de données Oracle
xs:ID	ROWID
xs:string	If no facets

© 2016-2022 Altova GmbH

	if UNICODE then NVARCHAR2 (255)
	else VARCHAR2 (255)
	else if UNICODE
	(Size = either length or maxLength)
	If Size <= 2000 then NCHAR
	if Size <= 4000 then NVARHCAR2
	if Size <= 4 Gigabytes then NCLOB
	else if NON-UNICODE
	(Size = either length or maxLength)
	If Size <= 2000 then CHAR
	if Size <= 4000 then VARCHAR2
	if Size <= 4 Gigabytes then CLOB
xs:normalizedString	*Comme* xs:string
xs:token	*Comme* xs:string
xs:Name	*Comme* xs:string
xs:NCName	*Comme* xs:string
xs:anyURI	*Comme* xs:string
xs:QName	*Comme* xs:string
xs:NOTATION	*Comme* xs:string
xs:boolean	NUMBER with constraint Boolean
xs:float	FLOAT
xs:double	FLOAT
xs:decimal	NUMBER
xs:duration	TIMESTAMP
xs:dateTime	TIMESTAMP
xs:time	DATE
xs:date	DATE
xs:gYearMonth	INTERVAL YEAR TO MONTH
xs:gYear	DATE
xs:gMonthDay	DATE
xs:gDay	DATE
xs:gMonth	DATE

xs:hexBinary	if no facets then RAW (255)
	(Size = either length or maxLength)
	If Size <= 2000 then RAW (X)
	else Size <= 2 Gigabytes then LONG RAW (X)
	if Size <= 4 Gigabytes then BLOB (X)
xs:base64Binary	BLOB
xs:integer	NUMBER
xs:int	NUMBER
xs:negativeInteger	NUMBER (limité à {...,-2,-1})
xs:positiveInteger	NUMBER (limité à {1,2,...})
xs:nonNegativeInteger	NUMBER (limité à {0,1,2,...})
xs:nonPositiveInteger	NUMBER (limité à {...,-2,-1,0})
xs:unsignedInt	NUMBER (contraintes supplémentaires)
xs:short	NUMBER
xs:unsignedShort	NUMBER (contraintes supplémentaires)
xs:long	NUMBER
xs:unsignedLong	NUMBER (contraintes supplémentaires)
xs:byte	BLOB
xs:unsignedByte	BLOB (contraintes supplémentaires)

© 2016-2022 Altova GmbH

30.5 Données techniques

Cette section contient des informations utiles concernant certains aspects techniques de votre logiciel. Cette information est organisée dans les sections suivantes :

- SE et exigences de mémoire
- Moteurs Altova
- Prise en charge Unicode
- Utilisation Internet

30.5.1 SE et exigences de mémoire

Système d'exploitation

Les applications logicielles d'Altova sont disponibles pour les plateformes suivantes :

- Windows 7 SP1 avec mise à jour de la plateforme, Windows 8, Windows 10, Windows 11
- Windows Server 2008 R2 SP1 avec mise à jour de la plateforme ou plus récent

Mémoire

Étant donné que le logiciel est rédigé en C++, il ne nécessite pas la performance d'un Java Runtime Environment et nécessite généralement moins de mémoire que les applications basées sur Java comparables. Néanmoins, chaque document est chargé entièrement dans la mémoire de manière à ce qu'il puisse être parsé complètement et pour améliorer la vitesse de la consultation et de l'édition. Les exigences de mémoire augmentent avec la taille du document.

Les exigences de mémoire sont aussi affectées par un historique de la fonction Annuler non limité. Si vous coupez/collez sans arrêt de larges sélections dans des documents volumineux, la mémoire disponible peut baisser rapidement.

30.5.2 Moteurs Altova

Validateur XML

Lors de l'ouverture d'un document XML, l'application utilise son validateur XML intégré pour vérifier la bonne formation, pour valider le document par rapport à un schéma (si spécifié), et de générer des arborescences et infosets. Le validateur XML est aussi utilisé pour fournir une édition intelligente pendant que vous éditez des documents et pour afficher dynamiquement toute erreur de validation qui peut se produire.

Le validateur XML intégré met en place les spécifications de la Final Recommendation of the W3C's XML Schema 1.0 et 1.1. Les nouveaux développements recommandés par le Groupe de travail de Schéma XML de W3C sont incorporés en continu dans le validateur XML, afin que les produits Altova vous donnent un environnement de développement de pointe.

Moteurs XSLT et XQuery

Les produits Altova utilisent les moteurs Altova XSLT 1.0, 2.0 et 3.0, et les moteurs Altova XQuery 1.0 et 3.1. Si un de ces moteurs est inclus dans le produit, la documentation concernant le comportement spécifique à la mise en place pour chaque moteur est indiquée dans les annexes de la documentation.

Note: Altova MapForce génère du code utilisant les moteurs XSLT 1.0, 2.0 et XQuery 1.0.

30.5.3 Prise en charge Unicode

Les produits XML d'Altova propose une prise en charge complète d'Unicode. Pour éditer un document XML, vous devrez utiliser une police d'écriture qui sera également prise en charge par ce document.

Veuillez noter que la plupart des polices ne contient qu'un sous-ensemble très spécifique de la plage Unicode et qu'elles sont donc généralement taillées à la mesure du système d'écriture correspondant. Si vous voyez apparaître un texte déformé, une des raisons peut être que la police que vous avez sélectionnée ne contient pas les symboles exigés. Il est donc utile de disposer d'une police qui couvre toute la plage Unicode, surtout lors de l'édition de documents XML dans des langues ou des systèmes d'écriture variés. Une police Unicode typique utilisée sur les PC Windows est Arial Unicode MS.

Dans le dossier `/Examples` de votre dossier d'application, vous trouverez un fichier XHTML appelé `UnicodeUTF-8.html` qui contient la phrase suivante dans plusieurs langues et systèmes d'écriture :

- *When the world wants to talk, it speaks Unicode*
- *Quand le monde veut communiquer, il parle en Unicode*
- *Wenn die Welt miteinander spricht, spricht sie Unicode*
- 世界的に話すなら、Unicode です。

Ouvrez ce fichier XHTML pour voir un aperçu des possibilités d'Unicode et pour indiquer que les systèmes d'écriture sont pris en charge par les polices disponibles sur votre PC.

30.5.4 Utilisation Internet

Les applications Altova initieront des connexions Internet pour vous dans les situations suivantes :

- Si vous cliquez sur "Demander code-clé d'évaluation" dans le dialogue d'Enregistrement (**Aide | Activation du logiciel**), les trois champs dans le dialogue d'enregistrement seront transférés vers notre serveur web au moyen d'une connexion http (port 80) normale et le code-clé d'évaluation sera renvoyé au client par le biais d'un e-mail SMTP normal.
- Dans certains produits Altova, vous pouvez ouvrir un fichier dans Internet (**Fichier | Ouvrir | Passer à URL**). Dans ce cas, le document est extrait à l'aide d'une des méthodes de protocole et connexions suivantes : HTTP (normalement port 80), FTP (normalement port 20/21), HTTPS (normalement port 443). Vous pouvez aussi exécuter un serveur HTTP sur le port 8080. (Dans le dialogue URL, spécifier le port après le nom de serveur et un double point.)
- Si vous ouvrez un document XML qui réfère à un Schéma XML ou à un DTD et que le document est spécifié par une URL, le document de schéma référencé est aussi extrait par le biais d'une connexion HTTP (port 80) ou un autre protocole spécifié dans l'URL (voir Point 2 ci-dessus). Un document de

© 2016-2022 Altova GmbH

schéma sera aussi extrait lorsqu'un fichier XML est validé. Veuillez noter que la validation peut avoir lieu automatiquement lors de l'ouverture d'un document si vous avez instruit l'application de procéder de cette manière (dans l'onglet Fichier du dialogue Options (**Outils | Options**)).

- Dans les applications Altova utilisant WSDL et SOAP, les connexions de service web sont définies par les documents WSDL.
- Si vous utilisez la commande **Envoyer par courrier électronique** (**Fichier | Envoyer par courrier électronique**) dans XMLSpy, la sélection ou le fichier actuel est envoyé avec tout programme d'e-mail conforme à MAPI et installé sur le PC de l'utilisateur.
- Fait partie de l'Activation du logiciel et du LiveUpdate tel que décrit ultérieurement dans l'Accord de licence de logiciel Altova.

30.6 Informations de licence

Cette section contient des informations concernant :

- la distribution de ce logiciel
- l'activation de logiciel et le license metering
- le contrat de licence régissant l'usage de ce logiciel

Veuillez lire ces informations attentivement. Elles ont force obligatoire puisque vous avez accepté ces termes lors de l'installation de ce logiciel.

Pour consulter les termes de toute licence Altova, rendez-vous sur la page des informations juridiques Altova sur le site web Altova.

30.6.1 Distribution électronique de logiciel

Ce produit est disponible par le biais de la distribution électronique de logiciel, une méthode de distribution qui fournit les avantages uniques suivants :

- Vous pouvez évaluer gratuitement le logiciel pendant 30 jours avant de vous décider à l'achat. (*Note : Altova MobileTogether Designer dispose d'une une licence gratuite.*)
- Une fois que vous avez décidé d'acheter le logiciel, vous pouvez passer vos commandes en ligne sur le site web Altova et vous obtiendrez en quelques minutes un produit bénéficiant d'une pleine licence.
- Lorsque vous passez une commande en ligne, vous disposerez toujours de la dernière version de nos logiciels.
- Le pack de produits comprend une aide sur écran qui peut être accédé depuis l'intérieur de l'interface de l'application. La dernière version du manuel d'utilisateur est disponible sous www.altova.com (i) sous format HTML pour une navigation en ligne, et (ii) sous format PDF pour le téléchargement (et pour imprimer si vous préférez avoir recours à une documentation en papier).

Période d'évaluation de 30 jours

Après avoir téléchargé le produit, vous pourrez évaluer celui-ci gratuitement pour une période de jusqu'à 30 jours. Au bout d'environ 20 jours de cette période d'évaluation, le logiciel commencera à vous rappeler qu'il n'est pas encore sous licence. Le message de rappel s'affichera une fois à chaque fois que vous démarrerez l'application. Si vous souhaitez continuer à utiliser le programme à l'issue de la période d'évaluation de 30 jours, vous devrez acheter une licence de produit, qui est fournie sous la forme d'un fichier de licence contenant un code-clé. Déverrouiller le produit en chargeant le fichier de licence dans le dialogue d'activation du logiciel de votre produit.

Vous pouvez acheter des licences de produit dans la boutique en ligne du https://shop.altova.com/.

Transmettre le logiciel à d'autres collaborateurs dans votre entreprise à des fins d'évaluation

Si vous souhaitez distribuer la version d'évaluation dans le cadre de votre réseau d'entreprise, ou si vous prévoyez de l'utiliser sur un PC qui n'est pas connecté à Internet, vous pourrez uniquement distribuer le fichier d'installation, à condition qu'il ne soit pas modifié de quelque manière que ce soit. Toute personne accédant au programme d'installation du logiciel que vous avez fourni doit demander son propre code-clé d'évaluation de 30

© 2016-2022 Altova GmbH

jours et devra aussi acheter une licence à l'issue de la période d'évaluation afin de pouvoir continuer à utiliser le produit.

30.6.2 Activation de logiciel et le license metering

En tant que partie intégrante de l'Activation du logiciel Altova, le logiciel peut utiliser votre réseau interne et votre connexion Internet à des fins de transmission des données relatives à la licence au moment de l'installation, de l'enregistrement, de l'utilisation ou de la mise à jour d'un serveur de licence utilisé par Altova et valider l'authenticité des données relatives à la licence pour protéger Altova contre une utilisation sans licence ou illégale du logiciel et pour améliorer le service clientèle. L'activation est basée sur l'échange des données relatives aux licences comme les systèmes d'exploitation, l'adresse IP, la date/heure, la version de logiciel et le nom de l'ordinateur, ainsi que d'autres informations échangées entre votre ordinateur et un serveur de licence Altova.

Votre produit Altova comporte un module intégré de contrôle des licences qui vous aide à éviter toute violation non-intentionnelle du contrat de licence de l'utilisateur final. Votre produit est licencé soit en tant qu'une installation utilisateur simple soit en tant qu'installation multi-utilisateur, et le module de contrôle des licences permet de vous assurer qu'aucune licence outre celles accordées pour le nombre d'utilisateurs sous licence n'utilise l'application simultanément.

Cette technologie de contrôle des licences utilise votre réseau local (LAN) pour communiquer entre les instances de l'application exécutée sur plusieurs ordinateurs.

Licence simple

Lorsque l'application est démarrée dans le cadre du processus de contrôle de la licence, le logiciel envoie un bref datagramme de diffusion pour trouver d'autres instances du produit exécuté sur d'autres ordinateurs dans le même segment de réseau. S'il n'obtient pas de réponses, il ouvrira un port pour écouter d'autres instances de l'application.

Licence utilisateurs multiples

Si plus d'une seule instance de l'application est utilisée dans le même LAN, ces instances communiqueront brièvement l'une avec l'autre lors du démarrage. Ces instances échangent des codes-clés afin de vous aider à mieux déterminer que le nombre de licences concurrentes achetées n'est pas violé accidentellement. Il s'agit de la même technologie de contrôle des licences généralement utilisée dans l'univers Unix et dans un certain nombre d'outils de développement de bases de données. Elle permet aux clients Altova d'acheter des licences multi-utilisateurs d'utilisation simultanée à des prix raisonnables.

