

Altova StyleVision 2027 Basic Edition



Manuel de l'utilisateur et de référence

Altova StyleVision 2027 Basic 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: 2026

© 2020-2026 Altova GmbH

Table des matières

1	Introduction	13
1.1	Fonctions de produit.....	14
1.2	Authentic View dans les Produits Altova.....	18
1.3	Qu'est-ce qu'une SPS?.....	19
1.4	Configurer StyleVision.....	20
1.5	Terminologie.....	21
1.6	À propos de cette documentation.....	24
2	Interface utilisateur	26
2.1	Fenêtre principale.....	27
2.1.1	Aperçu de design.....	28
2.1.2	Aperçus de sortie.....	30
2.2	Barres latérales.....	32
2.2.1	Aperçu du design.....	35
2.2.2	Arborescence de schéma.....	37
2.2.3	Arborescence de Design.....	40
2.2.4	Référentiel de style.....	44
2.2.5	Styles	46
2.2.6	Propriétés.....	48
2.2.7	Messages.....	52
2.2.8	Chercher et Remplacer.....	52
3	Tutoriel Démarrage rapide	54
3.1	Créer un nouveau fichier SPS.....	55
3.2	Insérer du contenu dynamique (depuis la Source XML).....	60
3.3	Insérer le Contenu statique.....	67
3.4	Formater le contenu.....	72
3.5	Utiliser les calculs automatiques.....	78

5.6.7	Tables CALS/HTML.....	141
5.7	Listes.....	147
5.7.1	Listes statiques.....	147
5.7.2	Listes dynamiques.....	149
5.8	Graphiques.....	152
5.8.1	Images : URI et données Inline.....	152
5.8.2	Types d'images et Sortie.....	154
5.8.3	Exemple : Un modèle pour les images.....	157
5.9	Contrôles de formulaires.....	158
5.9.1	Champs d'entrée, Champ de saisie multiligne.....	159
5.9.2	Cases à cocher.....	160
5.9.3	Listes de choix.....	161
5.9.4	Boutons radio, boutons.....	164
5.10	Liens.....	165
5.11	Codes-barres.....	166
5.12	Modules de mise en page.....	171
5.12.1	Conteneurs de mise en page.....	171
5.12.2	Cases de mise en page.....	175
5.12.3	Lignes.....	178
5.13	La fonction Change-To.....	181

6 Structure SPS 185

6.1	Sources de schéma.....	187
6.1.1	DTD et schémas XML.....	188
6.1.2	Schémas définis par l'utilisateur.....	193
6.1.3	Gestionnaire de schéma.....	196
6.2	Fusionner des données XML depuis des sources multiples.....	217
6.3	SPS modulaires.....	220
6.3.1	Objets modulaires disponibles.....	221
6.3.2	Créer un fichier SPS modulaire.....	224
6.3.3	Exemple : Un Carnet d'adresse.....	228
6.4	Fragments de modèles et de design.....	234
6.4.1	Modèle principal.....	234
6.4.2	Modèles globaux.....	234

6.4.3	Modèles définis par l'utilisateur.....	239
6.4.4	Modèles variables.....	241
6.4.5	Opérations Nœud-Modèle.....	242
6.4.6	Fragments de design.....	245
6.5	Modèles XSLT.....	249
6.6	Sortie multiple de document.....	251
6.6.1	Insérer un nouveau modèle de document.....	252
6.6.2	Nouveaux modèles de document et Structure de design.....	253
6.6.3	URL de nouveaux modèles de document.....	253
6.6.4	Fichiers d'aperçu et fichiers de documents de sortie.....	256
6.6.5	Propriétés du document et Styles.....	258
7	Fonctions avancées	260
7.1	Calcul automatique.....	261
7.1.1	Éditer et déplacer des calculs automatiques.....	261
7.1.2	Exemple : Un Invoice (une facture).....	263
7.2	Conditions.....	266
7.2.1	Définir les conditions.....	266
7.2.2	Éditer les conditions.....	269
7.2.3	Conditions et calculs automatiques.....	270
7.3	Regroupement.....	271
7.3.1	Exemple: Group-By (Persons.sps).....	274
7.3.2	Exemple: Group-By (Scores.sps).....	275
7.4	Trier.....	279
7.4.1	Le mécanisme de tri.....	279
7.4.2	Exemple : Trier sur de multiples Sort-Keys.....	281
7.5	Paramètres et variables.....	285
7.5.1	Paramètres déclarés par l'utilisateur.....	285
7.5.2	Paramètres pour les fragments de design.....	287
7.5.3	Paramètres pour les sources SPS.....	289
7.5.4	Variables.....	290
7.6	Table de contenus, Référencement, Signets.....	293
7.6.1	Items de bookmarking pour l'inclusion TOC.....	296
7.6.2	Créer le modèle TOC.....	303

7.6.3	Exemple : TOC simple.....	308
7.6.4	Exemple : TOC hiérarchiques et séquentielles.....	312
7.6.5	Numérotation automatique dans le corps du document.....	316
7.6.6	Référencement croisé.....	319
7.6.7	Signets et hyperliens.....	321
8	Présentation et Sortie	328
8.1	Formats prédéfinis.....	329
8.2	Échappement de sortie.....	331
8.3	Formatage de valeur (formater les types de données numériques).....	334
8.3.1	Mécanisme de formatage de valeur.....	334
8.3.2	Syntaxe du Formatage valeur.....	337
8.4	Travailler avec les Styles CSS.....	343
8.4.1	Feuilles de style externes.....	344
8.4.2	Styles Globaux.....	348
8.4.3	Styles locaux.....	350
8.4.4	Définir des valeurs de Style.....	352
8.4.5	Propriétés de style via XPath.....	354
8.4.6	Styles Composite.....	357
8.5	Propriétés Document HTML.....	360
9	Fonction additionnelle	362
9.1	URI d'entité non parsés.....	363
9.2	Nouveauté du fichier XSLT, XSL-FO ou FO.....	365
9.3	Fonctions XPath définies par l'utilisateur.....	369
9.3.1	Définir une fonction XPath.....	372
9.3.2	Réutiliser les fonctions pour localiser les nœuds.....	374
9.3.3	Paramètres dans les fonctions XPath.....	375
9.4	Travailler avec des dates.....	385
9.4.1	Formater les dates.....	385
9.5	Utiliser des Scripts.....	389
9.5.1	Définir fonctions JavaScript.....	390
9.5.2	Attribuer Fonctions comme gestionnaires d'événement.....	391

9.5.3	Fichiers JavaScript externes.....	392
9.6	Importation HTML.....	394
9.6.1	Créer un nouvel SPS via l'Importation HTML.....	394
9.6.2	Créer le schéma et le design SPS.....	396
9.6.3	Créer des Tables et Listes comme Éléments/Attributs.....	398
9.6.4	Générer la sortie.....	400
9.7	Interface ASPX pour les applications Web.....	401
9.7.1	Exemple : Localhost sur Windows 7.....	402
9.8	Fichier PXF : Conteneur pour fichiers SPS et associés.....	404
9.8.1	Créer un fichier PXF.....	404
9.8.2	Éditer un fichier PXF.....	407
9.8.3	Déployer un fichier PXF.....	408
10	Traitement automatisé	410
10.1	Interface de ligne de commande.....	411
10.1.1	StyleVision.....	411
10.1.2	StyleVision Server.....	412
10.2	Utiliser RaptorXML.....	414
10.2.1	Sortie PDF.....	414
10.3	Automatisation avec FlowForce Server.....	417
10.4	Comment automatiser le traitement.....	419
11	Commandes de menu et Référence	420
11.1	Symboles du Mode Design.....	421
11.2	Éditer le dialogue de l'expression XPath.....	426
11.2.1	Évaluateur.....	427
11.2.2	Débogueur.....	431
11.2.3	Générateur d'expression.....	438
11.3	Barres d'outils.....	443
11.3.1	Format.....	445
11.3.2	Table	446
11.3.3	Insérer Éléments de design.....	447
11.3.4	Filtre de design.....	449

11.3.5	Standard.....	450
11.4	Menu Fichier.....	452
11.4.1	Nouveau.....	452
11.4.2	Ouvrir, Recharger, Fermer, Fermer tout.....	458
11.4.3	Enregistrer design, Tout enregistrer.....	463
11.4.4	Enregistrer sous.....	468
11.4.5	Exporter en tant que fichier MobileTogether Design.....	469
11.4.6	Enregistrer les fichiers générés.....	469
11.4.7	Déployer vers FlowForce.....	470
11.4.8	Design web.....	472
11.4.9	Propriétés.....	472
11.4.10	Aperçu avant impression, Imprimer.....	474
11.4.11	Fichiers récemment utilisés, Quitter.....	475
11.5	Menu Éditer.....	476
11.5.1	Annuler, Rétablir, Sélectionner tout.....	476
11.5.2	Rechercher, Trouver suivant, Remplacer.....	476
11.5.3	Paramètres de feuille de style.....	482
11.5.4	Réduire/agrandir balisage.....	482
11.6	Menu Affichage.....	484
11.6.1	Barre d'outils et Barre de statut.....	484
11.6.2	Barres latérales de design.....	485
11.6.3	Filtre design, Zoom.....	486
11.6.4	Aperçu de sortie.....	486
11.7	Menu Insérer.....	488
11.7.1	Contenus.....	488
11.7.2	Reste des contenus.....	489
11.7.3	Commandes de formulaire.....	489
11.7.4	Calcul automatique.....	490
11.7.5	Paragraphe, Paragraphe spécial.....	491
11.7.6	Image.....	492
11.7.7	Ligne horizontale.....	495
11.7.8	Table.....	495
11.7.9	Puces et numérotation.....	496
11.7.10	Signet.....	498
11.7.11	Hyperlien.....	500

11.7.12	Condition, Condition sur base de sortie.....	501
11.7.13	Désactivé.....	503
11.7.14	Modèle.....	503
11.7.15	Modèle défini par l'utilisateur.....	504
11.7.16	Modèle de variable.....	505
11.7.17	Fragment de design.....	506
11.7.18	Conteneur de layout, Fenêtre de layout, Ligne.....	506
11.7.19	Table des matières.....	507
11.7.20	Nouveau document.....	507
11.7.21	Item défini par l'utilisateur.....	508
11.8	Menu Contenir dans.....	509
11.8.1	Modèle.....	509
11.8.2	Modèle défini par l'utilisateur.....	510
11.8.3	Modèle de variable.....	510
11.8.4	Paragraphe, paragraphe spécial.....	511
11.8.5	Puces et numérotation.....	511
11.8.6	Signet et lien hypertexte.....	512
11.8.7	Condition, condition sur base de sortie.....	512
11.8.8	Désactivé.....	514
11.8.9	Signet TdM et niveau TM.....	514
11.8.10	Nouveau document.....	515
11.8.11	Élément défini par l'utilisateur.....	515
11.9	Menu Table.....	516
11.9.1	Insérer table, Supprimer table.....	516
11.9.2	Ajouter en-têtes, pieds de page de table.....	517
11.9.3	Apposer/Insérer ligne/colonne.....	517
11.9.4	Supprimer ligne, colonne.....	518
11.9.5	Fusionner avec cellule gauche, droite, inférieure, supérieure.....	518
11.9.6	Fractionner la cellule horizontalement, verticalement.....	519
11.9.7	Afficher bordures de cellule, Afficher balisage de table.....	519
11.9.8	Propriétés de la table.....	519
11.9.9	Éditer tables CALS/HTML.....	520
11.9.10	Alignement vertical du contenu des cellules.....	521
11.10	Menu Propriétés.....	522
11.10.1	Éditer les puces et numérotation.....	522

11.10.2	Chaîne de formatage de valeur prédéfinie.....	523
11.11	Menu Outils.....	525
11.11.1	Orthographe.....	525
11.11.2	Options du vérificateur d'orthographe.....	526
11.11.3	Gestionnaire de schéma XML.....	529
11.11.4	Personnaliser.....	530
11.11.5	Restaurer barres d'outils et fenêtres.....	535
11.11.6	Options.....	535
11.12	Menu Fenêtre.....	542
11.13	Menu Aide.....	543
11.13.1	Aide	543
11.13.2	Activation, Bon de commande, Inscription, Mises à jour.....	543
11.13.3	Autres commandes.....	547

12 Annexes 548

12.1	Informations concernant le moteur XSLT et XQuery.....	549
12.1.1	XSLT 1.0.....	549
12.1.2	XSLT 2.0.....	549
12.1.3	XSLT 3.0.....	552
12.1.4	XQuery 1.0.....	553
12.1.5	XQuery 3.1.....	557
12.2	Fonctions XSLT et XPath/XQuery.....	558
12.2.1	Fonctions d'extension Altova.....	559
12.2.2	Fonctions d'extension diverses.....	635
12.3	Types de données dans les schémas XML générés à partir d'une BD.....	654
12.3.1	ADO	654
12.3.2	MS Access.....	655
12.3.3	MS SQL Server.....	656
12.3.4	MySQL.....	657
12.3.5	ODBC.....	657
12.3.6	Oracle.....	658
12.3.7	Sybase.....	659
12.4	Données techniques.....	661
12.4.1	SE et exigences de mémoire.....	661

12.4.2	Moteurs Altova.....	661
12.4.3	Prise en charge Unicode.....	662
12.4.4	Utilisation Internet.....	662
12.5	Informations de licence.....	664
12.5.1	Distribution électronique de logiciel.....	664
12.5.2	Activation de logiciel et le license metering.....	665
12.5.3	Altova Contrat de licence de l'utilisateur final.....	666
12.5.4	Packaging License Files with StyleVision Installer.....	666

Index	667
--------------	------------

1 Introduction

[Altova StyleVision 2027 Basic Edition](#) est une application pour concevoir et éditer graphiquement StyleVision Power Stylesheets. StyleVision® est exécuté sur Windows 10, Windows 11 et Windows Server 2016 ou plus récent. Quelques fonctionnalités de StyleVision et [Altova MissionKit](#) peuvent être intégrées avec les applications de Microsoft Office suite (MS Access, MS Excel, MS Word), version 2007 ou plus récentes.



StyleVision Power Stylesheet (SPS) peut être utilisé pour les objectifs suivants :

- Pour contrôler un aperçu WYSIWYG graphique de **documents XML dans Authentic View**, qui est un éditeur de document XML disponible de produits Altova : Altova XMLSpy, Altova StyleVision, Altova Authentic Desktop, and Altova Authentic Browser. Ceci vous permet de créer facilement des [formulaires électroniques](#) basés sur des documents XML.
- Pour générer les **Feuilles de style XSLT** basées sur le design SPS. (XSLT 1.0, XSLT 2.0, et XSLT 3.0 sont pris en charge.) Les feuilles de style XSLT peuvent être utilisées en dehors de StyleVision pour transformer les documents XML dans les sorties telles que HTML.
- Pour générer, directement de l'intérieur StyleVision, **HTML sortie** du document XML.

Dernière mise à jour : 21.10.2026

1.1 Fonctions de produit

Les principales fonctionnalités du produit StyleVision sont décrites de manière générale sur la [page d'accueil](#)¹³. Les fonctionnalités énumérées ci-dessous fournissent des détails sur divers aspects.

Fonctions de produit générales

Ci-dessous, vous trouverez une liste des fonctions principales de haut niveau de StyleVision.

- Les éditions Enterprise et Professional sont toutes deux disponibles comme applications séparées 64-bit et 32-bit.

Sources

Les designs SPS peuvent être basés sur les schémas XML et les DTD. Un design utilise d'autres fichiers de source, tels que les fichiers XML et CSS. Les fonctions supplémentaires suivantes concernant les sources sont prises en charge :

-
- Les documents HTML peuvent être [convertis en XML](#)³⁹⁴.

Interface

Ci-dessous, vous trouverez des fonctions GUI :

- Les [multiples designs SPS](#)²⁶ peuvent être ouverts simultanément, avec un étant actif à tout moment donné. Chaque design SPS est affiché dans une balise séparée.
- Les [filtres de modèle](#)⁴⁴⁹ vous permettent de personnaliser l'affichage du document de design. Avec cette fonction, vous pouvez désactiver l'affichage de modèles qui ne sont pas édités actuellement, augmentant ainsi l'efficacité.
- [Masquer les marques dans l'Aperçu Design](#)²⁸ : Les balises des marques dans l'Aperçu Design peuvent être masquées ou réduites, libérant ainsi de l'espace dans l'Aperçu Design.
- En concevant SPS, les [aperçus de sortie](#)³⁰ et les feuilles de style peuvent être affichés en cliquant sur les tables respectives. Ceci vous permet de prévisualiser rapidement la sortie et le code XSLT.

Sortie

Des formats de sortie variés sont pris en charge dépendant de la version qui a été installée. Les fonctions de sortie suivantes sont prises en charge :

- Les [versions XSLT 1.0, 2.0, et 3.0](#)⁹⁹ sont prises en charge.
- Dans les Éditions Enterprise et Professional, de [multiples formats de sortie](#)¹⁰² (HTML) sont générés depuis un seul design SPS.
- Les deux [fichiers XSLT et fichiers de sortie](#)¹⁰² peuvent être [générés et enregistrés](#)¹⁰², soit directement de l'intérieur de GUI ou via [StyleVision Server](#).
- Altova a élaboré un [format de fichier PXF](#)⁴⁰⁴ spécial qui permet l'enregistrement d'un fichier SPS ensemble avec les fichiers source et de données. Ceci permet à des projets SPS entiers d'être transportés plutôt que juste le fichier SPS.
- [Interface ASPX pour les applications Web](#)⁴⁰¹ : Avec cette fonction, les pages web HTML peuvent être mises à jour rapidement. StyleVision génère, depuis une SPS, tous les fichiers requis pour une

application ASPX. Quand la page web (un fichier .aspx) est actualisée, les données source (y compris les mises à jour) sont transformées de manière dynamique via XSLT dans la page web.

Fonctions design SPS

Ci-dessous, vous trouverez une liste des fonctions principales StyleVision spécifiques à la conception de SPS.

- SPS peut contenir du [texte statique](#)¹¹⁰, que vous saisissez dans SPS, et le [texte dynamique](#)¹¹⁰, qui est sélectionné du [document source](#)¹⁸⁷.
- Un [contenu dynamique](#)¹¹⁰ est inséré dans le design en glissant-déposant des nœuds depuis la [source de schéma](#)¹⁸⁷. Les éléments Design (paragraphe, listes, images, etc.) peuvent être insérés d'abord et un nœud XML depuis l'arborescence de schéma attribué à l'élément Design par la suite.
- Un [contenu dynamique](#)¹¹⁰ peut être inséré comme texte, ou sous forme d'un [appareil de saisie de données](#)¹⁵⁸ (tel qu'un [champ de saisie](#)¹⁵⁹ ou une [liste déroulante modifiable](#)¹⁶¹).
- La [structure du design](#)¹⁸⁵ est spécifiée et contrôlée dans un seul [modèle principal](#)²³. Cette structure peut être modifiée par des modèles optionnels pour des éléments individuels—connus comme [modèles globaux](#)²³⁴ car ils peuvent être appliqués globalement pour cet élément.
- Les [modèles globaux](#)²³⁴ peuvent être créés pour des types de données individuels, permettant le traitement à gérer également sur la bases de types.
- [Sortie multiple de document](#)²⁵¹ : La sortie générée par la SPS peut être conçue de manière à être partagée en de multiples documents. Dans le design, des modèles Nouveau Document sont créés et le contenu placé à l'intérieur. Chaque nouveau modèle de Nouveau Document génère un document séparé dans la sortie.
- [Modèles définis par l'utilisateur](#)¹¹⁹ : Un modèle peut être généré pour une séquence d'items par une expression XPath que vous précisez. Ces items peuvent être des valeurs atomiques ou des nœuds. Une expression XPath permet une plus grande précision de la sélection de nœuds, permettant aux conditions et filtres d'être utilisés pour la sélection.
- [Éléments définis par l'utilisateur](#)¹²² : Cette fonction a pour objectif de permettre la présentation d'éléments de langage (tels que HTML, XSLT et XSL-FO) afin d'être insérés librement à tout emplacement dans le design.
- [Blocs de texte définis par l'utilisateur](#)¹²² : Les blocs de texte XML peuvent être insérés librement dans tout emplacement dans le design, et ces blocs seront créés à cet emplacement dans la feuille de style XSLT.
- Les [fragments de design](#)²⁴⁵ permettent la modularisation et réutilisation des modèles à l'intérieur d'un SPS, et également dans de multiples SPS (voir [SPS modulaires](#)²²⁰), d'une manière similaire à la façon dont les fonctions sont utilisées.
- Les [modules SPS](#)²²⁰ peuvent être ajoutés à d'autres modules SPS, définissant donc des objets en un module SPS disponible pour d'autres modules. Ceci permet de réutiliser les objets de module dans de multiples SPS et rendent la maintenance plus facile.
- [Modèles XSLT](#)²⁴⁹ : Les fichiers XSLT peuvent être importés dans les feuilles de style générées. Si un nœud de document de l'instance XML correspond à un modèle dans le fichier XSLT importé et aucun autre modèle aura précedence sur le modèle importé, le modèle importé sera utilisé. De plus, les modèles nommés dans le fichier XSLT importé peuvent être appelés de l'intérieur du design.
- [Nouveautés de XSLT](#)³⁶⁵ : Une SPS peut être créée de XSLT-pour-HTML ou de XSLT-pour-FO. La structure et le style du modèle dans le XSLT seront créés dans le SPS. Vous pouvez ensuite modifier les composants SPS et ajouter du contenu et des formats au SPS.
- [Fonctions XPath définies par l'utilisateur](#)³⁶⁹ : L'utilisateur peut définir les fonctions XPath qui peuvent être utilisées partout dans le document où les fonctions XPath peuvent être utilisées.
- [Conteneurs Mise en page](#)¹⁷¹ : Un conteneur Mise en page est un bloc dans lequel les Éléments Design peuvent être affichés et positionnés de façon absolue au sein du bloc.

- [Blueprints](#) ¹⁷⁴ : Au sein d'un Conteneur Mise en page, une image d'un formulaire peut être utilisée comme blueprint en sous-couche pour le design. Avec l'aide d'un blueprint, un design existant peut être reproduit de manière précise.
- Une fonction commune des documents XML est la structure de données répétitive. Par exemple, un département de bureau a généralement plusieurs employés. Les données pour chaque employé seraient stockées dans une structure de données qui est répétée pour chaque employé. Dans SPS, le [traitement pour chacune de ces structures de données](#) ¹¹⁰ est défini une fois et appliqué dans chaque nœud pertinent à son tour (le nœud employé dans notre exemple).
- De multiples [tables de contenus](#) ²⁹³ peuvent être insérées dans XSLT 2.0 et 3.0 SPSs.
- Les structures de données répétitives peuvent aussi être insérées comme [tables dynamiques](#) ¹²⁵. Ceci permet un bouclage dans un format de table structuré, avec chaque boucle produisant une ligne par la structure de table (ou, si requis, une colonne) de la table.
- Un élément répétitif peut être [trié sur une ou plusieurs clé de tri](#) ²⁷⁹ que vous sélectionnez et l'élément trié défini est envoyé à la sortie (HTML).
- [Variables](#) ²⁹⁰ : Une variable peut désormais être déclarée sur un modèle et prendre une valeur qui est spécifiée avec une expression XPath. Précédemment, la valeur de la variable était limitée à la sélection du nœud sur lequel elle a été créée. Les variables dans la version 2010 permettent à toute expression XPath d'être spécifiées en tant que valeur de la variable.
- Les nœuds peuvent être [groupés](#) ²⁷¹ sur la base du contenu commun des données (par exemple, la valeur commune d'une valeur d'attribut) et leurs positions.
- La fonction [modèles conditionnels](#) ²⁶⁶ permet le traitement d'un ensemble de modèles à traiter selon que les conditions dans le document XML ou l'environnement de système document sont remplies. Ceci permet le traitement qui est conditionnel sur l'information contenue dans le document source ou qui ne peut pas être connue par le créateur du document SPS au moment de la création (par exemple, la date du traitement). Les conditions disponibles sont celles qui peuvent être testées en utilisant les expressions XPath.
- Les [calculs automatiques](#) ²⁶¹ vous permettent de manipuler les données depuis le/s document/s source et pour afficher le résultat. Ceci est utile quand vous voulez réaliser des calculs sur des numéros (par exemple, la somme des prix dans une facture), manipuler les strings (par exemple, changer les traits-d'union en barres obliques), générer du contenu, etc. Les manipulations disponibles sont celles qui peuvent être traitées utilisant les expressions XPath. Les fonctions Native Java et .NET peuvent être utilisées dans les expressions XPath des calculs automatiques.
- Les [images](#) ¹⁵² peuvent être insérées dans le design. L'URI pour l'image peut être statique (saisie dans SPS), ou dynamique (pris d'un nœud dans le document source), ou une combinaison des deux parties, statique et dynamique.
- [Images de données inline](#) ¹²⁵ : Les images peuvent être générées de texte encodé Base-16 et Base-64 dans le document XML. En conséquence, les images peuvent être stockées directement dans le document XML source comme texte. Une SPS peut décoder maintenant un tel texte et afficher l'image.
- Deux types de [listes](#) ¹⁴⁷ peuvent être créées : statique et dynamique. Dans une [liste statique](#) ¹⁴⁷, chaque item de liste est définie dans SPS. Dans une [liste dynamique](#) ¹⁴⁹, un nœud est créé comme item de liste ; les valeurs de toutes les instances de ce nœud sont créées comme items de la liste.
- Les [liens statiques et dynamiques](#) ³²¹ peuvent être insérés dans le design. L'URI cible peut être statique (saisie dans SPS), ou dynamique (pris d'un nœud dans le document source), ou une combinaison des deux parties, statique et dynamique.
- Des [signets](#) ³²¹ statiques peuvent être insérés. Ceux-ci servent d'ancres qui peuvent être mis en lien avec un hyperlien.
- Les [paramètres](#) ²⁸⁵ peuvent être déclarés globalement pour tout le SPS. Un paramètre est déclaré avec un nom et une valeur de string, et peut être utilisé dans les expressions XPath dans SPS. La valeur de paramètre que vous déclarez est la valeur par défaut. Elle peut être écrasée par une valeur transmise par le biais de [StyleVision Server](#).

- Avec la fonction [Formatage d'entrée](#)³³⁴, les contenus des nœuds de type de données de schéma XML numériques peuvent être formatés tel que requis pour l'affichage. Le formatage d'entrée peut aussi être utilisé pour formater le résultat d'un [calcul automatique](#)²⁶¹.
- Les [fonctions de JavaScript](#)³⁸⁹ peuvent être utilisées dans SPS afin de fournir une fonctionnalité définie par l'utilisateur pour Authentic View et la sortie HTML.
- Un nombre de [formats HTML prédéfinis](#)³²⁹ sont disponibles via la GUI et peuvent être appliqués dans des composants SPS individuels.
- Une grande quantité de formatage de texte CSS et des propriétés de mise en page peuvent être appliquées aux composants SPS individuels via la [barre latérale des Styles](#)³⁵⁰.
- De plus, ces styles CSS peuvent être définis pour les sélecteurs HTML au [niveau global](#)³⁴⁸ des feuilles de style SPS et dans des feuilles de style CSS externes. Ces règles de style s'appliqueront à la sortie HTML, permettant un formatage une flexibilité de mise en page substantiels.
- Les [styles peuvent également être assignés en utilisant les expressions XPath](#)³⁵⁴. Ceci permet aux valeurs des propriétés de style d'être sélectionnées et de définir les valeurs de propriété de manière conditionnelle.

1.2 Authentic View dans les Produits Altova

Authentic View est un éditeur graphique de documents XML disponible dans les produits Altova suivants :

- Altova XMLSpy
- Altova Authentic Desktop
- Altova StyleVision

Un StyleVision Power Stylesheet (SPS) créé dans StyleVision peut être utilisé comme modèle pour afficher et modifier des documents XML dans Authentic View. Ainsi, si vous appliquez un SPS à un document XML, vous pouvez ouvrir ce document dans Authentic View de n'importe quel produit mentionné ci-dessus et le modifier graphiquement, comme dans un traitement de texte tel que Microsoft Word. (Bien entendu, le SPS et le document XML doivent être basés sur le même schéma.)

1.3 Qu'est-ce qu'une SPS?

Une StyleVision Power Stylesheet (ou SPS) est une feuille de style XSLT élargie qui est utilisée pour créer de manière graphique un design pour un document de sortie HTML.

Une SPS est enregistrée avec une extension de fichier `.sps`.

Design du SPS

Une SPS est créée de manière graphique dans StyleVision. Elle est basée sur un schéma (DTD ou schéma XML). Le design de SPS est flexible. Il peut contenir du contenu dynamique et statique. Le [contenu dynamique](#)²² sont les données dans un document documents XML. Le [contenu statique](#)²³ est le contenu saisi directement dans SPS. Le contenu dynamique peut être inclus dans le design, soit en tant que texte droit ou au sein des composants tels que les champs d'entrée, les zones de liste déroulantes et les tables. De plus, le contenu dynamique peut être manipulé (utilisant les calculs automatiques) et affiché si certaines conditions dans le document source sont remplies. Différentes parties du contenu peuvent être placées à plusieurs emplacements variés dans SPS. Aussi, SPS peut contenir différents autres composants, tels que des images, hyperliens et des fonctions de JavaScript. Chaque composant de SPS peut ensuite être formaté pour la présentation, tel que requis.

Les feuilles de style SPS et XSLT

Après avoir complété le design de SPS, vous pouvez générer des feuilles de style XSLT basées sur le design que vous avez créé. StyleVision prend en charge XSLT 1.0, XSLT 2.0 et XSLT 3.0, et d'une seule SPS, vous pouvez générer des feuilles de style XSLT pour des sorties HTML, RTF, XSL-FO, Texte, ainsi que Word 2007-et-plus récent (*XSL-FO, Texte, et Word 2007-et-plus récent dans l'édition Enterprise uniquement ; RTF et Texte dans les éditions Enterprise et Professional ; dans l'édition Basic, uniquement la sortie HTML est prise en charge*). Les feuilles de style XSLT générées peuvent être utilisées dans des transformations externes pour transformer des documents XML basés sur le même schéma que SPS duquel la feuille de style XSLT a été générée. Pour plus d'informations sur les procédures utilisées avec les feuilles de style XSLT, voir la section [Fichiers générés](#)¹⁰².

SPS et la sortie

Vous pouvez aussi utiliser StyleVision pour générer directement la sortie (*HTML, RTF, Texte, XSL-FO, et PDF dans l'éditions Enterprise ; HTML et RTF dans l'édition Professional ; et HTML dans l'édition Basic*). Les onglets pour [Affichages de sortie](#)³⁰ affichent la sortie pour le document SPS actif directement dans la GUI de StyleVision. La sortie requise peut aussi être générée dans le fichier depuis l'intérieur de la GUI par le biais de la commande [Fichier | Enregistrer Fichiers générés](#)⁴⁶⁹ ou par le biais de [StyleVision Server](#).

1.4 Configurer StyleVision

Altova StyleVision est exécuté sur Windows 10, Windows 11. Après avoir téléchargé StyleVision du [site web d'Altova](#), double-cliquez sur le fichier exécutable (.exe) pour exécuter le programme de configuration. Le programme de configuration sera installé StyleVision à l'emplacement désiré. Les moteurs XSLT d'Altova (1.0 et 2.0) sont construits dans StyleVision et sont utilisés pour toutes les transformations internes. Pour cette raison, vous n'avez pas besoin d'installer un moteur XSLT en plus de l'installation de StyleVision.

Vous devrez, toutefois, avoir installé les composants suivants :

- Internet Explorer 5.5 ou plus récent, pour l'aperçu HTML et l'affichage Design. Internet Explorer 6.0 et plus récent a une meilleure prise en charge XML et est recommandé. Pour le [copier-coller depuis les documents Word](#)¹¹⁴ (et le contenu qui peut être collé dans les documents Word, tel que les tables Excel et le contenu de page HTML) Word 2007+ est requis

1.5 Terminologie

Cette section répertorie les termes utilisés dans l'interface graphique StyleVision et dans la présente documentation. Les termes sont organisés en groupes recensés ci-dessous et au sein de chaque groupe, ils sont triés par ordre alphabétique.

Termes associés aux produits d'Altova

Une liste de termes associés aux produits d'Altova.

- Authentic View** Un aperçu d'éditeur de document XML disponible dans les produits d'Altova suivants : Altova XMLSpy; Altova StyleVision; Altova Authentic Desktop; Altova Authentic Browser. Pour plus de détails sur Authentic View et les produits d'Altova, consultez le [site web d'Altova](#).
- SPS** Abréviation de StyleVision Power Stylesheet, ce terme est utilisé tout au long de cette documentation pour désigner le document de conception créé dans StyleVision et enregistré sous forme de fichier avec l'extension .sps. Pour une description détaillée, voir [Qu'est qu'un SPS?](#)¹⁹.
- Ressources globales** Un alias pour un ensemble de fichiers, un ensemble de dossiers, ou un ensemble de bases de données. Chaque alias a un ensemble de configurations et chaque configuration est mappée à une ressource. Dans StyleVision, lorsqu'une ressource globale est utilisée, la ressource peut être modifiée en changeant la configuration active dans StyleVision.

Termes XML de nature générale

Définitions de certains termes XML tels qu'utilisés dans cette documentation.

- schéma** Un schéma (avec « s » en minuscule) fait référence à tout type de schéma. Les schémas pris en charge par StyleVision sont le [Schéma XML](#)²¹ (en lettres majuscules) et DTD.
- Schéma XML** Dans cette documentation, le Schéma XML (en majuscules) est utilisé pour faire référence aux schémas qui sont en conformité avec la [spécification de schéma XML W3C](#). Le schéma XML est considéré comme un sous-ensemble de tous les [schémas](#)²¹ (en minuscules).
- URI et URL** Dans cette documentation, l'URI plus générale est utilisée exclusivement — même si l'identifiant a uniquement un aspect de « localisateur », et même pour les identifiants qui utilisent le schéma http.

Termes XSLT et XPath

Il y a eu des modifications dans la terminologie de XSLT 1.0 et XPath 1.0 à XSLT 2.0 et XPath 2.0. Par exemple, le nœud racine dans XPath 1.0 est le [nœud document](#)²² dans XPath 2.0. Dans cette documentation, nous utilisons la terminologie XSLT 3.0 et XPath 3.0 la plus récente.

- XPath absolu** Une expression de chemin qui commence au nœud racine de l'arborescence contenant le [nœud contextuel](#)²². Dans StyleVision, lorsque vous saisissez des expressions de chemin dans les dialogues, l'expression peut être saisie comme chemin absolu si vous vérifiez la case à cocher XPath Absolu dans le dialogue. Si cette case à cocher est décochée, le chemin est relatif au [nœud contextuel](#)²².

Item contextuel / nœud contextuel	L'item contextuel est l'item (nœud ou valeur de string) relatif par rapport l'évaluation de l'expression. Un nœud contextuel est un item contextuel qui est un nœud. Un item contextuel peut changer l'intérieur d'une expression, par exemple, avec chaque étape d'emplacement, ou à l'intérieur d'une expression de filtre (prédicat).
nœud actuel	Le nœud actuel est le nœud qui est actuellement traité. Le nœud actuel est le même que le nœud contextuel ²² dans les expressions qui n'ont pas de sous-expressions. Mais là où il y a des sous-expressions, le nœud contextuel ²² peut changer. Notez que la fonction actuelle() est une fonction XSLT, pas une fonction XPath.
élément du document	Dans un document XML de bonne formation, l'élément le plus extérieur est connu comme élément de document. Il s'agit d'un enfant du nœud du document ²² , et, dans un document XML de bonne formation, il n'existe qu'un seul élément de document. Dans la GUI, l'élément du document est appelé l'élément racine.
nœud du document	Le nœud du document représente et contient le document entier. Il s'agit du nœud de racine de la représentation d'arborescence du document, et il est représenté dans l'expression XPath comme : '/'. Dans la fenêtre d'arborescence du schéma de StyleVision, il est représenté par la légende : '/ Root elements'.

Termes spécifiques à StyleVision

Les termes qui se réfèrent aux mécanismes, concepts et composants de StyleVision.

Image blueprint	Une image blueprint est une image qui est utilisée comme image d'arrière-plan d'un conteneur de mise en page ²¹ , et serait généralement le « scan » d'un formulaire. Le design SPS peut être modelé sur l'image blueprint, recréant ainsi le design du formulaire.
items dynamiques.	Les items qui proviennent de sources de données XML. Les items dynamiques peuvent être du texte, des tables et des listes ; aussi des images et des hyperliens (quand les URI sont dynamiques).
élément global	Un élément dans la liste des éléments globaux dans la fenêtre de l'arborescence de schéma. Dans un Schéma XML, tous les éléments définis comme éléments globaux seront recensés dans la liste des éléments globaux. Dans un DTD, tous les éléments sont des éléments globaux et recensés dans la liste des éléments globaux. Les modèles globaux ²² peuvent être définis uniquement pour les éléments globaux.
modèle global	Un modèle global peut être défini pour un élément global ²² . Une fois défini, un modèle global peut être utilisé pour cet élément à chaque fois que cet élément apparaît dans le document. En alternative au modèle global, le traitement pour un élément global peut être défini dans un modèle local ²² .
Conteneur Mise en page :	Un conteneur de Mise en page est un bloc de design dans lequel les éléments de design peuvent être disposés et positionnés de manière absolue. Si un design doit être basé sur une formulaire, il peut être créé comme Conteneur de Mise en page pour que les éléments du formulaire peuvent être positionnés de manière absolue. En alternative, un design peut être fluide ou avoir des conteneurs de mise en page placés à l'intérieur du flux du document.
modèle local	Un modèle local est un modèle qui définit comment un élément (global ou non global) est traité à l'intérieur du modèle principal ²³ . Le modèle local s'applique à cette occurrence particulière de l'élément dans le modèle principal ²³ . À la place du modèle local, un modèle global ²² peut être appliqué dans une occurrence donnée d'un élément dans le modèle principal ²³ .

schéma principal	Une des sources de schéma assignées est désignée comme schéma principal ; le nœud du document du Fichier de travail XML ²³ associé avec le schéma principal est utilisé comme point de départ pour le modèle principal ²³ .
modèle principal	Le modèle principal de point d'entrée. Dans StyleVision, ce modèle correspond à l' élément du document ²² et est le premier à être évalué par le processeur XSLT. Dans la fenêtre d'arborescence de schéma, il est recensé comme l'enfant du nœud du document ²² . Le modèle principal ²³ définit la structure du document de sortie de base et définit comment le/s document/s d'entrée doivent être traités. Il peut contenir des modèles locaux ²² et peuvent référencer des modèles globaux ²² .
sortie	La sortie produite en traitant un document XML avec une feuille de style XSLT. Les fichiers de sortie qui peuvent être générés par StyleVision seraient en format HTML. Les feuilles de style XSLT générées par StyleVision ne sont pas non plus considérées comme sortie et sont référées séparément comme feuilles de style XSLT.
items statiques	Les items qui ont leur origine dans SPS et pas dans les sources de données XML. Les items statiques peuvent être du texte, des tables et des listes ; aussi des images, hyperliens et signets (quand les URI sont statiques).
Composant SPS	Un composant SPS peut être : (i) un nœud de schéma (par exemple, un nœud d'élément) ; (ii) un composant SPS statique tel qu'un calcul automatique ²⁶¹ ou un string de texte ; ou (iii) un format prédéfini ³²⁹ (représenté dans SPS par ses balises de début et de fin).
modèle	Défini librement comme ensemble d'instructions pour traiter un nœud ou un groupe de nœuds.
Fichier modèle XML	Un Fichier modèle XML est assigné à une SPS dans StyleVision (éditions Enterprise et Professional). Il s'agit d'un fichier XML qui fournit les données démarrant d'un nouveau document XML créé avec une SPS donnée quand cette SPS est ouverte dans Authentic View. Le fichier modèle XML doit être en conformité avec le schéma sur lequel la SPS est basée.
Élément défini par l'utilisateur	Un élément qui n'est ni un nœud dans l'arborescence de schéma ni un élément prédéfini ou un élément design, mais un qui est spécifié par l'utilisateur. Un élément peut être spécifié avec des attributs.
Modèle défini par l'utilisateur	Un modèle est créé pour une séquence spécifiée dans une expression XPath.
Blocs de texte définis par l'utilisateur	Les blocs de texte XML peuvent être insérés librement à tout emplacement dans le design
Fichier de travail XML/XBRL	Un fichier de travail XML/XBRL est un fichier de données XML qui est assigné à une SPS dans StyleVision afin de consulter la sortie output du document XML dans StyleVision. Sans fichier de travail XML/XBRL, la SPS dans StyleVision n'aura pas de données XML dynamiques à traiter. Si la SPS est basée sur un schéma qui contient plus d'un élément global, il peut y avoir une ambiguïté à propos de savoir quel élément global est l'élément du document. Assigner un fichier de travail XML/XBRL résolve une telle ambiguïté (car un document XML valide n'aura, par définition, qu'un élément de document ²²). Notez que cette fonction XBRL est disponible uniquement dans l'édition Enterprise.
Document XML	Le document XML est utilisé de deux manières : (i) pour se référer au document XML spécifique ; (ii) pour se référer à toute source de données XML . Quelle manière est intentionnée doit être claire du contexte.

1.6 À propos de cette documentation

Cette documentation est le manuel utilisateur livré avec <% StyleVision%>. Elle est disponible comme système d'Aide intégré de StyleVision, peut être consultée en ligne sur le [site web d'Altova](#), et peut aussi être téléchargé d'ici comme PDF que vous pouvez imprimer.

Le manuel utilisateur est organisée dans les sections suivantes :

- Une [introduction](#)¹³, qui explique ce qu'est une SPS et introduit les fonctions principales et les concepts de StyleVision.
- Une [description de l'interface utilisateur](#)²⁶, qui fournit un aperçu de la GUI de StyleVision.
- Une section du [tutoriel](#)⁵⁴, qui est un exercice pratique pour se familiariser avec les fonctions StyleVision.
- L'[aperçu d'utilisation](#)⁹⁶, qui décrit l'utilisation à un niveau élevé : par exemple, les sources de schéma utilisées pour créer un SPS, le large processus de design, le déploiement Authentic View et les projets.
- Le [contenu du fichier SPS](#)¹⁰⁹, qui explique comment les composants statiques (origine feuille de style) et dynamiques (origine document XML) sont créés et édités dans SPS.
- La [structure de fichier SPS](#)¹⁸⁵, qui affiche comment un fichier SPS peut être structuré et modulé, et décrit la gestion des modèles de StyleVision.
- Les [fonctions avancées du fichier SPS](#)²⁶⁰, qui décrivent des fonctions de design avancées, telles que la génération automatique de calculs, la définition de conditions, le groupement et le tri de critères définis par l'utilisateur, et comment créer des tables de contenus et des références croisées dans le document de sortie.
- La [présentation du fichier SPS](#)³²⁸, qui explique comment les composants SPS sont formatés et disposés.
- La [fonction SPS d'édition supplémentaire du fichier](#)³⁶², qui décrit une gamme de fonctions supplémentaires qui peuvent rendre vos SPS plus puissantes. Ces fonctions comprennent : des ressources globales pour tirer profit des fonctions dans d'autres produits d'Altova, une validation supplémentaire, des scripts, des variables et des paramètres.
- Une section de [référence](#)⁴²⁰ contenant les descriptions de tous les symboles et toutes les commandes utilisés dans StyleVision.
- Les [Appendices](#)⁵⁴⁸ contenant des renseignements sur l'information du moteur XSLT d'Altova ; des données techniques sur StyleVision ; et une information relative à la licence.

Comment utiliser

Nous suggérons que vous lisiez d'abord les sections '[Introduction](#)¹³, [Interface utilisateur](#)²⁶ et [Aperçu d'utilisation](#)⁹⁶ afin de vous familiariser avec les fonctions de StyleVision et l'utilisation générale. Puis, passez au [tutoriel](#)⁵⁴ pour une expérience pratique de création de SPS. Les sections de fichier SPS ([Contenu de fichier SPS](#)¹⁰⁹, [Structure de fichier SPS](#)¹⁸⁵, [Fonctions avancées de fichier SPS](#)²⁶⁰, [Présentation de fichier SPS](#)³²⁸, [Fonction supplémentaire de fichier SPS](#)³⁶²) fournissent des descriptions détaillées sur comment utiliser les différentes fonctions de StyleVision. Pour une référence suivante, la section [Référence](#)⁴²⁰ fournit une description concise de l'icône de barre d'outils, des symboles de design, et des commandes de menu, organisée selon la barre d'outils et le menu.

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 : StyleVision est aussi pris en charge sur Windows Server 2016 ou plus récent.

Options de prise en charge

Si vous avez des questions ou des problèmes liés à StyleVision, les options de prise en charge suivantes sont disponibles :

1. Consultez le fichier [Aide](#) ¹³ (cette documentation). Le fichier Aide contient une fonction de recherche en texte intégral, outre le fait d'être entièrement indexé.
2. Consultez la [FAQ](#) et le [forum de discussion](#) sur le [site Web d'Altova](#).
3. Contactez le [centre d'assistance d'Altova](#).

Abréviations utilisées les plus fréquemment

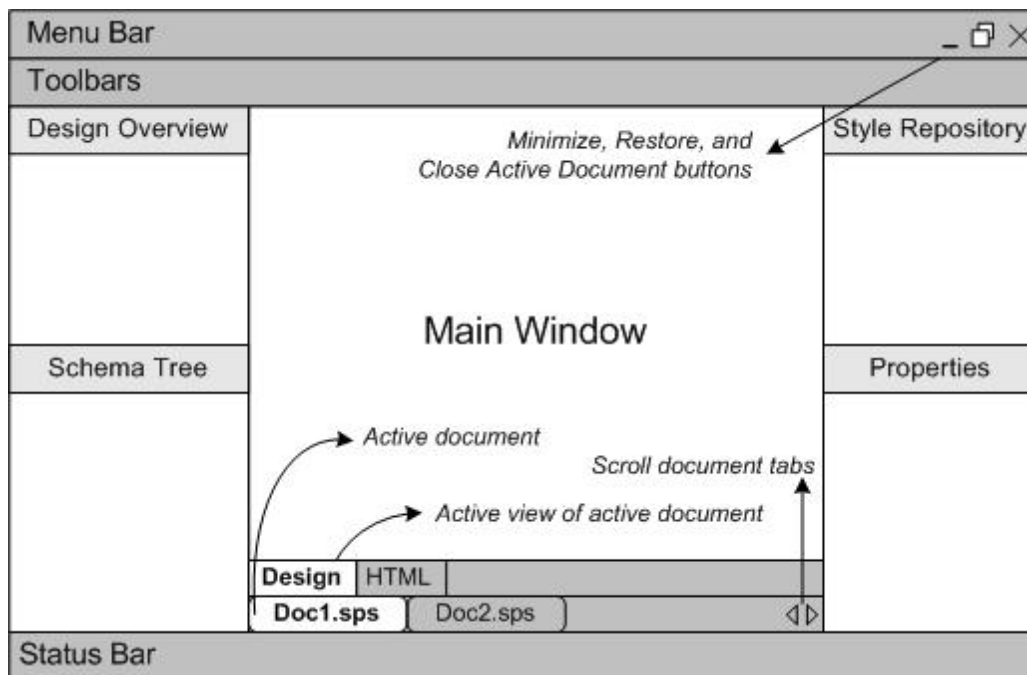
Les abréviations suivantes sont utilisées fréquemment dans cette documentation :

- **SPS** : StyleVision Power Stylesheet
- **CSS** : Feuilles de style en cascade
- **FAQ** : Foire Aux Questions

2 Interface utilisateur

La GUI de StyleVision (illustration ci-dessous, dans laquelle pas toutes les barres latérales sont affichées) consiste en les parties suivantes :

- Une **barre de menu**. Cliquez sur le menu pour afficher les items dans ce menu. Tous les menus et leurs items sont décrits dans la section [Référence de l'utilisateur](#)⁴²⁰. La barre de menu contient également les boutons Minimiser, Restaurer, et Fermer Document actif.
- Un **espace de la barre d'outils**. Les [barres d'outils](#)⁴⁴³ et les raccourcis de commande dans chaque barre d'outils sont décrits dans la section [Référence de l'utilisateur](#)⁴²⁰.
- Une **Fenêtre principale**²⁷ avec onglets qui affiche un ou plus de documents SPS ouverts à la fois. Dans cette fenêtre, vous pouvez [éditer le design de SPS](#)²⁸ et [prévisualiser les feuilles de style XSLT et la sortie](#)³⁰.
- Les [barres latérales de design](#)³² —l'[aperçu de design](#)³⁵, l'[arborescence de schéma](#)³⁷, l'[arborescence de design](#)⁴⁰, le [référentiel de style](#)⁴⁴, les [styles](#)⁴⁶, [propriétés](#)⁴⁸—qui peuvent être ancrés à l'intérieur de l'application GUI ou rendues flottants à l'écran.
- Une **barre de statut**, qui affiche l'information du statut de l'application.

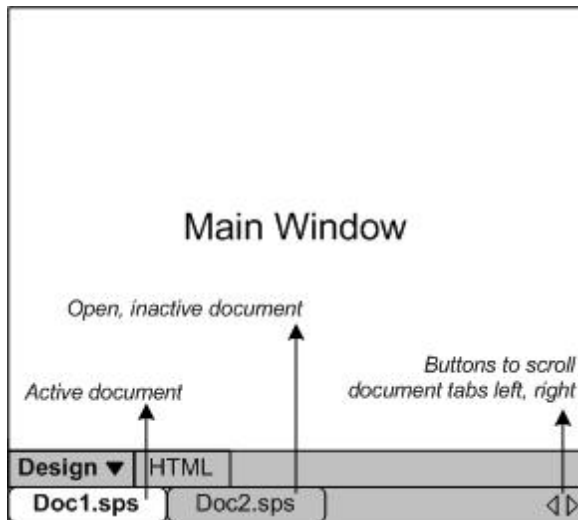


La [fenêtre principale](#)²⁷ et les [barres latérales de design](#)³² sont décrites en détail dans les sous-sections de cette section.

Note : La barre de menu et les barres d'outils peuvent être déplacées en glissant leurs poignées à l'emplacement désiré.

2.1 Fenêtre principale

La **fenêtre principale** (*illustration ci-dessous*) est là où le design SPS, les feuilles de style XSLT, et les aperçus de sortie sont affichés.



Fichiers SPS Design (.mtd) dans la fenêtre principale

- De multiples documents SPS peuvent être ouverts dans StyleVision, bien qu'un seul puisse être actif à tout moment. Les noms de tous les documents ouverts sont affichés dans les onglets en bas de la fenêtre principale, avec l'onglet du document actif étant mis en surbrillance.
- Pour activer un document ouvert, cliquez sur son onglet. En alternative, utilisez les options dans le menu Fenêtres.
- Si tant de documents sont ouverts qu'aucun onglet de document n'est visible dans la barre du document-onglet, cliquez sur le bouton de défilement approprié (à droite de la barre du document-onglet ; voir l'illustration ci-dessus) pour défiler sur les boutons dans l'affichage.
- Pour fermer le document actif, cliquez sur le bouton **Fermer Document** dans la barre de menu en haut à droite de la fenêtre d'application (ou sélectionnez [Fichier | Fermer](#)⁴⁵⁸).

Affichages de documents

Un document est affiché dans les affichages suivants, dont un peut être actif à la fois :

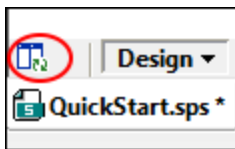
- L'[affichage de design](#)²⁸, dans lequel vous concevez une SPS et éditez les fonctions de JavaScript pour l'utilisation dans cette SPS. L'affichage peut être basculé entre le document de design et l'éditeur JavaScript en cliquant sur la flèche de menu de la liste déroulante et sélectionnant Design ou JavaScript, tel que requis.
- Les [Affichages de sorties](#)³⁰ (sorties HTML, Texte,). Ces affichages sont un aperçu du format de sortie actuel et de la feuille de style XSLT utilisée pour générer cette sortie. L'affichage peut être basculé entre l'aperçu de sortie et la feuille de style XSLT en cliquant sur la flèche de menu de la liste déroulante et en faisant la sélection appropriée.

Chacun des affichages listés ci-dessus est disponible comme onglet en bas de la fenêtre principale de l'onglet des affichages. Pour sélectionner un aperçu, cliquez sur son onglet. L'onglet de l'aperçu sélectionné est mis en surbrillance.

Aperçu de sortie

La fonctionnalité Aperçu de sortie vous permet de visualiser le design dans le volet gauche et de prévisualiser la sortie dans le volet droit. Cela vous permet de prévisualiser la sortie pendant que vous concevez, puis de modifier votre design en conséquence, non seulement en termes de présentation, mais également en termes de contenu.

Vous pouvez cliquer sur l'icône **Aperçu de sortie** (entourée en rouge dans la capture d'écran ci-dessous) pour diviser la fenêtre principale en deux volets verticaux : (i) la [vue Design](#) ²⁸ dans le volet gauche et (ii) les aperçus de la sortie dans le volet droit. L'icône **Aperçu de sortie** (voir capture d'écran ci-dessous) est située en bas à gauche de la fenêtre principale. Pour désactiver l'aperçu de la sortie, cliquez à nouveau sur l'icône **Aperçu de sortie**.

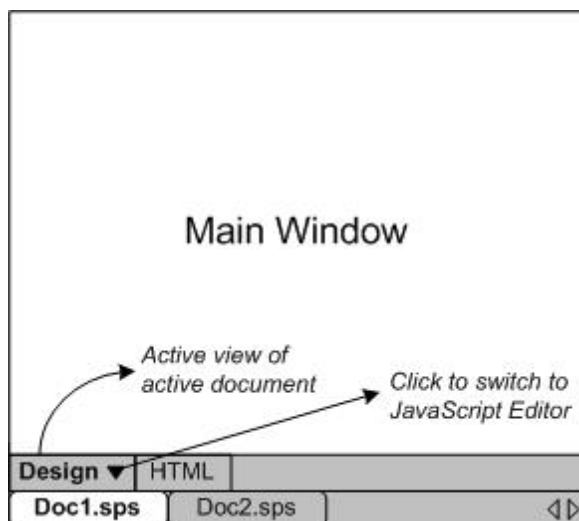


Dans le volet d'aperçu de sortie :

- Cliquez sur l'icône à l'extrême gauche pour régénérer l'aperçu de sortie actuellement sélectionné.
- Cliquez sur l'icône à l'extrême droite pour basculer entre les aperçus verticaux et horizontaux de sortie.

2.1.1 Aperçu de design

L'**Aperçu de design** (illustration ci-dessous) est l'aperçu dans lequel la SPS est conçue. Dans l'Aperçu de design, vous créez le design du document de sortie en (i) insérant le contenu (utilisant les barres latérales, le clavier, et la création de contenu variée ainsi que les fonctions d'édition fournies dans les menus et barres d'outils) ; et (ii) formatant le contenu utilisant les fonctions de formatage variées fournies dans les barres latérales et les menus. Ces aspects de l'Aperçu de design sont expliqués plus en détail ci-dessous.

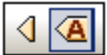


L'Aperçu de design peut aussi être basculé vers un [Éditeur JavaScript](#) ³⁹⁰. Dans l'Éditeur JavaScript, vous pouvez créer et éditer des [fonctions JavaScript](#) ³⁸⁹ qui deviennent ensuite disponibles dans la GUI pour utilisation dans la SPS. Pour basculer vers l'[Éditeur JavaScript](#) ³⁹⁰, cliquez sur le bouton de la liste déroulante dans l'onglet de design (*voir illustration*) et sélectionnez l'Éditeur JavaScript depuis le menu de la liste déroulante. Pour repasser à l'Affichage de design, cliquez sur le bouton de la liste déroulante dans l'onglet l'Éditeur JavaScript et sélectionnez le design depuis le menu de la liste déroulante.

Dans l'Aperçu de design, la SPS peut avoir plusieurs modèles : le modèle principal, les modèles globaux, et les fragments de design. Vous pouvez contrôler lesquels des types de modèles sont affichés dans l'Aperçu de design en utilisant les [filtres d'affichage du modèle](#) ⁴⁴⁹, qui sont disponibles comme les [icônes de la barre d'outils](#) ⁴⁴⁹. Ces filtres d'affichage vous aideront à optimiser et basculer entre les différents affichages de votre SPS.

Afficher des balises

L'affichage des balises dans l'Aperçu de design peut être contrôlé via les icônes de marque (*ci-dessous*).

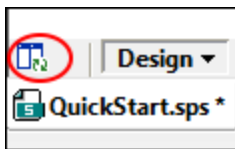


Les icônes affichées ci-dessus sont des boutons bascule. Elles, de gauche : (i) affichent de petites balises design (balises sans noms) ; et (ii) affichent de grandes balises design (balises avec noms). Lorsque des petites balises sont activées, le chemin vers un nœud est affiché quand vous survolez au-dessus de ce nœud avec la souris.

Aperçu de sortie

La fonctionnalité Aperçu de sortie vous permet de visualiser le design dans le volet gauche et de prévisualiser la sortie dans le volet droit. Cela vous permet de prévisualiser la sortie pendant que vous concevez, puis de modifier votre design en conséquence, non seulement en termes de présentation, mais également en termes de contenu.

Vous pouvez cliquer sur l'icône **Aperçu de sortie** (*entourée en rouge dans la capture d'écran ci-dessous*) pour diviser la fenêtre principale en deux volets verticaux : (i) la [vue Design](#) ²⁸ dans le volet gauche et (ii) les aperçus de la sortie dans le volet droit. L'icône **Aperçu de sortie** (*voir capture d'écran ci-dessous*) est située en bas à gauche de la fenêtre principale. Pour désactiver l'aperçu de la sortie, cliquez à nouveau sur l'icône **Aperçu de sortie**.



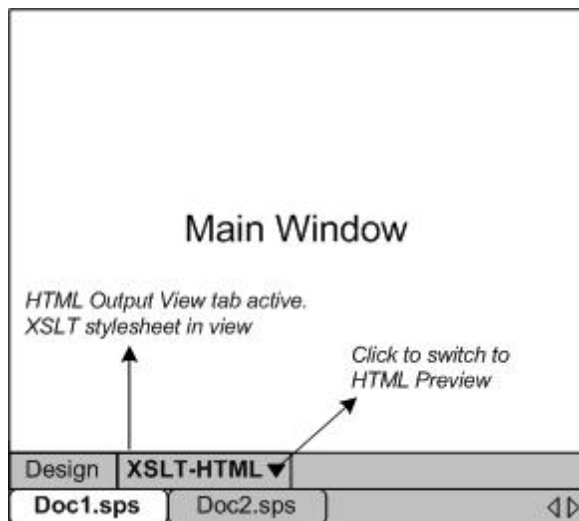
Dans le volet d'aperçu de sortie :

- Cliquez sur l'icône à l'extrême gauche pour régénérer l'aperçu de sortie actuellement sélectionné.
- Cliquez sur l'icône à l'extrême droite pour basculer entre les aperçus verticaux et horizontaux de sortie.

2.1.2 Aperçus de sortie

Chaque **Aperçu de sortie** (*illustration ci-dessous*) affiche : (i) la feuille de style XSLT-pour-HTML générée du design SPS ; et (ii) un aperçu de la sortie HTML, produite en transformant le [fichier de travail XML](#) ²³ avec la feuille de style XSLT.

Dans l'onglet de sortie HTML, l'aperçu peut être basculé entre la feuille de style XSLT-pour-HTML et l'aperçu de la sortie HTML en cliquant sur le bouton de la liste déroulante dans l'onglet de sortie HTML et en sélectionnant l'option XSLT ou l'option de l'aperçu de sortie, tel que requis.



Aperçu XSLT

L'aperçu XSLT affiche la feuille de style XSLT-pour-HTML généré depuis la SPS actuellement active. La feuille de style est générée à nouveau à chaque fois que l'aperçu XSLT est sélectionné.

Une feuille de style dans l'onglet de sortie est affichée avec les éléments « line-numbering » et « expandable/collapsible » ; cliquez sur les icônes + et – dans la marge gauche pour élargir/réduire les éléments. La feuille de style dans l'aperçu XSLT ne peut pas être éditée, mais peut être recherchée (sélectionnez [Éditer | Recherche](#) ⁴⁷⁶) et le texte peut être copié dans le presse-papiers (avec [Éditer | Copier](#) ⁴⁷⁶).

Note : Les feuilles de style XSLT générées depuis SPS peuvent être générées séparément et enregistrées utilisant la commande [Fichier | Enregistrer les fichiers générés](#) ⁴⁶⁹.

HTML

HTML affiche la sortie produite en transformant le [fichier de travail XML](#) ²³ avec la feuille de style XSLT-pour-HTML. La sortie est générée à nouveau à chaque fois que l'onglet de HTML est cliqué. Notez qu'il s'agit de la version enregistrée du fichier de travail XML qui est transformé—et non la version temporaire qui est éditée avec Authentic View.

Si aucun [fichier de travail XML](#) ²³ n'est assigné quand HTML est sélectionné dans l'aperçu HTML, vous serez invité à assigner le fichier de travail XML. Pour les SPS basées sur BD, il n'est pas nécessaire d'assigner un

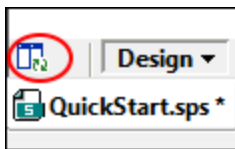
[fichier de travail XML](#) ²³ puisqu'un fichier XML temporaire non éditable est automatiquement généré quand une BB est chargée et que ce fichier XML est utilisé comme [fichier de travail XML](#) ²³.

Note : Les fichiers de sortie générés depuis SPS peuvent être générés et enregistrés utilisant la commande [Fichier | Enregistrer les fichiers générés](#) ⁴⁶⁹.

Aperçu de sortie

La fonctionnalité Aperçu de sortie vous permet de visualiser le design dans le volet gauche et de prévisualiser la sortie dans le volet droit. Cela vous permet de prévisualiser la sortie pendant que vous concevez, puis de modifier votre design en conséquence, non seulement en termes de présentation, mais également en termes de contenu.

Vous pouvez cliquer sur l'icône **Aperçu de sortie** (*entourée en rouge dans la capture d'écran ci-dessous*) pour diviser la fenêtre principale en deux volets verticaux : (i) la [vue Design](#) ²⁸ dans le volet gauche et (ii) les aperçus de la sortie dans le volet droit. L'icône **Aperçu de sortie** (*voir capture d'écran ci-dessous*) est située en bas à gauche de la fenêtre principale. Pour désactiver l'aperçu de la sortie, cliquez à nouveau sur l'icône **Aperçu de sortie**.



Dans le volet d'aperçu de sortie :

- Cliquez sur l'icône à l'extrême gauche pour régénérer l'aperçu de sortie actuellement sélectionné.
- Cliquez sur l'icône à l'extrême droite pour basculer entre les aperçus verticaux et horizontaux de sortie.

2.2 Barres latérales

Les **barres latérales** (aussi appelées fenêtres de barre latérale ou fenêtres) sont des composants GUI qui vous aident à concevoir SPS et vous fournir des informations associées à l'aperçu actif. Chaque barre latérale (*énumérée ci-dessous*) est décrite dans une sous-section de cette section.

- [Aperçu du design](#) ³⁵
- [Arborescence de Schéma](#) ³⁷
- [Arborescence de design](#) ⁴⁰
- [Référentiel de style](#) ⁴⁴
- [Styles](#) ⁴⁶
- [Propriétés](#) ⁴⁸
- [Messages](#) ⁵²
- [Chercher et Remplacer](#) ⁵²

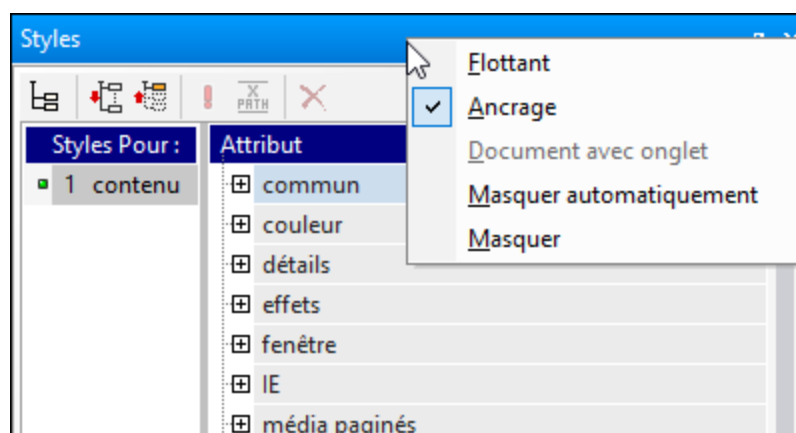
Mise en page des aperçus

La mise en page de l'aperçu fait référence à quelles barres latérales elle est disponible dans cet aperçu et comment ces barres latérales sont positionnées à l'intérieur de la GUI. Les mises en page peuvent être personnalisées pour des catégories d'aperçu séparées, et la personnalisation consiste en deux parties : (i) activer ou désactiver l'affichage des barres latérales individuelles dans un aperçu (par le biais du menu **Aperçu** ou en cliquant avec la touche de droite sur la barre de titre de la barre latérale et sélectionnant **Masquer**); (ii) positionnant la barre latérale à l'intérieur de la GUI, tel que requise. La mise en page définie de cette manière pour une catégorie d'aperçu est conservée pour cette catégorie d'aperçu particulière jusqu'au changement. Donc, si par exemple, dans l'Aperçu Design, toutes les barres latérales à l'exception de la barre latérale sont activées, alors cette mise en page est conservée pour l'Aperçu Design par rapport à de multiples changements d'aperçus, jusqu'au changement de la mise en page de l'aperçu de design. Les catégories d'aperçu sont : (i) aucun document ouvert ; (ii) Aperçu Design ; (iii) Aperçus de sortie.

Ancrer et faire flotter les fenêtres de la barre latérale

Les fenêtres de la barre latérale peuvent être ancrées dans la GUI StyleVision ou peuvent être créées pour flotter sur votre écran. Pour ancrer une fenêtre, glissez la fenêtre par sa barre de titre et déposez-la sur une des quatre pointes de flèche intérieures et extérieures qui apparaissent quand vous commencez à déplacer. Les pointes de flèche intérieures ancrent la fenêtre déplacée relative à la fenêtre dans laquelle les pointes de flèche intérieures apparaissent. Les quatre pointes de flèche extérieures ancrent la fenêtre déplacée dans chacun des quatre bords de la fenêtre d'interface. Pour faire flotter une fenêtre, (i) double-cliquez sur la barre de titre ; ou (ii) déposez-la n'importe où à l'écran, à l'exception des pointes de flèche qui apparaissent quand vous commencez à déplacer la fenêtre.

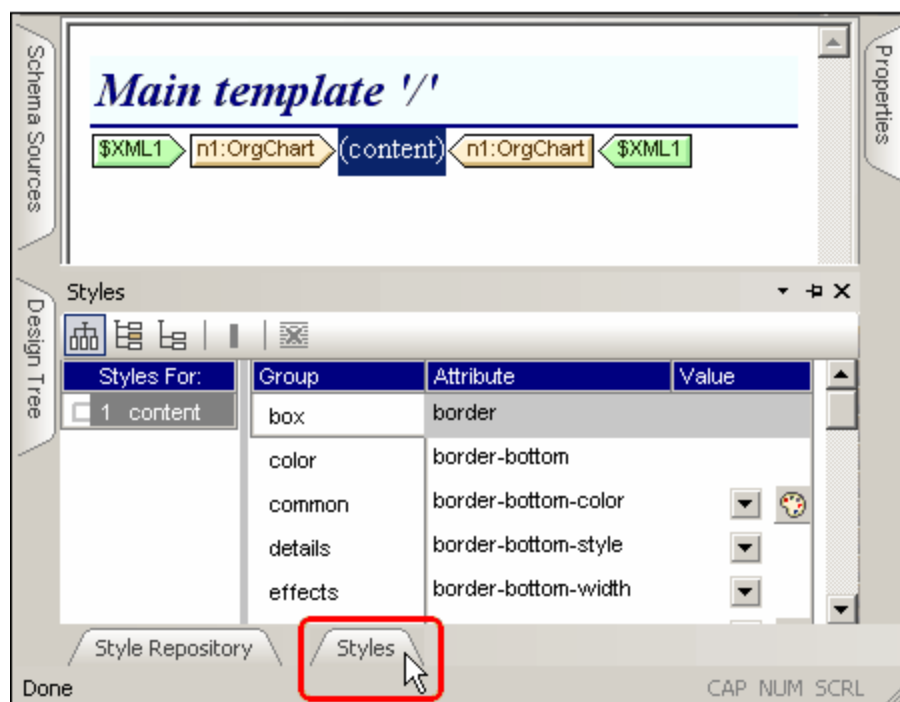
En alternative, vous pouvez aussi utiliser les mécanismes suivants. Pour faire flotter la fenêtre ancrée, cliquez sur le bouton **Menu** en haut à droite de la fenêtre ancrée (*voir la capture d'écran ci-dessous*) et sélectionnez Flottant. Ce menu peut aussi être accédé en cliquant avec la touche de droite sur la barre de titre de la fenêtre ancrée.



Pour ancrer la fenêtre flottante, cliquez avec la touche de droite sur la barre de titre de la fenêtre flottante et sélectionnez **Ancrage** depuis le menu qui apparaît ; la fenêtre sera ancrée dans la position dans laquelle elle a été ancrée en dernier.

Masquer automatiquement les fenêtres de la barre latérale Design

Une fenêtre ancrée peut être masquée automatiquement. Quand une fenêtre de la barre latérale est masquée automatiquement, elle est masquée dans un onglet au bord de la GUI. Placer le curseur au-dessus de l'onglet fait que cette fenêtre se déploie dans la GUI et dans la fenêtre principale. Dans la capture d'écran ci-dessous, placer le curseur sur l'onglet des Styles fait que la barre latérale des Styles se déroule dans la fenêtre principale.



Déplacer le curseur de la fenêtre déployée et de son onglet fait que la fenêtre se restaure dans l'onglet dans le coin de la GUI.

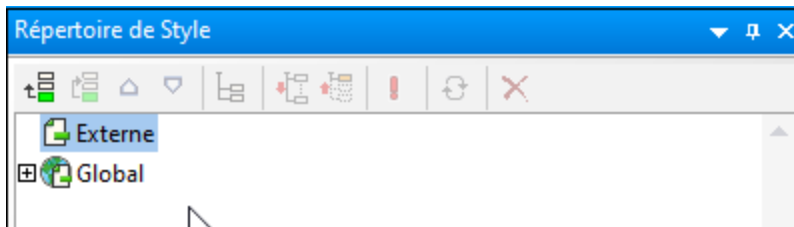
La fonction Masquer automatiquement est utile si vous souhaitez déplacer des barres latérales utilisées rarement de la GUI, tout en vous permettant d'y avoir un accès facile si vous en avez besoin. Ceci vous permet de créer plus d'espace d'écran pour la fenêtre principale tout en permettant un accès facile aux fenêtres de la barre latérale Design.

Pour masquer automatiquement une fenêtre, dans une fenêtre ancrée, cliquez sur le bouton Masquer automatiquement (icône de la punaise) en haut à droite de la fenêtre (*capture d'écran ci-dessous*). En alternative, dans le [Menu](#)³³, sélectionnez Masquer automatiquement ; (pour afficher le [Menu](#)³³, cliquez avec la touche de droite sur la barre de titre de la fenêtre ou cliquez sur le [bouton du Menu](#)³³ dans la barre de titre de la fenêtre ancrée).



La fenêtre sera masquée automatiquement.

Pour désactiver la fonction Masquer automatiquement pour une fenêtre particulière, placez le curseur sur un onglet pour que la fenêtre se déploie, puis cliquez sur le bouton Masquer automatiquement (*capture d'écran ci-dessous*). En alternative, dans le [Menu](#)³³, décochez Masquer automatiquement ; (pour afficher le [Menu](#)³³, cliquez avec la touche de droite sur la barre de titre de la fenêtre ou cliquez sur le [bouton du Menu](#)³³ dans la barre de titre de la fenêtre).



Note : lorsque la fonction Masquer automatiquement de la fenêtre de la barre latérale est désactivée, l'icône de la punaise de cette fenêtre pointe vers le bas ; quand la fonction est activée, l'icône de la punaise pointe à gauche.

Masquer (fermer) les fenêtres de la barre latérale

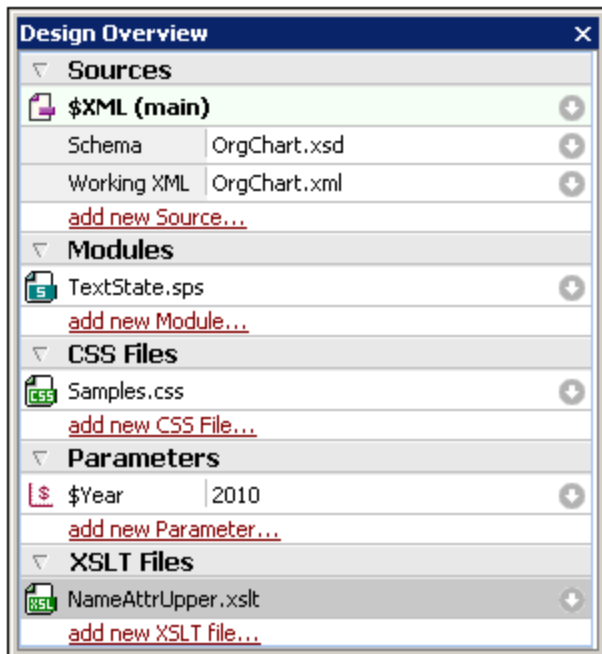
Quand une fenêtre de la barre latérale est masquée, elle n'est plus visible dans la GUI, soit sous sa forme maximisée (ancrée ou flottante), soit sous sa forme minimisée (comme onglet dans le coin de la GUI, ce qui est fait en utilisant la [fonction Masquer automatiquement](#)³³).

Pour masquer une fenêtre, cliquez sur le bouton **Fermer** en haut à droite de la fenêtre ancrée ou flottante. En alternative, dans le [Menu](#)³³, sélectionnez **Masquer**; (pour afficher le [Menu](#)³³, cliquez avec la touche de droite sur la barre de titre de la fenêtre ou cliquez sur le [bouton du Menu](#)³³ dans la barre de titre de la fenêtre).

Pour rendre une fenêtre visible à nouveau (ou fermer), sélectionnez le nom de la barre de titre Design dans le menu [Aperçu](#)⁴⁸⁵. La fenêtre de la barre de titre Design est rendue visible dans la position dans laquelle elle était (ancrée ou flottante) quand elle était hidden.

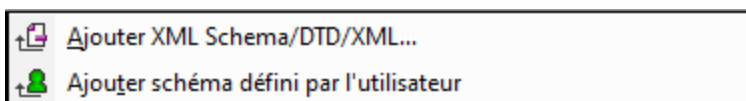
2.2.1 Aperçu du design

La barre latérale **Aperçu du design** (*capture d'écran ci-dessous*) vous permet d'ajouter des sources de schéma, paramètres globaux, SPS modules et fichiers CSS à la SPS active. Ceci vous donne un aperçu de ces composants et vous permet de les gérer de manière conviviale à un emplacement.



Ajouter des sources de schéma

Les sources de schéma peuvent être ajoutées dans une SPS. Une source de schéma est ajoutée en cliquant sur la commande **Ajouter Nouvelle Source** sous l'en-tête Sources. Ceci fait apparaître un menu (*capture d'écran ci-dessous*) qui vous permet d'ajouter un Schéma XML, DTD, schéma généré d'un fichier XML, ou d'un schéma défini par l'utilisateur.



Le fichier de travail XML

Lorsqu'un schéma est ajouté, il est recensé sous l'item Sources. Chaque schéma a une entrée pour le [fichier de travail XML](#) ²³ au sein de l'item XML.

Ajouter des modules, fichiers CSS files, paramètres et fichiers XSLT

Cliquez sur les commandes respectives **Ajouter Nouveau** en bas des sections Modules, Fichiers CSS, Paramètres et Fichiers XSLT pour ajouter un nouvel item à la section respective.

Fonctions d'Aperçu du design

Les fonctions suivantes sont communes à chaque section (Sources, Paramètres, etc.) dans la barre latérale de l'Aperçu du design :

- Chaque section peut être élargie ou réduite en cliquant sur la pointe de flèche triangulaire à gauche du nom de la section.
- Les Fichiers dans les sections Sources, Modules et Fichiers CSS sont recensés uniquement avec leurs noms de fichier. Quand vous défilez au-dessus du nom de fichier, la chemin de fichier entier est affiché dans une fenêtre popup.
- Les items qui sont grisés sont présents dans un module importé, pas dans le fichierSPS actuellement actif dans la GUI.
- Chaque section a également une commande **Ajouter Nouvel <Item>** en bas de la section, qui vous permet d'ajouter un nouvel item à cette section. Par exemple, cliquer sur la commande **Ajouter Nouveau Paramètre** ajoute un nouveau paramètre à la SPS et à la liste des Paramètres dans l'Aperçu du design.
- Chaque item dans une section a un menu contextuel qui peut être accédé soit en cliquant avec la touche de droite sur cet item ou en cliquant sur son icône **Menu contextuel**  (la flèche pointant vers le bas à droite de l'item).
- L'icône **Supprimer** dans le menu contextuel supprime l'item sélectionné. Cette commande est aussi disponible dans les menus contextuels si la commande est applicable.
- La commande du menu contextuel **Éditer Fichier dans XMLSpy** ouvre le fichier sélectionné dans l'application Altova XMLSpy.
- Les commandes du menu contextuel **Vers le haut** et **Vers le bas**, sont applicables uniquement quand un des [multiples modules](#) ²²⁴ dans la section des Modules est sélectionné. Chaque bouton déplace le module sélectionné, respectivement vers le haut ou le bas, par rapport au module immédiatement adjacent.

Sources

La section Sources recense le schéma sur lequel la SPS est basée ainsi que le fichier de travail XML attribué à la SPS. Vous pouvez changer chacune des ces sélections de fichier en accédant son menu contextuel (en cliquant avec la touche de droite ou en cliquant sur l'icône du Menu contextuel ) , puis en sélectionnant l'option **Assign...** appropriée.

Modules

La section Modules recense les [modules SPS](#) ²²⁰ utilisés par la SPS active. Les nouveaux modèles sont ajoutés à la liste en cliquant sur la commande **Ajouter Nouveau Module** et allez au fichier SPS requis. Étant donné que [l'ordre dans lequel les modules sont répertoriés](#) ²²⁴ est significatif, si plus d'un module est répertorié, la/les commande(s) **Vers le haut / Vers le bas** (dans le menu contextuel d'un module) peut/peuvent être utilisée/s pour changer l'ordre. Le menu contextuel fournit également une commande pour ouvrir le module sélectionné dans StyleVision.

Note : la barre latérale Aperçu du design fournit un aperçu des modules, vous permettant de gérer les modules au niveau du fichier. Les divers [objets de module](#) ²²¹ (objets à l'intérieur des modules), toutefois, sont répertoriés dans la [barre latérale de l'arborescence de design](#) ⁴⁰.

Fichiers CSS

La section des Fichiers CSS répertorie les fichiers CSS utilisés par la SPS active. Les nouveaux fichiers CSS sont ajoutés à la liste en cliquant sur la commande **Ajouter Nouveau Fichier CSS** et cherchez le fichier CSS requis. Étant donné que [l'ordre dans lequel les fichiers CSS sont répertoriés](#)³⁴⁴ est significatif, si plus d'un module est répertorié, la/les commande(s) **Vers le haut / Vers le bas** (dans le menu contextuel) devient/deviennent active/s quand un fichier CSS est sélectionné. Le fichier CSS sélectionné peut être déplacé vers le haut ou vers le bas en cliquant sur la commande requise. Le menu contextuel fournit également une commande pour ouvrir le module sélectionné dans <% SPY- GEN%>.

Note : la barre latérale Aperçu du design fournit un aperçu des fichiers CSS, vous permettant de gérer les fichiers CSS au niveau du fichier. Les [règles CSS](#)³⁴⁴ variées à l'intérieur des fichiers CSS, sont, cependant, répertoriées dans la [barre latérale du Référentiel de Style](#)⁴⁴.

Paramètres

La section Paramètres recense les paramètres globaux dans SPS. Vous pouvez ajouter des nouveaux paramètres utilisant la commande **Ajouter Nouveau Paramètre** en bas de la section. Double-cliquer sur le nom du paramètre ou la valeur vous permet d'éditer le nom ou la valeur, respectivement. Pour supprimer un paramètre, sélectionnez le paramètre, puis cliquez sur la commande **Supprimer** dans le menu contextuel.