Nous avons également conçu les applications de manière à ce qu'elles envoient des paquets de réseau peu importants et peu nombreux pour ne pas surcharger votre réseau. Les ports TCP/IP (2799) utilisés par votre produit Altova sont officiellement enregistrés auprès de l'IANA (*voir IANA Service Name Registry pour plus de détails*) et notre module le contrôle de licence est testé et éprouvé technologiquement.

Si vous utilisez un pare-feu, vous pourrez éventuellement apercevoir des communications sur le port 2799 entre les ordinateurs qui exécutent les produits Altova. Vous pouvez, bien évidemment, bloquer ce trafic entre les groupes différents dans votre entreprise, du moment que vous pouvez assurer par d'autres moyens que votre contrat de licence n'a pas été violé.

Vous noterez également que, si vous êtes en ligne, votre logiciel Altova contient de nombreuses fonctions utiles ; celles-ci ne concernent pas la technologie de contrôle des licences.

Note à propos des certificats

Votre application Altova contacte le serveur de mise sous licence Altova (`link.altova.com`) via HTTPS. Pour établir cette communication, Altova utilise un certificat SSL enregistré. Si ce certificat est remplacé (par exemple, par votre département IT ou une agence externe), votre application Altova vous avertira que la connexion n'est pas sûre. Vous pourriez utiliser le certificat de remplacement pour lancer votre application Altova, mais vous le ferez à vos propres risques et périls. Si vous voyez un message d'avertissement *Connexion non-sécurisée*, vérifiez l'origine du certificat et consultez votre équipe IT (qui sera en mesure de décider si l'interception et le remplacement du certificat Altova devrait continuer ou pas).

Si votre organisation nécessite d'utiliser son propre certificat (par exemple, pour surveiller la communication de et vers les machines client), nous vous recommandons d'installer le logiciel de gestion de licence gratuit d'Altova, Altova LicenseServer, dans votre réseau. Sous cette configuration, les appareils de client peuvent continuer d'utiliser les certificats de votre organisation, alors que l'Altova LicenseServer peut être autorisé à utiliser le certificat Altova pour une communication avec Altova.

30.6.3 Altova Contrat de licence de l'utilisateur final

- Le contrat Altova de licence de l'utilisateur final est disponible ici : https://www.altova.com/legal/eula
- La politique de confidentialité d'Altova est disponible ici : https://www.altova.com/privacy

© 2016-2022 Altova GmbH

Index

$

$ref (Schémas JSON), 725

.

.docx, 281, 950
.NET,
différences avec XMLSpy standalone, 1141
et débogueurs XMLSpy, 1144
intégration de XMLSpy avec, 1139
.pptx, 281, 950
.xslx, 281, 950

/

/XSLT,
et les transformations en lot, 494
processeur, 1631
transformations dans XMLSpy, 494

A

Activer le logiciel, 1651
Activer point d'arrêt- Débogueur XSLT, 1425
activer point de trace- Débogueur XSLT, 1425
ActiveX,
integration at application level, 2068
integration at document level, 2070
integration prerequisites, 2065
ActiveX controls,
adding to the Visual Studio Toolbox, 2066
support, 1687
ADO,
configurer une connexion, 972
en tant qu'interface de connexion de données, 965
ADO.NET,
configurer une connexion, 979
Affichage de la table du Mode Grid JSON,
éditer dans, 696
Importation/exportation de feuilles de calcul, 696
Affichage de la table du Mode Grid XML,
éditer dans, 309
Importation/exportation de feuilles de calcul, 309
Afficher des grandes balises, 620
Afficher des petites balises, 620
Afficher diagramme, 1398
Afficher nœuds d'exéc. actuels,
Débogueur XSLT, 1424
Afficher tous les globaux, 1398
Afficher un balisage mixte, 620
Agrandir,
complètement, 1504
Aide,
contenus, 1649
modèle de clavier, 1650
Ajouter,
ligne (dans Authentic View), 1437
Ajouter commande Enfant,
dans Mode Grille, 1348
Alias,
voir Ressources Globales, 1060
Aller à la ligne/au caract, 1505
Aller au fichier, 1506
Alternatives de type conditionnel, 203
Amélioration du Mode Grille,
voir Mode Grille, 88
Annotations dans l'aperçu de schéma, 181
Ant,
configurer les variables d'environnement pour,, 1255
Aperçu Attribut, 1616
Aperçu de la mise en page de table, 928
Aperçu de mise en page de table,
créer des formules depuis, 930
Aperçu de schéma, 49
éditer dans, 177
et affichage en ligne des commentaires globaux, 177
et commentaires globaux, 177
et Mode de Modèle de contenu, 176, 177
icônes dans, 177
trier les composants dans, 177
Aperçu du mécanisme du catalogue, 459
API,
documentation, 1702
JAVA, 2014

© 2016-2022 Altova GmbH

API,
JAVA Classpath, 2014
overview, 1703
Appeler des modèles nommés, 498
Application,
ActiveDocument, 1743
AddMacroMenuItem, 1744
AddXSLT_XQParameter, 1744
Application, 1744
ClearMacroMenu, 1744
CurrentProject, 1745
Dialogs, 1745
Documents, 1746
GetDatabaseImportElementList, 1746
GetDatabaseSettings, 1747
GetDatabaseTables, 1747
GetExportSettings, 1748
GetTextImportElementList, 1748
GetTextImportExportSettings, 1749
GetXSLT_XQParameterCount, 1749
GetXSLT_XQParameterName, 1750
GetXSLT_XQParameterXPath, 1750
ImportFromDatabase, 1750
ImportFromSchema, 1751
ImportFromText, 1752
ImportFromWord, 1753
NewProject, 1754
OnBeforeOpenDocument, 1741
OnBeforeOpenProject, 1742
OnDocumentOpened, 1742
OnProjectOpened, 1743
OpenProject, 1754
Parent, 1754
Quit, 1755
ReloadSettings, 1755
RemoveXSLT_XQParameter, 1755
RunMacro, 1756
ScriptingEnvironment, 1756
ShowApplication, 1756
ShowForm, 1757
URLDelete, 1758
URLMakeDirectory, 1758
WarningNumber, 1759
WarningText, 1759
Applications externes,
ouvrir les fichiers dans, 1587
Appliquer, 1601
Archive View, 281, 950
et fichiers EPUB, 958
et fichiers OOXML, 952
et fichiers ZIP, 956
Arrêter,
Débogueur XSLT, 1423
Arrière,
dans le Mode Schéma, 248
Assertions dans le Mode Schéma, 210, 215
Assertions de types simples, 232
Assignations de type conditionnel, 203
Assigner StyleVision Power Stylesheet au fichier XML, 1430
Assistance technique, 1654
Assistant à la saisie des Composants, 226
Assistant de saisie,
Détails, 52
Assistant de saisie Composants, 232
Assistant de saisie Détails, 230
Assistant pour les nouvelles taxonomie, 847
Assistants à la saisie,
pour XQuery, 511
Assistants à la saisie (Mode Texte, Mode Authentic), 334
Assistants à la saisie Attributs,
dans Authentic View, 623
Assistants à la saisie dans le Mode Schéma, 226
Assistants à la saisie dans Mode Texte, 164
Assistants à la saisie Éléments,
dans Authentic View, 623
Assistants à la saisie Entités,
dans Authentic View, 623
Assistants de saisie, 121
affichage de, 1647
dans Mode Grille, 98
mettre à jour, 1358
ATL,
plug-in sample files, 1690
Attribuer,
raccourci vers une commande, 1588
Attribuer des modèles de valeur,
Édition intelligente Xpath dans, 492
Attribut, 74
basculer dans le mode de Modèle de contenu, 74
dans les définitions de schéma, 74
AttributeFormDefault,
paramètres dans le Mode Conception de schéma, 1386
Attributs, 181
Attributs dans le Mode Schéma, 210, 212

© 2016-2022 Altova GmbH

Attributs Globaux, 181
Authentic View, 637
affichage de balises dans, 618
affichage des balises dans, 620
affichage du document, 620
afficher les balises de markup, 604
ajouter des nœuds, 607
aperçu de la GUI, 616
appareils de saisie des données dans, 610
appliquer des éléments, 607
assistants à la saisie, 604
assistants à la saisie dans, 623
caractères spéciaux dans, 610
coller en tant que XML/Texte, 627
éditer des données dans une BD XML, 1431
éditer les données de BD dans, 1429
entités dans, 610
fenêtre principale dans, 620
formater du texte dans, 618
Générer les documents de sortie du fichier PXF, 1438
icônes de barre d'outils, 618
Icônes de table XML, 641
imprimer un document XML depuis, 614
insérer des entités dans, 613
insérer des nœuds, 607
menu contextuel, 604
menus contextuels, 627
ouvrir un document XML dans, 603
ouvrir un nouveau fichier XML dans, 1429
passer au, 1503
retirer des éléments, 607
saisir des données dans, 610
saisir des valeurs d'attribut, 612
sections CDATA dans, 610
supprimer des nœuds, 607
tables (SPS et XML), 635
tables dans, 607
Tables SPS, 635
Tables XML, 637
utilisation de tables XML, 637
utilisation des fonctions importantes, 630
Authentic XML, 600
AuthenticDataTransfer,
dropEffect, 1761
getData, 1762
ownDrag, 1762
type, 1762
AuthenticRange,
AppendRow, 1767
Application, 1768
CanPerformAction, 1768
CanPerformActionWith, 1768
Close, 1769
CollapsToBegin, 1769
CollapsToEnd, 1769
Copy, 1770
Cut, 1770
Delete, 1770
DeleteRow, 1770
DuplicateRow, 1771
ExpandTo, 1772
FirstTextPosition, 1772
FirstXMLData, 1773
FirstXMLDataOffset, 1774
GetElementAttributeNames, 1775
GetElementAttributeValue, 1775
GetElementHierarchy, 1776
GetEntityNames, 1777
Goto, 1777
GotoNext, 1778
GotoNextCursorPosition, 1779
GotoPrevious, 1779
GotoPreviousCursorPosition, 1780
HasElementAttribute, 1780
InsertEntity, 1780
InsertRow, 1781
IsCopyEnabled, 1781
IsCutEnabled, 1782
IsDeleteEnabled, 1782
IsEmpty, 1782
IsEqual, 1782
IsFirstRow, 1783
IsInDynamicTable, 1783
IsLastRow, 1783
IsPasteEnabled, 1783
IsTextStateApplied, 1784
LastTextPosition, 1784
LastXMLData, 1785
LastXMLDataOffset, 1786
MoveBegin, 1787
MoveEnd, 1787
MoveRowDown, 1788
MoveRowUp, 1787
Parent, 1788
Paste, 1788
PerformAction, 1789

© 2016-2022 Altova GmbH

AuthenticRange,
Select, 1789
SelectNext, 1790
SelectPrevious, 1791
SetElementAttributeValue, 1791
SetFromRange, 1792
Text, 1793
AuthenticView, 1811
Application, 1804
AsXMLString, 1804
DocumentBegin, 1805
DocumentEnd, 1806
Event, 1807
Goto, 1808
IsRedoEnabled, 1808
IsUndoEnabled, 1809
MarkupVisibility, 1809
OnBeforeCopy, 1795
OnBeforeCut, 1795
OnBeforeDelete, 1796
OnBeforeDrop, 1796
OnBeforePaste, 1797
OnDragOver, 1798
OnKeyboardEvent, 1799
OnMouseEvent, 1800
OnSelectionChanged, 1801
Parent, 1809
Print, 1809
Redo, 1810
Selection, 1810
Undo, 1811
WholeDocument, 1812
XMLDataRoot, 1812
Avant,
dans le Mode Schéma, 248
Avro,
aperçu, 773
et RaptorXML, 773
structures de données en binaire, 780
structures des données dans JSON, 779
types de fichier, 773

B

Balisage (dans Authentic View),
afficher petit/grand/mixte, 1436
masquer, 1436
Balisage mixte (dans Authentic View), 1436
Balisage petit (dans Authentic View), 1436
Balises,
dans Authentic View, 620
in Authentic View, 618
Barre de menu, 132
Barre de statut, 132
Barre d'outils, 132, 1585
activer/désactiver, 1585
ajouter commande à, 1583
ajouter macro à, 1594
créer nouveau, 1585
réinitialiser barre d'outils & commandes de menu, 1585
Barre grise, 1504
Barre outils,
afficher grandes icônes, 1597
Base de données,
créer BD basée sur schéma, 1484
DB XML Oracle, 1465
éditer les enregistrements de, 1454
exporter des données XML vers, 1491
importer des données comme XML, 1473
structure importer en tant que Schéma XML, 1478
base de données Oracle,
se connecter par le biais d'ODBC, 1037
se connecter par le biais de JDBC, 1035
Base de données Progress OpenEdge,
se connecter par le biais d'ODBC, 1046
se connecter par le biais de JDBC, 1044
Bases de données,
éditer dans Authentic View, 1429
et ressources globales, 1076
prise en charge dans XMLSpy, 1058
voir aussi BD, 643
Bases de données dans XMLSpy, 963
Bases de lien,
référencer, 1539
BD, 643, 644
créer des requêtes, 644
éditer dans Authentic View, 643, 648
filtrer l'affichage dans Authentic View, 644
paramètres dans les requêtes BD, 644
requêtes dans Authentic View, 643
BD XML,
charger une nouvelle ligne de données dans Authentic View, 1431
charger une nouvelle ligne de données XML, 644

© 2016-2022 Altova GmbH

Bibliothèque, 1272
Big-endian, 1607
Bonne formation des documents XML, 288
BSON dans MongoDB, 747
build.xml,
activer le zip64mode dans,, 1255

C

C#,
code, 1159
integration of XMLSpy, 2073
paramètres de génération code, 1256
C++,
code, 1159
paramètres de génération code, 1256
Camemberts, 363
Caractère,
position, 1505
Catalogue,
Oasis XML, 1352
Catalogues, 459
Catalogues dans XMLSpy, 460
Catalogues et édition intelligente, 462
Catalogues et variables d'environnement, 463
Centre de support, 1654
Certificats, 420
Changements non enregistrés, 1602
Changer de mode,
en Authentic View, 618
Chapitres, 1649
Character-Set,
encodage, 1607
Chemins du fichier,
insérer dans le document XML, 290
Chercher,
et remplacer du texte dans le document, 1311
utiliser des expressions régulières, 155
Chercher dans les Schémas, 477
composants globaux, 489
définir l'étendue de, 485
exécuter la commande Chercher, 944
exécution des commandes Chercher et Remplacer, 486
fenêtre, 488
limiter la recherche à composants, 480
limiter la recherche à des propriétés et des valeurs de propriété, 482
renommer les composants globaux, 489
résultats, 488
terme de recherche, 479
terme de remplacement, 479
Chercher dans schémas,
voir Chercher dans les schémas, 477
Chercher dans XBRL, 941
chercher terme, 941
fenêtre, 946
résultats, 946
Class,
JAVA, 2014
Class ID,
in XMLSpy integration, 2076
Classes Wrapper,
dans code généré, 1256
ClassPath statement, 2014
Clé d'évaluation,
pour votre logiciel Altova, 1651
Clé privée de certificats, 420
Clé publique de certificats, 420
Clés étrangères,
désactiver dans SQLite, 999
Code,
SPL, 1258
types intégrés, 1271
CodeGeneratorDlg,
Application, 1813
CPPSettings_DOMType, 1814
CPPSettings_LibraryType, 1815
CPPSettings_UseMFC, 1815
CSharpSettings_ProjectType, 1816
OutputPath, 1816
OutputPathDialogAction, 1816
OutputResultDialogAction, 1817
Parent, 1817
ProgrammingLanguage, 1817
PropertySheetDialogAction, 1818
TemplateFileName, 1818
Codes-clé,
pour votre logiciel Altova, 1651
Coller,
en tant que Texte, 630
en tant que XML, 630
Coller en tant que,
Texte, 627

© 2016-2022 Altova GmbH

Coller en tant que,
XML, 627
Coloration de la syntaxe, 1616
Coloration syntaxique, 1605
pour XQuery, 511
COM API,
in Scripting Editor, 1668
COM-API,
documentation, 1702
Commande, 1592
ajouter à la barre d'outils/menu, 1583
menu contextuel, 1592
réinitialiser menu, 1592
supprimer du menu, 1592
Commande Annuler, 1299
Commande Apposer,
dans Mode Grille, 1348
Commande Coller, 1299
Commande Copier, 1299
Commande Couper, 1299
Commande Requête de base de données, 1440
Commande Rétablir, 1299
Commande Sélectionner tout, 1307
Commande Supprimer, 1299
Commander logiciel Altova, 1651
Commandes,
liste dans modèle de clavier, 1650
Commandes d'Affichage de la table (Mode Grid XML), 1349
Commandes de déplacement,
dans Mode Grille, 1349
Commentaires, 177
Commentaires dans l'Aperçu de schéma, 181
Commentaires dans les scripts SQL, 1449
Commentaires Globaux,
affichage en ligne de, 177
Commenter et décommenter,
dans les documents XML en Mode Texte., 290
Commenter/décommenter du texte XML, 1315
Comparaisons,
de fichiers, 1110
de fichiers et de répertoires, 1109
de répertoires, 1111
Comparaisons de fichiers, 1109, 1110
Comparaisons de répertoires, 1109, 1111
Comparer des fichiers, 1567
Comparer fichiers,
options, 1574
Comparer répertoires, 1571
Complétion automatique dans les scripts SQL, 1449
Composants de Schéma global,
chercher et renommer, 489
Compositeur,
pour séquences, 52
Compositeur de séquence,
utiliser, 52
Compositeurs dans Mode Schéma, 190
Configuration active,
pour les ressources globales, 1578
Configurations,
d'une ressource globale, 1061
Configurations dans ressources globales, 1077
Configure,
XMLSPY UI, 1687
Configurer mode,
dialogue pour Mode Modèle de contenu, 1394
Conformité XML, 1605
Connexion à SchemaAgent Server, 1403
Connexion de base de données,
exemples de configuration, 1001
Connexion de la base de données,
configuration, 965
lancer l'assistant, 966
réutilisation depuis les Ressources Globales, 1000
Content Model,
save diagram, 1389
Contenu JSON externe,
copier le Mode JSON Grille, 693
glisser-déposer dans le Mode Grille JSON, 693
Contenu XML externe,
copier le Mode XML Grille, 306
glisser-déposer dans le Mode Grille XML, 306
Context Window,
dans le débogueur XSLT/XQuery, 544
Contrainte d'identité,
basculer dans le mode de Modèle de contenu, 74
Contraintes d'identité dans le Mode Schéma, 210, 218
Contrat de licence de l'utilisateur final, 2252, 2254
Contrôle de source, 1643
activer, désactiver, 1322
afficher Historique, 1329
afficher les différences, 1330
ajouter au contrôle de source, 1326
annuler extraction, 1326
changer de fournisseur, 1333
extraire, 1324

© 2016-2022 Altova GmbH

Contrôle de source, 1643
fournisseurs pris en charge, 1320
installer un plug-in de contrôle de source, 1113
obtenir la dernière version, 1322
obtenir les fichiers, 1322
ouvrir le projet, 1321
partager depuis, 1328
propriétés, 1332
réinitialiser statut, 1333
supprimer du, 1327
Contrôler,
vérificateur orthographique, 1558
Convertir,
DB XML Oracle, 1465
données de base de données vers XML, 1473
données de MS Word de données vers XML, 1478
fichier de texte en XML, 1470
schéma à structure BD, 1484
Schéma de base de données en Schéma XML, 1478
Convertir en OIM xBRL-CSV, 1500
Convertir en OIM xBRL-JSON, 1500
Convertir en OIM xBRL-XML, 1500
Copier l'image dans le Mode Grille, 1301
Copier le texte du Mode Grille comme texte séparé par tabulation, 1301
Copier le texte en Mode Grille, 1300
Copier XPath, 1302
Copier XPointer, 1302
CoreCatalog.xml, 460
Côte-à-côte, 1616
Couleur, 1624
CR&LF, 1602
Création de documents XML,
tutoriel, 84
Créer,
DB basée sur schéma, 1484
CSS, 659
bordure du document, 662
coloration de syntaxe, 662
Fenêtre Info, 662
propriétés, 662
remplissage automatique, 662
CustomCatalog, 1352
CustomCatalog.xml, 460

D

DatabaseConnection,
ADOConnection, 1819
AsAttributes, 1819
CreateMissingTables, 1820
CreateNew, 1820
DatabaseKind, 1820
ExcludeKeys, 1821
File, 1821
IncludeEmptyElements, 1822
NumberDateTimeFormat, 1822
ODBCConnection, 1823
SQLSelect, 1824
TextFieldLen, 1824
DatabaseSpy,
arrière-plan graphique, 383
axe X du graphique, 391
axe Y du graphique, 394
axe Z du graphique, 395
couleurs de graphique, 389
fonctions du graphique à aires, 385
fonctions du graphique à barres, 385
fonctions du graphique à camembert, 385
fonctions du graphique à chandelles, 385
fonctions du graphique à jauge, 385
fonctions du graphique à ligne, 385
graphiques 3d, 396
grille du graphique, 394, 395
grille tailles de graphiques, 391
légende de graphique, 383
options de police de graphique, 398
polices de graphique, 398
tailles de graphiques, 397
titre de graphique, 383
Dates,
changer manuellement, 651
DB,
parcourir les tables dans Authentic View, 644
DB XML Oracle, 1465
gérer des Schémas XML, 1465
Parcourir base de données, 1468
db2-fn:sqlquery, 528
db2-fn:xmlcolumn, 528
Débogage XSLT/XQuery,

Débogage XSLT/XQuery,
fichiers utilisés, 533
Déboguer XPath/XQuery expressions, 584
Débogueur,
activer/désactiver point d'arrêt, 1425
activer/désactiver point de trace, 1425
afficher nœuds d'exéc. actuels, 1424
arrêter Débogueur XSLT, 1423
boîte de dialogue points d'arrêt/points trace, 1426
déboguer les fenêtres, 1427
démarrer débogueur XSLT, 1422
insérer/supprimer point d'arrêt, 1424
insérer/supprimer point de trace, 1425
options pour SOAP, 1532
paramètres, 1427
pas à pas détaillé, 1423
pas à pas principal, 1424
pas à pas sortant, 1424
pour SOAP, 796, 1529
redémarrer Débogueur XSLT, 1423
terminer session, 1423
Débogueur SOAP, 796
configurer les points d'arrêt, 808
dans Visual Studio .NET, 1144
Débogueur XQuery,
voir Débogueur XSLT/XQuery, 533
Débogueur XSLT,
activer/désactiver point d'arrêt, 1425
activer/désactiver point de trace, 1425
afficher nœuds d'exéc. actuels, 1424
arrêter Débogueur, 1423
boîte de dialogue points d'arrêt/points trace, 1426
dans Visual Studio .NET, 1144
déboguer les fenêtres, 1427
démarrer débogueur, 1422
insérer/supprimer point d'arrêt, 1424
insérer/supprimer point de trace, 1425
paramètres, 1427
pas à pas détaillé, 1423
pas à pas principal, 1424
pas à pas sortant, 1424
redémarrer Débogueur, 1423
terminer session débogueur, 1423
Débogueur XSLT/XQuery, 534
démarrer une session, 541
description de l'interface, 534
description du mécanisme, 534
fenêtre Contexte, 544
fenêtre Info, 547
fenêtre Messages, 546
fenêtre Modèles, 547
fenêtre Pile d'appel, 546
fenêtre Suivi XPath, 545
fenêtre Trace, 548
fenêtre Variables, 545
fenêtres d'information, 543
fonctions et utilisation, 533
icônes de la barre d'outils, 536
paramètres, 540
raccourcis de clavier, 559
utilisation points d'arrêt, 550
utilisation points de trace, 553
Déconnecter XMLSpy de SchemaAgent, 1404
defaultOpenContent, 181
Défaut,
encodage, 1607
menu, 1592
Définir,
couleur des graphiques, 389
couleurs de graphique, 389
fonctions du graphique à aires, 385
fonctions du graphique à barres, 385
fonctions du graphique à camembert, 385
fonctions du graphique à chandelles, 385
fonctions du graphique à jauge, 385
fonctions du graphique à ligne, 385
lignes de grille, 391, 394, 395
paramètre de l'axe Y, 394
paramètre de l'axe Z, 395
paramètres 3d, 396
paramètres axe X, 391
polices dans graphique, 398
polices de graphique, 398
tailles de graphiques, 397
titre de graphiques, 383
définition de composant,
réutiliser, 64
Définitions de schéma XML,
avancé, 64
Delete,
Application.URLDelete, 1758
Démarrage de groupe,
ajouter (menu contextuel), 1598
Démarrer,
Débogueur XSLT, 1422
Déplacer la ligne dans Authentic View, 1438