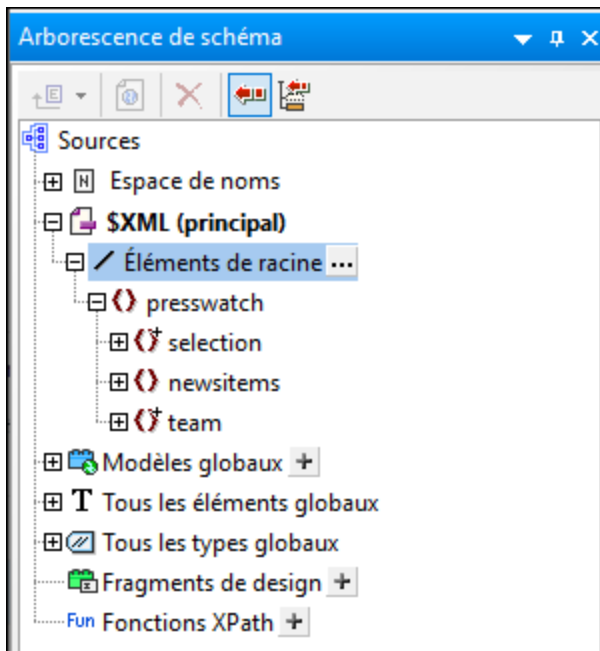
Fichiers XSLT

La section des Fichiers XSLT répertorie les fichiers XSLT qui ont été importés dans la SPS. Les modèles XSLT dans ces fichiers XSLT seront disponibles pour les feuilles de style comme modèles globaux. Pour une description complète sur comment cela fonctionne, voir [Modèles XSLT](#)²⁴⁹.

2.2.2 Arborescence de schéma

La barre latérale **Arborescence de schéma** (*capture d'écran ci-dessous*) vous permet d'exécuter les opérations suivantes :

- Sélectionner de multiples éléments de racine (éléments de document) pour un schéma.
- Glisser des nœuds (éléments, attributs, types globaux) d'une arborescence de schéma et déposez-les dans le design. Ces nœuds représentent le contenu XML qui doit être inclus dans la sortie.
- Afficher les listes de tous les éléments globaux et types dans toutes les sources de schéma la source de schéma. Permet qu'un élément global ou un type global puisse être créé comme modèle global.
- Afficher une liste de tous les noms d'espaces utilisés dans la SPS.
- Insérer et éditer [Fragments de design](#)²⁴⁵.
- Insérer et éditer les [fonctions XPath](#)³⁶⁹ définies par l'utilisateur pour la SPS.

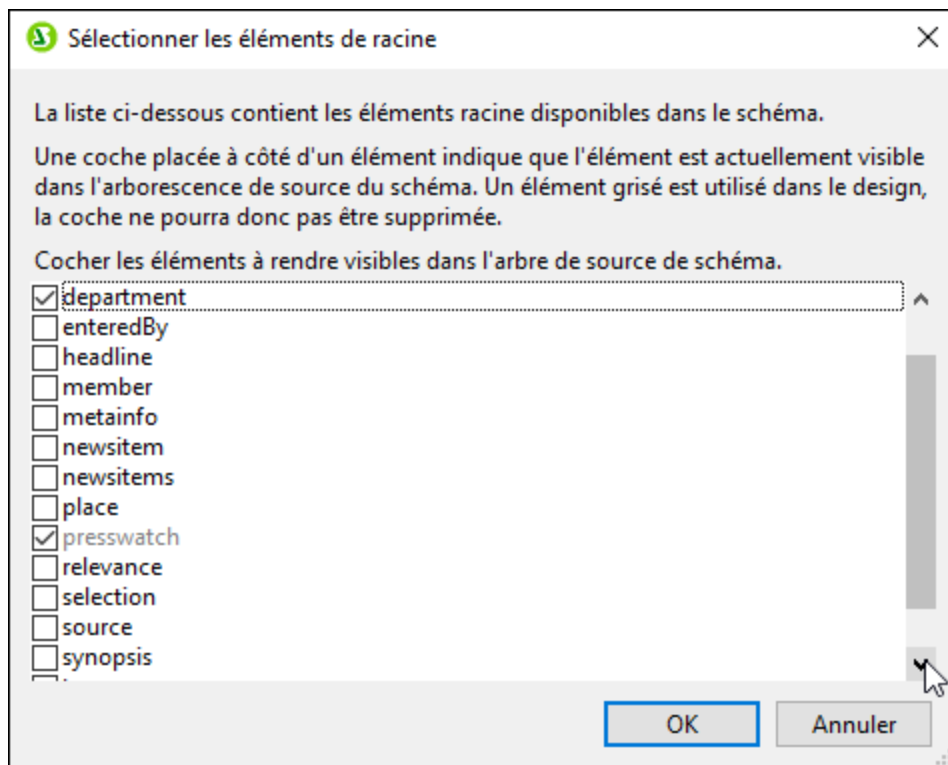


Éléments racine

Pour chaque schéma, sous l'en-tête \$XML, les [éléments de racine](#)²² sélectionnés (ou [éléments de document](#)²²) sont répertoriés. La liste consiste en tous les éléments de racine que vous sélectionnez pour le schéma (voir ci-dessous pour voir comment ceci fonctionne). Chaque élément de racine peut être élargi pour afficher son contenu d'arborescence de modèle. C'est à partir de ces nœuds dans les arborescences d'élément de racine que le contenu du modèle principal est créé. Notez que ce point d'entrée du modèle principal est le nœud du document du schéma principal que vous pouvez sélectionner ou modifier à tout moment (voir ci-dessous comment ceci fonctionne).

Pour sélectionner les éléments de racine pour un schéma, procédez comme suit : Cliquez sur le bouton

Sélectionner  à droite de l'item `Éléments racine`. Ceci fait apparaître le dialogue Sélectionner Éléments racine (*capture d'écran ci-dessous*), dans lequel vous pouvez sélectionner lequel des éléments globaux dans le schéma sera/seront les éléments racine. Voir [Structure SPS | Sources de schéma](#)¹⁸⁷ pour une explication des possibilités offertes par une sélection de multiples éléments racine.



De plus, tous les éléments globaux dans le schéma sont répertoriés sous l'item Tous les éléments globaux. Pour chaque élément global, un [modèle global](#) ²³⁴ peut être créé.

Éléments globaux et types globaux

Les éléments globaux et types globaux peuvent être utilisés pour créer des [modèles globaux](#) ²³⁴ qui peuvent être réutilisés dans d'autres modèles. Les types globaux supplémentaires peuvent aussi être utilisés directement dans des modèles.

Toutefois, les types globaux ne fonctionnent qu'avec des instances XML valides. Si l'instance XML est invalide, alors le type ne correspondra pas et aucun contenu ne sera affiché.

Fragments de design

Tous les [Fragments de design](#) ²⁴⁵ dans le document sont répertoriés sous cet item et peuvent être consultés quand l'item Fragments de design est élargi. La fonction Fragment de design suivante est disponible :

- Créer un Fragment de design en cliquant sur l'icône **Ajouter**  de l'item Fragments de design.
- Double-cliquer sur le nom d'un Fragment de design dans l'Arborescence de schéma permet au nom de ce Fragment de design d'être édité.
- Un Fragment de design peut être activé ou désactivé en cochant ou décochant, respectivement, la case à cocher du Fragment de design.
- Un Fragment de design peut être glissé depuis l'arborescence de schéma dans le design.

Voir la section [Fragments de design](#) ²⁴⁵ pour des informations sur le travail avec les Fragments de design.

Fonctions XPath définies par l'utilisateur

Une fonction XPath définie par l'utilisateur peut être ajoutée en cliquant sur l'icône **Ajouter**  de l'item des Fonctions XPath. Après avoir créé une fonction XPath, elle est répertoriée dans l'Arborescence de schéma. Double-cliquer sur une fonction XPath ouvre la fonction dans son dialogue pour l'édition. La fonction XPath suivante est disponible :

- Une fonction XPath de design peut être activée ou désactivée en cochant ou décochant, respectivement, la case à cocher à côté de l'item Fonctions XPath.
- En cliquant avec la touche de droite sur une fonction XPath, ceci ouvre également un menu contextuel qui contient des commandes pour renommer et déplacer une fonction XPath.

Voir la section [Fonctions XPath définies par l'utilisateur](#) ³⁶⁹ pour des informations détaillées sur le travail avec les fonctions XPath.

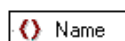
Espaces de noms

Les espaces de noms utilisés dans SPS sont répertoriés sous l'en-tête des espaces de noms ensemble avec leurs préfixes. Les espaces de noms dans cette liste proviennent de deux sources : les (i) espaces de noms définis dans le schéma référencé ou les schémas (*voir note ci-dessous*) ; et les (ii) espaces de noms qui sont ajoutés à chaque SPS nouvellement créée par défaut. Faire référence à une telle liste peut être très utile lors de l'écriture des expressions XPath. De plus, vous pouvez définir un espace de nom XPath par défaut pour toute la SPS en double-cliquant sur le champ de valeur de l'entrée `xpath-default-ns`, puis en saisissant l'espace de nom.

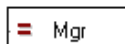
Note : si vous souhaitez ajouter un espace de nom à une feuille de style SPS ou à une feuille de style XSLT générée depuis une SPS créée, l'espace de nom doit être ajouté à l'élément de niveau supérieur `schema` du schéma XML sur lequel la SPS se base.

Les symboles utilisés dans les arborescences de schéma

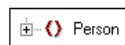
Ci-dessous, vous trouverez une liste des symboles d'arborescences de schéma.



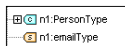
Élément.



Attribut.



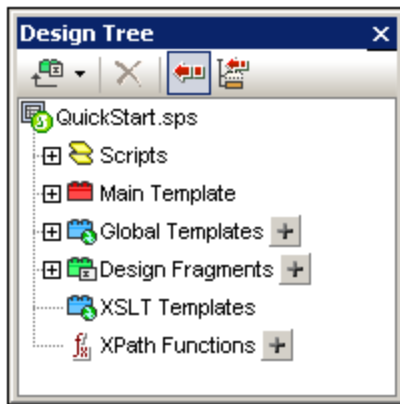
Élément avec des éléments enfant. Double-cliquer sur l'élément ou le symbole +/- à sa gauche fait que l'élément s'élargit/se réduit.



Les types globaux peuvent être soit de nature complexe soit simple. Les types complexes sont indiqués par une icône de couleur cyan, les types simples par une icône de couleur brune.

2.2.3 Arborescence de Design

La barre latérale **Arborescence de Design** (*capture d'écran ci-dessous*) fournit un aperçu de design SPS.



À la racine de l'arborescence de design, il se trouve le nom du fichier SPS ; l'emplacement du fichier est affiché dans une fenêtre pop up lorsque vous placez le curseur de la souris sur le fichier. Le prochain niveau de l'arborescence de design est organisé en catégories comme suit :

- Les [Scripts](#) ³⁸⁹, qui affichent toutes les fonctions de JavaScript qui ont été définies pour la SPS utilisant l'Éditeur JavaScript de StyleVision.
- Le [Modèle principal](#) ²³⁴, qui affiche une structure détaillée du modèle principal.
- Les [Modèles principaux](#) ²³⁴, qui recensent les modèles globaux dans la SPS actuelle, de même que les modèles globaux dans tous les modèles SPS inclus.
- Les [Fragments de design](#) ⁴², qui affichent tous les fragments de design dans le design, et vous permettent de les créer, éditer, renommer et supprimer.
- Les [Modèles XSLT](#) ⁴³, qui fournissent la capacité de consulter les modèles XSLT dans les fichiers XSLT importés.
- Les [Fonctions XPath définies par l'utilisateur](#) ³⁶⁹, qui permettent de créer, éditer, renommer et supprimer vos propres fonctions XPath.

Icônes de barre d'outils

Les icônes de barre d'outils suivantes sont des raccourcis de commandes communes de barres latérales pour de l'Arborescence de conception.



Ajoute un fragment de design, modèle principal ou item de mise à jour dans le design. Cliquez sur la partie gauche de l'icône ajoute un fragment de design. Cliquez sur la flèche vers le bas fait afficher une liste avec des commandes pour ajouter un fragment de design ou tous les différents items de mise en page.



Supprimez l'item sélectionné ; l'icône est active quand l'item dans les Modèles globaux ou les sous-arborescences de mise en page est sélectionné.



Synchroniser la bascule d'arborescence. Lorsque la bascule est activée (l'icône a des bordures), sélectionner un nœud dans l'arborescence sélectionne (i) le nœud correspondant dans le design, et (ii) le nœud correspondant dans l'Arborescence de design si l'icône Synchroniser l'Arborescence de design dans d'Arborescence de design est activée. Lorsque désactivé, les nœuds correspondants dans le design et l'arborescence de schéma ne sont pas sélectionnés.



Réduit automatiquement d'autres éléments dans l'Arborescence de design quand Synchroniser la bascule d'arborescence est activée et que l'item est sélectionné dans le design. Notez que cette bascule est activée uniquement lorsque Synchroniser la bascule d'arborescence est activée.

Modifier l'affichage de l'arborescence de design

L'affichage de l'arborescence de design peut être modifié via le menu contextuel (voir la capture d'écran ci-dessous), qui s'affiche en cliquant avec la touche de droite sur un item dans l'arborescence de design.



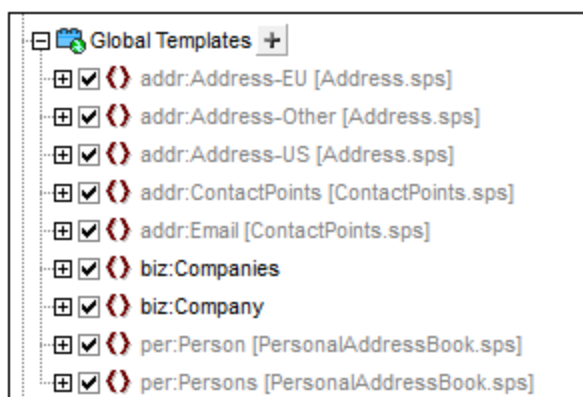
La commande **Supprimer** supprime l'item d'arborescence de design depuis l'arborescence de design. **Make Design Fragment** crée un [fragment de design](#) ⁴² dans le design et ajoute un item dans l'arborescence de design. **Élargir tout** élargit tous les items de l'arborescence de design.

Scripts et Modèle principal

La liste des scripts affiche tous les scripts dans le Design, y compris ceux dans les modules importés. La liste du Modèle principal affiche une arborescence du modèle principal. Les items dans l'arborescence et le design peuvent être supprimés en cliquant avec la touche de droite sur l'item et en sélectionnant **Supprimer**.

Modèles globaux

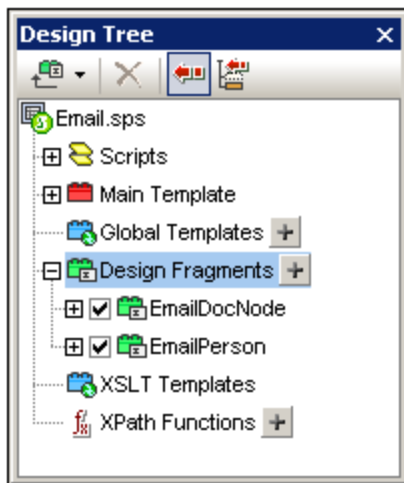
L'item [Modèles principaux](#) ²³⁴ recense tous les modèles globaux dans la SPS actuelle, de même que dans tous les modèles SPS ajoutés. Les modèles globaux définis dans la SPS actuelle sont affichés en noir, tandis que des modèles globaux qui ont été définis dans des modules ajoutés sont affichés en gris (voir la capture d'écran ci-dessous). Chaque modèle global a une case à cocher à sa gauche qui permet que vous l'activiez ou désactiviez. Quand un modèle global est désactivé, il est supprimé du design.



Un modèle global dans la SPS actuelle (pas un dans un module ajouté) peut être éliminé en sélectionnant et en cliquant sur le bouton **Supprimer** dans la barre d'outils ou la commande **Supprimer** dans le menu contextuel. Le composant est supprimé depuis le design et l'arborescence.

Fragments de design

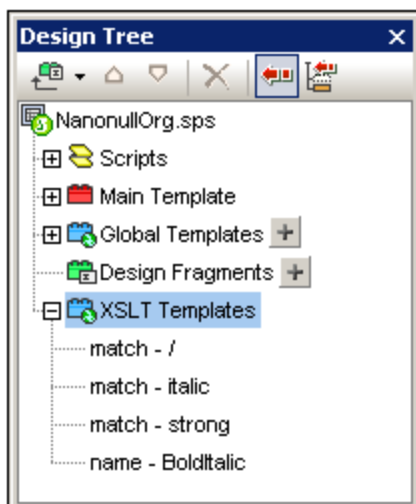
L'item [Fragments de design principaux](#)²⁴⁵ recense tous les Fragments de design dans la SPS actuelle, de même que dans tous les modèles SPS ajoutés. Les Fragments de design définis dans la SPS actuelle sont affichés en noir, tandis que Fragments de design qui ont été définis dans des modules ajoutés sont affichés en gris (*voir la capture d'écran ci-dessous*). Chaque Fragment de design a une case à cocher à sa gauche, qui permet que vous l'activiez ou désactiviez. Un fragment de design dans la SPS actuelle (pas un dans un module ajouté) peut être éliminé en sélectionnant et en cliquant sur la commande **Supprimer** dans le menu contextuel. Le composant est supprimé depuis le design et l'arborescence.



Un fragment de design peut être ajouté en cliquant sur l'icône Ajouter  à droite de l'item des Fragments de design. Chaque fragment de design est conçu comme une arborescence avec des nœuds qui s'élargissent/se réduisent. Tout composant dans une arborescence de fragment de design (qui est défini dans la SPS actuelle) peut être supprimé en le sélectionnant et en cliquant sur le bouton **Supprimer** dans la barre d'outils ou la commande **Supprimer** dans le menu contextuel. Le composant est supprimé depuis le design et l'arborescence.

Modèles XSLT

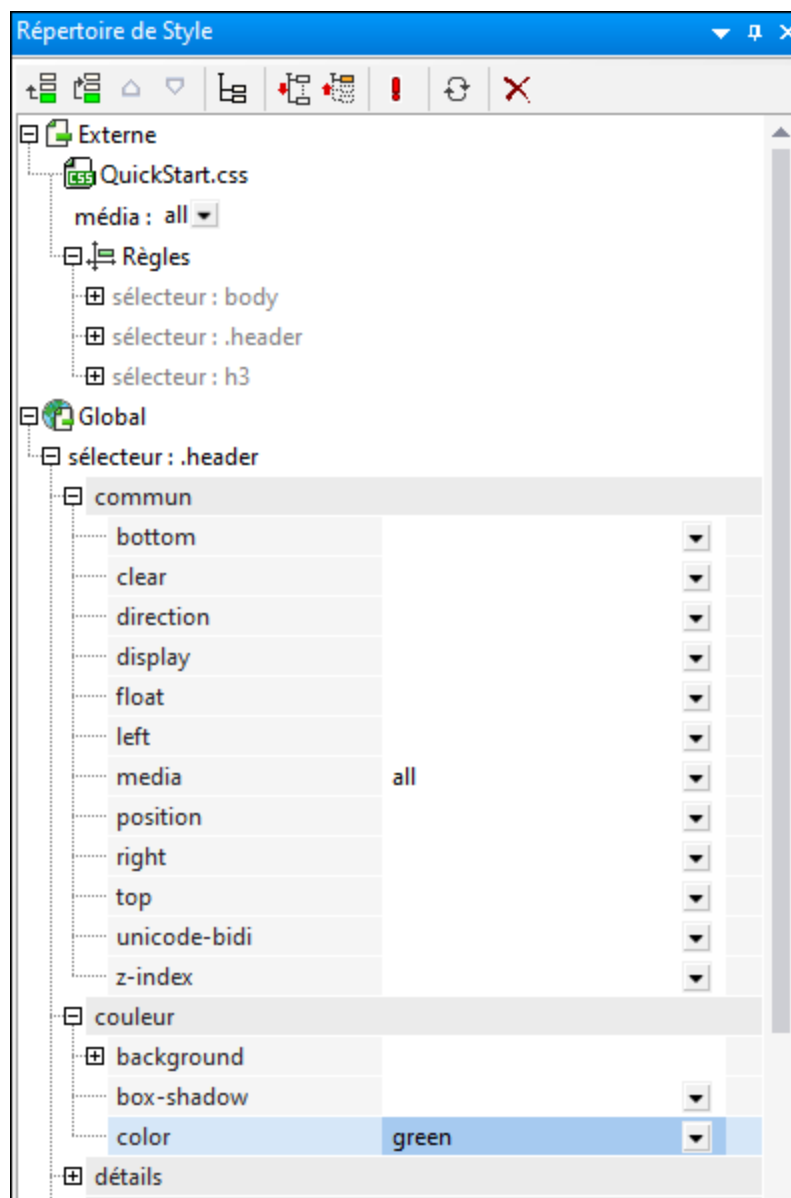
Dans la barre latérale de l'arborescence de design (*capture d'écran ci-dessous*), les modèles XSLT contenus dans le fichier XSLT importé sont affichés sous les en-têtes de modèles XSLT.



Il existe deux types de modèles XSLT importés : (i) modèles correspondants (indiqués par *Match*), et (ii) modèles nommés (indiqués par *Name*). Dans l'arborescence de design, ces deux types sont recensés avec la valeur de l'attribut `select` des modèles correspondants, et (ii) par la valeur de l'attribut `name` des modèles nommés, respectivement. Pour une description complète sur comment les Modèles XSLT fonctionnent, voir [Modèles XSLT](#)²⁴⁹.

2.2.4 Référentiel de style

Dans la barre latérale **Référentiel de style** (*capture d'écran ci-dessous*), vous pouvez assigner des feuilles de style CSS externes et définir des styles CSS globaux pour SPS. Les règles de style dans des feuilles de style CSS externes et styles CSS définis globalement sont appliquées à au document de sortie HTML.



La barre latérale du Référentiel de style contient deux listes, **Externe** et **Global**, chacune sous forme d'une arborescence. La liste externe contient une liste de feuilles de style CSS externes associées avec SPS. La liste globale contient une liste de tous les styles globaux associés avec SPS.

La structure de listes dans le Référentiel de style est comme suit :

Externe

- CSS-1.css (emplacement indiqué dans la popup qui apparaît en plaçant le curseur de la souris au-dessus)

- Media (peut être défini dans la fenêtre de le Référentiel de style)

- Règles (non-éritable ; doit être édité dans le fichier CSS)

- Selector-1

- Property-1

- ...

- Property-N

- ...

- Selector-N

- + ...

- + CSS-N.css

Global

- Selector-1

- + Selector-1 Properties

- ...

- + Selector-N

Précédence de règles de style

Si une règle de style globale et une règle de style dans une feuille de style CSS externe ont des sélecteurs qui identifient le même composant de document, alors la règle de style globale a précédence sur une feuille de style externe et sera appliquée. Si deux ou plusieurs règles de style globales sélectionnent le même composant de document, alors la règle qui est recensée en dernier de ces règles sera appliquée. De même, si deux ou plusieurs règles de style dans les feuilles de style externes sélectionnent le même composant de document, alors la dernière de ces règles dans les dernières feuilles de style sera appliquée.

Gérer les styles dans lu Référentiel de style

Dans la barre latérale du Référentiel de style, vous pouvez suivre les étapes suivantes, utilisant soit les icônes dans la barre d'outils et/ou les items dans le menu contextuel :

Ajouter : L'icône **Ajouter**  ajoute une entrée de feuille de style externe à l'arborescence externe ou une nouvelle entrée de style globale à l'arborescence globale, respectivement, selon si l'arborescence externe ou globale a été sélectionnée. La nouvelle entrée est ajoutée à la liste des entrées déjà existantes dans l'arborescence. La commande **Ajouter** est également disponible dans le menu contextuel. Pour plus de détails concernant les feuilles de style externes et les styles globaux, voir [Travailler avec les Styles CSS](#) ³⁴³. Notez qu'une feuille de style CSS externe peut aussi être ajoutée ou une feuille de style supprimée via la [barre latérale de l'Aperçu de Design](#) ³⁵.

Insérer : L'icône **Insérer**  insère une nouvelle entrée de feuille de style externe au-dessus de la feuille de style externe sélectionnée (dans l'arborescence externe) ou une nouvelle entrée de style globale au-dessus du style global sélectionné (dans l'arborescence globale). La commande **Insérer** est aussi disponible dans le

menu contextuel. Pour plus de détails concernant les feuilles de style externes et les styles globaux, voir [Travailler avec les Styles CSS](#) ³⁴³.

Déplacer Haut/Bas : L'icône **Déplacer vers le haut**  et **Déplacer vers le bas**  déplace la feuille de style externe sélectionnée ou le style global respectivement vers le haut et le bas de manière relative à d'autres entrées dans son arborescence. Ces commandes sont utiles pour changer la priorité des feuilles de style externes par rapport à elles-mêmes et des règles de style globaux relatives à elles-mêmes. Les commandes **Déplacer vers le haut** et **Déplacer vers le bas** sont aussi disponibles dans le menu contextuel. Pour plus de détails sur comment changer la précedence de styles, voir [Travailler avec les Styles CSS](#) ³⁴³.

Aperçus de styles du sélecteur : Tout sélecteur, soit dans une feuille de style externe ou définie globalement, peut être affichée dans un aperçu obtenu en utilisant les paramètres d'aperçu de l'arborescence. Ces

paramètres sont : **Liste non vide** , **Tout développer** , et **Tout réduire** , et ils sont disponibles dans les boutons de la barre d'outils et les commandes du menu contextuel : Activer le paramètre *List Non-Empty* fait que seules ces propriétés de style sont recensées qui ont une valeur définie pour elles. Autrement, toutes les propriétés de style disponibles sont affichées (ce qui pourrait rendre l'aperçu très encombré). Les paramètres *Expand All* et *Collapse All* combinent le paramètre *List Non-Empty* et respectivement élargit et réduit toutes les définitions de style du sélecteur sélectionné. Ces commandes sont aussi disponibles dans le menu contextuel.

« **Toggle Important** » : Cliquez sur l'icône Toggle Important  définit la valeur CSS `!important` comme activé ou désactivé pour la règle CSS sélectionnée.

« **Reload All** » : L'icône **Recharger tout**  recharge des feuilles de style CSS externes.

Réinitialiser : L'icône **Réinitialiser**  supprime la feuille de style externe sélectionnée ou le style global.

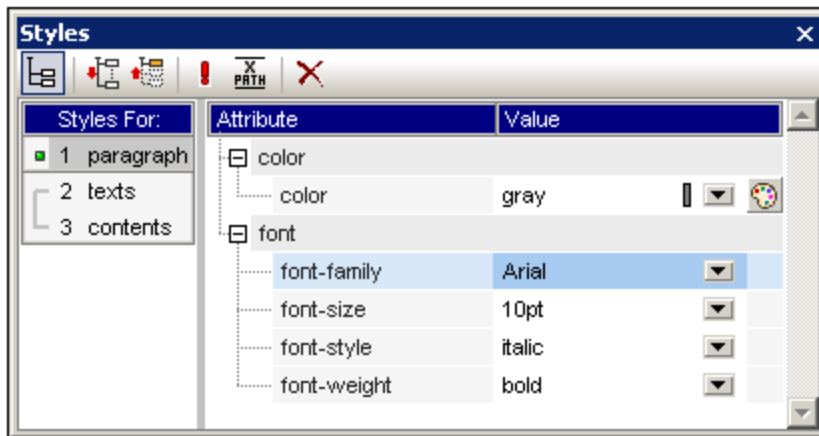
Éditer les styles CSS dans le Référentiel de style

Les mécanismes d'édition suivants sont fournis dans le Référentiel de style :

- Vous pouvez ajouter ou supprimer une feuille de style CSS et vous pouvez spécifier le média sur lequel la feuille de style CSS externe s'applique. Suivez les explications dans la section, [Feuilles de style CSS externes](#) ³⁴⁴.
- Les styles globaux peuvent avoir leurs sélecteurs et propriétés directement édités dans la fenêtre du Référentiel de style. Pour savoir comment cela fonctionne, consultez la section [Définir les Styles CSS globaux](#) ³⁴⁸.

2.2.5 Styles

La barre latérale **Styles** (*capture d'écran ci-dessous*) permet aux styles CSS d'être définis **localement** pour les composants SPS sélectionnés dans l'Aperçu de design. Ceci est opposé aux styles qui sont définis globalement dans la barre latérale du [Répertoire de Styles](#) ⁴⁴.



La barre latérale de Styles est divisée en deux grandes parties :

- Le côté gauche, **Styles-For column**, dans lequel les types de composant sélectionnés sont recensés. Vous devriez noter que quand une sélection est effectuée dans l'Aperçu de design, elle pourrait contenir plusieurs composants. Les composants sélectionnés sont recensés dans la colonne Styles-For, organisée par le type de composant. Un de ces types de composants peut être sélectionnée à la fois pour le styling. S'il existe uniquement un type de composant, alors cette instance est sélectionnée pour le styling. S'il existe plusieurs instances du type de composant, alors toutes les instances sélectionnées peuvent être stylées ensemble. Les styles définis sont appliqués localement à chaque instance. Si vous souhaitez styler uniquement une instance spécifique, sélectionnez cette instance de composant spécifique dans l'Aperçu de design et stylez-la dans la barre latérale des Styles. Vous pouvez aussi sélectionner une plage de composant en sélectionnant le composant « start-of-range » puis le composant « end-of-range » avec la clé **Shift** appuyée. Pour des informations détaillées sur la sélection des types de composant, voir [Définir les Styles CSS localement](#) ³⁵⁰.
- Le côté droit, **volet Définitions de style**, dans lequel les styles CSS sont définis pour le/s type/s sélectionné/s dans la colonne Styles-For. Le volet Définitions de style peut être affiché dans trois aperçus (voir ci-dessous pour la description). Pour les détails comment définir les définitions de style, voir [Définir les Valeurs de style CSS](#) ³⁵². L'icône XPath  active ou désactive l'application des expressions XPath comme source des valeurs de style. Si une propriété de style est sélectionnée et si l'icône XPath est activée, alors l'expression XPath peut être saisie pour cette propriété et la valeur de renvoi de l'expression XPath est utilisée comme valeur de cette propriété de style. De cette manière, la valeur d'un nœud dans un document XML peut être renvoyé au moment de l'exécution comme valeur d'une propriété. Quand l'icône XPath est activée, une valeur statique peut être saisie comme valeur de la propriété.

Paramètres pour l'affichage des définitions

L'affichage des définitions peut être modifié pour répondre à vos besoins d'édition. Trois paramètres d'affichage (énumérés ci-dessous) sont disponibles sous forme de boutons dans la barre d'outils et de commandes dans les menus contextuels.

- Liste non vide**  : lorsque ce paramètre est activé, seules les propriétés ayant des valeurs définies sont affichées, par ordre alphabétique, pour le type de composant sélectionné dans la colonne

de gauche. Sinon, toutes les propriétés sont affichées. Ce paramètre est très utile si vous souhaitez voir quelles propriétés sont définies pour un type de composant particulier. Si vous souhaitez définir de nouvelles propriétés pour le type de composant sélectionné, ce paramètre doit être désactivé afin que vous puissiez accéder à la propriété requise.

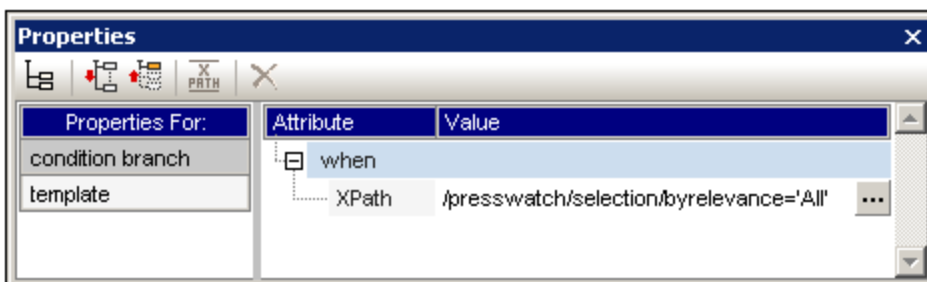
- **Tout développer**  : pour le type de composant sélectionné dans la colonne de gauche, toutes les propriétés affichées dans le volet de droite sont développées. Ce paramètre peut être combiné avec le paramètre **Liste non vide**.
- **Tout réduire**  : pour le type de composant sélectionné dans la colonne de gauche de la fenêtre, toutes les propriétés affichées dans le volet de droite sont réduites. Ce paramètre peut être combiné avec le paramètre **Liste non vide**.

Icônes des barres d'outils Toggle Important et Réinitialiser

Cliquez sur l'icône Toggle Important  définit la valeur CSS `!important` comme activé ou désactivé pour la règle CSS sélectionnée. Cliquez sur l'icône Réinitialiser  pour réinitialiser la valeur de la propriété sélectionnée.

2.2.6 Propriétés

La barre latérale **Propriétés** (capture d'écran ci-dessous) permet aux propriétés d'être définies pour les composants SPS sélectionnés dans l'Aperçu de design.



La barre latérale des Propriétés est divisée en deux grandes parties :

- La **colonne Styles-For**, dans laquelle les types de composant sélectionnés sont recensés. Un de ces types de composant peut être sélectionné à la fois et les propriétés y assignées. (Dans la capture d'écran ci-dessus, le composant *template* est sélectionné.) Pour des informations détaillées sur comment les composants avec des propriétés sont groupés, voir la section [Composants et leurs Groupes de propriété](#) ⁴⁹ ci-dessous.
- Le **volet Définitions de propriété**, dans lequel les propriétés de composant sont définies pour le type de composant sélectionné dans la colonne Properties-For. Le volet Définitions de Propriété peut être affiché dans trois aperçus (voir ci-dessous). Pour les détails des propriétés dans chaque groupe de propriété, voir la section [Groupes de propriété](#) ⁵¹ ci-dessous.

Paramètres pour l'affichage des définitions

L'affichage des définitions peut être modifié pour répondre à vos besoins d'édition. Trois paramètres d'affichage (*énumérés ci-dessous*) sont disponibles sous forme de boutons dans la barre d'outils et de commandes dans les menus contextuels.

- Liste non vide**  : lorsque ce paramètre est activé, seules les propriétés ayant des valeurs définies sont affichées, par ordre alphabétique, pour le type de composant sélectionné dans la colonne de gauche. Sinon, toutes les propriétés sont affichées. Ce paramètre est très utile si vous souhaitez voir quelles propriétés sont définies pour un type de composant particulier. Si vous souhaitez définir de nouvelles propriétés pour le type de composant sélectionné, ce paramètre doit être désactivé afin que vous puissiez accéder à la propriété requise.
- Tout développer**  : pour le type de composant sélectionné dans la colonne de gauche, toutes les propriétés affichées dans le volet de droite sont développées. Ce paramètre peut être combiné avec le paramètre **Liste non vide**.
- Tout réduire**  : pour le type de composant sélectionné dans la colonne de gauche de la fenêtre, toutes les propriétés affichées dans le volet de droite sont réduites. Ce paramètre peut être combiné avec le paramètre **Liste non vide**.

Icône de la barre d'outils « Reset »

Cliquer sur l'icône « Reset »  redéfinit la valeur d'une propriété sélectionnée à son défaut.

Composants et leurs groupes de propriété

La disponibilité des groupes de propriété est sensible au contexte. Quels groupes de propriété sont disponibles dépend du composant de design sélectionné. La table ci-dessous recense les composants SPS et les groupes de propriété qu'ils ont.

Composant	Groupe de propriété
Contenu	Content; Common; Event
Texte	Text ; Common ; Event
Calcul automatique	AutoCalc ; Common ; Event
Condition Branch	When
Appareil d'entrée de données	Common ; [Data-Entry Device] ; Event ; HTML
Image	Image ; Common ; Event ; HTML
Lien	Link ; Common ; Event ; HTML
Table	Table ; Common ; Event ; HTML ; Interactive

Paragraphe	Paragraphe ; Common ; Event ; HTML
------------	------------------------------------

Veillez noter les points suivants sur les types de composant :

- Les composants de contenu sont les espaces réservés `content` et `rest-of-contents`. Ceux-ci représentent le contenu de texte d'un nœud ou de nœuds depuis le document XML.
- Un composant texte est un string unique de texte statique. Un string unique s'élargit entre deux composants autres que les composants texte, et inclut les espaces réservés, si présents.
- Les appareils d'entrée de données sont un champ d'entrée, de multiples champs d'entrée, des zones de liste déroulantes, des cases à cocher, des cases d'option ; leurs propriétés couvrent un appareil d'entrée de données ainsi que des contenus de l'appareil d'entrée de données, le cas échéant.
- Un composant de table se réfère à la structure de table dans le design. Notez qu'il contient des sous-composants, qui sont considérés des composants de leur propre droit. Les sous-composants sont : ligne, colonne, cellule, en-tête et bas de page.
- Un composant de paragraphe est tout format prédéfini.

La table ci-dessous contient des descriptions de chaque groupe de propriété.

Groupe de propriété	Description
AutoCalc	Ces propriétés sont activées quand le calcul automatique est sélectionné. La propriété <code>Formatage de valeur</code> spécifie le formatage ³³⁴ d'un calcul automatique qui est numérique ou de type de données de date. La propriété <code>XPath</code> spécifie l'expression XPath qui est utilisée pour le Calcul automatique ²⁶¹ .
Commun	Le groupe de propriétés <i>Common</i> est disponible pour tous les types de composant à l'exception des types de composant Template et AutoCalc. Il contient les propriétés suivantes qui peuvent être définies pour le composant : <code>class</code> (un nom de classe), <code>dir</code> (la direction de l'écriture), <code>id</code> (une ID unique), <code>lang</code> (le langage), et <code>title</code> (un nom).
Appareil d'entrée de données	Spécifie la portée de la valeur des zones de liste, cases à cocher et cases d'option. Notez que le groupe de propriétés ne s'applique pas aux champs d'entrée et aux boutons.
Événements	Contient des propriétés qui permettent des fonctions JavaScript ³⁸⁹ à être définies pour les événements HTML « client-side » : <code>onclick</code> , <code>ondblclick</code> , <code>onkeydown</code> , <code>onkeypress</code> , <code>onkeyup</code> , <code>onmousedown</code> , <code>onmousemove</code> , <code>onmouseout</code> , <code>onmouseover</code> , <code>onmouseup</code> .
HTML	Disponible pour les types de composants suivants : appareils data-entry ¹⁵⁸ ; image ¹⁵² ; lien ³²¹ ; table ¹²⁵ ; paragraphes ³²⁹ . Notez qu'il existe différents types d' appareils « data-entry » ¹⁵⁸ et de paragraphes ³²⁹ , et que les tables ¹²⁵ ont des sous-composants. Ces propriétés sont de propriétés HTML qui peuvent être définies dans les éléments HTML correspondants (<code>img</code> , <code>table</code> , <code>p</code> , <code>div</code> , etc.). Les propriétés disponibles varient pour cette raison conformément au composant sélectionné. Les valeurs pour ces propriétés peuvent être sélectionnées utilisant les expressions XPath.

De plus, il existe des propriétés « component-specific » pour des [images](#)¹⁵², [liens](#)³²³, [paragraphes et d'autres formats prédéfinis](#)¹¹², et des [branches de condition](#)²⁶⁹. Ces propriétés sont décrites dans les sections respectives :

Définir des valeurs de propriété

Les valeurs de propriété peuvent être saisies d'une, de deux ou de trois manières, dépendant de la propriété :

- Saisir directement dans la colonne Valeur. À cette fin, sélectionnez une propriété, double-cliquez sur sa colonne Valeur, saisissez la valeur utilisant le clavier et appuyez sur **Saisir** ou cliquez n'importe où dans la GUI.
- En sélectionnant une valeur à partir de la liste déroulante dans la liste de choix pour cette propriété. Cliquez sur la flèche du bas de la zone de liste déroulante des options « property-value ».
- En utilisant le bouton Éditer  à droite de la colonne Valeur pour cette propriété. Cliquer sur le bouton Éditer ouvre une dialogue pertinent pour cette propriété.

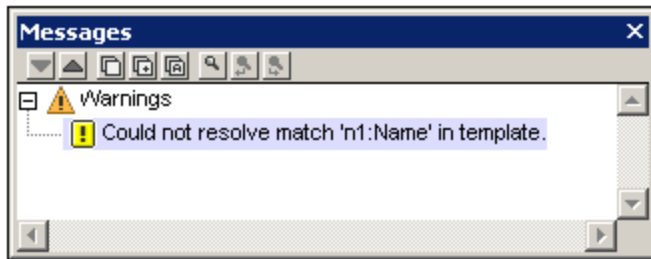
Pour certaines propriétés, dans les groupes de propriétés « Common » et HTML, les expressions XPath peuvent être utilisées pour fournir les valeurs de la propriété. L'icône XPath  active ou désactive l'application des expressions XPath comme source des valeurs de propriété. Avec une propriété sélectionnée, si l'icône XPath est activée, alors l'expression XPath peut être saisie pour cette propriété et la valeur de renvoi de l'expression XPath est utilisée comme valeur de cette propriété de style. De cette manière, la valeur d'un nœud dans un document XML peut être renvoyé, au moment de l'exécution, comme valeur d'une propriété. Quand l'icône XPath est activée, une valeur statique peut être saisie comme valeur de la propriété. *Voir aussi [Propriétés de style via XPath](#)* ³⁵⁴.