© 2016-2022 Altova GmbH

Dérivations Type Simple, 230
Dériver un type de schéma, 242
Désactiver,
point d'arrêt- Débogueur XSLT, 1425
point de trace-Débogueur XSLT, 1425
Description d'interface utilisateur, 115
Description GUI, 115
Détails Assistant de saisie, 52
Dialogs,
Application, 1825
CodeGeneratorDlg, 1825
DTDSchemaGeneratorDlg, 1827
FileSelectionDlg, 1825
GenerateSampleXMLDlg, 1826
Parent, 1826
SchemaDocumentationDlg, 1826
Dialogue Paramètres mode Texte, 1506
Dialogue points d'arrêt, 536
Dictionnaire, 1558
ajouter personnalisé, 1558
modifier existant, 1558
vérificateur orthographique, 1558
Dictionnaire personnalisé, 1558
Diff XML,
comparer des fichiers, 1574
comparer fichiers, 1567
comparer répertoires, 1571
directories,
creating with Application.URLMakeDirectory, 1758
Dissimuler les balises, 618, 620
Distribution,
des produits logiciels Altova, 2252
des produits logiciels d'Altova, 2252
DocEditEvent (obsolete),
altKey (obsolete), 1965
altLeft (obsolete), 1965
button (obsolete), 1966
cancelBubble (obsolete), 1967
clientX (obsolete), 1968
clientY (obsolete), 1968
ctrlKey (obsolete), 1969
ctrlLeft (obsolete), 1970
dataTransfer (obsolete), 1971
fromElement (obsolete), 1971
keyCode (obsolete), 1972
propertyName (obsolete), 1972
repeat (obsolete), 1973
returnValue (obsolete), 1973
shiftKey (obsolete), 1973
shiftLeft (obsolete), 1974
srcElement (obsolete), 1975
type (obsolete), 1976
DocEditView (obsolete),
ApplyTextState (obsolete), 1980
CurrentSelection (obsolete), 1980
EditClear (obsolete), 1981
EditCopy (obsolete), 1981
EditCut (obsolete), 1982
EditPaste (obsolete), 1982
EditRedo (obsolete), 1983
EditSelectAll (obsolete), 1983
EditUndo (obsolete), 1983
event (obsolete), 1984
GetAllowedElements (obsolete), 1984
GetNextVisible (obsolete), 1986
GetPreviousVisible (obsolete), 1987
IsEditClearEnabled (obsolete), 1988
IsEditCopyEnabled (obsolete), 1988
IsEditCutEnabled (obsolete), 1989
IsEditPasteEnabled (obsolete), 1989
IsEditRedoEnabled (obsolete), 1990
IsEditUndoEnabled (obsolete), 1990
IsRowAppendEnabled (obsolete), 1990
IsRowDeleteEnabled (obsolete), 1991
IsRowDuplicateEnabled (obsolete), 1991
IsRowInsertEnabled (obsolete), 1992
IsRowMoveDOwnEnabled (obsolete), 1992
IsRowMoveUpEnabled (obsolete), 1993
IsTextStateApplied (obsolete), 1993
IsTextStateEnabled (obsolete), 1994
LoadXML (obsolete), 1994
RowAppend (obsolete), 1995
RowDelete (obsolete), 1996
RowDuplicate (obsolete), 1996
RowInsert (obsolete), 1997
RowMoveDown (obsolete), 1997
RowMoveUp (obsolete), 1998
SaveXML (obsolete), 1998
SelectionMoveTabOrder (obsolete), 1999
SelectionSet (obsolete), 1999
XMLRoot (obsolete), 2000
Document, 1840
Application, 1833
AssignDTD, 1833
AssignSchema, 1833
AssignXSL, 1833

© 2016-2022 Altova GmbH

Document, 1840
AssignXSLFO, 1834
AuthenticView, 1834
Close, 1835
ConvertDTDOrSchema, 1835
CreateChild, 1837
CreateSchemaDiagram, 1838
CurrentViewMode, 1838
DataRoot, 1838
DocEditView, 1839
Encoding, 1839
EndChanges, 1839
ExecuteXQuery, 1840
ExportToDatabase, 1840
ExportToText, 1841
FullName, 1843
GenerateDTDOrSchema, 1843
GenerateProgramCode, 1844
GenerateSampleXML, 1844
GenerateSchemaDocumentation, 1844
GetExportElementList, 1846
GetPathName, 1847
GridView, 1847
IsModified, 1848
IsValid, 1848
IsWellFormed, 1849
Name, 1849
OnBeforeCloseDocument, 1831
OnBeforeSaveDocument, 1830
OnBeforeValidate, 1831
OnCloseDocument, 1832
OnViewActivation, 1832
parcourir DB XML Oracle, 1468
Path, 1850
RootElement, 1850
Save, 1850
SaveAs, 1851
Saved, 1851
SaveInString, 1851
SaveToURL, 1852
SetActiveDocument, 1852
SetEncoding, 1852
SetExternalIsValid, 1853
SetPathName, 1853
StartChanges, 1854
SwitchViewMode, 1854
Title, 1855
TransformXSL, 1855
TransformXSLFO, 1855
UpdateViews, 1856
UpdateXMLData, 1856
Vérificateur orthographique, 1558
XQuery, 1840
Document Word,
importer en tant que XML, 1478
Document XML,
assigner à la feuille de style XSLT, 1421
création d'un nouveau, 84
éditer dans le Mode Texte, 89
générer depuis un Schéma XML (éditions Enterprise et Professional), 447
ouvrir dans Authentic View, 603
parcourir DB XML Oracle, 1468
Documentation,
de fichiers de Schéma XML, 1389
de fichiers WSDL, 1517
de schéma, 79
de taxonomies XBRL, 1541
Document-level,
examples of integration of XMLSpy, 2073
Documents,
Count, 1857
Item, 1858
NewAuthenticFile, 1858
NewFile, 1858
NewFileFromText, 1859
OpenAuthenticFile, 1859
OpenFile, 1859
OpenURL, 1860
OpenURLDialog, 1860
Documents dans la fenêtre principale, 116
Documents HTML,
éditer, 660
Fenêtre Info, 660
Documents JSON,
convertir de et vers XML, 677, 772
créer nouveau, 668
éditer dans le Mode Grid, 682
éditer dans le Mode Texte, 677
expressions XQuery pour, 595
ouvrir dans XMLSpy, 668
valider, 759
Documents XML, 285
attribuer des schémas (y compr. des DTD), 288
automatiser les exécutions XQuery de, 342
automatiser les transformations XSLT de, 342

© 2016-2022 Altova GmbH

Documents XML, 285
chercher et remplacer dans, 425
contrôler la bonne formation, 288
éditer dans le Mode Grid, 293
éditer dans Mode Authentic, 332
encoder, 425
enregistrement, 286
et commenter en Mode Texte, 290
et expression XPath d'un nœud, 290
et lignes vides en Mode Texte, 290
et XQuery, 342
évaluer des expressions XPath sur, 425
Fonctions d'édition du Mode Texte pour, 290
générer des schémas depuis, 425
importer et exporter du texte, 425
insérer les chemins de fichier dans, 290
insérer XInclude, 290
modes par défaut de, 286
ouverture, 286
transformer avec XSLT, 342
validation automatique, 286
valider, 288
vérifier la validité, 94
Documents XQuery,
analyser le délai d'exécution de, 560
analyser le temps d'exécution de, 1420
documents XSLT,
éditer et gérer avec XSL Outline, 497
et fenêtre d'info, 502
Documents XULE,
éditer la prise en charge pour, 934
et des types de fichiers conformes, 934
et projets XMLSpy, 934
et taxonomie pour le remplissage automatique, 934
remplissage automatique, 934
validation de, 934
Données graphique,
exemple, 2227
Données JSON,
exemple, 671
objets, 671
tableaux, 671
types, 671
Données XML,
exporter vers fichier de texte, 1488
exporter vers la base de données, 1491
DSN Fichier,
configuration, 986
DSN Système,
configuration, 986
DSN utilisateur,
configuration, 986
DTD, 427, 1602, 1605
aller à depuis le document XML, 1372
aller à la définition depuis le document XML, 1372
attribuer au document XML, 1370
commandes de menu associées à, 1370
convertir à UML, 1378
convertir en Schéma XML, 1375
convertir en Schémas XML (éditions Enterprise et Professional), 445
éditer dans le Mode Grille (éditions Enterprise et Professional), 445
éditer dans le Mode Texte, 445
générer depuis le document XML, 1373
générer depuis un Schéma XML (éditions Enterprise et Professional), 447
générer du code depuis, 1384
générer fichier XML outline depuis, 1380
générer un document XML depuis, 445
inclure les entités, 1371
DTD et catalogues, 459
DTDSchemaGeneratorDlg,
Application, 1861
AttributeTypeDefinition, 1862
DTDSchemaFormat, 1862
FrequentElements, 1862
GlobalAttributes, 1863
MaxEnumLength, 1863
MergeAllEqualNamed, 1863
OnlyStringEnums, 1863
OutputPath, 1864
OutputPathDialogAction, 1864
Parent, 1864
ResolveEntities, 1864
TypeDetection, 1865
ValueList, 1865
Dupliquer,
ligne (dans Authentic View), 1437

E

Eclipse platform,
and XMLSpy Integration Package, 1147
Écran de démarrage, 1616

© 2016-2022 Altova GmbH

EDGAR validation on server, 1554
Éditer,
bouton macro, 1598
Éditer en tant que Texte brut,
dans Mode Grille, 1349
Éditer les enregistrements de base de données, 1454
Éditeur de définitions de table XBRL, 899
Éditeur de script,
lancer, 1566
Éditeur par défaut, 1605
Éditeur SQL,
créer la requête dans, 1453
dans la fenêtre de Requête de base de données, 1449
description de, 1449
Éditeur XBRL Formula, 873
Édition avec XMLSPY, 1602
Édition dans le Mode Texte, 149
Édition de Schéma,
modèles de contenu, 188
édition de schéma BSON, 747
Édition intelligente, 1608
Édition intelligente XPath, 492
Élément, 60
rendre optionnel, 60
restreindre le contenu, 60
Élément global,
utiliser dans le Schéma XML, 72
Élément optionnel,
rendre, 60
ElementFormDefault,
paramètres dans le Mode Conception de schéma, 1386
ElementList,
Count, 1866
Item, 1866
RemoveElement, 1866
ElementListItem,
ElementKind, 1866
FieldCount, 1867
Name, 1867
RecordCount, 1867
Éléments d'extensibilité, 1357
éléments Globaux, 181
Éléments répétés, 1608
Éléments vides, 1605
E-mail,
envoyer des fichiers par, 1293
Emballer dans Élément commande,
dans Mode Grille, 1348
Emplacements approuvés pour les scripts Authentic, 1438
emplacements de fichier SPP, 1316
En cours de validation,
documents XML, 94
Encodage,
de fichiers, 1286
défaut, 1607
Enregistrer comme image, 1306
Enregistrer fichiers,
encodage de, 1286
Enregistrer votre logiciel Altova, 1651
Entités,
dans XML basé sur Schéma XML, 288
définir dans Authentic View, 630, 652
insérer dans Authentic View, 630
insérer dans le Authentic View, 613
Entités parsées externement, 1605
Énumération,
définir pour les attributs, 74
Enumerations,
in XMLSpyControl, 2131
SPYAttributeTypeDefinition, 2001
SPYAuthenticActions, 2001
SPYAuthenticDocumentPosition, 2001
SpyAuthenticElementActions, 2002
SPYAuthenticElementKind, 2002
SPYAuthenticMarkupVisibility, 2002
SPYDatabaseKind, 2003
SPYDialogAction, 2003
SPYDOMType, 2003
SPYDTDSchemaFormat, 2004
SPYEncodingByteOrder, 2004
SPYExportNamespace, 2004
SPYFrequentElements, 2005
SPYKeyEvent, 2005
SPYLibType, 2006
SPYLoading, 2006
SPYMouseEvent, 2006
SPYNumberDateTimeFormat, 2007
SPYProgrammingLanguage, 2007
SPYProjectItemTypes, 2007
SPYProjectType, 2008
SPYSampleXMLGenerationOptimization, 2008
SPYSampleXMLGenerationSchemaOrDTDAssignment, 2009
SPYSchemaDefKind, 2009
SPYSchemaDocumentationFormat, 2010

© 2016-2022 Altova GmbH

Enumerations,
SPYTextDelimiters, 2011
SPYTextEnclosing, 2011
SPYTypeDetection, 2011
SPYURLTapes, 2011
SPYViewModes, 2012
SPYVirtualKeyMask, 2013
SPYXMLDataKind, 2013
Énumérations de types simples, 232
Esapce blanc dans XBRL Inline, 1639
Espace blanc,
supprimer, 1307
espace de noms,
dans schémas, 51
Espaces de noms,
dans des documents WSDL, 783
dans les taxonomies XBRL, 852
paramètres dans le Mode Conception de schéma, 1386
Espaces de noms cibles dans XBRL, 1538
Espaces de noms dans XBRL, 1537
Éteindre validation automatique, 1605
Évaluateur XPath,
utilisation, 1350
Évaluer un XPath, 124
Event, 1741, 1742, 1743, 1795, 1796, 1797, 1798, 1799, 1800, 1801, 1830, 1831, 1832, 1889, 1890, 1891
Events, 1709
Excel 2007, 281, 950
Exception Mémoire insuffisante,
résoudre, 1255
Exécuter XULE, 1552
Exécution XQuery, 518, 1417
Exécution XULE,
comment, 940
options, 940
Exigences de mémoire, 2249
Explorer, 1605
Exporter les données XML dans des fichiers de textes, 1488
Exporter les données XML dans la base de données, 1491
ExportSettings,
CreateKeys, 1868
ElementList, 1868
EntitiesToText, 1868
ExportAllElements, 1868
FromAttributes, 1869
FromSingleSubElements, 1869
FromTextValues, 1869
IndependentPrimaryKey, 1869
Namespace, 1869
SubLevelLimit, 1870
Expressions régulières, 1313
chercher et remplacer en utilisant, 155
Dans string recherche, 1307
Expressions Watch dans XPath/Débogueur XQuery, 584
Expressions XPath/XQuery,
Créé dans la fenêtre XPath/XQuery, 592
déboguer, 584
évaluation, 578
pour des documents JSON, 595
Expressions XQuery dans le Mode Grille JSON, 701
Expressions XQuery dans le Mode Grille XML, 314, 316
Extensions Altova,
fonctions graphiques (voir fonctions graphique), 2142
Extensions de ficher,
pour les fichiers XQuery, 510
Extensions de fichier,
personnaliser, 1352