Modifier ou supprimer une valeur de propriété

Pour modifier une valeur de propriété, utilisez toute méthode applicable décrite dans le paragraphe précédent, [Définir des valeurs de propriété](#) ⁵¹. Pour supprimer une valeur de propriété, sélectionnez la propriété et cliquez sur l'icône Reset  dans la barre d'outils de la barre latérale des Propriétés.

2.2.7 Messages

Quand StyleVision est ouvert pour la première fois, la barre latérale des Messages (*capture d'écran ci-dessous*) est affichée sous la Fenêtre principale de la GUI. Pour activer et désactiver la barre latérale des Messages, cliquez sur **Aperçu | Messages**.

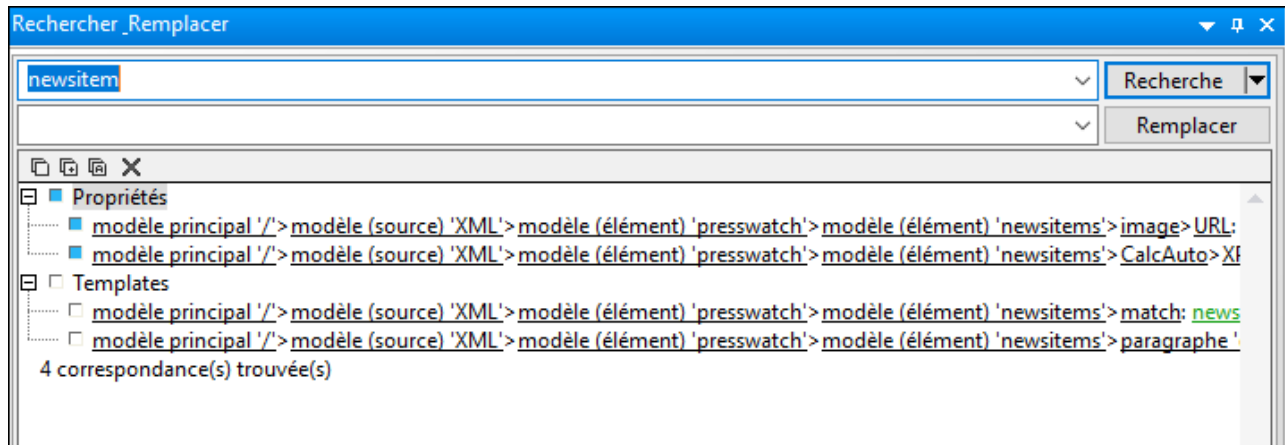


La barre latérale des Messages affiche des avertissements dans l'Aperçu de Design et Authentic View. Dans l'Aperçu de design, les avertissements sont liés à différents aspects associés au document SPS, d'un fichier de travail manquant aux erreurs dans la structure de design. Dans Authentic View, les avertissements concernent la validité des données XML saisies, qu'elles soient valides conformément au schéma sous-jacent ou selon le critère de validation supplémentaire.

2.2.8 Chercher et Remplacer

La barre latérale Chercher & Remplacer (*capture d'écran ci-dessous*) vous permet de chercher et remplacer du texte dans l'Aperçu de design. Cliquez sur la flèche vers le bas du bouton **Find** (*marque en bleu dans la capture d'écran ci-dessous*) pour sélectionner les options de recherche. Vous pouvez rechercher dans le texte, les styles, les propriétés, variables, correspondances de modèle, et expressions XPath pour les strings que vous saisissez directement dans le champ *Find* ou construisez avec les expressions régulières. Toutes les recherches réalisées dans cette barre latérale s'appliquent à l'Aperçu de design. Les commandes de menu [Éditer | Recherche](#) ⁴⁷⁶ et [Éditer | Remplacer](#) ⁴⁷⁶ définit le focus vers cette barre latérale et place le curseur dans le champ *Recherche*, vous permettant de procéder à une recherche dans l'Aperçu de design. Les résultats de la recherche sont affichés dans la barre latérale. Vous pouvez cliquer sur le résultat pour aller à

l'emplacement correspondant dans le design. Pour activer et désactiver la barre latérale Rechercher & Remplacer, cliquez sur **Aperçu | Rechercher & Remplacer**.



Pour des informations sur la recherche dans d'autres aperçus (Éditeur JavaScript et feuilles de style XSLT), voir le thème de référence du menu [Recherche. Trouver suivant. Remplacer](#)⁴⁷⁶.

Chercher

Saisir le terme que vous souhaitez chercher dans le champ *Rechercher*. Puis, cliquez sur la flèche pointant vers le bas du bouton **Recherche** (marqué en bleu dans la capture d'écran ci-dessus) pour sélectionner les options de recherche. Les options suivantes sont disponibles :

- *Où effecteur la recherche* : L'item respectif *Inclure <composant>* devrait être activé pour ce composant soit inclus dans la recherche.
- *Correspondances de la casse et/ou du mot entier* : Il s'agit d'options d'activation/désactivation.
- *Expressions régulières* : Votre entrée sera saisie comme expression régulière. Pour une description sur comment utiliser les expressions régulières, voir le sujet sur la référence de menu [Recherche. Trouver suivant. Remplacer](#)⁴⁷⁸.

Résultats

Les résultats sont organisés en groupes conformément au composant dans lequel le string correspondant apparaît (voir la capture d'écran ci-dessus). Chaque item de résultat est affiché comme chemin hiérarchique. Vous pouvez cliquer sur un lien quelconque pour aller à ce item dans l'Aperçu de design.

Le volet Résultats a une barre d'outils avec des icônes pour les commandes suivantes, de gauche : copier un item ou un groupe d'items dans le presse-papiers ; supprimer le volet de résultats.

Remplacer

Une fois que les résultats sont affichés, vous pouvez sélectionner un ou plusieurs items de résultats pour le/les remplacer. Le/s item/s sélectionnés seront indiqués par une puce bleue (voir la capture d'écran ci-dessus) et le bouton **Remplacer** sera activé. Saisissez le string de remplacement. Dans la case de texte *Remplacer* et cliquez sur **Remplacer**. Le remplacement est effectué et la puce bleue devient une puce verte.

3 Tutoriel Démarrage rapide

L'objectif de ce tutoriel est de vous guider rapidement parmi les étapes clés en créant une SPS efficace. Il débute avec une section de création et de définition de la SPS, vous montre comment insérer du contenu dans la SPS, comment formater les composants de la SPS, et comment utiliser deux fonctions SPS puissantes : Calculs automatiques et conditions. Tout au long du tutoriel, vous apprendrez comment structurer votre sortie de manière efficace et comment utiliser une variété de fonctions structurelles et de présentation.

Fichiers requis

Les fichiers associés à ce tutoriel Démarrage tutoriel sont dans le [dossier \(Mes\) Documents](#) ²⁴, C :

`\Documents and Settings\<username>\My`

`Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\QuickStart:`

- `QuickStart.xsd`, le fichier de schéma XML sur lequel SPS est basée.
- `QuickStart.xml`, le fichier de travail XML, qui est la source de données affichée dans les aperçus de sortie.
- `QuickStart.sps`, qui est le fichier SPS terminé ; vous pouvez comparer le fichier SPS que vous créez avec ce fichier.
- `QuickStart.css`, qui est la feuille de style CSS externe utilisée dans ce tutoriel.
- `NewsItems.BMP`, un fichier d'image qui est utilisé dans SPS.

Le tutoriel

Commençons par le début du tutoriel et passons de section en section. Aussi, vous devriez ouvrir les fichiers XSD et XML avant de démarrer le tutoriel et jetez un œil sur leur structure et leurs contenus. Gardez les fichiers XSD et XML ouverts en suivant le tutoriel, pour que vous puissiez vous y référer. Enregistrez votre document SPS avec un nom autre que `QuickStart.sps` (disons `MyQuickStart.sps`) pour ne pas écraser le fichier SPS fourni. Et, bien sûr, n'oubliez pas d'effectuer l'enregistrement avec succès après conclusion de chaque partie.

3.1 Créer un nouveau fichier SPS

Dans cette section, vous allez apprendre à :

- [Créer nouveau document <% SPS>](#) ⁵⁵
- [Ajouter une source de schéma pour SPS](#) ⁵⁷
- [Sélectionner une version XSLT de la SPS](#) ⁵⁸
- [Assigner le fichier de travail XML](#) ⁵⁸
- [Spécifier l'encodage de sortie](#) ⁵⁸
- [Enregistrer le document SPS](#) ⁵⁹

Fichiers dans cette section

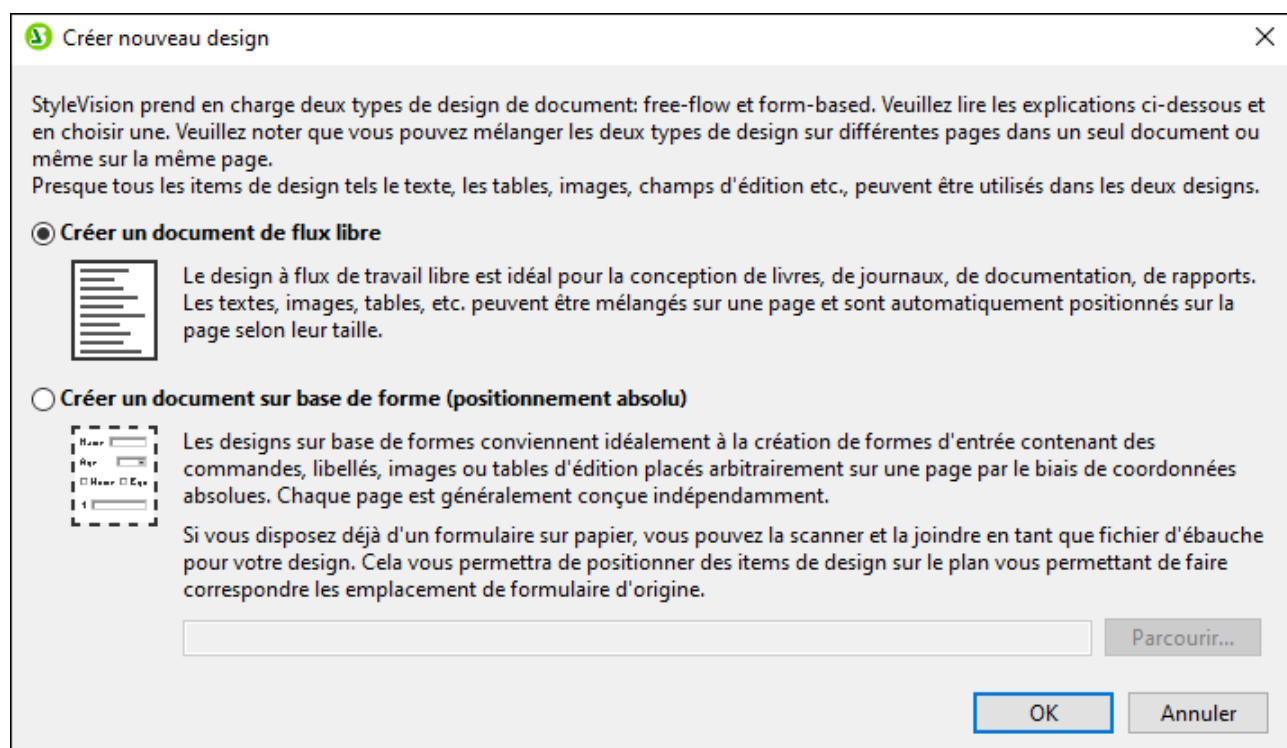
Les fichiers auxquels est fait référence dans cette section se trouvent sous le [dossier \(Mes\) Documents](#) ²⁴, C : \Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\QuickStart:

- QuickStart.xsd, le fichier de schéma XML sur lequel SPS est basée.
- QuickStart.xml, le fichier de travail XML, qui est la source de données affichée dans les aperçus de sortie.
- QuickStart.sps, qui est le fichier SPS terminé ; vous pouvez comparer le fichier SPS que vous créez avec ce fichier.

Créer un nouveau document SPS

Créer un nouveau document SPS en cliquant sur le [Fichier | Nouveau | Nouveau \(vide\)](#) ⁴⁵² ou sélectionnez **Nouveau (vide)**  dans la liste déroulante de la [nouvelle icône](#) ⁴⁵⁰  de la barre d'outils de l'application. Le dialogue Créer Nouveau design apparaît.

Le dialogue Créer nouveau design (*capture d'écran ci-dessous*) vous invite à sélectionner soit : (i) un design de document libre, soit (ii) un design de document basé sur un formulaire (dans lequel les composants sont positionnés de manière absolue, comme dans un programme de mise en page).

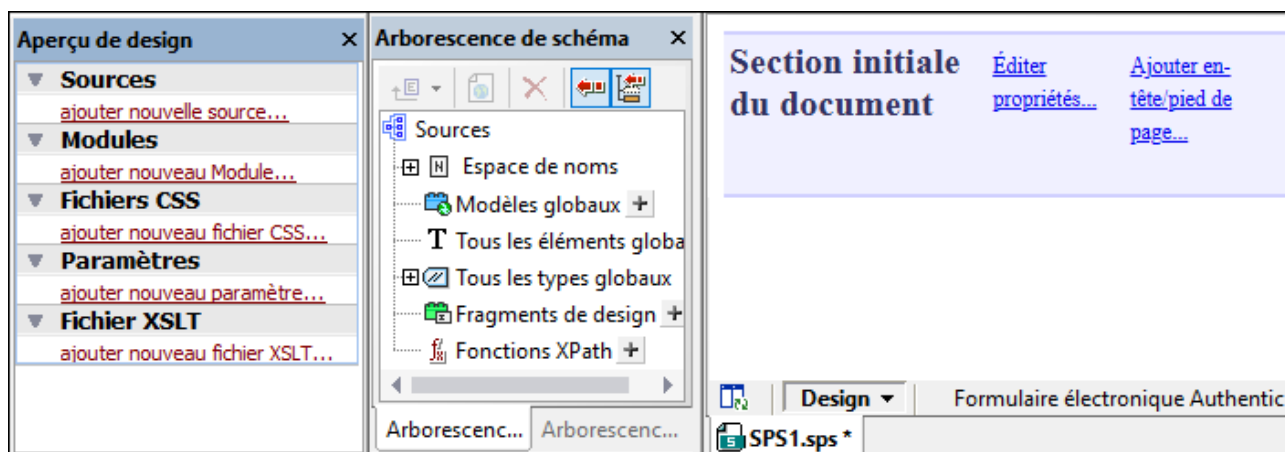


Dans une conception de document libre, le contenu du document est mis en page pour s'adapter à l'objet ou au visualiseur du support de sortie (papier ou écran). Les éléments du contenu du document ne peuvent être placés que les uns par rapport aux autres, et non de manière absolue. Ce type de conception convient aux documents tels que les rapports, les articles et les livres.

Dans un document basé sur un formulaire, un seul [conteneur de mise en page](#)¹⁷¹ est créé, dans lequel les composants de conception peuvent être positionnés de manière absolue. Les dimensions du conteneur de mise en page sont définies par l'utilisateur, et les boîtes de mise en page peuvent être positionnées de manière absolue dans le conteneur de mise en page et le contenu du document peut être placé dans des boîtes de mise en page individuelles. Si vous souhaitez que la conception de votre SPS reproduise un design spécifique basé sur un formulaire, vous pouvez utiliser une image du formulaire original comme [image de référence](#)¹⁷¹. L'image de référence peut ensuite être incluse comme image d'arrière-plan du conteneur de mise en page. L'image de référence est utilisée pour vous aider à concevoir votre formulaire ; elle ne sera pas incluse dans la sortie.

Vous créez un document « free-flowing », sélectionnez donc cette option en cliquant sur la case d'option *Créer un document « free-flow »*, puis cliquez sur **OK**.

Un nouveau document intitulé `SPS1.sps` est créé et affiché dans l'[Aperçu de design](#)²⁸ (capture d'écran ci-dessous).

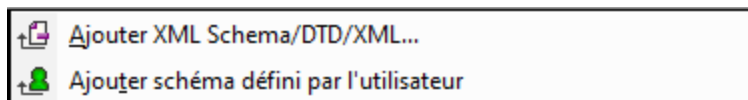


Dans l'[Aperçu de design](#) ²⁸, un modèle principal vide est affiché. Dans les barres latérales de la [présentation de design](#) ³⁵ et l'[arborescence de schéma](#) ³⁷, il n'existe pas d'entrées de schéma.

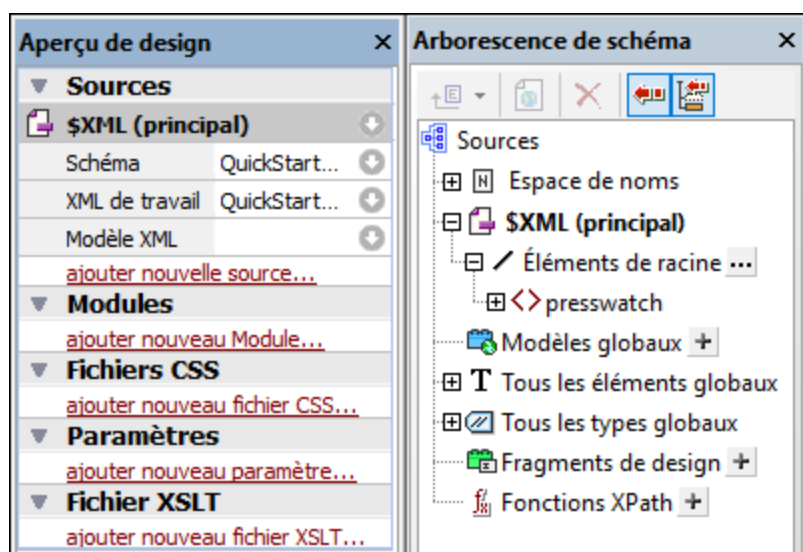
Ajouter une source de schéma

Pour cette SPS, vous utiliserez le schéma `QuickStart.xsd`. Pour ajouter ce schéma comme source de schéma, suivez les étapes suivantes :

1. Dans la barre latérale de l'Aperçu de design, sous l'en-tête Sources, cliquez sur la commande **Ajouter Nouvelle Source** (capture d'écran ci-dessus). Dans le menu qui apparaît (capture d'écran ci-dessous), sélectionnez **Ajouter Schéma XML/DTD/XML**.



2. Dans le dialogue Ouvrir qui apparaît pour le fichier `QuickStart.xsd` dans le [dossier \(Mes\) Documents](#) ⁵⁵ (voir ci-dessus), et cliquez sur **Ouvrir**.
3. Vous serez invité à sélectionner un fichier de travail XML. Sélectionnez l'option pour sélectionner le fichier depuis le « filesystem », puis parcourez pour trouver le fichier `QuickStart.xml` dans le [dossier \(Mes\) Documents](#) ⁵⁵ (voir ci-dessus) et cliquez sur **Ouvrir**. Le schéma sera ajouté comme source de schéma dans la barre latérale de l'Aperçu du design et dans la barre latérale de l'arborescence de schéma (voir la capture d'écran ci-dessous). Aussi, dans l'Aperçu de design, le fichier de travail XML que vous choisirez sera assigné au schéma.



Vous devez noter les points suivants : (i) dans l'Aperçu du design, l'entrée \$XML pour la source de schéma recense le schéma et le [fichier de travail XML](#) ²³ ; (ii) dans la barre latérale de l'arborescence de schéma, l'arborescence des éléments racine recenserait un ou plusieurs [éléments racine \(éléments du document\)](#) ²² vous sélectionnez depuis des [éléments globaux](#) ²² définis dans le schéma. Dans le cas de ce schéma, l'élément `presswatch` est sélectionné par défaut car l'[élément global](#) ²² dans le schéma qui se trouve clairement au niveau supérieur de la hiérarchie définie dans le schéma ; (iii) Tous les [éléments globaux](#) ²² dans le schéma sont recensés dans l'[arborescence de « Tous les Éléments globaux »](#) ³⁷.

Sélectionner la version XSLT

Pour cette SPS, vous utiliserez XSLT 2.0. Pour spécifier la version XSLT, dans la barre d'outils de l'application, cliquez sur l'icône .

Assigner ou changer le fichier de travail XML

En ajoutant le schéma XML à la SPS dans l'étape précédente, vous avez aussi assigné un [fichier de travail XML](#) ²³ au schéma. Un fichier de travail XML fournit la SPS avec une source de données XML à traiter. Pour assigner, changer ou désassigner un [fichier de travail XML](#) ²³ pour un schéma donné, dans la barre latérale de l'Aperçu du design, cliquez avec la touche de droite partout dans la ligne du fichier de travail XML que vous souhaitez modifier (ou cliquez sur l'icône du menu contextuel  à droite), et sélectionnez la commande requise depuis le menu contextuel qui apparaît. Le [fichier de travail XML File](#) ²³ est désormais assigné et le nom de fichier est saisi dans l'Aperçu du design. Avant le traitement, assurez-vous d'avoir correctement attribué le fichier `QuickStart.xml`, qui se trouve dans le [dossier \(Mes\) Documents](#) ⁵⁵, comme le fichier de travail XML.

Spécifier l'encodage de la sortie

Dans l'onglet d'encodage par défaut du dialogue Options ([Outils | Options](#) ⁵³⁵), définissez l'encodage HTML à Unicode UTF-8.

Enregistrer le document SPS

Après avoir défini la SPS telle que décrite ci-dessus, enregistrez-la comme `MyQuickStart.sps` dans le [dossier \(Mes\) Documents](#) ⁵⁵. Faites ceci via la commande de menu [Fichier | Enregistrer Design](#) ⁴⁶³ ou **Ctrl+S**.

Dans le dialogue Enregistrer Design qui apparaît, sélectionnez l'option *Enregistrer comme SPS*, et saisissez le nom du fichier SPS à enregistrer.

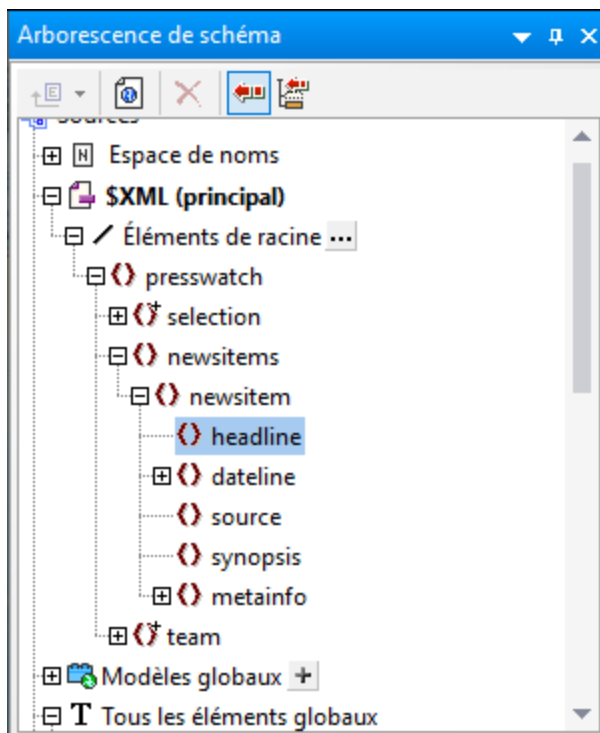
3.2 Insérer du contenu dynamique (depuis la Source XML)

Cette section introduit les mécanismes pour insérer les données depuis des nœuds dans le document XML. À l'intérieur de celle-ci, vous apprendrez à glisser des nœuds élément et attribut depuis l'arborescence de schéma dans le design et créer ces nœuds comme contenus. Quand un nœud est créé comme contenu, les données contenues sont sa sortie comme string qui est la concaténation du contenu de ces nœuds de texte enfant de l'élément et les nœuds texte de ses éléments descendant.

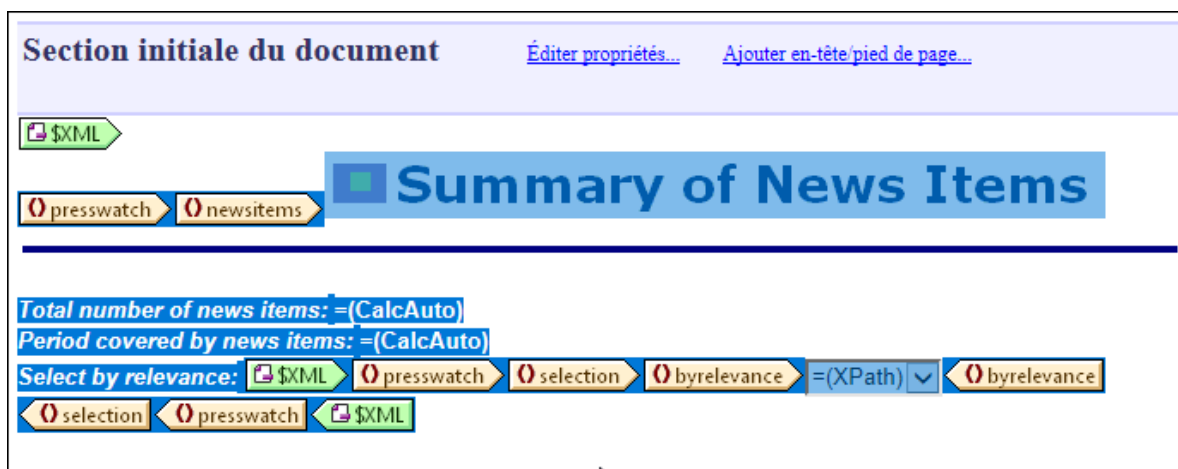
Insérer les contenus d'élément

Dans votre SPS, faites la chose suivante :

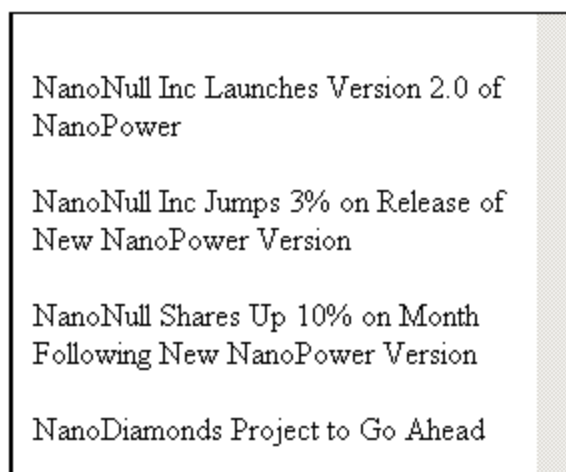
1. Dans la [barre latérale de l'arborescence de schéma](#)³⁷, élargissez l'arborescence de schéma jusqu'aux enfants de l'élément `newsitem` (voir la capture d'écran ci-dessous).



2. Sélectionnez l'élément `headline` (notez que le type de données de l'élément est affiché dans une pop-up quand vous passez au-dessus avec la souris ; voir la capture d'écran ci-dessus). Glissez l'élément dans l'[Aperçu de design](#)³² et, quand la flèche du curseur se tourne vers un point d'insertion, déposez-le dans le modèle principal.
3. IDans le menu contextuel qui s'ouvre, sélectionnez **Créer Contenus**. Les balises de début et de fin de l'élément `headline` sont insérées au point où vous avez déposé l'élément `headline`, et ils contiennent l'espace réservé le contenu. Les balises `headline` sont enveloppées par les balises de début et de fin des éléments ancêtre de `headline` (capture d'écran ci-dessous).
4. Dans les éléments « put » du design sur différentes lignes (en appuyant **Enter**) tel qu'affiché dans la capture d'écran ci-dessous.



Cliquez sur la balise HTML pour voir un [aperçu de la sortie HTML](#) ³⁰ (*capture d'écran ci-dessous*). L'aperçu HTML affiche les contenus des éléments enfant `headline` de `newsitem`, chacun comme string de texte.

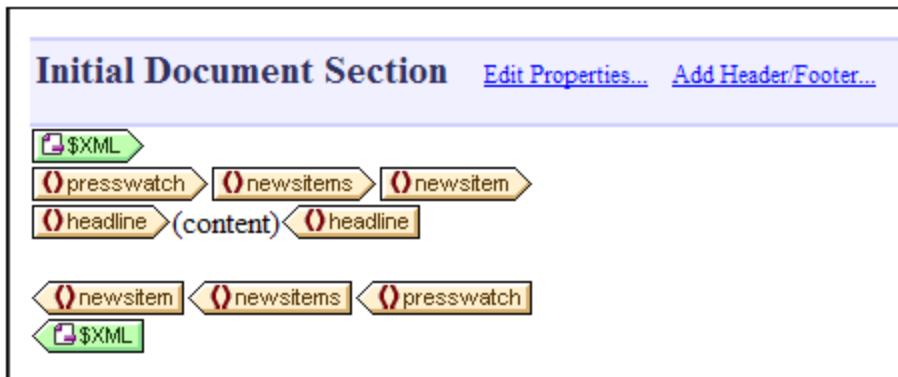


Note : vous pouvez aussi créer les contenus d'un nœud en suivant les étapes suivantes : (i) Cliquez sur l'icône des Contenus Insérer dans [insérer la barre d'outils des Éléments de design](#) ⁴⁴⁷, (ii) cliquez sur l'emplacement dans le design là où vous souhaitez insérer le contenu, (iii) sélectionnez depuis l'arborescence du Sélecteur de schéma qui apparaît, le nœud pour lequel vous souhaitez créer des contenus.

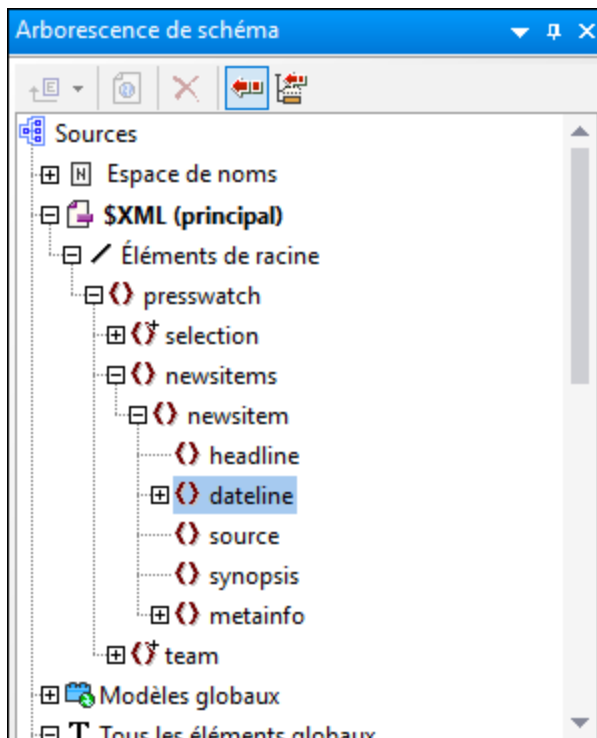
Insérer les contenus d'attribut

Quand un élément est inséré dans le design comme contenus, les contenus de ses attributs ne sont pas insérés automatiquement. Vous devez glisser le nœud attribut dans le design de la valeur de l'attribut pour qu'il soit le résultat. Dans votre SPS, faites la chose suivante :

1. Placez le curseur à la fin de la balise de fin de l'élément `headline` et appuyez sur **Enter**. Ceci produit une ligne vide (*capture d'écran ci-dessous*).

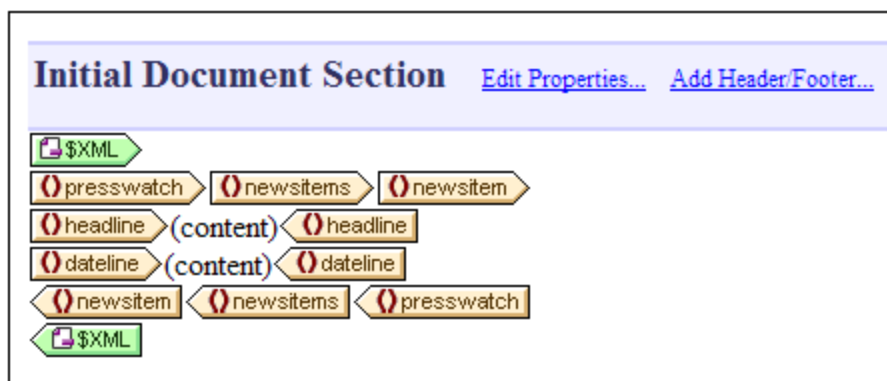


2. Dans la barre latérale de l'arborescence de schéma, élargissez l'élément `dateline` (capture d'écran ci-dessous).



Notez que l'élément `dateline` a deux éléments enfant, `date` et `place`, et que l'élément `place` a deux attributs, `city` et `country`.

3. Glissez l'élément `dateline` dans le design et déposez-le au début de la ligne vide nouvellement créée (capture d'écran ci-dessous).

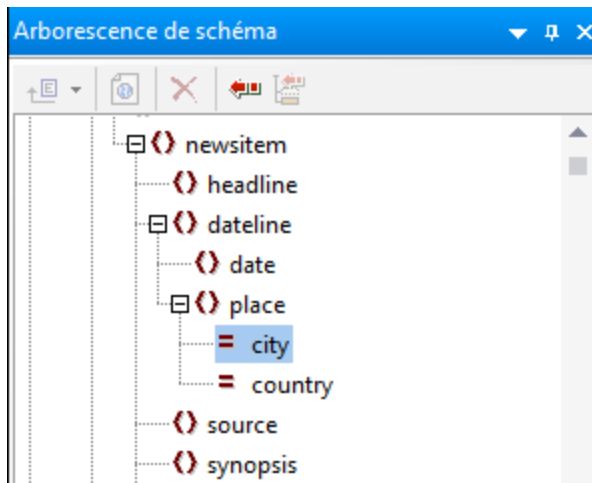


4. Basculez vers l'[Aperçu HTML](#) ³⁰ et regardez attentivement la sortie de `dateline` (*capture d'écran ci-dessous*).



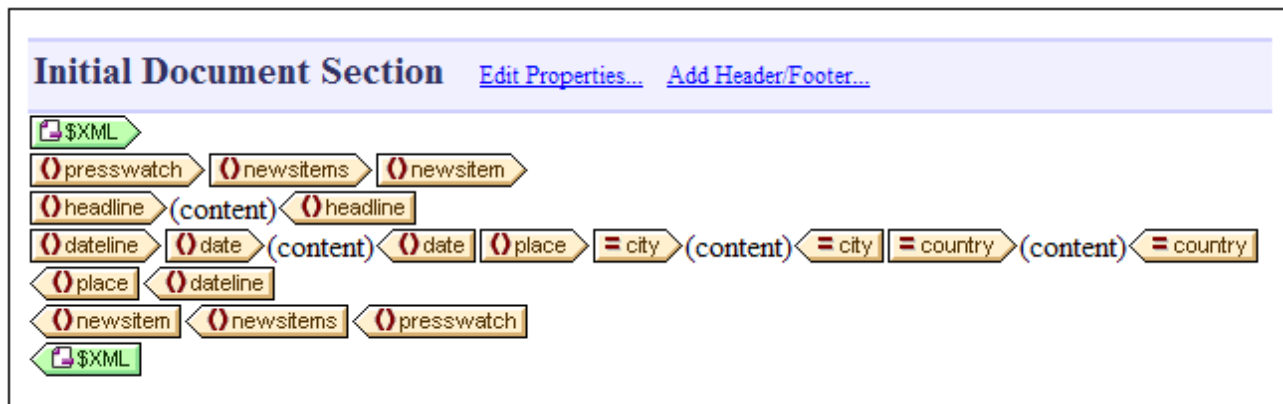
Notez que tandis que les contenus des enfants `date` des éléments `dateline` sont des sorties, aucun contenu n'est une sortie pour l'enfant `place` de `dateline`. Ceci est dû au fait que les données `place` sont contenues dans les attributs de l'élément `place` (dans les attributs `city` et `country`) et les contenus d'attribut ne pas des sorties quand l'élément parent de l'attribut est traité.

5. Glissez l'élément `date` de la [barre latérale de l'arborescence de schéma](#) ³⁷ et déposez-le (créez-le comme contenus) entre les balises de début et de fin de l'élément `dateline`.
6. Sélectionnez l'attribut `city` de l'élément `dateline/place` (*capture d'écran ci-dessous*) dans la [barre latérale de l'arborescence de schéma](#) ³⁷.



7. Glissez le nœud attribut @city dans l'[Aperçu de design](#) ²⁸, et déposez-le (créer comme contenus) juste après la balise de fin de l'élément date.
8. Glissez le nœud attribut @country dans l'[Aperçu de design](#) ²⁸, et déposez-le (créer comme contenus) juste après la balise de fin de l'attribut @city.

Quand vous avez terminé, le design SPS devrait avoir l'aspect suivant :



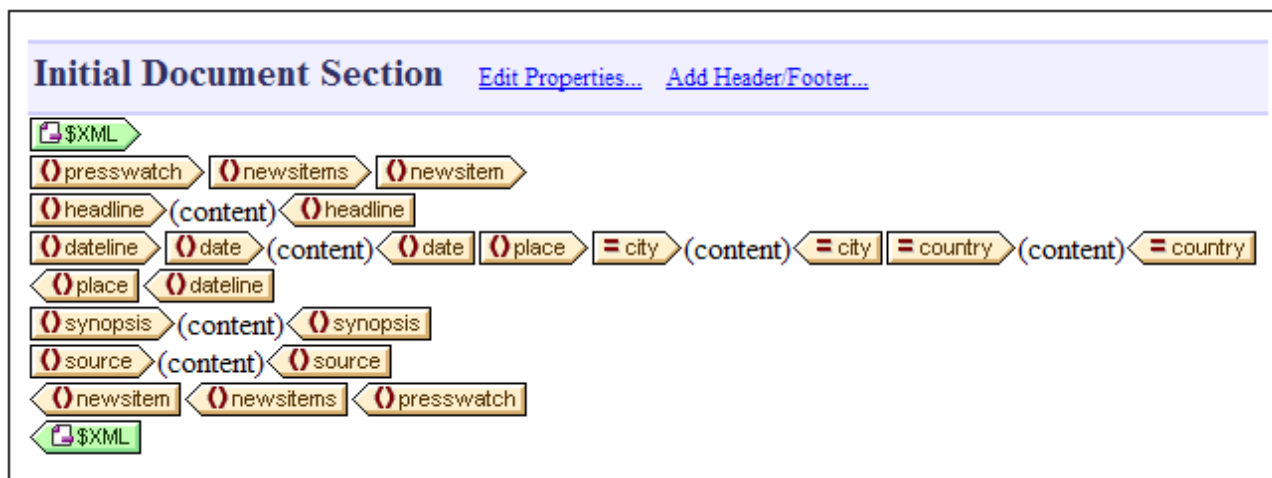
L'[Aperçu HTML](#) ³⁰ aura lieu de ceci :



Notez que les valeurs des attributs @city et @country sont désormais incluses dans la sortie.

Ajouter plus de contenu dynamique

Les contenus des éléments et attributs de la source de données XML peuvent être insérés partout dans le design utilisant la méthode décrite ci-dessus. Pour compléter cette section, ajoutez les éléments `synopsis` et `source` au design pour que le design ait l'aspect suivant :



Notez que l'élément `synopsis` a été placé avant l'élément `source`, qui n'est pas l'ordre dans lequel les éléments se trouvent dans le schéma. Après avoir ajouté les éléments `synopsis` et `source` au design, vérifiez l'[aperçu HTML](#) ⁽³⁰⁾ pour voir la sortie. Veuillez noter les points suivants importants : L'ordre dans lequel les nœuds sont placés dans le [modèle principal](#) ⁽²³⁾ est l'ordre dans lequel ils apparaîtront dans la sortie (voir la section, [Modèles et Fragments de design](#) ⁽²³⁴⁾, pour plus d'information sur la structure du document de sortie).

Un autre point important à noter à cette étape est la manière dont un nœud est créé dans le design. Dans l'[Aperçu HTML](#) ⁽³⁰⁾, vous verrez que tous les nœuds inclus dans le design ont été envoyés dans la sortie comme strings de texte. Alternativement à la sortie comme string de texte, un nœud peut être une sortie sous

une autre forme, par exemple, comme table ou comme zone de liste. Dans cette section, en créant tous les nœuds comme `(contenus)`, spécifiez que le format de sortie de tous les nœuds sont des strings de texte. Dans la section [Utiliser les Conditions](#)⁶², vous apprendrez comment créer un nœud dans la zone de liste modifiable, et dans la section, [Utiliser les modèles globaux et « Reste-de-Contenus »](#)⁹⁰, comment créer un nœud comme table (dynamique).

Veillez à enregistrer le fichier avant de basculer vers la prochaine section.