F

Facettes de types simples, 232
Favoris, 1649
Fenêtre d'ancrage, 115
Fenêtre d'info, onglet XSLT,
description de, 502
et création de dossiers zip, 502
et documents XSLT, 502
et projets, 502
voir aussi XSL Outline, 502
fenêtre de débogueur,
disposition, 548
Fenêtre de masquage automatique, 115
Fenêtre de message HTTP,
envoyer la requête, 813
et codes de statut HTTP, 821
et en-tête Accept, 821
et WADL, 819
importer une requête dans, 819
journaux, 821
Méthodes HTTP, 813
recevoir une réponse, 821
Fenêtre de messages, 122

© 2016-2022 Altova GmbH

Fenêtre de messages, 122
affichage de, 1647
Fenêtre de projet, 118
affichage de, 1647
Fenêtre d'info, 120
Fenêtre Graphiques, 129
Fenêtre Info,
affichage de, 1647
dans le Débogueur XSLT/XQuery, 547
pour documents CSS, 662
pour documents HTML, 660
Fenêtre Info CSS, 662
Fenêtre Info HTML, 660
fenêtre Messages,
dans Débogueur XSLT/XQuery, 546
fenêtre Modèles,
dans Débogueur XSLT/XQuery, 547
Fenêtre Outline XSL, 125, 498
fenêtre Pile d'appel,
dans Débogueur XSLT/XQuery, 546
Fenêtre principale, 116
Fenêtre Requête de base de données, 1440
Fenêtre Sortie,
affichage de, 1647
fenêtre Suivi XPath,
dans Débogueur XSLT/XQuery, 545
fenêtre trace, 553
dans le Débogueur XSLT/XQuery, 548
Fenêtre Trouver dans fichier, 126
Fenêtre Trouver dans GetXBRL, 129
Fenêtre Trouver dans Schémas, 128
fenêtre Variables,
dans le Débogueur XSLT/XQuery, 545
Fenêtre XPath, 124
Fenêtre XPath/XQuery, 575
Fenêtre XULE, 131
pour la requête interactive d'instances XBRL, 936
Fenêtres d'Information,
disposer, 548
Fenêtres flottantes, 115
Feuille de style XSLT,
assigner au document XML, 1420
assigner document XML à, 1421
ouvrir, 1422
Feuille de style XSLT pour FO,
assigner au document XML, 1421
Feuilles de style XSLT,
analyser le délai d'exécution de, 560
analyser le temps d'exécution de, 1420
Fichier, 1602
création d'un nouveau, 1277
encodage, 1286
encodage par défaut, 1607
enregistrer, 1287
envoyer par e-mail, 1293
fermer, 1287
onglet, 1602
options d'impression, 1294
Ourvrir les options, 1602
ouvrir, 1281
Fichier CSV,
importer en tant que XML, 1470
fichier de métadonnées, 314, 316, 701
fichier de métadonnées JSON pour filtres et formules, 701
fichier de métadonnées XML pour filtres, 314
fichier de métadonnées XML pour les formules, 316
Fichier PXF,
générer les documents de sortie de <% AUTH-VIEW%>, 1438
Fichier SPS,
assigner au fichier XML, 1430
Fichier texte,
exporter des données XML vers, 1488
importer en tant que XML, 1470
Fichier XML,
générer depuis DTD ou Schéma XML, 1380
Fichier XML de modèle,
dans Authentic View, 603
Fichier XML Ressources, 1061
Fichiers,
ajouter au contrôle de source, 1326
comparer deux, 1567
options de comparaison, 1574
utilisés les plus récemment, 1297
Fichiers binaires Avro, 780
Fichiers EPUB, 958
Fichiers modèles,
pour de nouveaux documents, 1277
Fichiers non-XML, 1605
Fichiers utilisés les plus récemment,
liste de, 1297
fichiers WSDL,
Validation, 1357
Fichiers XQuery,
configurer les extensions de fichier dans XMLSpy, 510

© 2016-2022 Altova GmbH

Fichiers ZIP, 281, 950
création dans Archive View, 956
édition dans Archive View, 956
FileSelectionDlg,
Application, 1870
DialogAction, 1870
FullName, 1871
Parent, 1871
Filtres dans le Mode Grid XML, 314
Filtres dans le Mode Grille XML,
paramètres de, 326
Filtres dans le Mode JSON Grid, 701
paramètres de, 713
Firebird,
Se connecter par le biais de JDBC, 1002
se connecter par le biais de ODBC, 1003
Fonction zoom,
dans le Mode Design de schéma, 1398
Fonctions d'extension dans les scripts MSXSL, 2231
Fonctions d'extension pour XSLT et XQuery, 2230
fonctions du graphique à aires, 385
fonctions du graphique à barres, 385
fonctions du graphique à camembert, 385
fonctions du graphique à chandelles, 385
fonctions du graphique à jauge, 385
fonctions du graphique à ligne, 385
Fonctions graphique,
liste, 2217
structure de donnée graphique pour, 2221
Fonctions XSLT,
Fenêtre XSL Outline, 498
FOP,
polices, 344
Formatage dans le Mode Texte, 144
Formats d'image,
dans Authentic View, 655
Formats graphiques,
dans Authentic View, 655
formats xBRL OIM, 948
Formules,
créer depuis l'Aperçu de mise en page de table, 930
Formules dans le Mode Grille XML, 316
paramètres de, 326
Formules dans le Mode JSON Grid, 701, 702
paramètres de, 713

G

Generate Sample XML, 2001, 2008, 2009
GenerateSampleXMLDlg,
Application, 1883
FillWithSampleData, 1885
NonMandatoryAttributes, 1886
NonMandatoryElements, 1886
Parent, 1887
RepeatCount, 1887
TakeFirstChoice, 1888
Générateur d'expression, 592
Générateur de Code, 1159, 1384
Générer,
du code depuis le schéma, 1159
structure BD basée sur schéma, 1484
Gestion de l'espace blanc, 338
Gestion de projet dans XMLSpy, 110
Gestionnaire de Contrôle de source, 1333
Gestionnaire de schéma,
aperçu CLI, 438
aperçu de, 428
Commande Aide CLI, 438
Commande de mise à jour CLI, 443
Commande de mise à niveau CLI, 444
Commande désinstallation CLI, 442
Commande Info CLI, 439
Commande Initialiser CLI, 439
Commande Installer CLI, 440
Commande Liste CLI, 441
Commande Réinitialiser CLI, 442
comment exécuter, 431
corriger un schéma, 436
désinstaller un schéma, 437
installer un schéma, 436
mettre à niveau un schéma, 436
recenser les schémas par statut dans, 434
réinitialiser, 437
statut de schémas dans, 434
Gestionnaire de taxonomie,
aperçu CLI, 835
aperçu de, 826
Commande Aide CLI, 836
Commande de mise à jour CLI, 841
Commande de mise à niveau CLI, 841

© 2016-2022 Altova GmbH

Gestionnaire de taxonomie,
Commande désinstallation CLI, 840
Commande Info CLI, 836
Commande Initialiser CLI, 837
Commande Installer CLI, 837
Commande Liste CLI, 838
Commande Réinitialiser CLI, 839
comment exécuter, 830
corriger une taxonomie, 833
désinstaller une taxonomie, 834
installation d'une taxonomie, 833
mettre à niveau une taxonomie, 833
recenser les taxonomies par statut dans, 832
réinitialiser, 834
statut de taxonomies dans, 832
Glisser-déposer dans le Mode Grille JSON, 693
Glisser-déposer dans le Mode Grille XML, 306
Global,
paramètres, 1601
Grammaire, 1605
Grand balisage (dans Authentic View), 1436
Graphiques,
ajouter une légende, 383
aperçu, 347
apparence, 374
axe X, 391
axe Y, 394
axe Z, 395
configuration de graphique, 370
couleur d'arrière-plan, 383
couleur de série, 389
dans le Profileur XSLT/XQuery, 571
définir les couleurs, 389
données de graphique, 367
exemple (avancé), 402
exemple (chandelier), 408
exemple (simple), 400
exporter, 399
fonctions du graphique à aires, 385
fonctions du graphique à barres, 385
fonctions du graphique à camembert, 385
fonctions du graphique à chandelles, 385
fonctions du graphique à jauge, 385
fonctions du graphique à ligne, 385
gamme de couleur, 389
lignes de grille, 391, 394, 395
marges, 397
onglets multiples pour, 347
paramètres 3d, 396
polices, 398
recharger, 347
schéma de couleur, 389
Sélection de l'Axe X, 358
Sélection de l'Axe Y, 363
superpositions, 369
supprimer la légende, 383
taille de la jauge, 397
tailles, 397
titre, 383
XPath de source, 354
Graphiques dans le Mode Grille, 320
Graphiques dans le Mode JSON Grid, 707
GridView,
CurrentFocus, 1891
Deselect, 1891
IsVisible, 1891
OnBeforeDrag, 1889
OnBeforeDrop, 1889
OnBeforeStartEditing, 1890
OnEditingFinished, 1890
OnFocusChanged, 1891
Select, 1892
SetFocus, 1892
Grille d'aperçu des définitions, 718
Groupes d'attribut dans le Mode Schéma, 212
Groupes d'attributs Globaux, 181
Groupes de modèle, 181
Guides de retrait, 1506

H

Help menu, 1649
Historique de navigation, 248
Hotkey, 1588
HTML, 659
integration of XMLSpy, 2078
HTML example,
of XMLSpyControl integration, 2076, 2077, 2078
HTTP, 812

© 2016-2022 Altova GmbH

I

IBM DB2,
attribuer Schémas XML au fichier XML, 1460
gérer les schémas XML, 1457
gestion et attribution de schéma, 1456
se connecter par le biais de JDBC, 1006
se connecter par le biais de ODBC, 1008
IBM DB2 for i,
se connecter par le biais d'JDBC, 1014
IBM DB2 pour i,
se connecter par le biais d'ODBC, 1015
IBM Informix,
se connecter par le biais de JDBC, 1018
Icône,
afficher grandes, 1597
ajouter la barre d'outils/menu, 1583
Images dans le JSON Grid View, 706
Images dans le Mode Grid, 319
Importation XML,
basée sur schéma, 1483
Importer,
données de base de données basées sur Schéma XML, 1483
données de base de données comme XML, 1473
données de MS Word de document en tant que XML, 1478
importer en tant que XML, 1470
structure de base de données en Schéma XML, 1478
Importer des taxonomies, 1539
Imprimer,
depuis Authentic View, 614
Indentation dans le Mode Texte, 144
Infobulle, 1597
afficher, 1597
afficher raccourcis dans, 1597
Information de Copyright, 2252
Information juridique, 2252
Informations générales, 2249
Informations techniques, 2249
Inline XBRL in XMLSpy, 872
Insérer,
ligne (dans Authentic View), 1437
point d'arrêt - Débogueur XSLT, 1424
point de trace - Débogueur XSLT, 1425
Insérer avant/après commande,
dans Mode Grille, 1348
Integrating,
XMLSpy in applications, 2065
Integration Package for Eclipse,
installing, 1147
Internet, 1654
Internet usage,
dans les produits Altova, 2250
IT Feuille de style, 1420, 1421

J

Java, 2082
API, 2014
aux classes générées, 1239
ClassPath, 2014
code, 1159
éviter des exceptions dans le code généré, 1255
Java settings, 1636
Java virtual machine,
path setting, 1636
JDBC,
configurer une connexion (Windows), 990
en tant qu'interface de connexion de données, 965
se connecter à Teradata, 1051
JRE,
for XMLSpy Integration Package for Eclipse, 1147
JScript,
scripting with XMLSpy, 1659
json,
convertir instance en/de instance XML, 1494
Convertir Schéma JSON en/de Schéma XML, 1497
et XPath, 763
et XPath/Fenêtre de sortie XQuery, 763
et XQuery, 763
XQuery expressions for, 766
JSON Lines, 676
JSON Schema, 676

L

Langage de script, 1642
Largeur maximum de la cellule, 1616
Largeurs optimales, 1505, 1616
Licence, 2254

© 2016-2022 Altova GmbH

Licence, 2254
information à propos de, 2252
Licence de produit de logiciel, 2254
Licences,
pour votre logiciel Altova, 1651
License metering,
des produits Altova, 2253
Liens,
suivre dans Authentic View, 630
Ligne,
ajouter (dans Authentic View), 1437
aller à, 1505
déplacer haut/bas, 1438
dupliquer (dans Authentic View), 1437
insert (in Authentic View), 1437
supprimer, 1438
supprimer (dans Authentic View), 1437
Lignes vides,
dans les documents XML en Mode Texte, 290
Linkbases dans taxonomies, 854
Little-endian, 1607
loading, 1860
Localisateur d'objet,
dans la Fenêtre Requête de base de données, 1445
Longueur de ligne,
retour à la ligne dans le mode texte, 1505