3.3 Insérer le Contenu statique

Le contenu statique est un contenu que vous saisissez ou insérez directement dans le design—comme opposé au contenu qui vient de la source XML. Une variété de composants statiques peut être placée dans un design SPS. Dans cette partie du tutoriel, vous apprendrez comment insérer les composants statiques suivants :

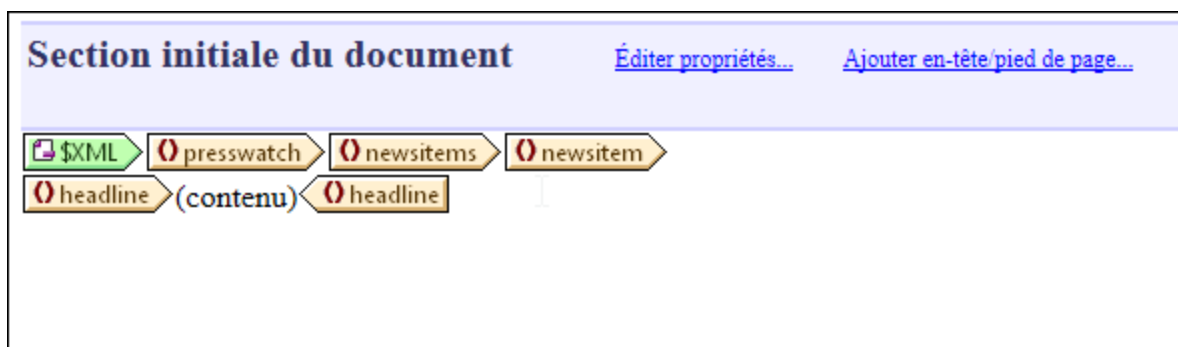
- [Une image](#) ⁶⁷
- [Une ligne horizontale](#) ⁶⁸
- [Texte](#) ⁶⁹

Insérer une image statique

L'image statique à insérer est dans le [dossier \(Mes\) Documents](#) ²⁴ : C:\Documents and Settings\<username>\My

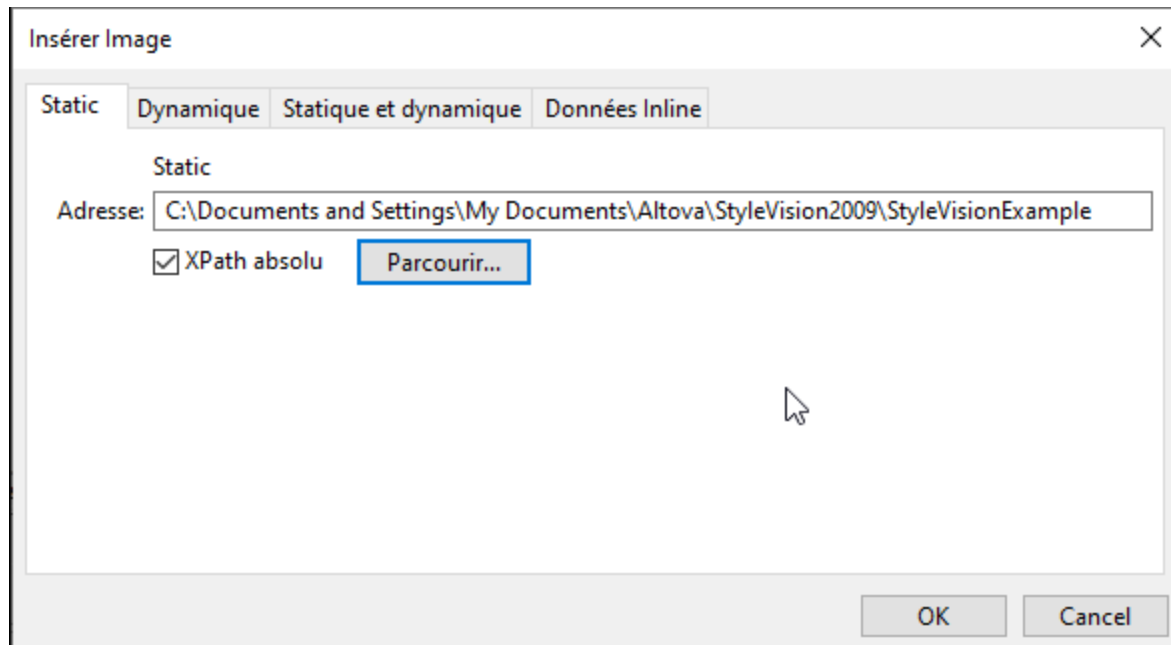
Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\QuickStart\NewsItems.BMP. Elle sera utilisée comme en-tête du document. Pour insérer cette image dans l'en-tête du document, suivez les étapes suivantes :

1. Placez le curseur entre les balises de début de `newsitems` et `newsitem` (*capture d'écran ci-dessous*).



Notez que le curseur est au sein de l'élément `newsitems`, mais à l'extérieur de l'élément `newsitem`. Il sera pour cela inséré dans la sortie une fois, au début du traitement de l'élément `newsitems` (car il existe un élément `newsitems` défini dans le schéma).

2. Cliquez avec la touche de droite et sélectionnez [Insérer | Image](#) ⁴⁹². Le dialogue Insérer Image s'ouvre (voir la *capture d'écran ci-dessous*).



3. Dans la barre statique, cliquez sur la case à cocher Chemin absolu, puis recherchez le fichier `NewsItems.BMP` et sélectionnez-le.
4. Cliquez sur **OK** pour terminer.

Le fragment HTML ressemblera à l'exemple ci-dessous :



Insérer les lignes horizontales

La première ligne horizontale que vous allez insérer se trouve entre l'en-tête du document et le corps du document. Procédez comme suit :

1. Placez le curseur immédiatement après avoir inséré l'image statique.
2. Cliquez avec la touche de droite et sélectionnez **Insérer | Ligne horizontale** ⁴⁹⁵. Une ligne horizontale est insérée.

Définir les propriétés pour la ligne comme suit :

1. Avec la ligne sélectionnée dans l'[Aperçu de design](#) ²⁸, dans la [barre latérale des Propriétés](#) ⁴⁸, sélectionnez le composant *ligne* (dans les « Properties For column »), puis le groupe de propriétés *HTML*.
2. Assignez les propriétés de *couleur* et de *taille* pour la ligne.
3. Avec la ligne sélectionnée dans l'[Aperçu de design](#) ²⁸, dans la [barre latérale des Propriétés](#) ⁴⁶, sélectionnez le composant *ligne* (dans les « Properties For column »), puis le groupe de propriétés *déroulantes*. Définir une propriété *margin-bottom* de 12pt.
4. Vérifiez la sortie dans l'[Aperçu HTML](#) ³⁰.

Insérez maintenant une ligne horizontale à la fin de chaque newsitem. À cette fin, le curseur devrait être placé immédiatement avant la balise de fin de l'élément `newsitem`. Ceci fait que la ligne devient la sortie à la fin de chaque élément `newsitem`. Vous pouvez changer l'épaisseur de la propriété de la ligne en définissant la propriété *taille* de la ligne à un nombre sans unité (dans la barre latérale des Propriétés, sélectionnez *ligne*, et définissez une valeur, disons 3).

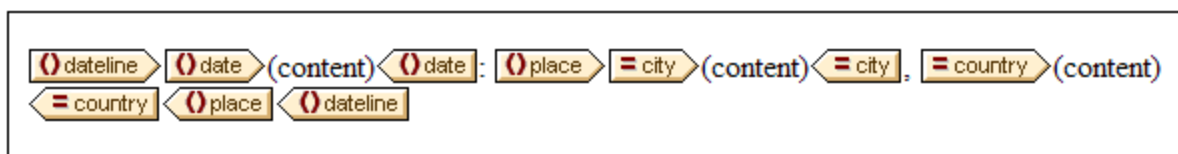
Insérer le texte statique

Vous avez déjà ajouté du texte statique à votre design. Quand vous avez appuyé sur la clé **Enter** pour obtenir de nouvelles lignes (dans la section [Insérer Contenu dynamique \(depuis la Source XML\)](#) ⁶⁰), un espace (texte statique) a été ajouté. Dans cette section, vous ajouterez quelques caractères de texte statique à votre design.

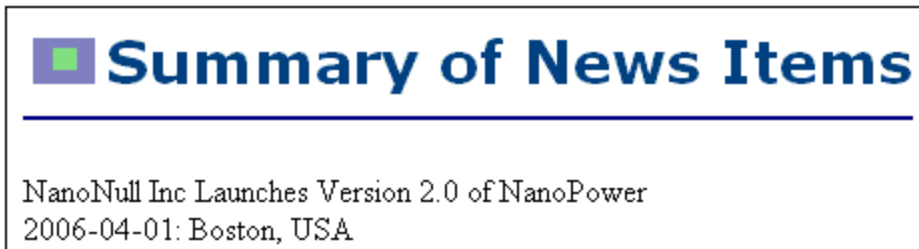
La SPS que vous avez conçue jusqu'ici produira une sortie qui aura l'aspect suivant :



Notez que dans la sortie de l'élément `dateline`, les contenus de l'élément `date` et les attributs `place/@city` et `place/@country` sont exécutés ensemble sans espace. Vous pouvez ajouter l'espace comme texte statique. Dans le design, placez le curseur après l'élément `date` et saisissez deux points et un espace. Ensuite, saisissez une virgule et un espace après l'attribut `@city` (voir la capture d'écran ci-dessous)

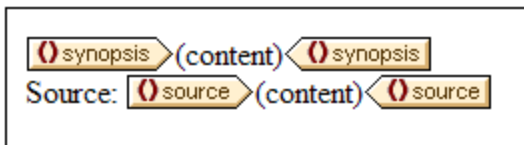


Cette partie de la sortie aura l'aspect suivant :



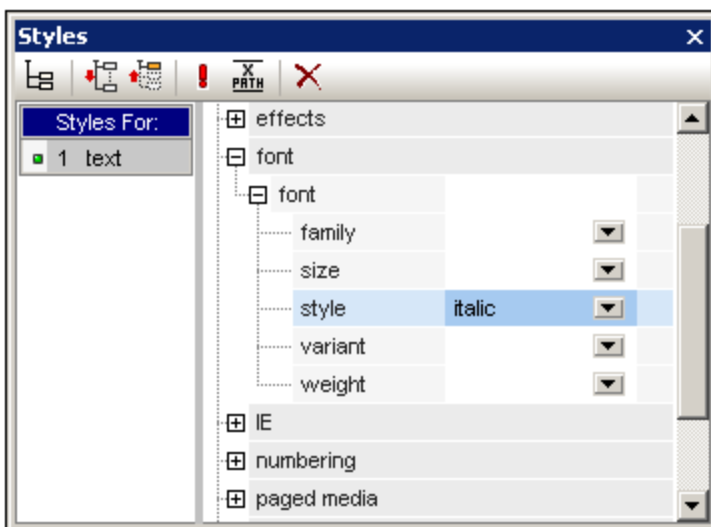
Notez les deux points, l'espace et la virgule dans la sortie `dateline`. Tous ces items de texte sont des items de texte statique qui ont été insérés directement dans le design.

Vous ajouterez maintenant un item de plus de texte statique. Dans le design, tapez le string "Source : " juste avant la balise de début de l'élément `source` (*capture d'écran ci-dessous*).

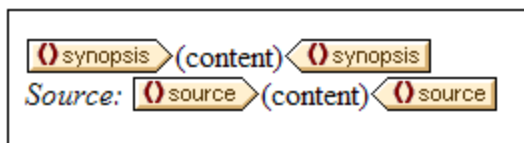


Formater le texte statique

Pour formater le texte statique, mettez le texte à formater en surbrillance et spécifiez les propriétés de style local. Dans le design, mettez le texte en surbrillance "Source : " que vous venez de saisir. Dans la [barre latérale de Styles](#) ⁴⁶ (*capture d'écran ci-dessous*), notez que le composant *1 text* est sélectionné. Élargissez maintenant le groupe de propriétés *font* tel que choisi dans la capture d'écran ci-dessous, et, pour la propriété *font-style*, sélectionnez l'option *italic* depuis le menu de la liste déroulante.



Le texte statique (qui est le string "Source:") sera en italique dans le design et aura l'aspect suivant :



La sortie aura l'aspect suivant dans l'Aperçu HTML :



Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower
2006-04-01: Boston, USA
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.
Source: NewTech Online

Si vous pensez qu'il n'y a pas assez d'espace vertical entre l'item source et la ligne horizontale séparant deux éléments `newsitem`, alors, dans le design, insérez une ligne vide entre la ligne source et horizontale (en appuyant sur **Enter**).

Une fois que avez terminé, enregistrez le fichier.

Dans cette section, vous avez appris comment insérer le contenu statique et comment le formater. Dans la prochaine section, vous apprendrez davantage sur la conception des composants et comment ils peuvent être formatés utilisant des principes et propriétés CSS.

3.4 Formater le contenu

StyleVision offre un [mécanisme de styling](#)³⁴³ puissant et flexible, basé sur CSS, pour formater les composants dans le design. Les aspects clés du mécanisme de style de StyleVision sont les suivants :

- Les règles de style CSS peuvent être définies pour les composants en bloc et inline.
- Les [formats prédéfinis](#)³²⁹ sont des composants en bloc qui ont des styles inhérents et peuvent être utilisés comme wrappers pour un groupe de composants qui doivent être traités comme bloc. Les styles inhérents de ces formats prédéfinis peuvent être écrasés par les styles que vous pouvez spécifier localement sur chaque composant. En maintenant le principe en cascade des CSS.
- Les attributs de classe peuvent être déclarés sur des composants dans le design et la classe peut être utilisée comme sélecteur de règles de style [externe](#)³⁴⁴ ou [global](#)³⁴⁸.
- Vous pouvez préciser les styles à trois niveaux. Ceux-ci sont, par ordre croissant de priorité : (i) les règles de style dans des [feuilles de style externes](#)³⁴⁴, (ii) les [règles de style global](#)³⁴⁸, et (iii) les [règles de style local](#)³⁵⁰.

Dans cette section, vous allez apprendre à :

- [Assigner des formats prédéfinis](#)⁷²
- [Assigner un composant comme attribut de classe](#)⁷³
- [Définir les styles dans une feuille de style CSS externe](#)⁷⁴ et ajoutez cette feuille de de style au référentiel de style de la SPS
- [Définir les règles de style global](#)⁷⁵
- Définir les [styles locaux pour une sélection de multiples composants design](#)⁷⁶
- Définir les [styles locaux pour une sélection de composant unique](#)⁷⁶

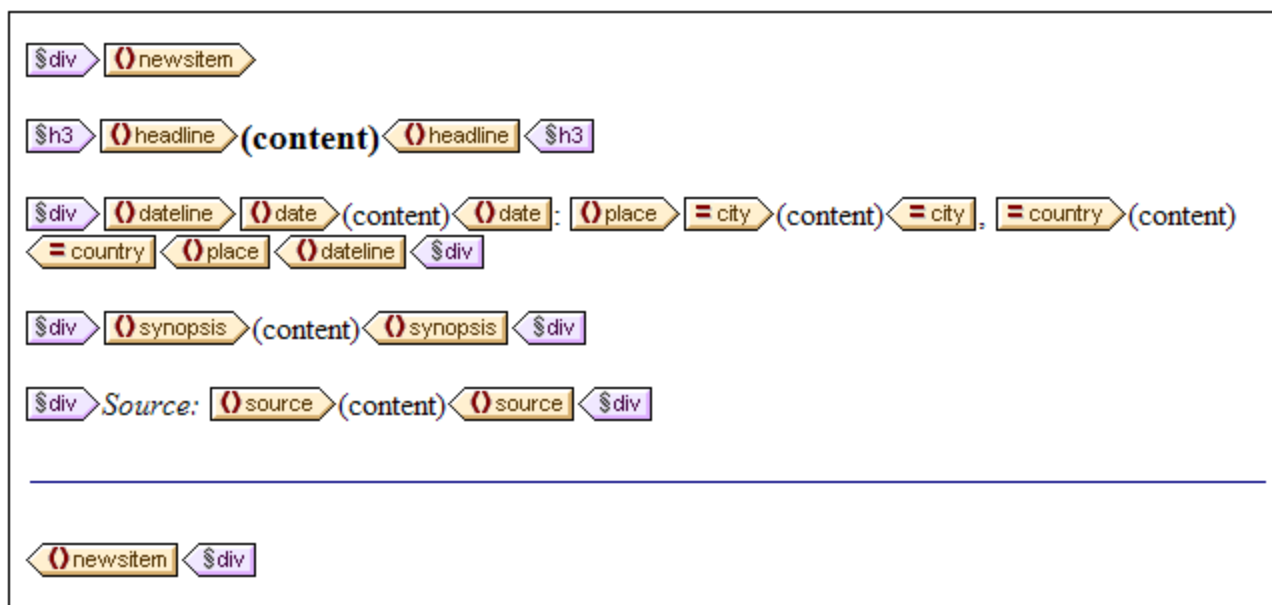
Assigner des formats prédéfinis

Une des raisons pour assigner un [format prédéfini](#)³²⁹ est de donner un composant au style inhérent de ce [format prédéfini](#)³²⁹. Dans le design, sélectionnez l'élément `headline`, puis sélectionnez **Enclose with | Special Paragraph | Heading 3 (h3)** (en alternative, utilisez la zone de liste Formats prédéfinis dans la barre d'outils). Les balises Format prédéfini sont créées autour de l'élément `en-tête` (*capture d'écran ci-dessous*).



Notez que les propriétés de police des contenus changent et que l'espace vertical est ajoutée au-dessus et sous le format prédéfini. Ces valeurs de propriété sont inhérentes dans le format prédéfini `h3`.

Une autre utilisation de formats prédéfinis est de grouper les composants de design en un bloc pour qu'ils puissent être formatés comme bloc ou attribués des propriétés inline comme groupe. La propriété prédéfinie la plus opportune à cette fin est le format prédéfini `div`, qui crée un bloc sans espace ci-dessus ou en-dessous. Dans votre design, attribuez `newsitem`, `dateline`, `synopsis`, et `source` aux composants séparés des nœuds `div`. Votre design devrait ressembler à la capture d'écran ci-dessous. Notez que le texte statique "Source: " est aussi inclus dans le composant `div` qui contient l'élément `source`, et que tout l'élément `newsitem` se trouve à l'intérieur du composant `div`.

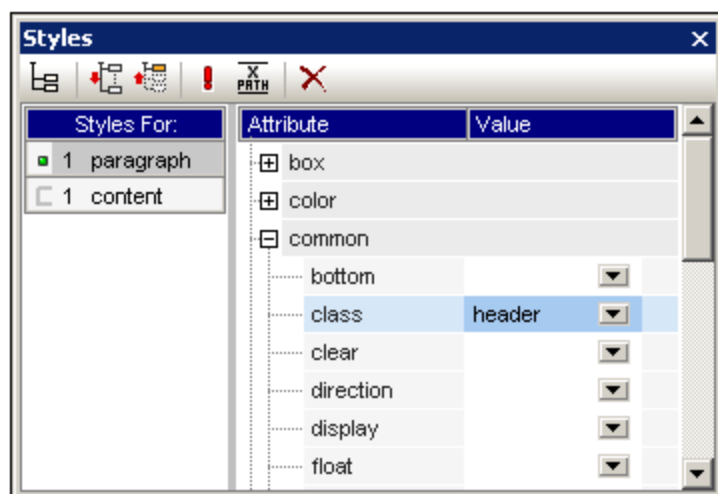


Vous avez désormais des composants groupés ensemble dans différents blocs.`div`. [Plus tard dans cette section](#) ⁷⁶, vous apprendrez comment attribuer les styles à de tels blocs de composants groupés.

Assigner les composants aux attributs de classe

Une règle de style peut être définie pour la classe de composants. Par exemple, tous les en-têtes peuvent être définis pour avoir un ensemble de propriétés communes (par exemple, une font-family, font-weight et couleur particuliers). À cette fin, vous devez faire deux choses : (i) attribuer les composants qui doivent avoir des propriétés communes dans une classe unique ; (ii) définir les propriétés de style pour cette classe.

Dans votre design, sélectionnez la balise `h3`, et, dans la barre latérale de Styles, sélectionnez *1 paragraph* (pour sélectionner le format prédéfini), et le groupe de propriétés *common*. Élargissez le groupe des propriétés *common*, puis double-cliquez sur le champ de valeur de la propriété `class` et saisissez `header`.



Cette instance particulière du format `h3` est désormais assignée à la classe appelée `header`. Quand vous définissez des propriétés de styling pour la classe `header` (styles depuis une feuille de style externe ou style global SPS), ces propriétés seront appliquées à tous les composants dans la classe SPS `header`.

Ajoutez une feuille de style CSS externe au référentiel de style

Les règles de style dans une feuille de style CSS externe peuvent être appliquées aux composants dans le design SPS. Des feuilles de style externe doivent, toutefois, d'abord être ajoutées au référentiel de style pour que les règles puissent être appliquées aux composants. Dans la [barre de style du Référentiel Style](#)⁴⁴ (dans l'Aperçu de design), suivez l'étape suivante :

1. Sélectionnez l'item `External`.
2. Cliquez sur le bouton **Ajouter** dans la barre d'outils de la [barre latérale du Référentiel de style](#)⁴⁴. Le dialogue Ouvrir s'ouvre.
3. Parcourez pour le fichier `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\QuickStart\QuickStart.css`, qui est dans le [dossier \(Mes\) Documents](#)²⁴, et cliquez sur **Ouvrir**.

La feuille de style est ajoutée au référentiel de style. Elle contient les règles suivantes qui sont pertinentes à cette étape.

```
.header {
    font-family: "Arial", sans-serif;
    font-weight: bold;
    color: red;
}

h3 {
    font-size: 12pt;
}
```

Les règles de style pour la classe `header` et l'élément `h3` sont combinées et produisent la même sortie HTML suivante pour l'élément `headline`.

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA

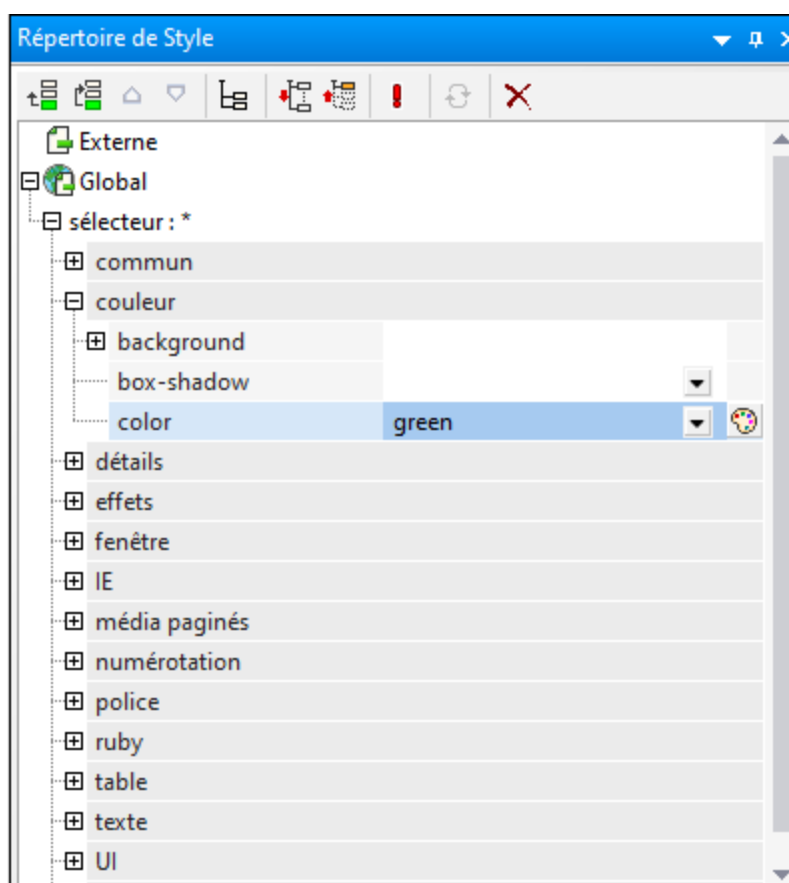
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

Définir les règles de style global

Les [règles de style global](#) ³⁴⁸ peuvent être définies pour SPS utilisant les sélecteurs CSS. Les règles sont définies directement dans la [barre latérale du Répertoire de style](#) ⁴⁴. Créer un style global pour la classe `header` comme suit :

1. Avec l'[Aperçu de design](#) ²⁸ actif, dans la [barre latérale du Répertoire de style](#) ²⁸, sélectionnez l'item global.
2. Cliquez sur le bouton **Ajouter** dans la barre d'outils. Ceci crée une règle vide pour le sélecteur de caractère générique (*), qui est en surbrillance.
3. Tapez `.header` pour remplacer le caractère générique tel le sélecteur.
4. Élargissez le groupe de propriétés *couleur* et sélectionnez *vert* depuis la liste déroulante de valeurs de propriété *couleur* (*capture d'écran ci-dessous*).



Là où la règle de style global définit une propriété qui est également définie dans une feuille de style externe (la propriété *couleur*), la valeur de propriété dans la règle globale prend précedence. Dans l'aperçu HTML, les contenus de l'en-tête seront pour cette raison en vert. D'autres propriétés de définitions depuis une feuille de style externe (pas « over-riden » par une propriété dans une règle de style global) sont conservées (dans le cas, *font-family* et *font-weight*).

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

Définir des styles locaux pour de multiples composants en une fois

Les styles locaux peuvent être définis pour de multiples composants en une fois. Dans votre design, pour préciser que tout le contenu de texte d'un item « news » devrait avoir Arial comme sa police, cliquez sur le composant `div` contournant l'élément `newsitem` et, dans la [barre d'outils des Styles](#)⁴⁶, dans la colonne Styles For, sélectionnez 1 paragraph. Ensuite, dans le groupe des propriétés *font*, assignez Arial comme `font-family`. Le paramètre des propriétés sera hérité par les cinq formats prédéfinis descendant.

Maintenant, dans le design, sélectionnez les trois composants `div` contournant les nœuds `dateline`, `synopsis`, et `source` (en gardant la clé **Shift** appuyée quand vous cliquez sur chaque composant `div`). Dans la [barre latérale des Styles](#)⁴⁶, sélectionnez 3 paragraphs, puis le groupe de propriétés *font* et définissez une taille de police `font-size` of 10pt. (Le composant `h3` n'a pas été sélectionné car il a déjà la taille de police `font-size` requise 12pt.)

Enfin, dans le design, sélectionnez le composant `div` contournant l'élément `dateline`. Dans la colonne Styles For de la [barre latérale Styles](#)⁴⁶, sélectionnez 1 paragraph. Dans le groupe des propriétés *font*, définir `font-weight` vers **bold** et `font-style` vers *italic*. Dans le groupe de propriétés *color*, définir la couleur comme grise. La sortie de la dateline aura l'aspect suivant

2006-04-01: Boston, USA

Notez que le styling défini pour le composant `div` a été appliqué au texte statique également au sein du composant `div` (c'est-à-dire un point et la virgule).

Définir les styles locaux pour une sélection de composant unique

Un style local défini sur un composant écrase d'autres styles définis à des niveaux plus élevés de SPS pour ce composant. Dans le design, sélectionnez l'élément `headline` et attribuez-le dans une couleur bleue marine (propriété de `couleur` dans le groupe des propriétés de style de *couleur*). La propriété définie localement (`color:navy`) écrase le style global pour la classe `.header` (`color:green`).

Sélectionnez le composant `div` contournant l'élément `source`. Dans la [barre latérale des Styles](#)⁴⁶, avec l'item 1 paragraph dans la colonne Styles For sélectionnée, définissez la propriété de `couleur` (dans le groupe de *couleur* des propriétés de style) vers le gris. Dans le groupe de propriétés de style *font*, définissez `font-weight` comme **bold**. Ces valeurs sont appliquées au texte statique. Souvenez-vous que dans la dernière

section du texte statique "Source: " a été attribuée comme valeur `font-style` de `italic`. Ces nouvelles propriétés (`font-weight:bold` et `color:gray`) viennent en plus de la propriété `font-style:italic`.

Maintenant, dans l'Aperçu de design, sélectionnez l'espace réservé (contenu) de l'élément `source`. Dans la colonne Styles For, avec 1 *content* sélectionné, définissez la propriété `couleur` (dans le groupe de propriétés de style *color*) comme `black`. Dans le groupe de propriétés de *font*, définissez `font-weight` comme `normal`. Les nouvelles propriétés sont définies dans le nœud de l'espace réservé contenus de l'élément `source` et écrasez les propriétés dans le composant `div` (voir la capture d'écran ci-dessous).

Compléter le formatage

Pour compléter le formatage dans cette section, sélectionnez le composant `div` dans l'élément `synopsis` et, dans la zone de liste [Formats prédéfinis](#)³²⁹ dans la barre d'outils, sélectionnez `p`. Ceci donne au bloc les styles inhérents de l'élément HTML `p`. L'Aperçu HTML ressemblera à l'exemple ci-dessous :

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

Une fois que avez terminé, enregistrez le fichier.

3.5 Utiliser les calculs automatiques

Les [calculs automatiques](#) ²⁶¹ sont un mécanisme puissant pour fournir des informations supplémentaires depuis les données XML disponibles. Dans cette section, vous ajouterez deux informations au design : le nombre total d'items news et la période de temps couverte par les items news dans le document XML. Aucune pièce d'information n'est directement disponible dans le document XML mais doit être calculée ou manipulée depuis les données disponibles.

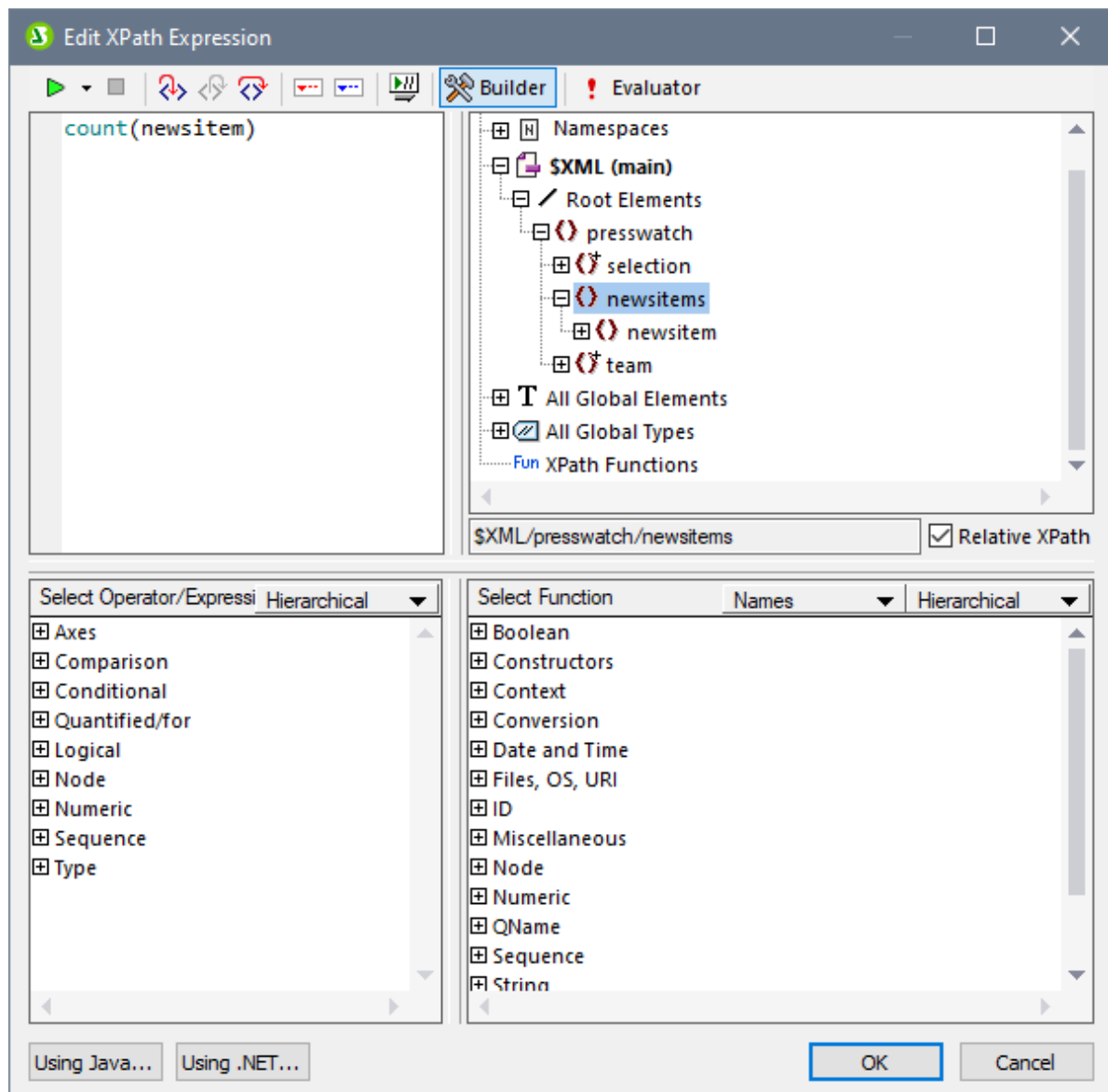
Compter les nœuds de l'item news

Dans le design, faites la chose suivante :

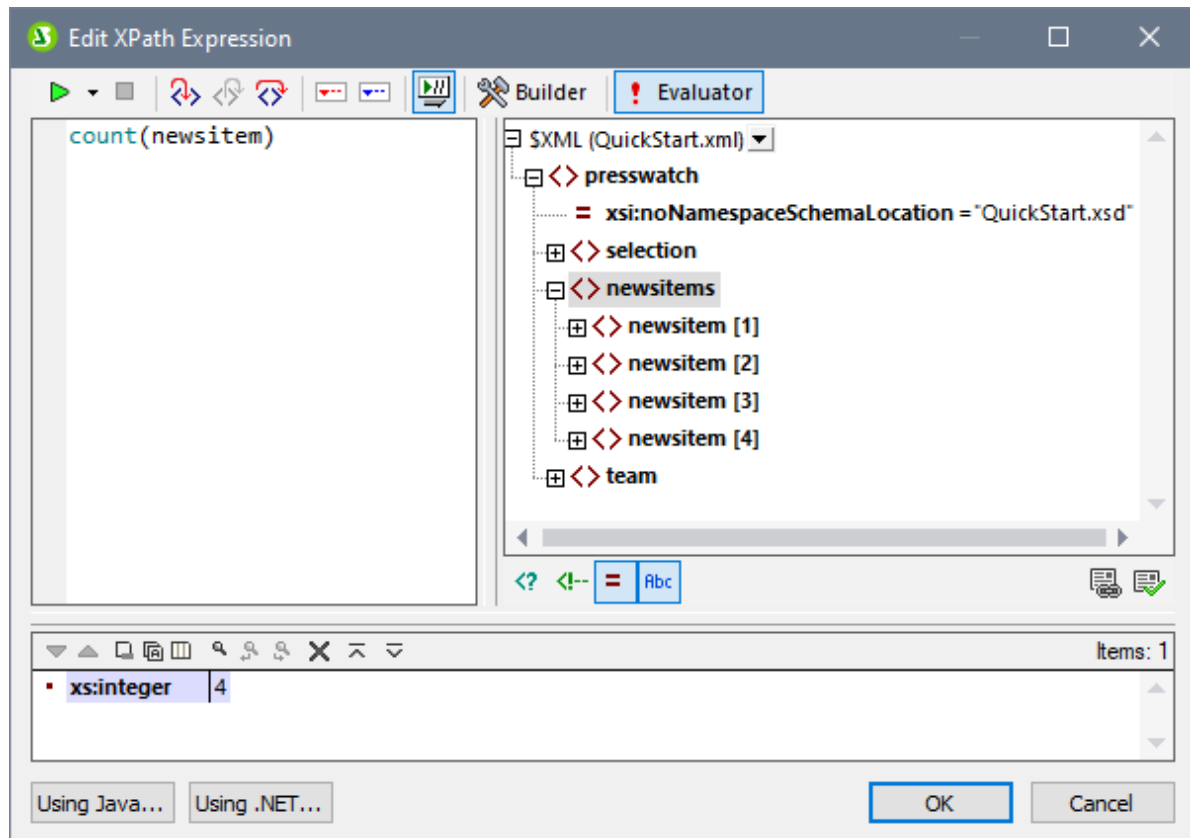
1. Créer un espace, tel qu'affiché dans la capture d'écran ci-dessous, pour une ligne de texte statique (sur laquelle le calcul automatique sera également placé). Utilisez la clé **Return** pour ajouter de nouvelles lignes et insérer une ligne horizontale en-dessous de l'espace que vous créez (*voir la capture d'écran*).



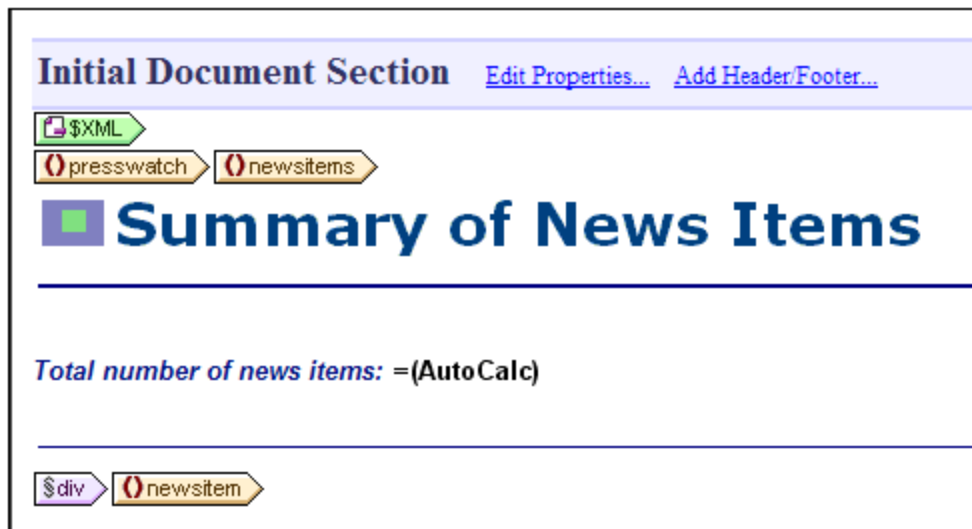
2. Tapez le texte statique "Nombre total d'items news: « tel qu'affiché dans la capture d'écran ci-dessus.
3. Appliquer le style local de votre choix au texte statique. Suivez l'étape tel que décrit dans la section [Formater le Contenu](#) ⁷⁶.
4. Placez le curseur après le point et sélectionnez **Insérer | Calcul automatique | Valeur**. Apparaîtra le [dialogue Éditer Expression XPath](#) ⁴²⁶ (*capture d'écran ci-dessous*). (En alternative, vous pouvez cliquer avec la touche de droite et sélectionner la commande dans le menu contextuel.)



5. Dans l'arborescence de schéma, notez que le nœud contextuel est `newsitems`, qui est en surbrillance. Maintenant, dans le champ de texte *Expression*, saisissez soit l'expression `count(newstitem)` ou créez l'expression utilisant les volets entry-helper en dessous du champ de texte *Expression*. (Double-cliquez sur la fonction `count` (trouvée dans le groupe de fonctions *Sequence*) pour la saisir, puis (dans l'expression dans le champ de texte) placez le curseur à l'intérieur des parenthèses de la fonction et double-cliquez sur le nœud `newsitem` dans l'arborescence de schéma. Vous pouvez voir ce que l'expression XPath renvoie en cliquant sur le bouton **Évaluateur**. Les résultats de l'évaluation seront affichés dans le volet *Résultats* (voir capture d'écran ci-dessus). Pour une description détaillée du dialogue Éditer l'expression XPath, voir la section [Éditer l'expression XPath](#) ⁴²⁶.



6. Cliquez sur **OK** pour terminer. Le calcul automatique est inséré dans le design à l'emplacement du curseur (*capture d'écran ci-dessous*). Formater le calcul automatique utilisant les [styles locaux](#) ⁷⁶.



Votre sortie HTML aura l'air de ceci :

Summary of News Items

Total number of news items: 4

Afficher la période couverte par les newsitems

La période couverte par tous les newsitems ensemble peut être obtenue en obtenant la date du newsitem la plus récente et la date des newsitems la plus ancienne. Ceci peut être obtenu avec des expressions XPath comme celles affichées ci-dessous. La première expression ci-dessous sort les contenus du nœud `date`. La deuxième expression est un raffinement, sortant juste les valeurs du mois et de l'année dans le nœud `date`. Vous pouvez utiliser les deux.