M

Macro,
ajouter au menu/barre d'outils, 1594
bouton éditer, 1598
Macros,
developing, 1659, 1664
enabling, 1671, 1683
exécuter les macros d'application, 1567
running, 1684
MainCatalog, 1352
Manuel de l'utilisateur, 28
MapForce, 1380
Marge de ligne, 1506
Marge de signet, 1506
Marge pliante, 1506
MariaDB,
se connecter par ODBC, 1020
Marqueurs de fin de ligne, 1506
Marqueurs Espaces blancs, 1506
Masquer le balisage (dans Authentic View), 1436
Mémpoire,
stockage de l'information de schéma, 1385
Menu, 1592
ajouter macro à, 1594
ajouter/supprimer commande, 1583
Authentic, 1428
Conception de schéma, 1386
Convertir, 1470
Défaut/XMLSPY, 1592
DTD/Schéma, 1370
Éditer, 1298
Help, 1649
json, 1366
Mode, 1501
Outils, 1557
personnaliser, 1592
Projet, 1316
SOAP, 1522
supprimer les commandes depuis, 1592
WSDL, 1509
XML, 1347
XSL/XQuery, 1409
Menu Authentic, 1428
affichage de la balise, 618
édition des tables dynamiques, 618
Menu Conception de schéma, 1386
Menu contextuel,
commandes, 1592
Menu Convertir, 1470
menu DTD/Schéma, 1370
Menu du Menu Contextuel,
pour personnalisation, 1598
Menu Édition, 1298
Menu Fenêtre, 1647
Menu Fichier, 1277
Menu JSON, 1366
Menu Mode, 1501
Menu Outils, 1557
Menu projet, 1316
Menu SOAP, 1522
Menu WSDL, 1509
Menu XBRL, 1533
Menu XML, 1347
Menu XSL/XQuery, 1409
Menus contextuels,
dans Authentic View, 627

© 2016-2022 Altova GmbH

Messages d'assertion, 238
Messages de validation, 122
Mettre à jour la commande des assistants de saisie, 1358
Microsoft Access,
se connecter par le biais d'ADO, 972, 1022
Microsoft Office 2007, 281, 950
Microsoft SQL Server,
se connecter par le biais d'ODBC, 1027
se connecter par le biais d'ADO, 1025
Microsoft® SharePoint® Server, 1339
MIME, 1605
Mise à jour XQuery, 518
Mode,
Agrandir, 1504
Aller à la ligne/au caract, 1505
Aller au fichier, 1506
Largeurs optimales, 1505
Mode Design Schéma, 1502
Mode Grille amélioré, 1502
Mode Navigateur, 1503
Mode Texte, 1501
Réduire, 1504
Mode Authentic,
éditer XML dans, 332
Mode Avro, 780
Mode Base de données/Table,
comment utiliser, 100
Mode de design WSDL,
consulter le fichier dans, 250
description de, 250
Fenêtre principale, 251
fonction, 250
Liaisons, 251
PortTypes, 251
Services, 251
Mode de Modèle de contenu,
assigner des types conditionnels, 203
description générale, 188
éditer dans, 197
Mode de Schéma JSON, 718
Mode Design de schéma,
Afficher diagramme, 1398
Afficher tous les globaux, 1398
fonction zoom, 1398
Mode Écran divisé dans le Mode Texte, 165
Mode générateur, 592
Mode Grid (JSON),
voir le Mode JSON Grid, 682
Mode Grid (XML), 293
Affichage du document, 295
ajouter suppression nœuds, 299
assistants d'entrée, 299
créer des graphiques dans, 320
éditer la structure du document, 299
Édition de l'Affichage de la table, 309
Expressions XQuery dans, 314, 316
fichier de métadonnées, 314, 316
filtrage de l'affichage, 316
fonctions de, 295
formules pour calculer une sortie, 314
Glisser-déposer dans, 306
images dans, 319
Menu contextuel, 323
Mode Grid des documents JSON,
voir le Mode JSON Grid, 682
Mode Grille, 98, 168, 1502, 1504
ajouter des éléments et des attributs, 98
entrée des données dans, 88
et Mode Table, 100
utiliser des Assistants de saisie, 98
Mode Grille (XML),
éditer le contenu du document, 302
éditer les types de nœuds, 302
rechercher et remplacer, 302
validation, 302
Mode Grille amélioré, 1502
Mode JSON Grid,
ajouter des composants en tant qu'enfant, 1367
apposer des composants, 1367
boutons de commande dans la cellule, 690
Commande Type, 1366
créer des graphiques dans, 707
document affiché dans, 685
éditer le contenu du document dans, 690
éditer structure document dans, 689
insérer des composants, 1367
Remplissage automatique, 690
symboles des types différents, 685
touches de commande dans les cellules, 689
wrapper composant dans array, 1367
wrapper composant dans objet, 1367
Mode JSON Grille, 682
affichage de l'image dans, 706
Affichage de la table, 696
Commande Affichage de la table, 1368

© 2016-2022 Altova GmbH

Mode JSON Grille, 682
commandes de menu contextuel de, 710
Expressions XQuery dans, 701, 702
fichier de métadonnées, 701, 702
filtrer le mode, 701
formules pour calculer une sortie, 701, 702
Glisser-déposer dans, 693
Icônes de l'Affichage de la table, 696
réévaluer tous les filtres et formules, 1369
renverser les lignes et les colonnes de la table, 1368
trier colonne dans la table, 1368
Mode Modèle de contenu, 49
compositeurs et composants, 190
configurer, 1394
description d'interface, 190
objets de diagramme, 190
Mode Navigateur,
actualiser le contenu, 1508
arrêter le chargement de page, 1508
déplacement avant et arrière, 1508
fenêtre séparée, 1508
taille de police, 1508
Mode par défaut,
paramètres dans la Fenêtre principale, 1605
Mode Quitter, 1602
Mode Schéma, 170
Assistant à la saisie des Composants, 226, 232
Assistant de saisie Détails, 230
assistants à la saisie, 226
configurer l'affichage, 58
déplacement avant et arrière, 248
Mode Schéma JSON,
$ref, 725
ajouter des schémas externes, 722
Assistant à la saisie Contraintes, 722
Assistant à la saisie Détails, 722
Assistants à la saisie, 722
configurer, 755
définitions globales et locales, 725
Mode Design, 727
objets, 729
propriétés d'objet, 729
Références élargies aux Schémas JSON, 725
Mode Schéma, chercher dans,
voir Chercher dans les schémas, 477
Mode Table, 1608
comment utiliser, 100
Mode Texte, 143, 1501
activer/désactiver remplissage automatique, 1608
Assistants à la saisie dans, 164
éditer dans, 89
et commenter dans les documents XML, 290
et lignes vides dans les documents XML, 290
Fonctions d'édition spéciales pour les documents XML, 290
fonctions d'édition intelligentes, 149
formatage de texte, 144
Mode Écran divisé dans, 165
numérotation de ligne dans, 146
pliage de source dans, 146
polices de schéma, 1626
propriétés de police, 144
raccourcis clavier, 167
retrait, 144
retrait dans, 146
signets dans, 146
word-wrapping, 144
Mode WSDL,
Assistant à la saisie Aperçu, 255
Assistant à la saisie Détails, 255
assistants à la saisie, 255
importer dans le document WSDL, 255
Mode XBRL, 263
assistants à la saisie dans, 271
Onglet Calcul dans la fenêtre principale, 268
Onglet Définition dans la fenêtre principale, 268
Onglet Éléments dans la fenêtre principale, 263
Onglet Présentation dans la fenêtre principale, 268
paramètres, 1545
Mode XBRL, recherche dans,
voir chercher dans XBRL, 941
Mode XML Grid,
Voir Mode Grid, 293
mode XSD, 172
modèle Authentic View, 603
Modèle de clavier, 1650
Modèle de contenu,
basculer les attributs, 74
créer un modèle de base, 52
Modèles,
de documents XML dans Authentic View, 1429
Modèles de contenu,
de composants de schéma, 188
Modèles de contenu ouverts, 207
Modèles de contenu ouverts par défaut, 207
Modèles de types simples, 232

© 2016-2022 Altova GmbH

Modèles de valeur (XSLT 3.0),
Édition intelligente XPath dans, 492
Modèles nommés, 498
Modèles XSLT,
Fenêtre XSL Outline, 498
Modes d'édition, 139
MongoDB et BSON, 747
Moteurs Altova,
dans les produits Altova, 2249
MS SQL Server,
extensions de schéma, 1401
paramètres de schéma, 1402
MSXML, 1631
générer du code pour, 1256
msxsl:script, 2231
Multiutilisateur, 1602
MySQL,
se connecter par le biais d'ODBC, 1033

N

Navigateur, 1616
Mode, 1503
Navigation,
raccourcis dans le design de schéma, 77
Nœud,
afficher nœud d'exéc. actuel, 1424
Noms de méthode lors de la génération de code,
réserver, 1255
Non sélectionnés, 1504
Notations dans le Mode Schéma, 181
Nouveau document XML,
créer, 84
Nouveau fichier,
créer, 1277
Nouvelles fonctions, 29
Numérotation de ligne dans Mode Texte, 146

O

OASIS,
Catalogue XML, 1352
Objets globaux,
dans SPL, 1262
Occurrences,
nombre de, 52
ODBC,
configurer une connexion, 986
en tant qu'interface de connexion de données, 965
se connecter à MariaDB, 1020
se connecter à Teradata, 1053
Office Open XML, 281, 950
OIM, 948, 1500
OLE DB,
en tant qu'interface de connexion de données, 965
OOXML,
voir sous Office Open XML, 281, 950
Open Office XML,
créer dans Archive View, 952
éditer dans Archive View, 952
fichiers d'exemples, 954
openContent, 181
OpenJDK,
en tant que Java Virtual Machine, 990
Optimisation de la vitesse XSL, 505
Optimiseur de vitesse XSL, 1411
Options,
arrière-plan graphique, 383
axe X du graphique, 391
axe Y du graphique, 394
axe Z du graphique, 395
couleurs de graphique, 389
fonctions du graphique à aires, 385
fonctions du graphique à barres, 385
fonctions du graphique à camembert, 385
fonctions du graphique à chandelles, 385
fonctions du graphique à jauge, 385
fonctions du graphique à ligne, 385
graphiques 3d, 396
grille du graphique, 391, 394, 395
légende de graphique, 383
polices de graphique, 398
tailles de graphiques, 397
titre de graphique, 383
Options de XQuery Update, 1634
options d'ouverture,
fichier, 1602
Options de formatage, 1602
Options de prise en charge, 137
options de validation XBRL, 1637
Options d'impression, 1294
Options du vérificateur orthographique, 1562

© 2016-2022 Altova GmbH

Options Optimiseur de vitesse XSL, 1634
Options XQuery, 1634
Oracle,
extensions de schéma, 1399
Paramètres de schéma, 1400
Outils,
voir aussi Applications externes, 1587
Ouvrir,
fichier, 1281
Overrides, 181
Overview,
of XMLSpy API, 1703

P

Pack d'intégration XMLSpy, 1140
Page de code, 1624
Panneau AACID, 210
Paramètres, 134, 1601
aperçu de, 46
arrière-plan graphique, 383
axe X du graphique, 391
axe Y du graphique, 394
axe Z du graphique, 395
couleurs de graphique, 389
dans les requêtes BD, 644
Débogueur XSLT, 1427
fonctions du graphique à aires, 385
fonctions du graphique à barres, 385
fonctions du graphique à camembert, 385
fonctions du graphique à chandelles, 385
fonctions du graphique à jauge, 385
fonctions du graphique à ligne, 385
graphiques 3d, 396
grille du graphique, 391, 394, 395
légende de graphique, 383
passer à la feuille de style via l'interface, 1413
polices de graphique, 398
script, 1642
tailles de graphiques, 397
titre de graphique, 383
Paramètres de comparaison de fichier, 1574
Paramètres de programme, 1601
Paramètres de sauvegarde automatique, 1602
Paramètres de table, 924
Paramètres de validation, 1602
Paramètres d'impression, 1296
Paramètres du Mode Grid (JSON), 713
Paramètres du Mode Grille (XML), 326
paramètres XSLT,
passer à la feuille de style via l'interface, 1413
Parcourir,
DB XML Oracle, 1468
Parent, 1850
Parseur,
/XSLT, 1631
Pas à pas détaillé,
XSLT debugger, 1423
Pas à pas principal,
Débogueur XSLT, 1424
Pas à pas sortant,
Débogueur XSLT, 1424
PDF,
transformer dans XMLSpy, 494
Pending Update List (PUL), 521
Période d'évaluation,
des produits logiciels Altova, 2252
des produits logiciels d'Altova, 2252
Personnalisation, 134
Personnalisation de catalogues, 462
Personnaliser, 1592
commandes barre d'outils/menu, 1583
macros, 1594
menu, 1592
menu contextuel, 1592
Personnaliser le menu contextuel, 1598
perspective XMLSpy dans Eclipse, 1149, 1155
Perspectives de débogage dans Eclipse, 1154
perspectives du débogueur XMLSpy dans Eclipse, 1158
Pile d'appel (XPath/XQuery), 584
Pilotes de base de données,
Aperçu, 969
Pilotes ODBC,
vérifier la disponibilité de, 986
Plateforme Eclipse,
et XMLSpy, 1146
perspective XMLSpy dans, 1149, 1155
Perspectives de débogage de XMLSpy, 1154
perspectives du débogueur XMLSpy, 1158
Points d'entrée XMLSpy dans, 1152
Plateformes,
pour les produits Altova, 2249
Pliage de source dans Mode Texte, 146
Plug-in,