- `concat(min(//date), ' to ', max(//date)).`
- `concat(month-from-date(min(//date)), '/', year-from-date(min(//date)), ' to ', month-from-date(max(//date)), '/', year-from-date(max(//date)))`

Dans le design, insérez le texte statique et le calcul automatique tel qu'affiché dans la capture d'écran ci-dessous. Appliquez un style local que vous aimez.

Total number of news items: =(AutoCalc)
Period covered by news items: =(AutoCalc)

Le fragment HTML ressemblera à l'exemple ci-dessous :

Total number of news items: 4
Period covered by news items: 4/2006 to 5/2006

Une fois que avez terminé, enregistrez le fichier.

3.6 Utiliser des Conditions

Si vous consultez `QuickStart.xml`, vous verrez que chaque élément `newsitem` a un élément enfant `metainfo`, qui, à son tour, peut contenir un ou plusieurs éléments enfant `relevance`. Chaque nœud `relevance` contient une en-tête sous laquelle la pertinence de l'item `news` est indexée. De plus, il existe un nœud `/presswatch/selection/byrelevance`. Le contenu de ce nœud contient une des en-têtes pertinentes et détermine quels items `news` sont affichés. Par exemple, si le contenu du nœud `byrelevance` est `NanoPower`, alors tous les items `news` qui ont un nœud `relevance` contenant `NanoPower` sont affichés. Une condition peut tester ce que le contenu du nœud `byrelevance` représente (en recherchant ce nœud) et fournir le traitement approprié (affichages) dans le modèle conditionnel. Dans cette section, vous créerez un modèle conditionnel qui affiche ces items `news` ayant un élément `relevance` qui correspond au contenu de `byrelevance`.

Nous procéderons comme suit :

1. Créez une zone de liste déroulante qui affiche la valeur du nœud `byrelevance`. Les valeurs de la liste déroulante sont obtenues en utilisant l'expression XPath, qui compile une liste des valeurs de nœuds `relevance` uniques.
2. Insérer une condition autour de l'élément `newsitem`. Cette condition sélectionne tous les éléments `newsitem` qui ont un élément `relevance` avec un contenu correspondant au contenu du nœud `byrelevance`. Le contenu entouré par une branche d'une condition est connu comme modèle conditionnel.
3. Avec le modèle conditionnel, recensez chaque nœud `relevance` de cet item `news`.
4. Mettez en surbrillance l'élément `relevance` (dans la liste des éléments `relevance`) qui correspond à l'élément `byrelevance`. Ceci est fait en créant une condition pour sélectionner de tels éléments `relevance`, puis d'appliquer un formatage spécial à ce modèle conditionnel.
5. Dans la condition pour l'élément `newsitem`, insérez une branche qui sélectionne tous les items `news`.

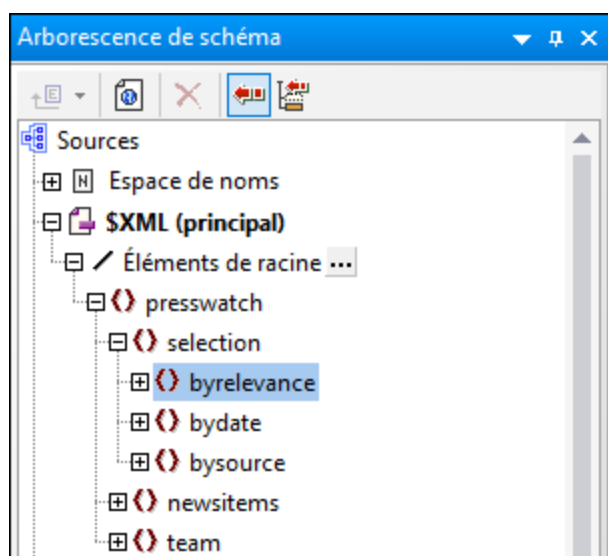
Créer une zone de liste pour sélectionner des valeurs de nœuds uniques

Dans le document XML, le nœud qui contiendra la sélection de l'utilisateur est `/presswatch/selection/byrelevance`. Créez ce nœud comme zone de liste déroulante. Procédez comme suit :

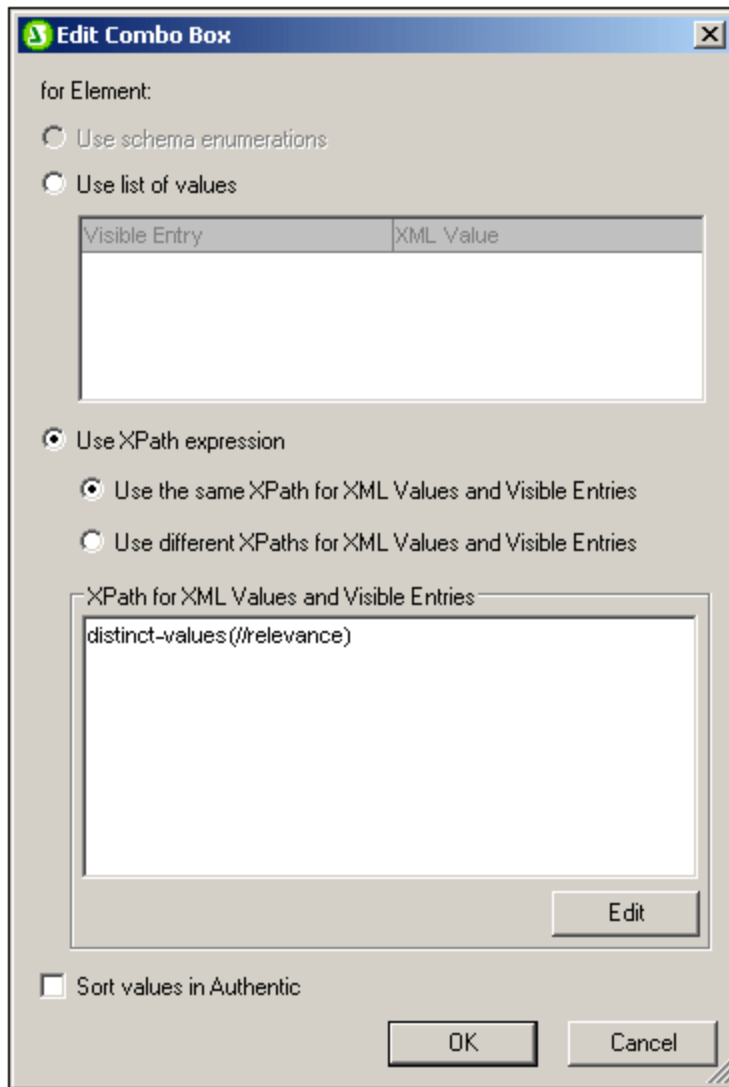
1. Insérez le texte statique "Sélectionner par pertinence : " au niveau supérieur du document et juste en-dessous du [deuxième calcul automatique](#)⁷⁸ (capture d'écran ci-dessous).

Total number of news items: =(AutoCalc)
Period covered by news items: =(AutoCalc)
Select by relevance: |

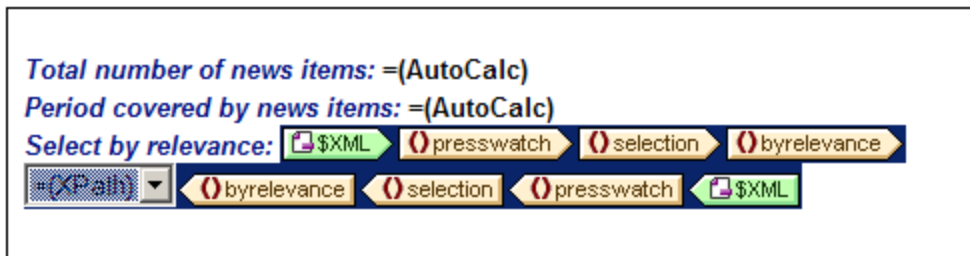
2. Glissez le nœud `byrelevance` de la [barre latérale de l'arborescence de schéma](#)³⁷ (capture d'écran ci-dessous), et déposez-le après le nouveau texte statique saisi.



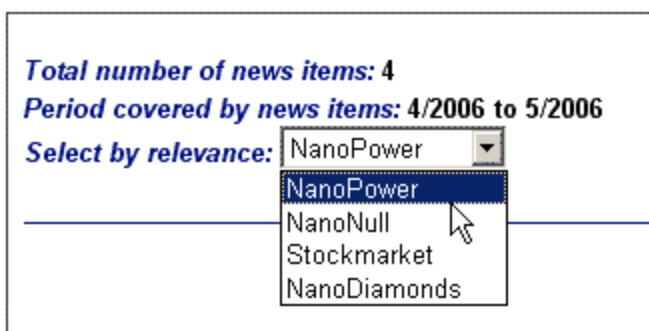
3. Dans le menu contextuel qui apparaît, sélectionnez « Créer Combo Box ». Le dialogue affiché ci-dessous apparaît.



4. Dans le dialogue « Éditer Zone de liste » (*capture d'écran ci-dessus*), sélectionnez Utiliser expression XPath , puis Utiliser la Même XPath pour les Valeurs XML et Entrées Visibles. Dans le champ XPath pour les valeurs XML et les entrées visibles, saisissez l'expression XPath suivante : `distinct-values(//relevance)`. Cette expression sélectionne des valeurs uniques des éléments `relevance` dans le document XML. Notez que malgré que les valeurs de tous les nœuds `relevance` apparaîtront dans la zone de liste HTML ; en sélectionner une dans l'Aperçu HTML n'aura aucun effet sur le contenu du nœud dans le document XML (c'est sur quoi SPS agit). Le document HTML est une sortie obtenue en transformant le document XML ; il n'accepte pas d'entrée. La liste déroulante est utilisée ici pour démontrer des manières alternatives de présenter le contenu.
5. Cliquez sur **OK** pour terminer. La liste déroulante est insérée et le design aura l'aspect suivant :



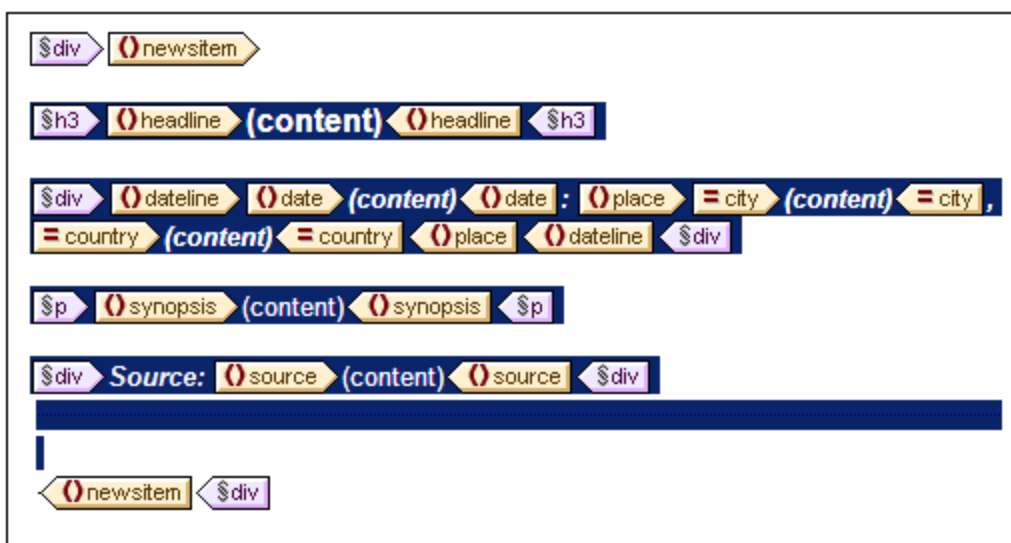
6. Basculer vers [Aperçu HTML](#) ³⁰. Quand vous cliquez sur la flèche vers le bas de la zone de liste déroulante, notez que la liste contient des valeurs uniques de tous les nœuds *relevance* (capture d'écran ci-dessous). Vérifier par rapport au document XML. Il s'agit d'une liste dynamique qui sera augmentée chaque fois qu'une nouvelle valeur *relevance* est ajoutée au document XML.



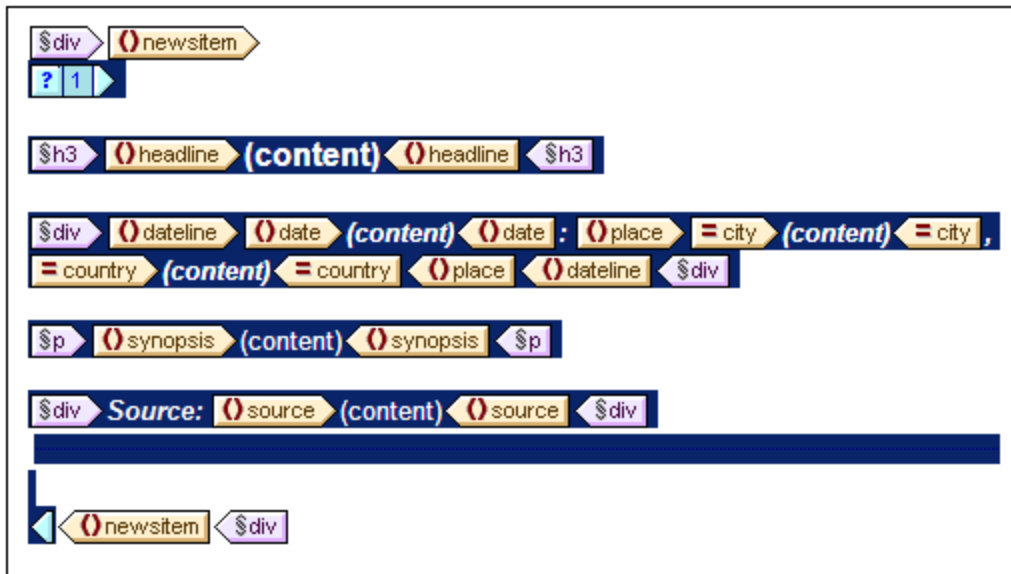
Insérer une condition pour afficher des news items ayant la **relevance** sélectionnée.

La condition sélectionne des éléments *newsitem* qui ont un élément *metainfo/pertinence* avec une valeur qui est la même que dans l'élément */presswatch/selection/byrelevance*. Insérer la condition comme suit :

1. Sélectionner les contenus de la partie *newsitem* du design qui doit être contenue à l'intérieur de la condition (en surbrillance dans la capture d'écran ci-dessous).



2. Sélectionnez la commande de menu (ou la commandes de menu contextuel) [Contenir dans | Condition](#) ⁵¹². Le [dialogue Éditer l'expression Xpath](#) ⁴²⁶ s'affiche..
3. Saisissez l'expression `metainfo/relevance=/presswatch/selection/byrelevance`. Cette expression évalue la valeur true quand la valeur du descendant `metainfo/relevance` du `newsitem` actuel est la même que la valeur de l'élément `/presswatch/selection/byrelevance` (la sélection de l'utilisateur).
4. Cliquez sur **OK**. La condition est créée autour des contenus de l'élément `newsitem` (*capture d'écran ci-dessous*).

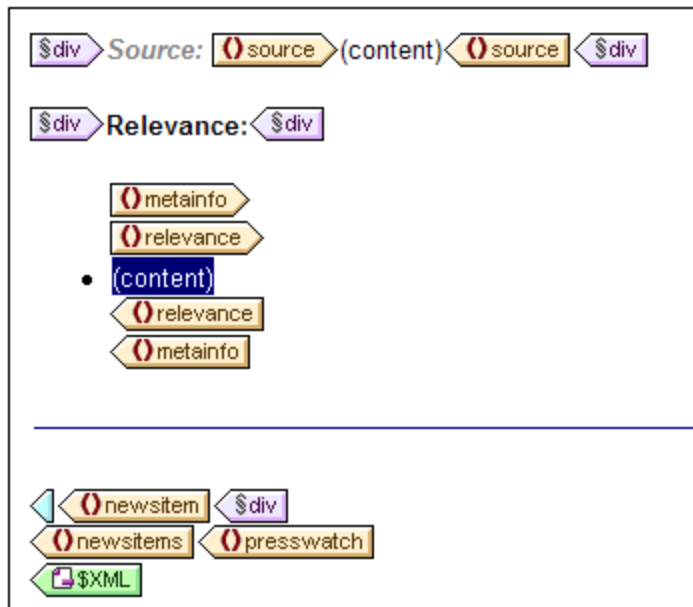


Notez qu'il y a une branche unique dans cette condition. Les News items pour lesquels le test de la condition évalue « true » sont affichés, ceux pour lesquels la condition test n'évalue pas « true » ne sont pas affichés. Dans ce cas, la condition fonctionne comme filtre. Plus tard dans cette section, vous ajouterez une deuxième branche à cette condition.

Insérer le nœud `relevance` comme liste

Afin d'afficher les nœuds `relevance` de chaque élément `newsitem`, insérez-les dans le design comme suit (voir la capture d'écran ci-dessous) :

1. Créer une espace vertical en-dessous du composant `div` composant pour l'élément `source` et à l'intérieur de la balise de fin du modèle conditionnel.
2. Saisissez le texte statique « Relevance : » et créez un format prédéfini de `div` autour (mettez en surbrillance le texte statique et insérez le format prédéfini).
3. Glissez l'élément `relevance` de l'arborescence des éléments Racine dans la [barre latérale de l'arborescence de schéma](#) ³⁷ et déposez-le dans le design ci-dessous du texte statique Relevance :.
4. Créez-le comme liste. (Dans le menu contextuel qui apparaît quand vous déposez le nœud dans le design, sélectionnez les Bullets et Numbering, puis sélectionnez le format de liste désiré.)
5. Appliquer un formatage de texte aux contenus de la liste. Quand vous avez terminé, le design devrait avoir l'aspect suivant :

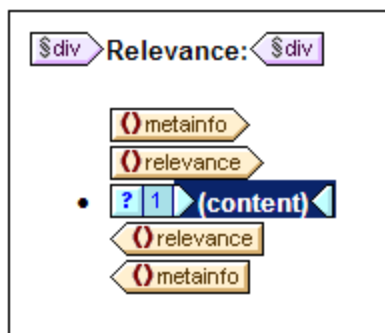


Désormais, dans Aperçu HTML, vérifier les résultats pour différentes sélections de `relevance`; Faites ceci en : (i) changeant la valeur du nœud `byrelevance` dans le document XML ; (ii) enregistrant le document XML ; (iii) et la ré-ouverture du fichier SPS dans StyleVision.

Mettant l'élément sélectionné `relevance` en gras

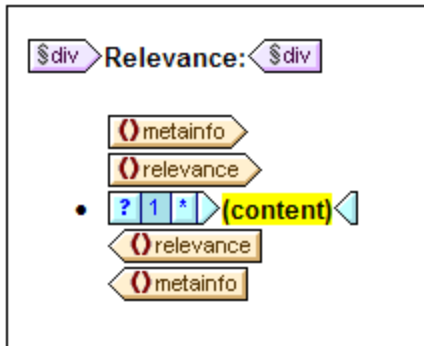
Quelques news items ont plus d'élément `relevance`. Dans de tels cas, le design serait amélioré si la pertinence qui correspond à la sélection de l'utilisateur serait mis en surbrillance tandis que les autres ne le seraient pas. Vous pouvez procéder des manières suivantes :

1. Sélectionnez l'élément `relevance` dans le design.
2. Insérer une condition, lui donnant une expression XPath de `./=presswatch/selection/byrelevance`. Ceci crée une condition avec une branche unique (*capture d'écran ci-dessous*) qui sélectionne les éléments `relevance` correspondant à l'élément `byrelevance`.



3. Sélectionnez l'espace réservé `contents` et donnez-lui un formatage local (dans la barre d'outils (*grou*) et une couleur d'arrière-plan jaune (*groupe color*).
4. Cliquez avec la touche de droite dans la condition et, depuis le menu contextuel, sélectionnez **Copier Branche**.

5. Dans le [dialogue Éditer l'expression XPath](#) ⁴²⁶ qui apparaît, vérifiez la case à cocher Otherwise (du côté droit, en haut).
6. Cliquez sur **OK** pour terminer. Une nouvelle branche (Otherwise) est créée (*capture d'écran ci-dessous*). La branche condition sélectionne tous les éléments `relevance` qui ne correspondent pas à l'élément `byrelevance`.



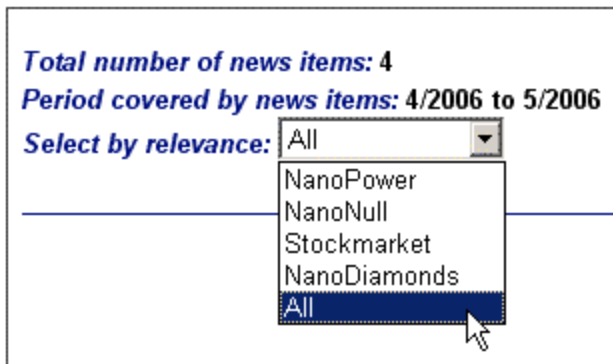
7. Notez que les contenus de la branche Otherwise sont une copie de la première branche ; l'espace réservé `contents` est en gras et a un arrière-plan jaune. Supprimez ce formatage (gras et couleur d'arrière-fond) de l'espace réservé `contents`.

Vous avez défini une condition avec deux branches (chacune avec son modèle conditionnel) qui réalise le test suivant sur chaque élément `relevance` : Si les contenus de `relevance` correspondent à `/presswatch/selection/byrelevance`, alors les contenus de `relevance` sont affichés en gras et avec un arrière-fond jaune. Autrement, (la deuxième branche) ils sont affichés normalement. Vérifiez ceci dans Aperçu HTML.

Modifiez la zone de liste déroulante et insérez une deuxième branche de condition

Dans la zone de liste déroulante, il n'existe pas d'option de liste déroulante pour sélectionner tous les news items. Pour inclure cette option, procédez comme suit :

1. Dans l'Aperçu de Design, sélectionnez la zone de liste déroulante.
2. Dans la barre latérale des Propriétés, avec `combobox` sélectionné dans les Propriétés For column, cliquez sur le bouton **Éditer** de la propriété *valeur d'entrée Zone de liste* (dans le groupe de propriétés *zone de liste*).
3. Dans la [Éditer Zone de liste](#) ¹⁶¹ qui apparaît, modifiez l'expression XPath de `distinct-values(//relevance)` à `distinct-values(//relevance), 'All'`. Ceci ajoute le string `All` à la séquence des items renvoyés par une expression XPath.
4. Vérifiez la liste déroulante de la zone de liste dans Aperçu HTML (*capture d'écran ci-dessous*).



La valeur `All` peut désormais être saisie dans le nœud `byrelevance`. L'idée est que lorsque le nœud `byrelevance` contient la valeur `All`, tous les news items devraient être affichés.

La condition qui affiche le modèle news item a une branche unique avec l'expression `metainfo/relevance=/presswatch/selection/byrelevance`. Puisqu'aucun nœud `metainfo/relevance` n'a la valeur `All`, aucun news item ne sera affiché quand `All` est la valeur du nœud `byrelevance`. Ce que vous devez faire c'est créer une deuxième branche pour la condition, qui testera une valeur pour `All`. En créant le modèle de l'item news au sein de cette branche, vous sortirez l'item news si le test est « true ». Procédez comme suit :

1. Dans l'Aperçu de Design, sélectionnez la condition de news item.
2. Cliquez avec la touche de droite dans la condition et, depuis le menu contextuel, sélectionnez **Copier Branche**.
3. Dans le [dialogue Éditer l'expression XPath](#) ⁴²⁶ qui s'affiche là, saisissez l'expression `: /presswatch/selection/byrelevance='All'`.
4. Cliquez sur **OK** pour terminer. Une deuxième branche est créée.

La deuxième branche a comme contenu le même modèle que la première branche. La deuxième branche sort le modèle de l'item news si le contenu du nœud `byrelevance` est `All`.

Après avoir complété cette section, enregistrez le design.

3.7 Utiliser des modèles globaux et Rest-of-Contents

Des [modèles globaux](#) ²² sont utiles pour spécifier le traitement d'un élément globalement. Ceci fait que les règles du modèle global (défini dans un emplacement) doivent être utilisées aux mêmes endroits dans la feuille de style. Un modèle global peut être utilisé de deux manières :

- Les règles du modèle global peuvent être copiées dans le modèle global.
- Un modèle local (dans le modèle principal) peut transmettre le traitement de ce nœud dans le modèle global. Une fois que le modèle global est exécuté, le traitement continue dans le modèle principal. Dans ce cas, le modèle global est invoqué ou utilisé depuis le modèle principal.

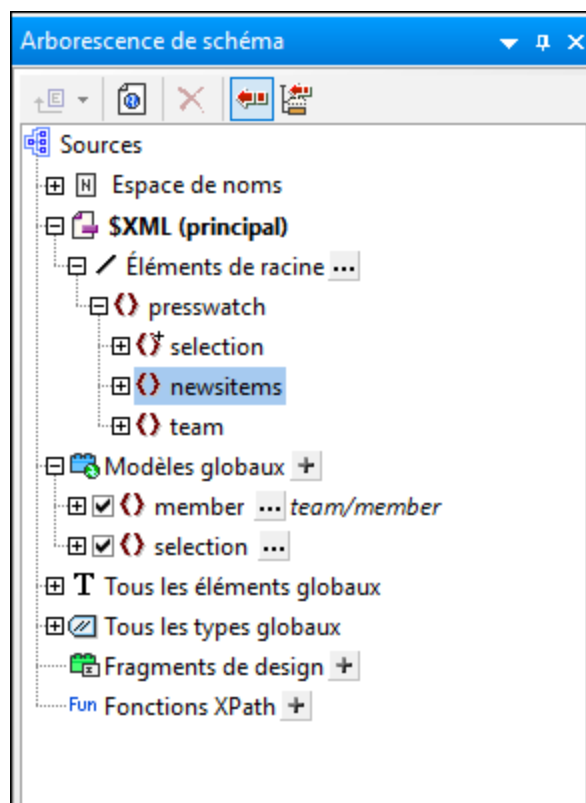
Il existe deux mécanismes qui sont utilisés pour invoquer un modèle global depuis le modèle principal :

- Un modèle local référence un modèle global.
- Une instruction (`rest-of-contents`) du modèle principal applique des modèles aux éléments modèles de l'élément actuel (c'est-à-dire des `rest-of-contents` de l'élément actuel). Si un modèle global existe pour un des éléments descendants, le modèle global est appliqué pour cet élément. Autrement le modèle intégré pour les éléments est appliqué. (Le modèle intégré pour les éléments traite des éléments enfant et sort du contenu texte des éléments. En guise de résultat, le contenu de texte de tous les éléments descendants sera la sortie. Notez que les valeurs d'attributs ne sont **pas** la sortie.)

Dans cette section, vous créerez un design pour le modèle des membres de l'équipe utilisant l'instruction `rest-of-contents` et un modèle global pour le membre de l'[élément global](#) ²² `member`.

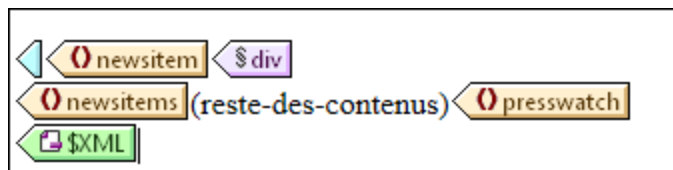
Insérer une instruction de `rest-of-contents`

La grande structure du schéma est affichée dans la capture d'écran ci-dessous.



L'élément du document `presswatch` contient trois enfants : (i) `sélection`; (ii) `newsitems` ; et (iii) `team`. Le modèle principal que vous avez créé jusqu'à ce point traite l'élément `/presswatch`. Au sein de l'élément `presswatch`, uniquement l'élément `newsitems` est traité. La `sélection` et les éléments `team` ne sont pas traités au sein de l'élément `presswatch` (bien que la `sélection` ait été traitée au sein de l'élément `newsitems`). Insérer l'instruction `rest-of-contents` à l'intérieur de `presswatch` causera pour cette raison les éléments `sélection` et `team` à être traités.

Insérer l'instruction `rest-of-contents` dans le design en plaçant le curseur entre les balises de fin de `newsitems` et `presswatch`, et sélectionnez la commande de menu ou la commande du menu contextuel [Insérer | Reste de Contenu](#)⁴⁸⁹. L'espace réservé `rest-of-contents` est inséré (capture d'écran ci-dessous).



Si vous regardez l'aperçu HTML, vous verrez un string de texte (capture d'écran ci-dessous) :

```
AllAndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.edwar
```

Le string est le résultat de l'application des modèles intégrés dans les éléments `sélection` et `team`. Le modèle built-in pour les éléments traite les éléments enfant. Le modèle built-in pour les nœuds de texte sort le texte dans le nœud de texte. L'effet combiné de ces deux modèles built-in est de sortir le contenu du texte de

tous les nœuds descendants des éléments `selection` et `team`. Le texte `All` vient de `selection/byrelevance`, et est suivi par la sortie de texte des nœuds descendants `team/member`, `premier`, `dernier`, `e-mail`, dans l'ordre du document. Notez que l'attribut `id` de `membre` n'est pas une sortie (car, comme attribut, elle n'est pas considérée comme enfant de `membre`).

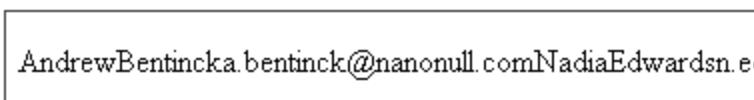
Créer un modèle global pour `selection`

Puisque le contenu de `selection` n'est pas requis dans la sortie, vous devriez créer un modèle global vide pour `selection` pour que son contenu ne soit pas traité. Procédez comme suit :

1. Dans le Mode Design, cliquez avec la touche de droite sur `selection` dans l'arborescence « Tous les éléments globaux » dans la [barre latérale de l'arborescence de schéma](#) ³⁷.
2. Dans le menu contextuel qui s'ouvre, sélectionnez **Faire / Supprimer Modèle global**. Un modèle global pour `selection` est créé (*capture d'écran ci-dessous*).



3. Dans le modèle global, cliquez sur l'espace réservé `contents` et appuyez sur la clé **Supprimer** de votre clavier. L'espace réservé de contenus est supprimé.
4. Vérifiez l'Aperçu HTML. Le texte `All` n'est plus présent dans le ligne de la sortie de texte par les modèles built-in (*capture d'écran ci-dessous*).

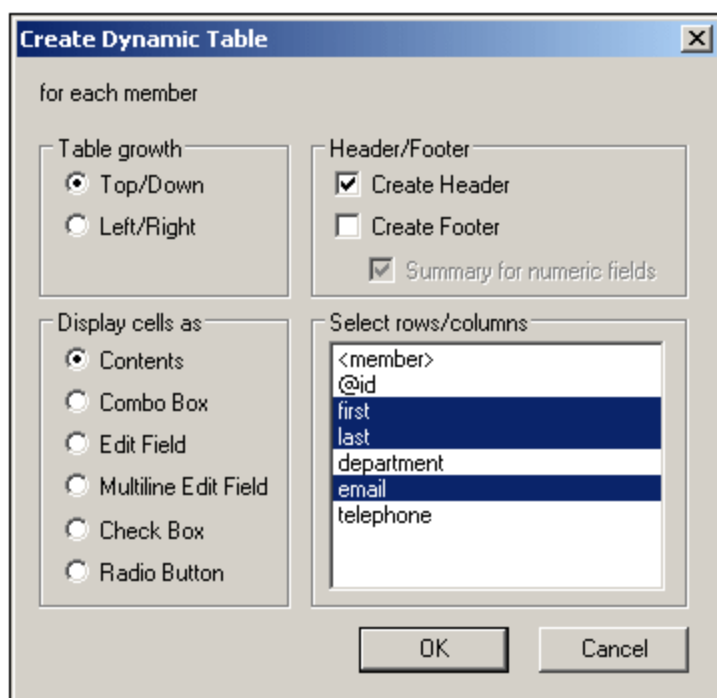


Puisque le modèle global pour `selection` est vide, les éléments enfant de `selection` ne sont pas traités.

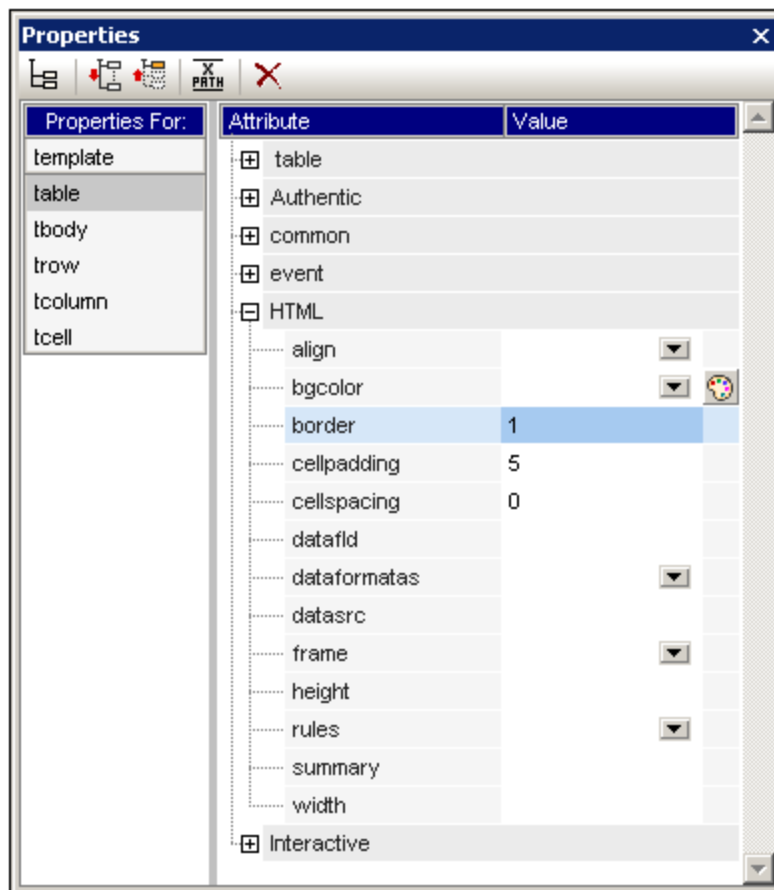
Créer un modèle global pour `team/member`

L'objectif est de créer une table qui affiche les détails des membres de l'équipe « press monitoring ». Cette table sera créée dans le modèle global pour l'élément `team`. Procédez comme suit :

1. Créer un modèle global pour l'élément `team` (cliquez avec la touche de droite sur `team` dans la liste Tous les éléments globaux de la barre latérale de l'Arborescence de schéma et sélectionnez **Faire / Supprimer Modèle global**).
2. Dans la liste Tous les éléments globaux, élargissez l'élément `team` et glissez son élément enfant `member` dans le modèle global de `team` (dans le design).
3. Dans le menu contextuel qui apparaît quand vous déposez l'élément dans le modèle global `team`, sélectionnez **Créer Table**. Ceci fait apparaître le dialogue Créer Table dynamique (*capture d'écran ci-dessous*).



4. Dans la liste des attributs/éléments décochez @id, department et telephone (voir la capture d'écran), et cliquez sur **OK**. La table dynamique est créée.
5. Placez le curseur dans une cellule du corps de table, et dans la [barre latérale des Propriétés](#) ⁴⁸, avec la table sélectionnée dans les Propriétés Pour colonne, précisez les propriétés de table telles qu'affichées dans la capture d'écran ci-dessous.



6. Définissez des propriétés additionnelles tel que requis dans les Propriétés et barres latérales de Styles. Par exemple, une coloration d'arrière-plan peut être définie pour la ligne d'en-tête en plaçant le curseur dans la ligne de l'en-tête, et avec `tr` sélectionné dans Styles Pour colonne de la barre latérale des Styles, précisez une valeur pour la propriété `couleur d'arrière-plan` (groupe *couleur*). Vous pouvez éditer les en-tête, qui sont des strings de texte statique. Aussi, si l'espace réservé `content` de l'élément `team` est toujours présent dans le modèle global, supprimez-le.

L'aperçu HTML de la table aura l'air de ceci :

First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

3.8 Et voilà !

Félicitations ! Vous avez accompli ce tutoriel avec succès. Vous avez appris les aspects les plus importants de la création d'une SPS :

- Comment [créer la structure](#)⁶⁰ du document ([modèle principal](#)⁶⁰ et [modèles globaux](#)⁹⁰).
- Comment insérer le contenu [dynamique](#)⁶⁰ et [static](#)⁶⁷ dans le design, utilisant une variété de composants SPS dynamiques et statiques.
- Comment utiliser les [styles CSS](#)⁶⁰, dans [feuilles de style externes](#)⁷⁴, dans [règles de style global](#)⁷⁵, et dans [règles de style local](#)⁷⁶.
- Comment utiliser des [Auto-Calculations](#)⁷⁸ pour dériver des informations additionnelles des données XML disponibles.
- Comment utiliser les [conditions](#)⁸² pour filtrer les données XML et comment obtenir différentes sorties dépendant des valeurs dans les données XML.
- Comment utiliser les [modèles globaux](#)⁹² et [rest-of-contents](#)⁹⁰.

Pour une description plus détaillée des fonctions, voir les sections correspondantes dans les quatre sections :

- [Fichier SPS : Contenu](#)¹⁰⁹
- [Fichier SPS : Structure](#)¹⁸⁵
- [Fichier SPS : Fonctions avancées](#)²⁶⁰
- [Fichier SPS : Présentation](#)³²⁸
- [Fichier SPS : Fonction additionnelle](#)³⁶²

Ces sections contiennent également des descriptions de plusieurs autres fonctions de StyleVision non rencontrées dans le tutoriel de Démarrage rapide.

4 Aperçu d'utilisation

Objectifs

Les documents SPS que vous créez dans StyleVision peuvent être utilisés pour générer les feuilles de style XSLT pour HTML. Une feuille de style générée 'une SPS peut être utilisée pour transformer tout document XML dans le même schéma que SPS.

Étapes pour créer une SPS

Ci-dessous, vous trouverez une explication d'étapes impliquées dans la création d'une nouvelle feuille de style SPS.

1. [Attribuer un schéma](#)³⁷ à la SPS vide nouvellement créée. Le schéma pourrait être le suivant : (i) un fichier de schéma (Schéma DTD ou XML ; (ii) un schéma XML généré d'une BD (*éditions Enterprise et Professional uniquement*) ; (iii) un schéma basé sur une taxonomie XBRL (*édition Enterprise uniquement*) ; (iv) un schéma défini par l'utilisateur (créé directement dans StyleVision). Ceci est fait dans la [barre latérale de l'Aperçu de design](#)³⁵. En alternative, une nouvelle SPS peut être créée directement avec un schéma par le biais de la commande **Fichier | Nouveau**.
2. [Attribuer un fichier de travail XML](#)³⁷ à la SPS. Le [fichier de travail XML](#)²³ fournit les données XML traitées par SPS lorsqu'il génère. Le [fichier de travail XML](#)²³ est attribué dans la [barre latérale de l'Aperçu de design](#)³⁵. Le fichier de travail XML vous permet de prévisualiser la sortie dans StyleVision.
3. [Sélectionnez la version XSLT requise](#)⁹⁹. Afin de générer la sortie Texte, la version XSLT doit être XSLT 2.0 ou XSLT 3.0
4. Sélectionnez la [Compatibilité d'Internet Explorer](#)¹⁰⁰ pour faire correspondre à la version d'Internet Explorer installée.
5. Le document SPS est conçu dans le [Mode Design](#)²⁸ utilisant les différents composants de design disponibles dans le designer. Le [processus de design](#)⁹⁸ consiste à créer une structure de document et définir les [propriétés de présentation](#)³²⁸.
6. Les sorties sont testées. Si des modifications au design sont requises, elles sont faites et le document SPS est re-testé.
7. Si les [fichiers XSLT ou fichiers de sortie](#)¹⁰² sont requis, ceux-ci sont [générés](#)¹⁰².

4.1 SPS et Sources

Créer un nouveau fichier SPS

Pour créer un nouveau document SPS, sélectionnez une option de la commande **Fichier | Nouveau** (**Ctrl+N**)⁴⁵² ou cliquez sur l'icône **Nouveau design**  dans la **barre d'outils standard**⁴⁵⁰. Un nouveau document SPS est créé et affiché en Mode Design. Le nouveau document obtient un nom provisoire de `SPSX.sps`, où `x` est un entier correspondant à la position de ce document SPS dans la séquence des nouveaux documents créés depuis le démarrage de l'application.

Une fois qu'un nouveau document SPS a été créé, les fichiers source pour SPS doivent être attribués.

Attribuer les fichiers source pour SPS

Il existe deux types de fichiers source qui peuvent être attribués à un SPS:

- [Sources de schéma](#)⁹⁷
- [Fichier de travail XML](#)⁹⁷

Ces attributions de fichier source sont faites dans la **barre latérale de l'Aperçu de design**³⁵. Comment réaliser ces attributions est décrit dans la section, [Aperçu de design](#)³⁵. Les points significatifs de chaque type de fichier source sont donnés ci-dessous.