© 2016-2022 Altova GmbH

Plug-in,
ATL sample files, 1690
registration, 1686
User interface configuration, 1687
XMLSPY, 1686
Plugin XMLSpy pour VS .NET,
installer, 1140
Point d'arrêt,
Boîte de dialogue, 1426
Point de trace,
boîte de dialogue, 1426
Points d'arrêt,
utiliser dans le débogueur SOAP, 808
utiliser dans le débogueur XSLT/XQuery, 550
Points d'arrêt dans XPath/Débogueur XQuery, 584
Points de débogage dans XPath/Débogueur XQuery, 584
Points de trace,
utiliser dans le Débogueur XSLT/XQuery, 553
Points de trace dans XPath/Débogueur XQuery, 584
Police, 1626
Police, 1626
schéma, 1626
Polices dans le Mode Texte, 144
Polices de Grille, 1624
Polices de schéma, 1626
Polices PDF, 344
Polices WSDL, 1629
Polices XBRL, 1630
Position, 1505
Caractère, 1505
Ligne, 1505
PostgreSQL,
se connecter nativement, 995
se connecter par le biais d'ODBC, 1043
PowerPoint 2007, 281, 950
Préfixe d'espace de noms,
insérer dans le Mode Grille, 1358
Présentation, 1616
Pretty-print,
dans Mode Texte, 1307
Prise en charge Altova, 137
Prise en charge Unicode,
dans les produits Altova, 2250
Processeur externe XSL, 1631
Processeur XQuery,
dans les produits Altova, 2249
Processeurs XSLT,
dans les produits Altova, 2249
Processus de communication,
Débogueur SOAP, 798
Processus de communication SOAP, 798
Produits Altova, 137
profil décompte des arrêts, 560
profil graphe d'appel, 560
Profilage, 1420
Décompte des arrêts, 560
Graphe d'appels, 560
Profiler, 560
Profileur, 560
Profileur XQuery,
graphiques des résultats, 571
Profileur XSLT,
graphiques des résultats, 571
Profileur XSLT/XQuery, 560
Programmers' Reference, 1657
Projet,
propriétés, 1343
Projets, 1335
ajouter au contrôle de source, 1326
ajouter des fichiers à, 1334
ajouter des ressources globales à, 1335
ajouter dossier à, 1336
ajouter dossiers externes à, 1336
ajouter dossiers Web externes à, 1339
ajouter fichiers actifs à, 1335
ajouter fichiers liés à, 1335
ajouter l'URL à, 1335
aperçu, 1316
aperçu de, 1079
Avantages d'utiliser, 1085
comment créer et éditer, 1080
création d'un nouveau, 1319
enregistrer, 1080, 1320
fermer, 1319
nommer, 1080
ouvrir, 1319
propriétés de, 1080
recharger, 1319
Traitement par lots avec, 1085
utiliser, 1085
utilisés les plus récemment, 1346
Projets dans XMLSpy,
avantages de, 110
comment créer, 110
PUBLIC,

© 2016-2022 Altova GmbH

PUBLIC,
identificateur - catalogue, 1352
PUL dans XQuery Update, 521

R

Raccourci, 1588
afficher dans infobulle, 1597
attribuer/supprimer, 1588
Raccourci de clavier, 1588
Raccourcis, 283
Recharger, 1602
fichiers modifiés, 1286
Recherche,
texte dans document, 1307
Voir Rechercher, 1311
Recherche dans XBRL,
voir chercher dans XBRL, 941
Redefines, 181
Redémarrer,
Débogueur XSLT, 1423
Réduire,
non sélectionnés, 1504
Référence C#,
aux classes générées, 1223
Référence C++,
aux classes générées, 1208
Référence de l'utilisateur, 1276
Référencer les Schémas JSON, 725
Régions dans les scripts SQL, 1449
Register,
plug-in, 1686
Registre,
paramètres, 1601
Règles,
de validation de schéma (voir Règles de schéma), 452
Règles de schéma,
ajouter des ensembles de règles à un schéma, 452
définir, 454
Règles de schémas, 452
Réinitialiser,
barre d'outils & commandes de menu, 1585
commandes de menu, 1592
raccourci, 1588
Relations dans Taxonomies, 860, 862, 865
Relations de schéma nommées,
paramètres de schéma MS SQL Server, 1402
Relations de schéma sans nom,
paramètres de schéma MS SQL Server, 1402
Remplacer, 126
texte, 1307
Texte dans de multiples fichiers, 1313
texte dans document, 1311
utiliser des expressions régulières, 155
Remplissage automatique,
activer/désactiver mode texte, 1608
Répertoires,
comparer deux, 1571
Requête,
voir sous Requête de base de données, 1440
voir sous Fenêtre Requête de base de données, 1440
voir sous XQuery, 1440
Requête de base de données,
créer la requête, 1453
Résultats de, 1454
Résultats d'édition, 1454
Se connecter à une BD pour la requête, 1442
Volet de Messages, 1454
Volet Navigateur dans dans la Fenêtre Requête de base de données, 1445
Requêtes,
pour afficher les BD dans Authentic View, 644
Resources globales,
configuration active pour, 1578
Ressources globales, 1060
barre d'outils activation, 1585
changer les configurations, 1077
définir, 1061, 1577
définir le type de dossier, 1068
définir le type de fichier, 1063
définissant le type de base de données, 1070
utiliser, 1073, 1076, 1077
utiliser le type de fichier et le type de dossier, 1073
Ressources Globales Altova,
voir sous Ressources Globales, 1060
Retour à la ligne, 1505
activer/désactiver, 1505
retour à la ligne activer/désactiver, 1505
Retrait,
dans Mode Texte, 1307
RichEdit, 1437
Rôles d'arc dans XBRL, 1533
Rôles de lien dans les taxonomies XBRL, 860
Rôles de lien dansn XBRL, 1535

© 2016-2022 Altova GmbH

RootCatalog.xml, 460

S

Sauts de ligne, 1602
Sauvegarde automatique, 141
Sauvegardes, 141
save, 1852
schema, 1751
génération de Code, 1159
Schéma,
attribuer à XML BD, 1460
convertir à UML, 1378
créer BD basée sur schéma, 1484
documentation, 79
gestion et attribution dans les bases de données IBM DB2, 1456
Mode Design, 1502
ouvrir schéma WSDL, 789
paramètres, 1602
Police de documentation, 1626
Voir aussi Schéma XML, 1370
voir Schéma XML, 49
Schéma Avro, 668
description, 776
terminologie, 776
Schéma JSON, 668
ajouter des définitions globales, 721
allOf, 751
any, 745
anyOf, 751
conditionnels, 753
définition de chaîne, 743
définitions numériques, 743
description, 674
forbidden, 745
Générer depuis une instance JSON, 768
if-then-else, 753
multiple, 745
not, 751
objets, 734
objets et dépendances, 737
oneOf, 751
opérateurs, 751
propriétés non spécifiées, 734
sans contrainte, 745
sélecteurs de type (any, multiple), 745
sélecteurs de type (unconstrained, forbidden), 745
tableaux, 741
terminologie, 674
types atomiques, 743
types primitifs, 743
types simples, 743
Schéma XML, 49, 1370, 1386
ajouter des composants, 52
ajouter les élément avec, 57
aller à depuis le document XML, 1372
aller à la définition depuis le document XML, 1372
attribuer à XML BD, 1460
attribuer au document, 1371
commandes de menu associées à, 1370
composants globaux, 181
configurer l'affichage, 58
configurer Mode de Modèle de contenu, 1394
convertir à DTD, 1375
créer un nouveau fichier, 49
créer un schéma de base, 49
Définir les espaces de noms dans, 51
éditer les modèles de contenu, 188
élément <alternative>, 203
extensions MS SQL Server, 1401
extensions Oracle, 1399
générer depuis le document XML, 1373
générer depuis une DTD (éditions Enterprise et Professional), 445
générer du code depuis, 1384
générer fichier XML outline depuis, 1380
générer une documentation de, 1389
gestion et attribution dans les bases de données IBM DB2, 1456
modifier lors de l'édition de documents XML, 103
navigation dans le mode design, 77
paramètres dans le Mode Conception de schéma, 1386
paramètres de schéma MS SQL Server, 1402
Paramètres de schéma Oracle, 1400
paramètres d'espace de noms dans le Mode Conception de schéma, 1386
Smart Fix, 236
tutoriel, 49
validation, 236
voir aussi Schéma, 1370
SchemaAgent, 474
afficher les schémas dans, 1405
fonctionne avec, 469, 474

© 2016-2022 Altova GmbH

SchemaAgent, 474
ouvrir des schémas depuis XMLSpy, 474
se connecter au serveur depuis XMLSpy, 1403
se déconnecter du serveur, 1404
validation étendue, 1405
SchemaAgent dans XMLSpy, 465
SchemaAgent Server,
se connecter à, 466
SchemaDocumentationDlg,
AllDetails, 1893
Application, 1893
IncludeAll, 1895
IncludeAttributeGroups, 1895
IncludeComplexTypes, 1896
IncludeGlobalElements, 1896
IncludeGroups, 1897
IncludeIndex, 1897
IncludeLocalElements, 1897
IncludeRedefines, 1898
IncludeSimpleTypes, 1898
OptionsDialogAction, 1899
OutputFile, 1899
OutputFileDialogAction, 1899
OutputFormat, 1900
Parent, 1900
ShowAnnotations, 1900
ShowAttributes, 1901
ShowChildren, 1901
ShowConstraints, 1902
ShowDiagram, 1901
ShowEnumerations, 1901
ShowNamespace, 1902
ShowPatterns, 1902
ShowProgressBar, 1903
ShowProperties, 1903
ShowResult, 1903
ShowSingleFacets, 1903
ShowSourceCode, 1904
ShowType, 1904
ShowUsedBy, 1904
schemanativetype, 1259
Schémas,
consulter par le biais des catalogues, 462
en mémoire, 1385
gestion pour IBM DB2, 1457
Schémas et catalogues, 459
Schémas XML, 427
aplatir des schémas inclus, 448
aplatir les schémas inclus, 1407
convertir en DTD (éditions Enterprise et Professional), 447
éditer dans le Mode Grille (éditions Enterprise et Professional), 447
éditer dans le Mode Schéma (éditions Enterprise et Professional), 447
éditer dans le Mode Texte, 447
et ressources globales, 288
générer un document XML depuis, 447
gestion pour IBM DB2, 1457
plus DTDs, 288
répartir en sous-ensembles, 1405
y compris d'autres schémas dans, 448
Schémas, chercher dans,
voir Chercher dans les schémas, 477
Script, 1642
Scripting Editor,
overview, 1659, 1661
Scripts Authentic,
Emplacements approuvés, 1438
Paramètres de sécurité, 1438
Scripts dans XSLT/XQuery,
voir sous Fonctions d'extension, 2230
SE,
pour les produits Altova, 2249
Se connecter à SchemaAgent Server, 466
Sections CDATA,
insérer dans Authentic View, 630
Sélection de thème pour XMLSpy dans Eclipse, 1152
Sélectionneur de date,
utiliser dans Authentic View, 650
Serveur SQL,
se connecter par le biais d'ADO, 972
se connecter par le biais d'ADO.NET, 979
se connecter via JDBC, 990
Serveur Web, 1654
service web,
se connecter à, 789
SharePoint® Server, 1339
Show large markup, 618
Show mixed markup, 618
Show small markup, 618
Signature,
voir Signature XML, 1359
Signature XML, 1432
Signatures,
voir Signatures XML, 412
Signatures XML, 412, 654

© 2016-2022 Altova GmbH

Signatures XML, 412, 654
création, 1359
créer, 414
vérification, 1362
vérifier, 417
Signet, 1649
Signets,
insérer et supprimer, 1314
naviguer, 1314
Signets dans les scripts SQL, 1449
Signets dans Mode Texte, 146
Smart Fix pour les Schémas XML, 236
Smart Restrictions, 242
SOAP, 782, 795, 1524
arrêter le serveur proxy, 1531
créer une nouvelle requête, 1522
envoyer la requête au serveur, 1524
envoyer requête depuis WSDL, 791
lancer le serveur proxy, 1530
options de débogueur, 1532
paramètres requête, 1525
request, 1531
requests, 1531
requêtes, 1524, 1531
session de débogage, 1529
SOAP validation, 795
Sortie HTML,
générer dans Authentic View depuis le fichier PXF, 1438
Sortie PDF,
générer dans Authentic View depuis le fichier PXF, 1438
Sortie RTF,
générer dans Authentic View depuis le fichier PXF, 1438
sortie Word 2007+,
générer dans Authentic View depuis le fichier PXF, 1438
Sous-ensembles de schéma, 448, 1405, 1407
SPL, 1258
blocs de code, 1258
conditions, 1266
foreach, 1267
objets globaux, 1262
sous-routines, 1268
utiliser des fichiers, 1263
variables, 1261
SPS,
Assigner à nouveau fichier XML, 1277
SPS tables,
editing dynamic tables, 618
SpyProject,
CloseProject, 1905
ProjectFile, 1906
RootItems, 1906
SaveProject, 1906
SaveProjectAs, 1906
SpyProjectItem,
ChildItems, 1907
FileExtensions, 1907
ItemType, 1907
Name, 1908
Open, 1908
ParentItem, 1908
Path, 1908
ValidateWith, 1908
XMLForXSLTransformation, 1908
XSLForXMLTransformation, 1909
XSLTransformationFileExtension, 1909
XSLTransformationFolder, 1909
SpyProjectItems,
AddFile, 1909
AddFolder, 1910
AddURL, 1910
Count, 1910
Item, 1911
RemoveItem, 1911
SQL Server,
manage XML Schemas, 1462
SQLite,
désactiver les clés étrangères, 999
Stockages de certificat, 420
Structure du document XSLT, 125
Style, 1624
StyleVision, 1380
et XBRL, 1546
pour l'édition StyleVision Power Stylesheet, 1430
StyleVision Power Stylesheet,
assigner au fichier XML, 1430
éditer dans StyleVision, 1430
Suivez les changements, 1602
Sujet,
consulter sur TDM, 1649
Supprimer, 1583
barre d'outils, 1585
commande de la barre d'outils, 1583
commande depuis le menu contextuel, 1592
icône de la barre d'outils, 1583
ligne (dans Authentic View), 1437
point d'arrêt - Débogueur XSLT, 1424