Sources de schéma

Un fichier source de schéma doit être attribué à un SPS pour qu'une structure pour le document de design puisse être créé. Les sources de schéma sont attribuées dans la **barre latérale de l'Aperçu de design**³⁵. Un schéma peut être un fichier de schéma XML (fichier `.xsd`), un schéma XML généré par un fichier XML, un DTD ou un schéma défini par l'utilisateur. Pour chaque schéma, un [fichier de travail XML](#)⁹⁷ peut être assigné.

Note : si vous souhaitez ajouter un espace de nom à une feuille de style SPS ou à une feuille de style XSLT générée depuis une SPS créée, l'espace de nom doit être ajouté à l'élément de niveau supérieur `schema` du schéma XML sur lequel la SPS se base.

Fichier de travail XML

peut, en option, avoir un [fichier de travail XML](#)²³ y associé. La fonction du [fichier de travail XML](#)²³ est de fournir la source de données XML pour les aperçus de sortie dans StyleVision, et doit être valide conformément au schéma avec lequel il est associé. Le [fichier de travail XML](#)²³ est attribué dans la **barre latérale de l'Aperçu de design**³⁵.

4.2 Créer le design

Dans le design SPS, vous précisez :

1. [Quel contenu](#) ⁹⁸ (du document XML ou BD) devrait aller à la sortie ; de plus, du contenu peut être ajouté directement dans SPS pour l'inclusion dans la sortie ;
2. [Comment la sortie doit être structurée](#) ⁹⁸ ; et
3. [Quelles propriétés de présentation \(formatage\)](#) ⁹⁸ sont appliquées dans les différentes parties de la sortie.

Contenu pour la sortie

Le contenu pour la sortie peut venir du :

1. document XML auxquels la SPS est appliquée. Le contenu du [document XML](#) ²³ est inclus dans SPS en glissant le nœud de données XML requis de l'arborescence de schéma pertinente dans la [barre latérale de l'arborescence de schéma](#) ³⁷ et le déposant à l'emplacement désiré dans SPS.
2. Un document XML externe qui est accessible dans l'application (c'est-à-dire dans StyleVision). En utilisant la fonction doc() de XPath 2.0 dans l'Auto-Calcul, le contenu de sources du document XML externe peut être accédée. Un document XML accédé par le biais de la fonction doc() dans une expression XPath ne doit pas être référencée par le biais des associations [Sources de schéma](#) ³⁷.
3. Le design SPS lui-même Le texte et d'autres contenus (tels les images et tables) peuvent être insérés directement dans SPS utilisant le clavier et d'autres fonctions GUI. De telles entrées sont indépendantes du document XML.
4. Les données dynamiques manipulées (source XML), avec les manipulations étant obtenues utilisant les expressions XPath. Les manipulations sont généralement atteintes avec [Auto-Calculations](#) ²⁶¹.
5. Pour la sortie HTML, les [fonctions JavaScript](#) ³⁸⁹ peuvent être utilisées pour générer du contenu.

Structure de la sortie

Dans le design SPS, la [structure de la sortie](#) ²³⁴ peut être contrôlée en utilisant soit : (i) une approche procédurale, dans laquelle la structure de sortie est précisée dans un [modèle au niveau d'entrée](#) ²³⁴ du modèle principal de (StyleVision ([modèle principal](#) ²³⁴)) et peut être indépendante de la structure du document XML ; (ii) une approche déclarative, dans laquelle les [règles de modèle sont déclarées pour de nombreux nœuds](#) ²³⁴ (les modèles globaux de StyleVision ([modèles globaux](#) ²³⁴)), générant ainsi une sortie qui suit la structure du document XML ; ou (iii) une combinaison d'approches procédurales et déclaratives. Dans le Mode Design, vous pouvez utiliser un mélange de [modèle principal](#) ²³⁴ et [modèles globaux](#) ²³⁴ pour obtenir la structure désirée du document de sortie. L'utilisation de [SPS modulaires](#) ²²⁰ et de [Fragments de design](#) ²⁴⁵ fournit une flexibilité supplémentaire de la manière dont un SPS est structuré.

Présentation (ou formatage) de la sortie

Dans le Mode Design, les propriétés de présentation sont appliquées dans les composants de design utilisant les styles CSS. Les styles peuvent être définis localement dans le composant, pour des sélecteurs HTML déclarés au niveau du document, et pour des sélecteurs HTML déclarés dans une feuille de style CSS externe. De plus, certains éléments HTML peuvent être appliqués dans les composants utilisant des [formats prédéfinis](#) ³²⁹. Pour voir comment préciser les propriétés de présentation en détail dans la section, voir [Procédures de présentation](#) ³²⁸.

4.3 Versions XSLT et XPath

Un SPS est essentiellement une feuille de style XSLT. Pour chaque SPS, vous devez définir la version XSLT : 1.0, 2.0, ou 3.0. Faites ceci en cliquant sur l'icône de la barre d'outils appropriée :  ou  ou . La sélection que vous faites détermine deux choses :

- Lequel des trois moteurs XSLT dans StyleVision est utilisé pour les transformations ; StyleVision a des moteurs XSLT 1.0, XSLT 2.0 et XSLT 3.0.
- Laquelle des fonctionnalités XSLT (1.0, 2.0 ou 3.0) est affichée dans l'interface et autorisée dans SPS. Par exemple, XSLT 3.0 utilise XPath 3.0, qui est un langage bien plus puissant que XPath 1.0 (qui est utilisé dans XSLT 1.0) ou XPath 2.0 (qui est utilisé dans XSLT 2.0). De plus, quelques fonctions SPS, telles que la fonction table-des-matières, qui est uniquement disponible avec XSLT 2.0 et XSLT 3.0.

Note : Afin de générer la sortie Texte, la version XSLT doit être XSLT 2.0 ou XSLT 3.0

Transformations XSLT

Les transformations XSLT dans StyleVision sont utilisées : (i) pour générer les [affichages de sortie](#)³⁰ dans l'interface; et (ii) pour [générer et enregistrer les fichiers de sortie](#)¹⁰² (HTML) [au sein de l'interface](#)⁴⁶⁹ et via [StyleVision Server](#). Le moteur XSLT utilisé pour les transformations (moteurs Altova XSLT 1.0, 2.0, ou 3.0) correspond à la version XSLT sélectionnée dans SPS.

Fonctions XSLT dans GUI

La fonction appropriée pour chaque version XSLT concerne principalement l'utilisation de la version XPath correcte (XPath 1.0 pour XSLT 1.0, XPath 2.0 pour XSLT 2.0, XPath 3.0 pour XSLT 3.0). Les expressions XPath sont largement utilisées dans StyleVision—plus généralement dans des fonctions telles que [Auto-Calculations](#)²⁶¹ et [Modèles conditionnels](#)²⁶⁶—et il existe des mécanismes d'interface qui requièrent et vous aident à créer des expressions XPath. La fonction de la version correcte XPath est rendue automatiquement disponible dans l'interface conformément à la version XSLT que vous sélectionnez.

4.4 Compatibilité Internet Explorer

Internet Explorer (IE) doit être installé sur l'appareil StyleVision pour afficher correctement le design SPS (dans Design View) et les aperçus de sortie (dans Aperçu HTML). Ci-dessous, vous trouverez les notes sur les versions IE qui sont prises en charge :

- Internet Explorer 5.5 ou supérieur
- Internet Explorer 6.0 et supérieur a une meilleure prise en charge XML et est recommandé.
- Internet Explorer 9 (IE9) ou supérieur fournit des fonctions supplémentaires, telles qu'une prise en charge pour plus de formats d'image et pour de nouveaux styles CSS. Si vous planifiez l'utilisation de ces fonctions supplémentaires dans votre design, vous allez éventuellement vouloir utiliser IE9.

Prise en charge de la fonction IE9 dans StyleVision

Les fonctions suivantes pour IE9 ou supérieur sont prises en charge dans StyleVision:

- Formats d'image supplémentaires pris en charge : TIFF, JPEG XR et SVG. (Les documents SVG doivent être en format XML et doivent être dans l'espace de nom SVG.) Ces formats d'image seront affichés dans IE9, mais pas dans des versions plus anciennes de IE. Pour une liste complète d'images prises en charge dans diverses sorties, voir [Types d'image et sortie](#)¹⁵⁴.
- Prise en charge de nouveaux styles CSS (y compris les styles CSS3), qui sont énumérés ci-dessous. L'application de ces styles est limitée à la sortie HTML.
 - background-clip
 - background-origin
 - background-size
 - box-sizing
 - box-shadow
 - border-radius (border-*-radius)
 - font-stretch
 - ruby-align
 - ruby-overhang
 - ruby-position
 - overflow-x, overflow-y
 - outline (outline-color, outline-style, outline-width)
 - text-align-last (partial)
 - text-overflow (partial)
- Prise en charge pour la nouvelle fonction longueur CSS `calc()`
- Prise en charge pour les nouvelles fonctions de couleur CSS `rgba()`, `hsl()` et `hsla()`
- Prise en charge pour les nouvelles unités de longueur CSS `rem`, `vw`, `vm`, `vh` et `ch`
- Les éléments HTML5 qui sont pris en charge par IE9 peuvent être insérés dans le design comme [éléments définis par l'utilisateur](#)¹²².

Mode Design et versions IE

Vous pouvez configurer le Mode Design pour une version IE spécifique en précisant, dans le dialogue des [Propriétés](#)⁴⁷², la version IE avec laquelle vous souhaitez que le Mode Design soit compatible. Ceci a les effets suivants :

- Tous les styles CSS qui peuvent être rendus seront automatiquement affichés dans les barres latérales des Styles de StyleVision. (Notez, toutefois, que si IE9 a été sélectionnée, alors IE9 doit être installée pour les styles CSS appuyés par IE9 pour être disponible dans l'interface de design.) Par exemple, si la version IE9 est installée et que IE9 est sélectionnée comme version compatible, alors les styles CSS3 pris en charge dans IE9 seront disponibles dans l'interface de design.
- Les éléments HTML correspondant à la version IE sélectionnée peuvent être saisis comme [formats prédéfinis](#)¹¹² ou comme des [éléments définis par l'utilisateur](#)¹²². L'élément HTML sera rendu dans l'aperçu HTML conformément à la manière dont la version IE installée rend cet élément. Par exemple, si la version IE9 est installée et que IE9 est sélectionnée comme version compatible, alors les éléments HTML5 pris en charge sont rendus dans l'aperçu HTML.

Définir le Mode Design pour une version IE spécifique

Pour définir le Mode Design pour une version IE spécifique, sélectionnez la commande de menu Fichier | Propriétés et, dans l'onglet Sortie, sélectionnez la version IE requis (compatible). Voir [Fichier | Propriétés](#)⁴⁷² pour les détails.

Compatibilité d'anciens designs SPS avec IE9

Si vous ouvrez un design SPS qui a été créé pour une plus ancienne version IE, et si la plus récente version IE9 ou supérieure est installée sur l'appareil StyleVision, alors StyleVision détectera la version la plus récente et demandera dans un dialogue si vous souhaitez changer la compatibilité à la comptabilité IE9. Changer à la nouvelle compatibilité fournira des options supplémentaires du Mode Design tel qu'indiqué ci-dessus.

L'apparence du document dans le Mode Design et la sortie HTML restera inchangée pour les **colonnes de table**, qui sont gérées différemment par IE9. Si vous modifiez la comptabilité IE en compatibilité IE9, alors vérifiez si les colonnes de table sont générées tel que requis. Si ce n'est pas le cas, vous pouvez modifier les propriétés des colonnes de table ou remettre, dans le dialogue [Propriétés](#)⁴⁷², la compatibilité IE à celle de la version IE sélectionnée précédemment.

4.5 Fichiers générés

Dans StyleVision, les feuilles de styles et fichiers de sortie peuvent être générés utilisant la commande [Fichier | Enregistrer Fichiers générés](#)⁴⁶⁹ ou [StyleVision Server](#).

Les fichiers suivants peuvent être générés depuis StyleVision :

- Feuilles de style XSLT basées sur le design SPS.
- Les fichiers de sortie générés en traitant le [fichier de travail XML](#)²³ attribués dans SPS avec les feuilles de style générées du SPS.

La balise pour la sortie est contenue dans SPS. Les données pour la sortie sont contenues dans le document XML. C'est la feuille de style XSLT qui joint la balise et les données ensemble dans la sortie. Les deux feuilles de style XSLT de même que la sortie actuelle peuvent être consultées dans StyleVision dans les [Aperçus de sortie](#)³⁰.

Note : si vous souhaitez ajouter un espace de nom à une feuille de style SPS ou à une feuille de style XSLT générée depuis une SPS créée, l'espace de nom doit être ajouté à l'élément de niveau supérieur `schema` du schéma XML sur lequel la SPS se base.

Site web d'Altova : [Reporting XML](#)

Documents de sortie

Ci-dessous, vous trouvez des points importants à noter en ce qui concerne les documents générés :

- **Sortie HTML et feuilles de sortie** : (1) le formatage et la mise en page du document HTML généré seront identiques à l'Aperçu HTML de StyleVision. (2) les appareils d'entrée de données (champs d'entrée de texte, cases à cocher, etc.) dans le fichier HTML ne permettent pas d'entrée. Ces appareils d'entrée des données sont prévus pour l'entrée des données XML dans Authentic View et, bien qu'ils soient traduits comme inchangés dans les équivalents HTML graphiques, ils ne peuvent pas être utilisés pour l'entrée des données dans le document HTML.

Site web d'Altova : [XML en HTML](#)

4.6 Catalogues dans StyleVision

Le mécanisme de catalogue XML permet aux fichiers d'être extraits de dossiers locaux, donc augmentant la vitesse de traitement général, ainsi qu'améliorant la portabilité des documents—puisque uniquement les URI de fichiers catalogue doivent être modifiés. Voir la section [Le fonctionnement des catalogues](#)¹⁰³ pour tout détail.

Les produits XML d'Altova utilisent un mécanisme de catalogue pour accéder rapidement à et charger des fichiers communément utilisés, tels que les DTD et Schémas XML. Ce mécanisme de catalogue peut être personnalisé et étendu par l'utilisateur et est décrit dans la section [Structure du catalogue dans StyleVision](#)¹⁰⁴ et [Personnaliser vos catalogues](#)¹⁰⁶. La section [Variables pour les emplacements système de Windows](#)¹⁰⁷ rassemble les variables Windows pour les emplacements de systèmes communs. Ces variables peuvent être utilisées dans les fichiers de catalogue pour situer des dossiers utilisés communément.

Cette section est organisée dans les sous-sections suivantes :

- [Comment fonctionnent les catalogues](#)¹⁰³
- [Structure du catalogue StyleVision](#)¹⁰⁴
- [Personnaliser vos catalogues](#)¹⁰⁶
- [Variables pour les emplacements de Système](#)¹⁰⁷

Pour plus d'informations sur les catalogues, voir la [spécification de catalogues XML](#).

4.6.1 Comment fonctionne 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 StyleVision :

```
<?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 StyleVision, 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.

4.6.2 Structure du catalogue dans StyleVision

Lorsque StyleVision 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 -->
<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.
- `CoreCatalog.xml` est situé dans le dossier d'application in the StyleVision et est utilisé pour localiser les schémas et les feuilles de style utilisés par des processus spécifiques de StyleVision, 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 :

<code>%PersonalFolder%</code>	Dossier personnel de l'utilisateur actuel, par exemple c : \\Users\\<name>\\Documents
<code>%CommonSchemasFolder%</code>	C:\ProgramData\Altova\Common2027\Schemas
<code>%ApplicationWritableDataFolder%</code>	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 StyleVision.
- `CustomCatalog.xml` est situé dans votre dossier `MyDocuments\Altova\StyleVision`.
- Les fichiers `catalog.xml` sont chacun dans un dossier de schéma spécifique, ces dossiers de schéma étant dans le dossier des Schémas communs.

4.6.3 Personnaliser vos Catalogues

Lorsque vous créez des entrées dans `CustomCatalog.xml` (ou tout autre fichier catalogue qui doit être lu par StyleVision), 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"/>`

Veillez 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, StyleVision 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 StyleVision 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 StyleVision trouve un schéma référencé

Un schéma est référencé dans un document XML par le biais de l'attribut `xsi:schemaLocation` (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 séquentielle par StyleVision. 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.

Spécifications de schéma XML

L'information de spécification de schéma XML est prédéfinie dans StyleVision 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 StyleVision des informations sur les assistants de saisie à des fins d'édition si vous vouliez créer des documents conformément à ces recommandations antérieures.

4.6.4 Variables pour les emplacements système de Windows

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 :

<code>%PersonalFolder%</code>	Chemin complet vers le dossier Personnel de l'utilisateur actuel, par exemple <code>c:\Users\<name>\Documents</code>
<code>%CommonSchemasFolder%</code>	<code>C:\ProgramData\Altova\Common2027\Schemas</code>
<code>%ApplicationWritableDataFolder%</code>	<code>C:\ProgramData\Altova</code>
<code>%AltovaCommonFolder%</code>	<code>C:\Program Files\Altova\Common2027</code>
<code>%DesktopFolder%</code>	Chemin complet vers le dossier Bureau de l'utilisateur actuel.
<code>%ProgramMenuFolder%</code>	Chemin complet vers le dossier Menu Programme pour l'utilisateur actuel.
<code>%StartMenuFolder%</code>	Chemin complet vers le dossier Démarrage pour l'utilisateur actuel.
<code>%StartUpFolder%</code>	Chemin complet vers le dossier Démarrage pour l'utilisateur actuel.
<code>%TemplateFolder%</code>	Chemin complet vers le dossier Modèle pour l'utilisateur actuel.
<code>%AdminToolsFolder%</code>	Chemin complet vers le répertoire de système de fichier qui stocke des outils administratifs pour l'utilisateur actuel.
<code>%AppDataFolder%</code>	Chemin complet vers le dossier Données d'Application pour l'utilisateur actuel.
<code>%CommonAppDataFolder%</code>	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.

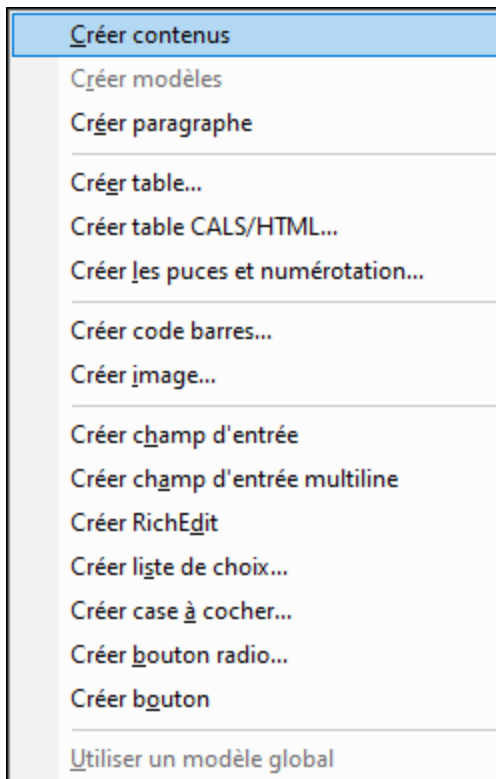
5 Contenu SPS

Cette section décrit en détails les procédures majeures utilisées pour créer et éditer des composants de document SPS qui sont utilisés pour créer des emplacements dans le design du document pour le contenu des données XML. Les procédures sont recensées ci-dessous et sont décrites en détail dans les sous-sections de cette section. Ces mécanismes sont utilisés pour concevoir tout type de modèle : [principal](#)²³⁴, [global](#)²³⁴, or [nommé](#)²⁴⁵.

- [Insérer du contenu XML comme texte](#)¹¹⁰. Les données XML peuvent être insérées dans le design en glissant les nœuds pertinents (élément, attribut, type ou CDATA) dans le design et créez-les comme (contenus) ou (reste-de-contenus).
- [Insérer le contenu MS Word](#)¹¹⁴
- [Modèles définis par l'utilisateur](#)¹¹⁹
- [Éléments définis par l'utilisateur. XML. blocs de texte](#)¹²²
- [Travailler avec des tables](#)¹²⁵. Les tables peuvent être insérées par (i) le designer SPS, directement dans le design SPS (tables statiques) ou utilisant des sous-structures de document XML, et (ii) l'utilisateur Authentic View.
- [Créer des listes](#)¹⁴⁷. Les listes statiques, où la structure de liste est saisie dans le design SPS et les listes dynamiques, où une sous-structure de document XML est créée comme liste, fournit des capacités de tri des données puissantes.
- [Utiliser des graphiques](#)¹⁵². Les graphiques peuvent être insérées dans le design SPS utilisant une variété de méthodes à déterminer l'URI cible (statique, dynamique, une combinaison des deux, et une entité non parsée d'URI).
- [Utiliser les appareils d'entrée des données \(ou contrôles de formes\)](#)¹⁵⁸. Les données XML peuvent être saisies par l'utilisateur Authentic View via les appareils d'entrée de données tels que les champs d'entrée et zones de liste déroulante. Ceci fournit une couche d'aide à l'utilisateur ainsi que des contraintes d'entrée. Des nœuds individuels dans le document XML peuvent être créés comme appareils d'entrée de données.
- [Liens](#)¹⁶⁵
- [Codes-barres](#)¹⁶⁶
- [Modules de mise en page](#)¹⁷¹
- [La fonction Change-To](#)¹⁸¹. Cette fonction permet qu'un nœud différent soit sélectionné comme correspondance pour un modèle et permet qu'un nœud soit changé en un autre type de contenu.

5.1 Insérer du contenu XML comme texte

Les données du nœud dans le document XML sont incluses dans le design en glissant le nœud de schéma correspondant depuis la fenêtre de l'arborescence de schéma et en le déposant dans le design. Lorsque le nœud de schéma est déposé dans le design, un menu s'ouvre avec les options afin de savoir comment le nœud doit être créé dans le design (*capture d'écran ci-dessous*).



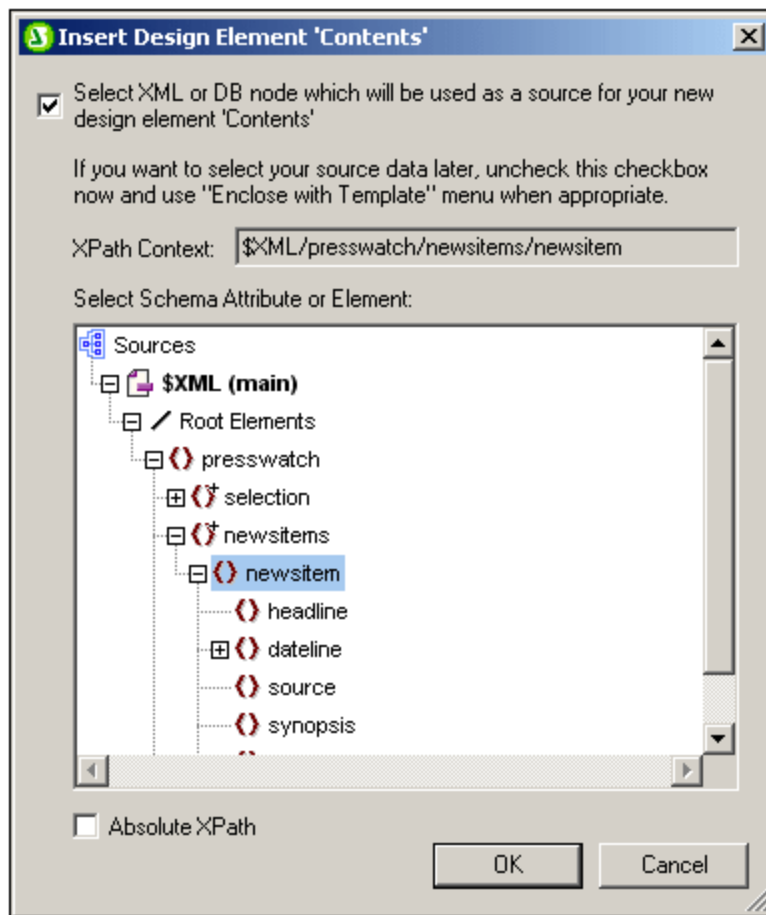
Types de nœuds de schéma

Les nœuds de schéma qui peuvent être déposés depuis la barre latérale de l'arborescence de schéma dans le design ont trois types : (i) les nœuds élément ; (ii) les nœuds attribut ; et (iii) les nœuds datatype.

Utiliser Insérer icône de la barre d'outils des Contenus

L'icône **Insérer Contenu** dans [Insérer barre d'outils des éléments de design](#) ⁴⁴⁷ vous permet également d'insérer du contenu d'un nœud dans le design. Insérer contenus comme suit :

1. Sélectionner Insérer icône des Contenus.
2. Cliquez sur l'emplacement du design où vous souhaitez insérer des contenus. Le Sélecteur Insérer Contenus s'ouvre (*capture d'écran ci-dessous*).



3. Le contexte de l'emplacement de l'insertion dans le design est affiché dans le champ *Contexte XPath*. Sélectionnez le nœud pour lequel vous souhaitez créer des contenus.
4. Cliquez sur **OK**. L'espace réservé de `contenus` est créé. Si le nœud que vous sélectionnez est tout autre chose que le nœud contextuel, des tags modèle supplémentaires avec le chemin vers le nœud sélectionné seront créés autour de l'espace réservé `contenus`.

Sortie du contenu texte des nœuds

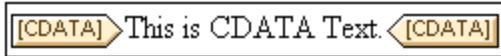
Pour sortir des contenus de texte du nœud, le nœud devrait être créé comme `contenus`. Quand un nœud est créé comme `contenus`, le nœud aura l'air de ceci dans le document design :



Dans la capture d'écran ci-dessous, l'élément `Desc` a été créé comme `contenus`. La sortie affichera le contenu de texte de `Desc`. Si `Desc` a des éléments descendants, tels que *Gras* et *Italique*, alors le contenu du texte des éléments descendants sera également une sortie faisant partie des contenus de `Desc`. Notez que les nœuds attribut de `Desc` ne sont pas considérés comme nœuds enfant et les contenus des nœuds attribut ne doivent pas être des sorties comme faisant partie des contenus de `Desc`. Les nœuds d'attribut doivent être insérés explicitement afin d'être traités.

Sections CDATA

Si les sections CDATA sont présentes dans le document XML, elles seront une sortie.



Note : dans Authentic View, les sections CDATA ne peuvent pas être insérées dans les champs d'entrée (c'est-à-dire des zones de texte et zones de texte multiligne). Elles ne peuvent pas être saisies à l'intérieur des éléments qui sont affichés dans Authentic View comme composants de contenu de texte.

Dans cette section

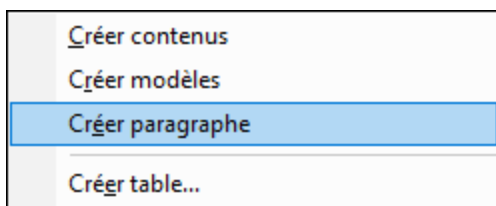
Dans les sous-sections de cette section, nous allons décrire d'autres aspects d'insertion du contenu XML comme texte :

- Comment le contenu de texte d'un nœud peut être [balisé avec un format directement prédéfini](#) ¹¹² quand le nœud est inséré.
- Comment les nœuds descendants qui ne sont pas inclus explicitement au sein d'un nœud peuvent être inclus pour le traitement. Voir [Reste-de-Contenus](#) ¹¹³.

Note : vous pouvez créer une **règle de modèle vide** en supprimant l'espace réservé (contenu) d'un nœud. Une règle modèle vide est utile si vous souhaitez définir qu'un nœud ne soit pas traité, à savoir ne pas produire de sortie.

5.1.1 Insérer du contenu avec un format prédéfini

Le contenu de texte d'un nœud peut être directement inséré avec la balise d'un des formats prédéfinis de StyleVision. À cette fin, glissez le nœud depuis la fenêtre de l'arborescence de schéma et déposez-le à l'emplacement désiré. Dans le menu qui apparaît, sélectionnez **Créer paragraphe** (capture d'écran ci-dessous).



Le format prédéfini peut être modifié en sélectionnant la balise de format prédéfini et en choisissant un autre format prédéfini depuis [Formater zone de liste dans la barre d'outils](#) ⁴⁴⁵ (capture d'écran ci-dessous) ou utiliser la commande de menu **Insérer | Format**.



Le format prédéfini peut aussi être modifié en modifiant la valeur de la propriété `type` de paragraphe du groupe de propriétés *paragraphe* dans la fenêtre des propriétés, ou en changeant le type de paragraphe `type` par le biais de la [commande du menu contextuel](#).²⁴² [Enclose With | Special Paragraph](#)²⁴² du nœud modèle.

Chaque type de paragraphe possède des caractéristiques de formatage particulières qui peuvent être utilisées à votre avantage. Notez que le type de format `pre` permet des retours chariots et des sauts de ligne pour être des sorties au lieu d'être normalisées à des espaces blancs.

5.1.2 Reste-des-Contenus

L'espace réservé `reste-des-contents` applique les modèles à tous les éléments enfant restants de l'élément pour lequel le modèle a été créé. Comme exemple, prenez en compte la chose suivante :

- Un élément `parent` a 4 éléments enfant, `child1` à `child4`.
- Dans le modèle pour l'élément `parent`, le traitement a été explicitement défini pour les éléments enfant `child1` et `child4`.

Ceci résulte en uniquement les éléments enfant `child1` et `child4` étant traités. Les éléments `child2` et `child3` ne seront pas traités. Maintenant, si l'espace réservé `reste-des-contents` est inséré à l'intérieur du modèle pour le `parent`, alors non seulement `child1` et `child4` sont traités utilisant les règles de traitement définies dans le modèle. De plus, les modèles seront appliqués pour les éléments enfant `child2` et `child3`. Si les [modèles globaux](#)²³⁴ pour ceux-ci sont définis, alors les modèles globaux seront utilisés. Autrement, les modèles built-in par défaut (pour l'élément, attribut et nœuds de texte) seront appliqués.

Important : Il est important de noter quels nœuds sont sélectionnés pour le `reste-des-contents`.

- Tel que décrit avec l'exemple ci-dessus, tous les nœuds de l'élément enfant et nœuds de texte enfant sont sélectionnés par l'espace réservé du `reste-des-contents`. (Même des nœuds enfant invalides dans le document XML seront traités.)
- Les nœuds attribut ne sont pas sélectionnés ; ils ne sont pas des nœuds enfant, c'est-à-dire qu'ils ne sont pas sur l'axe enfant de XPath.
- Si un modèle global d'un élément enfant est utilisé dans le modèle parent, alors l'élément enfant ne compte pas comme avoir été utilisé localement. En conséquence, l'espace réservé `reste-des-contents` sélectionnera aussi de tels éléments enfant. Toutefois, si un modèle global d'un élément enfant est « copié localement » alors cette utilisation compte comme utilisation locale, et l'élément enfant ne sera pas sélectionné par l'espace réservé `reste-des-contents`.

Note : vous pouvez créer une **règle de modèle vide** en supprimant l'espace réservé (`contenu`) d'un nœud. Une règle modèle vide est utile si vous souhaitez définir qu'un nœud ne soit pas traité, à savoir ne pas produire de sortie.

5.2 Insérer le contenu MS Word

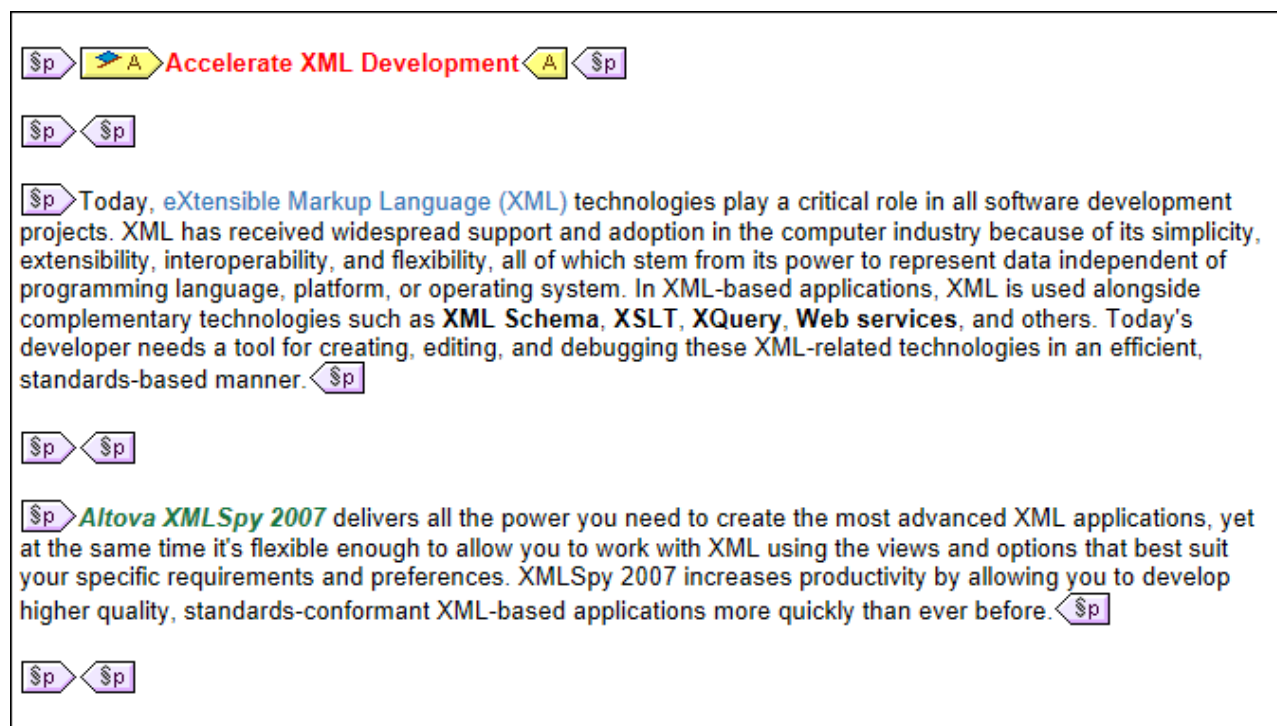
Si Microsoft Word 2007+ est installé sur votre appareil, alors le contenu peut être collé de documents Word dans le design comme **contenu statique**. Le contenu Word sera inséré à l'intérieur des composants de design correspondants appropriés et les propriétés de formatage du texte seront reportées depuis le contenu Word. Par exemple, le contenu texte qui se trouve dans un bloc de paragraphe sera inséré à l'intérieur du [composant de paragraphe](#)¹¹², et le formatage du texte sera préservé (*voir captures d'écran ci-dessous*).

Accelerate XML Development

Today, eXtensible Markup Language (XML) technologies play a critical role in all software development projects. XML has received widespread support and adoption in the computer industry because of its simplicity, extensibility, interoperability, and flexibility, all of which stem from its power to represent data independent of programming language, platform, or operating system. In XML-based applications, XML is used alongside complementary technologies such as **XML Schema**, **XSLT**, **XQuery**, **Web services**, and others. Today's developer needs a tool for creating, editing, and debugging these XML-related technologies in an efficient, standards-based manner.

Altova XMLSpy 2007 delivers all the power you need to create the most advanced XML applications, yet at the same time it's flexible enough to allow you to work with XML using the views and options that best suit your specific requirements and preferences. XMLSpy 2007 increases productivity by allowing you to develop higher quality, standards-conformant XML-based applications more quickly than ever before.

Contenu Word.



Contenu Word collé dans un design. Un format de paragraphe adapté a été appliqué et un formatage de texte a été préservé.

Note : en plus du contenu Word, *tout contenu qui a été collé dans un document Word* peut aussi être collé dans un design StyleVision. Ceci inclut les **tables MS Excel** et le **contenu de la page HTML**.

Note : pour créer un SPS qui contient du contenu statique depuis tout le document Word, créez un nouvel SPS avec la commande [Fichier | Nouveau | Nouveau à partir de Word 2007+](#) ⁴⁵².

Fonctions Word prises en charge

Les structures et formats Word suivants sont pris en charge quand un contenu Word est copié-collé dans le design :

- Texte formaté
 - Différentes polices, taille, poids, style, décoration de texte, etc.
 - Couleur
 - Couleur d'arrière-fond :
 - Bord autour du texte
- Paragraphes
- Sauts de page
- Ligne horizontale
- Lien hypertexte
- Signet
- Tables
 - « Rowspans », « colspans »
 - Contenu formaté/riche
 - Tables intégrées

- *En-têtes, bas de page*
- Listes, sous-listes
 - *À puces : différents styles*
 - *Énuméré : différents styles*
- Images

5.3 Insérer Contenu MS Excel

Si Microsoft Excel 2007+ est installé sur votre appareil, alors le contenu peut être collé de documents Excel dans le design comme **contenu statique**. Le contenu Excel sera inséré comme tables statiques et d'autres composants design correspondants. Les propriétés de formatage seront préservées (*voir les captures d'écran ci-dessous*). Chaque feuille Excel est insérée comme table statique séparée.

	A	B
1	09.03	Euclid's Elements
2	11.09	English Phrasal Verbs in Use
3	08.86	Code Book
4	09.80	Foundations and Fundamental Concepts of Mathematics
5	18.38	Style
6	08.72	The English Language
7	18.36	History of Mathematics
8	06.09	QED
9	07.89	Fowler's Modern English Usage
10	05.30	Oxford Guide to Plain English
11	21.88	Rediscover Grammar
12	11.97	How to Solve It
13	21.54	Advanced Learner's Grammar
14	19.02	Macmillan English Grammar in Context
15	22.57	Oxford Style Manual
16	16.44	Proofreading
17	20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren
18	97.50	Oxford Companion to the Book

Feuille Excel.

Sheet1	
09.03	Euclid's Elements
11.09	English Phrasal Verbs in Use
08.86	Code Book
09.80	Foundations and Fundamental Concepts of Mathematics
18.38	Style
08.72	The English Language
18.36	History of Mathematics
06.09	QED
07.89	Fowler's Modern English Usage
05.30	Oxford Guide to Plain English
21.88	Rediscover Grammar
11.97	How to Solve It
21.54	Advanced Learner's Grammar
19.02	Macmillan English Grammar in Context
22.57	Oxford Style Manual
16.44	Proofreading
20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren
97.50	Oxford Companion to the Book

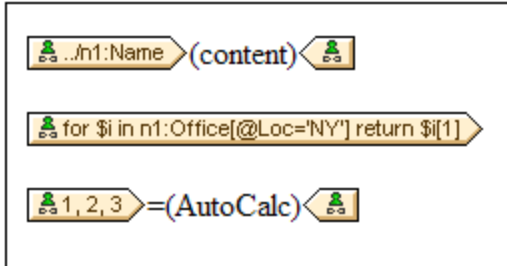
Le contenu Excel importé dans le design comme table statique avec du formatage de texte est préservé.

Note : en plus du contenu Excel, *tout contenu qui peut être collé dans le document Excel* peut aussi être collé dans un design StyleVision. Ceci inclut le **contenu MS Word** et le **contenu de la page HTML**.

Note : pour créer un SPS qui contient du contenu statique depuis tout le document Excel, créez un nouvel SPS avec la commande [Fichier | Nouveau | Nouveau à partir de Excel 2007+](#) ⁴⁵².

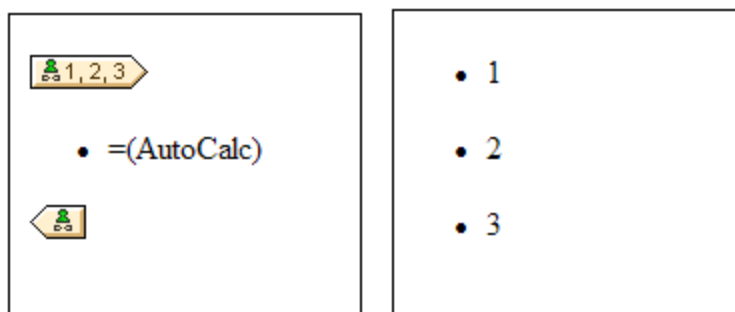
5.4 Modèles définis par l'utilisateur

Les modèles définis par l'utilisateur sont des modèles pour des items générés par une expression XPath que vous spécifiez. Ces items peuvent être des valeurs atomiques ou des nœuds. Dans la capture d'écran ci-dessous, qui affiche trois modèles définis par l'utilisateur, notez l'icône du modèle défini par l'utilisateur à gauche des balises (un symbole de personne en vert). Les modèles définis par l'utilisateur sont très utiles car ils fournissent une flexibilité extraordinaire pour créer les modèles.



L'expression XPath des trois modèles définis par l'utilisateur affichés dans la capture d'écran ci-dessus fait les choses suivantes :

- Sélectionne un nœud dans le schéma de source. En utilisant une expression XPath, tout nœud dans les sources de schéma peut être atteint de l'intérieur de chaque nœud contextuel. Si StyleVision peut cibler sans ambiguïté le nœud spécifié, le modèle sera modifié automatiquement depuis un modèle défini par l'utilisateur vers un modèle normal. S'il s'agit d'un modèle défini par l'utilisateur, ceci sera indiqué par une icône de modèle défini par l'utilisateur en vert du côté gauche des balises de modèle.
- Sélectionne un nœud qui remplit une condition spécifiée par la construction `pour` de XPath 2.0 et XPath 3.0. De tels modèles ne peuvent jamais être résolus en modèles normaux (mais resteront des modèles définis par l'utilisateur) car la construction `for` n'autorise pas StyleVision à résoudre sans ambiguïtés la cible de l'information de schéma uniquement de l'information de schéma dont elle dispose actuellement.
- Sélectionne une séquence de valeurs atomiques {1, 2, 3}. Tandis qu'il est permis de créer un modèle pour une valeur atomique, vous ne pouvez pas utiliser l'espace réservé `contents` à l'intérieur d'un tel modèle. Ceci est dû au fait que l'instruction `xsl:apply-templates` (c'est ce que l'espace réservé `contents` génère) ne peut être appliquée qu'aux items de nœud (pas les valeurs atomiques). Vous devriez, toutefois, utiliser un calcul automatique en combinaison avec un élément de design tel qu'une liste. Par exemple, le modèle défini par l'utilisateur à gauche générerait la sortie à droite.



Note : si le fichier SPS utilise XSLT 1.0, alors l'expression XPath que vous saisissez doit retourner un ensemble de nœud. Autrement une erreur est reportée.

L'avantage d'utiliser XPath pour sélectionner un nœud modèle

L'avantage de sélectionner un nœud de schéma via une expression XPath (modèles définis par l'utilisateur) est que le mécanisme du sélecteur de chemin XPath peut être utilisé pour sélectionner tout nœud ou séquence d'items, de même que filtrer ou définir les conditions pour la sélection de nœud. Le résultat étant que des nœuds de document XML spécifiques peuvent être ciblés pour tout modèle donné. Par exemple, l'expression XPath `//Office/Department[@Location="NY"]` sélectionnera uniquement ces nœuds `Department` qui ont un attribut `Location` avec une valeur de `NY`. Voir également d'autres exemples dans cette section.

Note : Si une expression XPath contient plusieurs étapes de chemin d'accès, il est important — en particulier pour le regroupement et le tri — de déterminer si les parenthèses doivent être placées autour des différentes étapes du chemin d'accès ou non. Par exemple, l'expression XPath `/Org/Office/Dept` sera traitée différemment que `(/Org/Office/Dept)`. Pour l'expression précédente (sans crochets pointus), le processeur transite à travers l'étape de l'emplacement. Pour la dernière expression (avec des crochets pointus), tous les éléments `Dept` de tous les éléments `Office` sont renvoyés en un ensemble de nœud non différencié.

Crochets pointus	Mécanisme XSLT sous-jacent	Effet
Non	<pre><xsl:for-each select="Org"> <xsl:for-each select="Office"> <xsl:for-each select="Dept"> ... </xsl:for-each> </xsl:for-each> </xsl:for-each></pre>	Chaque élément <code>Office</code> a sa propre population <code>Dept</code> . Donc regrouper et trier peut être effectué à l'intérieur de chaque <code>Office</code> .
Oui	<pre><xsl:for-each select="/Org/Office/Dept"> ... </xsl:for-each></pre>	La population <code>Dept</code> s'élargit sur tous les éléments <code>Office</code> et à travers <code>Org</code> .

La différence d'évaluation des expressions XPath peut être significative pour regrouper et trier.

Insérer un modèle défini par l'utilisateur.

Pour insérer un modèle défini par l'utilisateur, faites comme suit :

1. Cliquez sur l'icône **Insérer modèle défini par l'utilisateur** dans la barre d'outils Insérer éléments de design, puis cliquez dans l'emplacement du design là où vous souhaitez insérer le modèle. En alternative, cliquez avec la touche de droite dans l'emplacement de design où vous souhaitez insérer le modèle et, depuis le menu contextuel qui apparaît, sélectionnez la commande **Insérer modèle défini par l'utilisateur**.
2. Dans le dialogue [Éditer expression XPath](#) ⁴²⁶ qui s'ouvre, saisissez l'expression XPath que vous souhaitez et cliquez sur **OK**. Notez que le nœud contextuel de l'expression XPath sera le nœud dans lequel vous avez cliqué. Un modèle de nœud vide sera créé. Dès fois, un nœud joint est créé. Lorsqu'un nœud est joint, les nœuds d'instance ciblés sont sélectionnés comme s'ils étaient à un seul niveau, tandis que si un nœud n'est pas joint (c'est-à-dire s'il est fractionné en de multiples niveaux hiérarchiques), alors la sélection du nœud est effectuée en passant à travers chaque instance à chaque niveau de fractionnement. L'ensemble de nœud dans les deux cas de la sélection (joint et fractionné) est le même sauf si le critère de groupement ou de tri est spécifié. Pour une discussion de l'effet qu'ont des nœuds joints sur le groupement et le tri des mécanismes, voir [Opérations Nœud-](#)

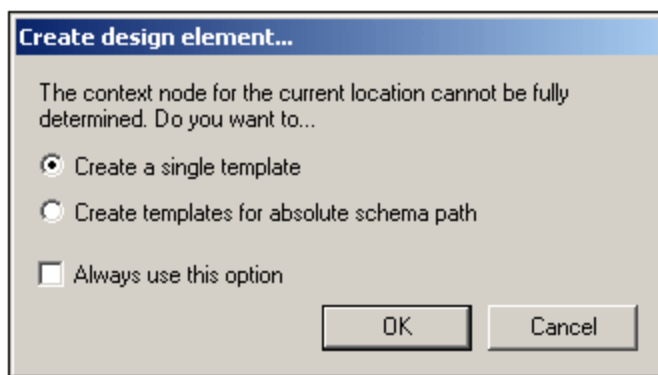
[Modèle](#) ²⁴².

Éditer une correspondance de modèle

La sélection du nœud de tout modèle de nœud (défini par l'utilisateur ou normal) peut être modifiée en utilisant une expression XPath pour sélectionner la nouvelle expression de correspondance. Pour éditer la correspondance de modèle d'un modèle de nœud, cliquez avec la touche de droite sur le modèle de nœud, puis sélectionnez la commande **Éditer correspondance de modèle**. Ceci ouvre le dialogue Éditer expression XPath, dans lequel vous saisissez l'expression XPath peut être saisie. Ensuite, cliquez sur **OK**.

Ajouter des nœuds aux modèles définis par l'utilisateur

Si un nœud de l'arborescence de schéma est ajouté à un modèle défini par l'utilisateur, le contexte pour un nouveau nœud ne sera pas connu si le modèle défini par l'utilisateur a été créé pour un nœud ou une séquence qui ne peut pas être placé dans le contexte de la source de schéma du SPS. Vous serez donc invité pour savoir comment le nouveau nœud (*capture d'écran ci-dessous*) doit être référencé: (i) par son nom (essentiellement, un chemin relatif), ou (ii) par un chemin entier de la racine à la source de schéma.



Le comportement par défaut est de demander un avis afin de voir comment procéder. Ce comportement par défaut peut être changé dans l'onglet de design du [dialogue Outils | Options](#) ⁵³⁵.

5.5 Éléments définis par l'utilisateur, XML, blocs de texte

Les [éléments définis par l'utilisateur](#)¹²² et [blocs de texte définis par l'utilisateur](#)¹²³ permettent, respectivement, à (i) tout élément et (ii) bloc de texte XML d'être insérés dans le design. L'avantage de ces fonctions est que les designers ne sont pas restreints à ajouter des éléments XML et éléments de design depuis les schémas de source et la palette des éléments de design StyleVision. Ils peuvent créer des (i) modèles qu'ils définissent (éléments définis par l'utilisateur), et (ii) du code XML indépendant et autonome (blocs définis par l'utilisateur) qui crée des objets de manière indépendante (par ex., ActiveX objects).

Il existe une différence importante entre les éléments définis par l'utilisateur et les blocs de texte définis par l'utilisateur. Un élément défini par l'utilisateur est créé dans le design comme nœud modèle pour un seul élément XML (avec attributs). Tout le contenu de ce modèle doit être créé de manière explicite. Ce contenu consiste en différents éléments design disponibles dans SPS. Un bloc de texte XML défini par l'utilisateur peut contenir un élément de design ; il s'agit d'un bloc indépendant et autonome. Puisque l'élément défini par l'utilisateur est créé en étant vide, il ne se prête pas à la création d'un objet requérant un nombre de lignes de code. À des fins ultérieures, les blocs de texte définis par l'utilisateur devraient être utilisés.

Note : les éléments définis par l'utilisateur et les blocs de texte définis par l'utilisateur sont pris en charge dans Authentic View uniquement dans les produits de Enterprise Editions d'Altova.

5.5.1 Éléments définis par l'utilisateur

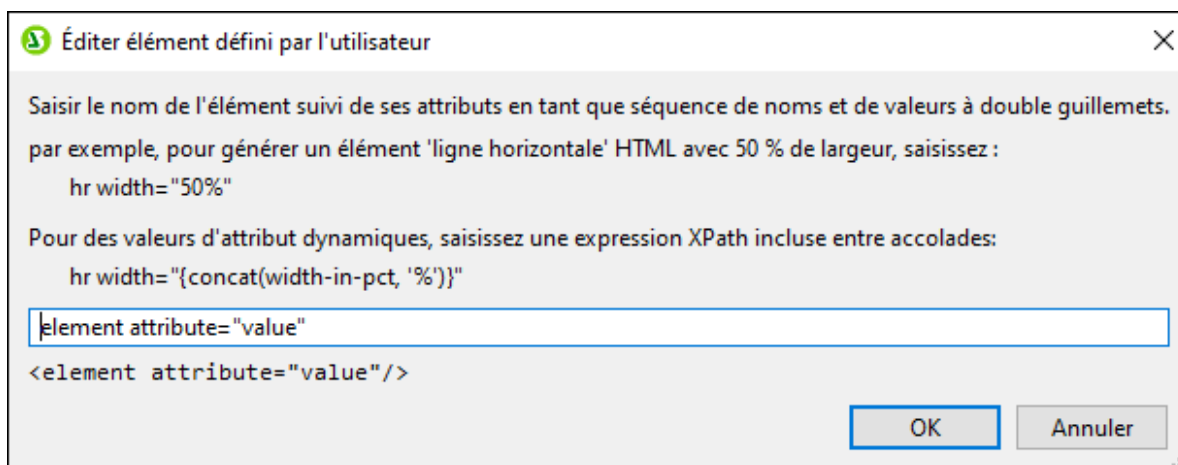
Les éléments définis par l'utilisateur sont des éléments que vous pouvez générer dans la sortie sans que ces éléments aient besoin d'être dans une des sources de schéma de SPS. Ceci signifie qu'un élément de tout espace réservé (HTML ou XSL-FO par exemple) peut être inséré dans tout emplacement dans le design. Les éléments de design SPS peuvent ensuite être insérés à l'intérieur de l'élément inséré.

Note : les éléments définis par l'utilisateur sont pris en charge dans Authentic View uniquement dans les produits de Enterprise Editions d'Altova.

Insérer Éléments définis par l'utilisateur

Le mécanisme pour utiliser les éléments définis par l'utilisateur sont comme suit :

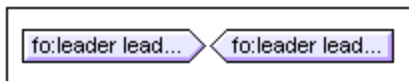
1. Cliquez avec la touche de droite dans l'emplacement du design où vous souhaitez insérer l'élément défini par l'utilisateur.
2. Depuis le menu contextuel qui apparaît, sélectionnez **Insérer élément défini par l'utilisateur | Éléments définis par l'utilisateur**.
3. Dans le dialogue qui apparaît (*capture d'écran ci-dessous*), saisissez le nom de l'élément, les paires valeur-attribut souhaitées, et une déclaration d'espace de nom pour l'élément si le document n'en contient pas.



Dans la capture d'écran ci-dessus, un élément XSL-FO appelé `leader` est créé. Il a reçu un préfixe de `fo:`, qui est lié à la déclaration de l'espace de nom

`xmlns:fo="http://www.w3.org/1999/XSL/Format"`. L'élément a un nombre d'attributs, y compris `leader-length` et `rule-style`, chacun avec sa valeur respective. L'élément, son attribut et la déclaration d'espace de nom doivent être saisis sans les parenthèses angulaires de la balise.

4. Cliquez sur **OK** pour insérer l'élément dans le design. L'élément est affiché dans le design comme modèle vide avec le début et la fin des balises (*capture d'écran ci-dessous*).



5. Vous pouvez ajouter maintenant du contenu au modèle comme pour tout autre modèle. L'élément défini par l'utilisateur peut contenir du contenu statique et dynamique du document XML, de même que plus d'éléments définis par l'utilisateur.

Note : un élément défini par l'utilisateur qui est prévu pour une sortie particulière devrait être enveloppé dans une condition basée sur la sortie pour éviter des résultats inattendus dans les sorties alternatives.

5.5.2 Blocs de texte XML définis par l'utilisateur

Un bloc de texte XML défini par l'utilisateur est un fragment XML qui sera inséré dans le code XSLT généré par SPS. Il est placé dans le design SPS comme bloc autonome auquel aucun élément de design ne peut être ajouté. Un tel bloc de texte XML devrait être applicable pour cette raison comme code XSLT à l'emplacement de la feuille de style dans lequel il apparaît.

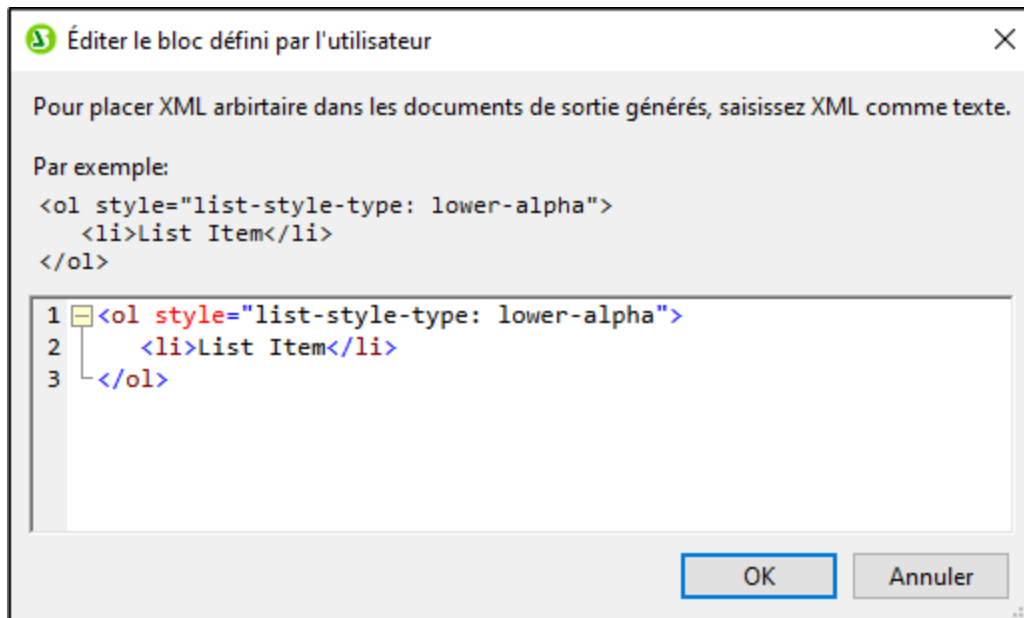
L'utilité de cette fonction est qu'elle fournit un mécanisme au designer de la feuille de style avec lequel vous pouvez insérer des fragments XSLT et un code personnalisé dans le design. Par exemple, un objet ActiveX peut être inséré à l'intérieur de l'élément HTML `SCRIPT`.

Note : cette fonction sera activée **uniquement dans les Enterprise editions d'Authentic View** (à savoir, dans les Enterprise editions de StyleVision, Authentic Desktop, Authentic Browser et XMLSpy).

Insérer les blocs de texte XML définis par l'utilisateur

Pour Insérer un bloc de texte XML, procédez comme suit :

1. Cliquez avec la touche de droite sur l'emplacement dans le design où vous souhaitez insérer le bloc défini par l'utilisateur.
2. Depuis le menu contextuel qui apparaît, sélectionnez **Insérer item défini par l'utilisateur | Bloc défini par l'utilisateur**.
3. Dans le dialogue qui apparaît maintenant (*capture d'écran ci-dessous*), saisissez le bloc de texte XML que vous souhaitez insérer. Notez que le bloc de texte XML devrait être un XML bien formé pour être accepté par le dialogue.



Dans la capture d'écran ci-dessus, un bloc de texte XML est ajouté qui génère une liste HTML ordonnée.

4. Cliquez sur **OK** pour insérer l'élément dans le design. Le bloc de texte XML est affiché dans le design en tant que zone de texte.

Note : un bloc de texte XML qui est prévu pour une sortie particulière devrait être enveloppé dans une condition basée sur la sortie afin d'éviter des résultats inattendus dans les sorties alternatives.

5.6 Tables

Dans un design SPS, deux types de tables peuvent être utilisés : les **tables SPS** et les **tables CALS/HTML**. Il existe des différences entre les deux types, et il est important de les comprendre. Cette section contient une description détaillée pour savoir comment utiliser les deux types de tables.

Tables SPS

Une **table SPS** est le composant d'un design de SPS. Elle est structurée et formatée dans le design. Elle peut être créée partout dans le design et tout nombre de tables SPS peut être créé.

Les tables SPS sont des outils purement de présentation et sont représentées utilisant le vocabulaire de présentation de le format de sortie. La structure d'une table SPS n'est pas **représentée par les nœuds dans le document XML**—bien que le contenu des cellules de la table peut provenir de nœuds dans le document XML.

Il existe deux types de tables SPS :

- Les **tables statiques** sont créées étape par étape par la personne qui conçoit un SPS. Une fois que la structure de table est créée, le contenu de chacune des cellules est défini séparément. Le contenu des cellules peut provenir d'emplacements aléatoires dans l'arborescence de schéma et peut être de types différents. Notez que les lignes d'une table statique ne représentent pas de structure de données répétitive. C'est la raison pour laquelle la table est dite être statique : elle a une structure fixe qui ne change pas avec le contenu XML.
- Les **tables dynamiques** sont prévues pour les structures de données dans le document XML qui se répètent. Elles peuvent être créées pour les éléments de schéma qui ont une sous-structure - c'est-à-dire au moins un attribut enfant ou un élément. Tout élément avec une sous-structure se répète s'il existe plus d'instance de celui-ci. Chaque instance de l'élément serait une ligne dans la table dynamique, et tous ou quelques éléments ou attributs de son enfant seraient les colonnes de la table. Une structure de table dynamique reflète pour cette raison le contenu du fichier XML et change de manière dynamique avec le contenu.

Tables CALS/HTML

Le modèle de contenu d'une table CALS ou d'une table HTML est défini dans le document XML—par extension dans le DTD ou schéma - et suit la spécification respective (CALS ou HTML). Dans le design SPS, vous pouvez préciser que la/les table/s CALS/HTML table/s doit/doivent être traitée/s comme tables. La structure des données XML qui représente la table CALS/HTML générera dans ces cas le balisage de table pour les formats de sortie respectifs. Le formatage des tables CALS/HTML peut être spécifié dans le document d'instance XML ou SPS ou les deux.

Veuillez trouver ci-dessous l'Aperçu HTML d'une table HTML.

Name	Phone
John Merrimack	6517890
Joe Concord	6402387

Le fragment de code HTML pour la table XML affiché dans l'illustration ci-dessous a l'aspect suivant :