© 2016-2022 Altova GmbH

Supprimer, 1583
point de trace - Débogueur XSLT, 1425
raccourci, 1588
Supprimer espace blanc, 1307
Supprimer la ligne dans Authentic View, 1438
Sybase,
se connecter par le biais de JDBC, 1049
Système aide, 1649

T

Table,
générer automatiquement, 1605
Table de données de graphique,
comment la construire, 350
Table des matières, 1649
Table Display commands (JSON Grid View), 1368
Tables,
dans Authentic View, 607
editing dynamic (SPS) tables, 618
Tables dans Authentic View,
icônes pour éditer des tables XML, 641
usage de, 635
utiliser des tables SPS (statiques et dynamiques), 635
utiliser des tables XML, 637
Tables dans le Mode Grille JSON,
copie comme TSV ou JSON, 713
Tables dans le Mode Grille XML,
copie comme TSV ou XML, 326
Tables du Mode Grid (JSON),
copie comme TSV ou JSON, 713
Tables du Mode Grille (XML),
copie comme TSV ou XML, 326
Tables dynamiques,
éditer, 618
Tables dynamiques (SPS) dans Authentic View,
usage de, 635
Tables SPS dans Authentic View,
usage de, 635
Tables Statiques (SPS) tables dans Authentic View,
usage de, 635
Tables XML dans Authentic View,
usage de, 637
Tabulations, 1602
Taille d'onglet,
et pretty-printing, 1506
paramètre, 1506
Taxonomies,
ajouter des éléments dans, 857
créer nouveau, 847
espace de noms cible de, 852
espaces de noms cibles dans, 1538
espaces de noms dans, 852, 1537
et Assistant de Nouvelle taxonomie, 847
et linkbases, 854
et rôles de lien, 860
étapes de création, 843
fichiers dans, 844, 854
importer, 848, 1539
relations dans, 860, 862, 865
Taxonomies dans XBRL, 843, 844
Taxonomies XBRL, 843
voir aussi Taxonomies, 844
Teradata,
se connecter par JDBC, 1051
se connecter par ODBC, 1053
terminate, 1755
Terminer,
session débogueur, 1423
Test de bonne formation des documents JSON, 677, 682
Texte,
formater dans Authentic View, 630
pretty-printing, 1307
rechercher et remplacer, 1311
Trouver dans document, 1307
Texte JSON de sources externes,
ajouter rapidement au document, 761
Texte structuré, 1608
Texte XML, 1608
Texte XML de sources externes,
ajouter rapidement au document, 340
TextImportExportSettings,
DestinationFolder, 1912
EnclosingCharacter, 1912
Encoding, 1912
EncodingByteOrder, 1912
FieldDelimiter, 1912
FileExtension, 1912
HeaderRow, 1913
ImportFile, 1913
Touche Entrée,
effets de l'utilisation, 656
Touche Retour,

© 2016-2022 Altova GmbH

Touche Retour,
voir touche Entrée, 656
Touche Retour chariot,
voir touche Entrée, 656
Traitement XULE, 1552
Traiter les Instructions dans le Mode Schema, 181
Transformation,
voir Transformation XSLT, 1412
Transformation XSL,
voir XSLT, 105
Transformation XSLT, 1410, 1412
attribuer un fichier XSLT, 105
dans XMLSpy, 106
en FO, 1412
en PDF, 1412
tutoriel, 105
Transformations JSON avec XSLT/XQuery, 763
Trier dans les Tables (Mode Grid XML), 1349
Trouver dans la commande de fichiers, 1313
TSV,
copie les tables du Mode Grid JSON comme, 713
copie les tables du Mode Grille XML comme, 326
Tutoriel,
pour WSDL, 783
Tutoriel WSDL, 783
type,
extension dans le document XML, 86
Type complexe, 64
dans les définitions de schéma, 64
étendre la définition, 64
type d'élément,
spécifier dans le document XML, 86
Type de base,
modifier, 240
Type simple,
dans les définitions de schéma, 64
Type simples,
assertions de, 232
définir des facettes de, 232
énumérations de, 232
modèles de, 232
valeurs échantillon de, 232
Types,
intégrés, 1271
Types complexes,
anonyme, 181
global, 181
nommé, 181
Types de fichier, 1605
Types dérivés,
modifier le type de base de, 240
Types simples,
anonyme, 181
global, 181
nommé, 181

U

UCS-2, 1607
UML,
convertir schémas en, 1378
URL, 1758, 1852, 1860
envoyer par e-mail, 1293
User interface,
configure using plug-in, 1687
UTF-16, 1607

V

Valeurs d'attribut,
saisir dans le Authentic View, 612
Valeurs d'échantillon de types simples, 232
Validate EDGAR, 1554
Validation, 134, 1352
attribuer DTD au document XML, 1370
attribuer Schéma XML au document XML, 1371
des schémas liés utilisant SchemaAgent, 474
élargir avec les Règles de schéma, 452
extension avec Règles de schémas, 452
fichiers WSDL, 1357
Validation automatique, 1605
Validation de schéma élargies, 452
Validation de schémas étendue, 452
Validation de Schémas XML, 236
Validation étendue, 474
dans SchemaAgent, 1405
validation XSD, 236
Valider,
fichier WSDL, 789
Valider des documents XML, 288
Valider sur la commande modification, 1358
Variables,

© 2016-2022 Altova GmbH

Variables,
dans SPL, 1261
Variables d'environnement, 463
ANT_OPS, 1255
Variables d'Environnement utilisées dans les catalogues, 460
Variables dans XPath/Débogueur XQuery, 584
VBScript,
scripting with XMLSpy, 1659
vérificateur orthographique, 1558
dictionnaire personnalisé, 1558
Vérification de la bonne formation, 1350
Pour les documents XML, 94
Vérification de la syntaxe des documents JSON, 677, 682
Version de Schéma JSON, 719
Visual Studio,
adding the XMLSpy ActiveX Controls to the toolbox, 2066
générer du code pour, 1256
Visual Studio .Net,
et débogueurs XMLSpy, 1144
et différences XMLSpy, 1141
et XMLSpy, 1139
Volet Navigateur,
dans la Fenêtre Requête de base de données, 1445
Volet Requête,
dans la fenêtre de Requête de base de données, 1449
VS .NET,
et Pack d'intégration XMLSpy, 1140

W

WADL,
utiliser pour les requêtes HTTP, 819
Windows,
activer/désactiver l'affichage, 1647
disposer, 1647
en cascade, 1647
en mosaïque, 1647
Flottant, ancré, tabulé, 115
gérer l'affichage de, 115
Masquer automatiquement, 115
prise en charge pour les produits Altova, 2249
Word 2007, 281, 950
Word-wrapping dans le Mode Texte, 144
WSD,
service web, 1522
WSDL, 782, 789
2.0 composants, 1512
composants 1.1, 1509
convertir de 1.1 en 2.0, 1521
créer des liaisons, 786
créer des messages, 784
créer des opérations, 784
créer des paramètres, 784
créer des ports, 788
créer des PortTypes, 784
créer des services, 788
créer documentation, 792
créer nouveau document, 783
débogueur SOAP, 796
envoyer requête SOAP, 791
espace de noms, 783
générer documentation, 1517
interface dans 2.0, 1513
liaison dans 1.1, 1511
liaison dans 2.0, 1514
messages dans 1.1, 1510
opérations dans 1.1, 1510
ouvrir schéma, 789
reparser document, 1521
se connecter à un service web, 789
service dans 1.1, 1512
service dans 2.0, 1515
Type de port dans 1.1, 1511
types, 1516
utiliser pour les requêtes HTTP, 819
valider, 789

X

XBRL, 825
espaces de noms, 1537
espaces de noms cibles, 1538
et MapForce, 1546
et StyleVision, 1546
générer une documentation, 1541
rôles d'arc, 1533
rôles de lien, 1535
validation, 949
XBRL Inline,
Gestion de l'espace blanc, 1639

© 2016-2022 Altova GmbH

xBRL-CSV (OIM), 1500
xBRL-JSON (OIM), 1500
xBRL-XML (OIM), 1500
Xerces,
générer du code pour, 1256
XInclude,
insérer dans le document XML, 290
insérer dans Mode Grille, 1302
insérer dans Mode Texte, 1302
XML,
Catalogue Oasis, 1352
vérificateur orthographique, 1558
XML BD,
attribuer Schémas XML à, pour IBM DB2, 1460
gérer les schémas XML pour IBM DB2, 1457
XML document,
générer depuis une DTD, 445
XML Schema,
content model diagram, 1389
XML tables dans Authentic View,
icônes pour éditer, 641
xml:base, 246
xml:id, 246
xml:lang, 246
xml:space, 246
XMLData,
AppendChild, 1953
EraseAllChildren, 1954
EraseCurrentChild, 1955
GetChild, 1956
GetChildKind, 1957
GetCurrentChild, 1957
GetFirstChild, 1958
GetNextChild, 1958
HasChildren, 1960
HasChildrenKind, 1960
InsertChild, 1960
IsSameNode, 1961
Kind, 1962
MayHaveChildren, 1962
Name, 1962
Parent, 1962
TextValue, 1963
XMLSpy, 1276
Aide, 137
fonctions, 137
integration, 2065
plug-in registration, 1686
XMLSpy API,
documentation, 1702
overview, 1703
XMLSpy command table :, 2092
XMLSpy dans Eclipse, 1146
XMLSpy Enterprise Edition,
manuel de l'utilisateur, 28
XMLSpy integration,
example of, 2076, 2077, 2078
XMLSPY plug-in, 1686
XMLSpy Plugin for Eclipse,
see Integration Package for Eclipse, 1147
XMLSpyCommand,
in XMLSpyControl, 2111
XMLSpyCommands,
in XMLSpyControl, 2113
XMLSpyControl, 2114
documentation of, 2065
example of integration at application level, 2076, 2077, 2078
examples of integration at document level, 2073
integration using C#, 2073
integration using HTML, 2078
object reference, 2111
XMLSpyControlDocument, 2122
XMLSpyControlPlaceHolder, 2128
XMLSpyDocumentEditor,
MarkUpView, 1995
XMLSpyLib, 1702, 1703
Application, 1740
AuthenticDataTransfer, 1761
AuthenticRange, 1766
AuthenticSelection (obsolete), 1976
AuthenticView, 1794
CodeGeneratorDlg, 1812
DatabaseConnection, 1818
Dialogs, 1824
DocEditEvent (obsolete), 1963
DocEditView (obsolete), 1978
Document, 1828
Documents, 1857
DTDSchemaGeneratorDlg, 1861
ElementList, 1865
ElementListItem, 1866
ExportSettings, 1867
FileSelectionDlg, 1870
GenerateSampleXMLDlg, 1883
GridView, 1888
ProjectItem, 1907

© 2016-2022 Altova GmbH

XMLSpyLib, 1702, 1703
SchemaDocumentationDlg, 1892
SpyProject, 1905
SpyProjectItems, 1909
TextImportExportSettings, 1911
XMLData, 1952
XPath,
évaluation, 124
générer un nœud dans un document XML, 290
XPath 1.0,
dans l'évaluateur XPath, 1350
XPath 2.0,
dans l'évaluateur XPath, 1350
XPath du nœud sélectionné dans le document XML,
copier dans le presse-papier, 1302
XPath vers le nœud sélectionné, 616
XPath/XQuery expressions,
evaluating and debugging, 574
in XPath/XQuery Window, 598
XPath/XQuery Window, 574
XPaths,
paramètres pour les points de trace, 553
XPointer,
générer un nœud dans un document XML, 290
XPointer du nœud sélectionné dans le document XML,
copier dans le presse-papier, 1302
XQuery, 517
assistants à la saisie, 511
coloration de la syntaxique, 511
éditer dans le Mode Texte, 507
exécution, 517
Fonctions d'édition intelligentes, 513
Fonctions d'extension, 2230
fonctions pour IBM DB2, 528
ouvrir le fichier, 510
passer les variables au document XQuery, 1413
pour la requête de bases de données XML, 528
prise en charge BD, 528
validation de document, 517
XQuery expressions for JSON, 766
XQuery Update,
prévisualisation, 521
PUL, 521
XQuery Update dans XMLSpy, 521
XQuery Update Facility, 521
XSD 1.0 et 1.1,
modes d'édition, 172
xsi:type,
utilisation, 86
XSL,
voir XSLT, 1422
XSL Outline, 497
xsl:- param, 498
xsl:call-template, 498
XSL:FO,
et les transformations XSLT, 494
xsl:param, 498
XSLT, 1505
Assistants à la saisie pour, 492
documents, 492
fonctions dans XMLSpy, 491
Fonctions d'extension, 2230
modifer dans XMLSpy, 107
Saisie semi-automatique dans le Mode Texte, 492
valider, 492

Z

zip64mode,
activer dans le fichier build.xml, 1255
Zoom dans Mode Texte, 146

© 2016-2022 Altova GmbH