```
<table border="1" width="40%">
  <tbody>
    <tr>
      <td>Name</td>
      <td>Phone</td>
    </tr>
    <tr>
      <td>John Merrimack</td>
      <td>6517890</td>
    </tr>
    <tr>
      <td>Joe Concord</td>
      <td>6402387</td>
    </tr>
  </tbody>
</table>
```

Le document XML original peut avoir l'air de ceci :

```
<phonelist border="1" width="40%">
  <items>
    <person>
      <data>Name</data>
      <data>Phone</data>
    </person>
    <person>
      <data>John Merrimack</data>
      <data>6517890</data>
    </person>
    <person>
      <data>Joe Concord</data>
      <data>6402387</data>
    </person>
  </items>
</phonelist>
```

Notez que les noms d'élément dans le document XML n'ont pas besoin d'avoir de la sémantique de table ; la **structure de table**, toutefois, doit correspondre au modèle de table HTML ou CALS. Veuillez noter les éléments suivants :

- Notez que seul un élément XML peut correspondre à l'élément de colonne HTML <td/>.
- Une table CALS/HTML peut être insérée dans tout emplacement dans le document XML où, selon le schéma, l'élément correspondant à l'élément de table est permis.
- Dans Authentic View, les données sont saisies directement dans les cellules de table. Ces données sont stockées comme contenu de l'élément de table CALS/HTML correspondant.
- Les propriétés de formatage de la table CALS/HTML pourraient provenir du document XML ou pourraient être spécifiées dans le design SPS.

Résumé pour le designer

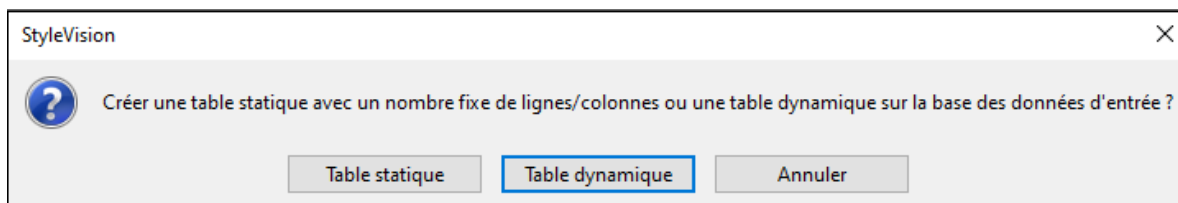
De la perspective du concepteur du document, les points suivants devraient être notés :

- La **structure** d'une table SPS est définie dans la SPS. D'autre part, la structure d'une table CALS/HTML est spécifiée dans le schéma et doit suivre le modèle de table CALS/HTML ; les noms d'éléments dans le schéma peuvent toutefois être différents que ceux dans les modèles de table CALS ou HTML.
- Les **Colspans et rowspans** dans les tables SPS sont spécifiés dans la SPS. Mais dans les tables CALS/HTML, les colspans et rowspans sont spécifiés dans le document d'instance XML.
- Le **formatage de table** de tables SPS est spécifié dans la SPS. Le formatage des tables CALS/HTML est spécifié dans le document d'instance XML et/ou la SPS.

5.6.1 Tables statiques

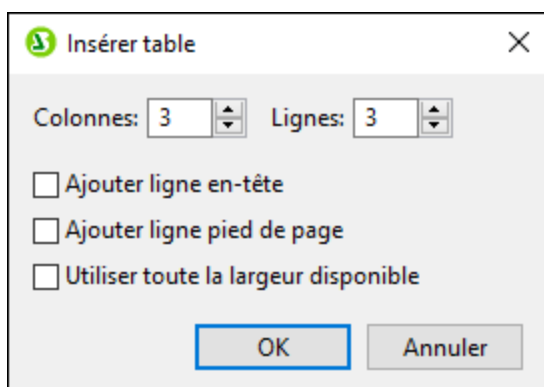
Pour créer une table statique, procédez comme suit :

1. Utiliser une des commandes suivantes : **Table | Insérer Table** ou **Insérer | Table** ou cliquez sur l'icône **Insérer Table**  dans la barre d'outils Insérer des Éléments Design.
2. Toutes ces commandes ouvre le dialogue Créer Table (*capture d'écran ci-dessous*).



Cliquez sur **Table statique**.

3. La dialogue Insérer Table (*capture d'écran ci-dessous*) s'ouvre, dans lequel vous précisez les dimensions de la table et précisez si la table pourrait occuper toute la largeur disponible.



4. Cliquez sur OK. Une table vide avec les dimensions spécifiées, telle qu'affichée ci-dessous, est créée.

5. Vous pouvez désormais saisir du contenu dans les cellules de table utilisant des fonctions régulières StyleVision. Le contenu de la cellule pourrait être du texte ou des éléments déplacés de l'arborescence de schéma ou des objets or tels que des images ou tables intégrées. La figure ci-dessous affiche une table contenant des tables intégrées.

Personne	Téléphone	Fax												
	<table><tr><td></td><td></td></tr><tr><td>Bureau</td><td>Domicile</td></tr><tr><td></td><td></td></tr></table>			Bureau	Domicile			<table><tr><td></td><td></td></tr><tr><td>Bureau</td><td>Domicile</td></tr><tr><td></td><td></td></tr></table>			Bureau	Domicile		
Bureau	Domicile													
Bureau	Domicile													

Les tables statiques SPS sont idéales pour organiser les données XML qui sont situées dans la hiérarchie de schéma de manière aléatoire, ou pour le contenu statique (contenu qui n'est pas dérivé depuis une source XML).

Supprimer les colonnes, lignes et tables

Pour supprimer une colonne, une ligne ou une table, placez le curseur dans la colonne, la ligne ou la table à supprimer et cliquez sur l'item de menu **Table | Supprimer colonne**, **Table | Supprimer ligne**, ou **Table | Supprimer table**, respectivement. Si vous avez des tables intégrées, ces commandes s'appliqueront respectivement, à la colonne, la ligne et la table contenant le curseur.

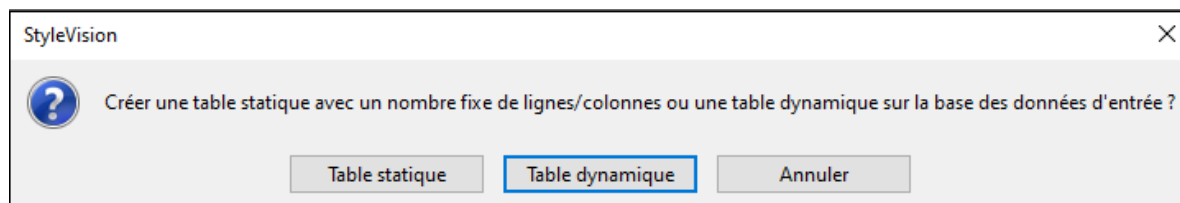
Icônes d'édition de la table de la barre d'outils

La table éditant les icônes, qui sont par défaut dans la deuxième ligne de la barre d'outils, sont des raccourcis des commandes de menu de la **Table**. Ces commandes vous permettent d'insérer, de supprimer, d'éditer la structure des et d'assigner les propriétés de formatage à la table statique. Ces icônes peuvent aussi être utilisées pour des tables SPS dynamiques.

5.6.2 Tables dynamiques

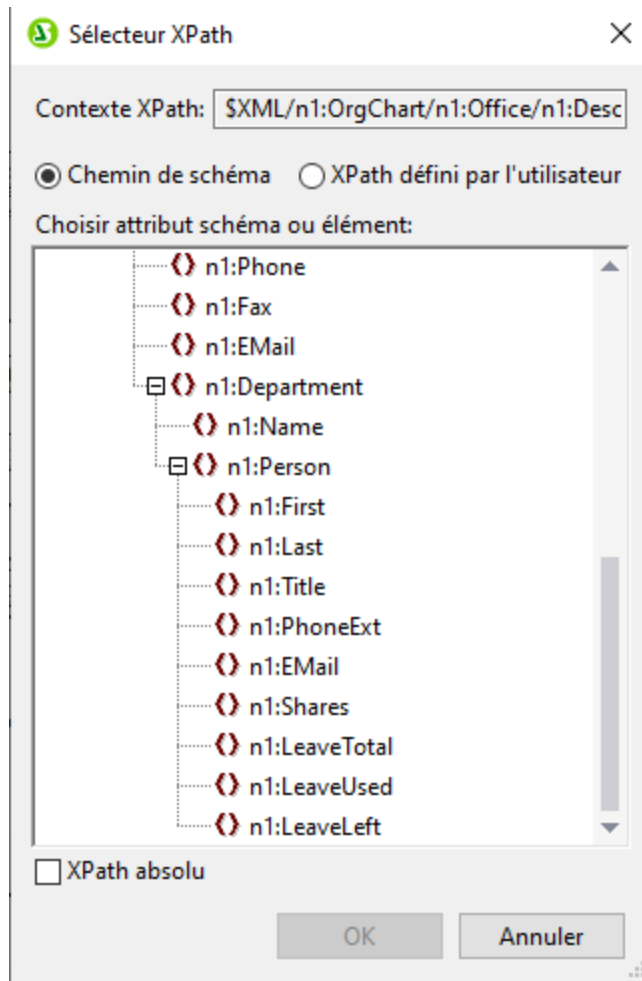
Pour Insérer une table dynamique, procédez comme suit :

1. Utiliser une des commandes suivantes : **Table | Insérer Table** ou **Insérer | Table** ou cliquez sur l'icône **Insérer Table**  dans la barre d'outils Insérer des Éléments Design.
2. Toutes ces commandes ouvre le dialogue Créer Table (*capture d'écran ci-dessous*). Si vous avez cliqué sur l'icône Insérer Table dans la barre d'outils, le dialogue Créer Table s'ouvre quand vous cliquez sur l'emplacement dans le design où vous voulez insérez la table.



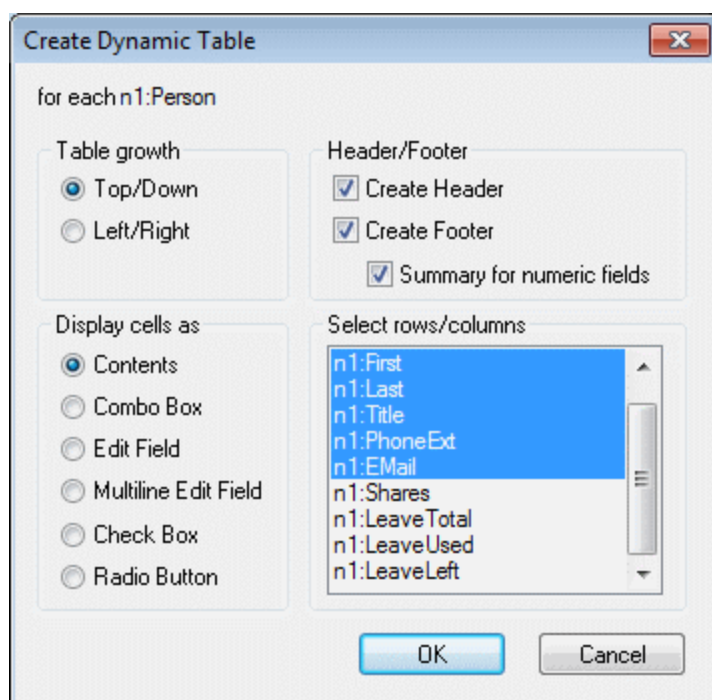
Cliquez sur **Table dynamique**.

3. Dans le dialogue Sélecteur XPath (*capture d'écran ci-dessous*) qui s'ouvre, notez que le contenu XPath est le contexte de l'emplacement d'insertion, et ne peut pas être modifié dans le dialogue. Sélectionnez le nœud à créer comme table dynamique. Dans la capture d'écran ci-dessous, le nœud contextuel est `n1:Department`, et le nœud `n1:Person` a été sélectionné comme nœud à créer pour une table.



Si vous sélectionnez l'option XPath défini par l'utilisateur, vous pouvez alors saisir une expression XPath pour sélectionner le nœud à créer en tant que table dynamique.

4. Cliquez sur **OK**. Ceci fait apparaître le dialogue Créer Table dynamique (*capture d'écran ci-dessous*).



5. Les éléments enfant et attributs de l'élément qui ont été glissés ans la fenêtre Design sont affichés dans la liste « Sélectionner lignes/colonnes » et peuvent être créés comme colonnes de la table. Décochez les nœuds enfant que vous ne voulez pas et sélectionnez tout attribut/élément que vous souhaitez inclure comme colonnes. (Dans la figure ci-dessus, les éléments *Shares*, *LeaveTotal*, *LeaveUsed* et *LeaveLeft* ont été décochés.) Une explication des autres options est donnée ci-dessous. Cliquez sur **OK** une fois que vous avez terminé. Notez que les colonnes sont créées pour des éléments enfant et attributs et pour aucun descendant à un niveau inférieur.

Note : si vous avez précisé une expression XPath définie par l'utilisateur pour sélectionner le nœud à créer comme table dynamique, alors StyleVision ne saura probablement pas de manière explicite quel nœud est ciblé. De manière conséquente, Créer Table dynamique n'affichera pas dans de tels cas une liste d'attributs/éléments enfant à sélectionner comme champs (colonnes) de la table. La table qui est créée devra donc être peuplée avec du contenu de nœud. Ce contenu de nœud devrait être des attributs/éléments enfant du nœud sélectionné à créer comme table.

Note : une autre manière de créer un nœud de schéma comme table est de glisser le nœud depuis l'arborescence de schéma dans le design et de spécifier qu'il soit créé comme table lorsque vous le déposez.

La table s'agrandit vers le bas et vers la droite

Lorsqu'une table s'agrandit du haut vers le bas, ceci est l'aspect qu'elle a :

name	street	city	state	zip
ipo:name (contents) ipo:name	ipo:street (contents) ipo:street	ipo:city (contents) ipo:city	ipo:state (contents) ipo:state	ipo:zip (contents) ipo:zip

Lorsqu'une table s'agrandit de gauche à droite, elle a l'air de ceci :

name	 (contents) 
street	 (contents) 
city	 (contents) 
state	 (contents) 
zip	 (contents) 

En-têtes et pieds de page

Les colonnes et lignes peuvent avoir des en-têtes qui seront les noms de la colonne et les éléments de ligne. Les en-têtes de colonnes sont créées au niveau supérieur de chaque colonne. Les en-têtes de lignes sont créées à gauche de la ligne. Pour inclure les en-têtes, vérifiez la case à cocher **Créer en-tête**. Si la table s'agrandit du haut vers le bas, créant une en-tête, ceci crée une ligne d'en-tête au-dessus du corps de la table. Si la table s'agrandit du gauche vers la droite, créant une en-tête, ceci crée une colonne d'en-tête au-dessus du corps de la table.

Pour inclure les bas de page, vérifiez la case à cocher **Créer bas de page**. Les bas de page, tout comme les en-têtes, peuvent être créés pour des colonnes (en bas de colonnes) et lignes (à droite d'une ligne). Le bas de page des colonnes numériques ou lignes additionnera chaque colonne ou ligne si la case à cocher *Résumé pour des champs numériques* est cochée.

Par le biais du menu **Table**, les cellules d'en-tête et de bas de page peuvent être jointes et fractionnées, et les lignes et colonnes peuvent être insérées, ajoutées et supprimées ; ceci donne une flexibilité considérable pour la structure des en-têtes et des bas de page. De plus, les en-têtes et bas de page peuvent contenir tout type de contenu statique ou dynamique, y compris des modèles conditionnels et des calculs automatiques.

Note : Les en-têtes et bas de page doivent être créés quand la table dynamique est définie. Vous faites ceci en vérifiant les options **Créer en-tête** et **Créer bas de page** dans le dialogue **Créer Table dynamique**. Ajouter ou insérer une ligne à l'intérieur d'une table dynamique ne crée pas d'en-têtes ou de bas de page mais une ligne supplémentaire. La différence est significative. Avec les commandes **Créer En-tête/Base de page**, de vrais en-têtes et bas de page sont ajoutés en haut et en bas de la table, respectivement. Si une ligne est insérée ou ajoutée, alors la ligne apparaît pour chaque occurrence de l'élément qui a été créée comme table dynamique.

Tables dynamiques intégrées

Vous pouvez intégrer une table dynamique à l'intérieur d'une autre table dynamique si l'élément pour lequel la table dynamique doit être créé est un enfant de l'élément qui a été créé comme table dynamique contenant. Procédez comme suit :

1. Créez la table dynamique extérieure pour que l'élément enfant à créer comme table dynamique est créé comme colonne.
2. Dans la table dynamique du Mode Design, cliquez avec la touche de droite sur l'élément enfant.
3. Sélectionnez **Changer à | Table**. Ceci fait apparaître le dialogue **Créer Table dynamique**.
4. Définissez les propriétés de la table dynamique intégrée.

Pour intégrer la table dynamique dans une table statique, glissez l'élément à créer comme table dynamique dans la cellule requise de la table statique. Quand vous la déposez, sélectionnez **Créer Table** depuis le menu contextuel qui apparaît.

Tables pour les éléments avec contenu textuel

Pour créer des colonnes (ou lignes) pour les éléments enfant, l'élément étant créé comme table doit avoir un **élément enfant ou nœud attribut**. Avoir un **nœud de texte enfant** ne fonctionne pas. Si vous vous retrouvez dans une telle situation, créez un élément enfant appelé, disons `Text`, et mettez votre nœud de texte dans les éléments `TableElement/Text`. Désormais, vous serez capable de créer `TableElement` comme table dynamique. Cette table aura une colonne pour les éléments `Text`. Chaque ligne contiendra pour cette raison une cellule contenant le nœud texte dans `Text`, et les lignes de la table correspondront aux occurrences de l'élément `TableElement`.

Contenus des cellules de corps de la table

Quand vous créez une table dynamique, vous pouvez créer le contenu du nœud comme tout nombre des composants `StyleVision`. Dans les exemples ci-dessus, les cellules du corps de la table ont été créées comme contenus ; dans le dialogue Créer Table dynamique, l'option pour Afficher cellules comme est *contenus*. Elles ont pu aussi être créées comme outils de saisie des données. Il y a deux points à noter ici :

- La paramètre que vous sélectionnez est un paramètre global pour toutes les cellules du corps de la table. Si vous souhaitez avoir une cellule individuelle qui apparaît différemment, éditez la cellule après avoir créé la table : cliquez avec la touche de droite sur la cellule et, dans le menu contextuel qui apparaît, sélectionnez « Changer à », puis sélectionnez le type de contenu de la cellule requis.
- Si vous créez des cellules comme contenus d'éléments, et si l'élément a des éléments descendants, alors le contenu de la cellule sera une concaténation de strings de texte de l'élément et de tous ses éléments descendants.

Supprimer les colonnes, lignes et tables

Pour supprimer une colonne, une ligne ou une table, placez le curseur dans la colonne, la ligne ou la table à supprimer et cliquez sur l'item de menu **Table | Supprimer colonne**, **Table | Supprimer ligne**, ou **Table | Supprimer table**, respectivement. Si vous avez des tables intégrées, la table contenant immédiatement le curseur sera supprimée quand la commande **Table | Supprimer Table** est utilisée.

Icônes d'édition de la table de la barre d'outils

La table éditant les icônes dans la barre d'outils sont des raccourcis des commandes de menu de la **Table**. Ces commandes vous permettent d'insérer, de supprimer, d'éditer la structure des et d'assigner les propriétés de formatage à la table dynamique. Ces icônes peuvent aussi être utilisées pour des tables statiques.

Créer des tables dynamiques dans les modèles globaux

Vous pouvez également créer des tables dynamiques sur des éléments à l'intérieur des modèles globaux. Le processus fonctionne de la même manière que les éléments du modèle principal (décrits ci-dessus). Le point important à noter est que, dans le modèle global, une table dynamique ne peut être créée que pour des **éléments descendants** du nœud modèle global ; il ne peut pas être créé pour le nœud modèle global lui-même. Par exemple, si vous souhaitez créer une table dynamique pour l'élément `authors` à l'intérieur du modèle global, alors la table dynamique doit être créée à l'intérieur du modèle global de l'élément parent de `authors`, disons `contributors`. Elle ne peut pas être créée à l'intérieur du modèle global de l'élément `authors`.

5.6.3 Traitement conditionnel dans les tables

Le traitement conditionnel peut être défini dans des colonnes et lignes individuelles de tables statiques et dynamiques, de même que dans une colonne et des en-têtes de ligne afin d'afficher ou de masquer la colonne, la ligne ou l'en-tête dépendant de la vérité de la condition. Si la condition évalue en tant que true, les colonne, ligne ou en-tête est affichée. Autrement, elle ne l'est pas.

Ajouter et éditer le traitement conditionnel

Pour ajouter le traitement conditionnel à une colonne, ligne ou un en-tête, cliquez avec la touche de droite dans le composant de design respectif et sélectionnez **Éditer traitement conditionnel**. (Dans la capture d'écran ci-dessous, l'en-tête de colonne et composant de design en haut à gauche est affiché en bleu ; le composant de design de la deuxième colonne est affiché en bleu ; le seul composant de ligne se trouve sous le composant design de l'en-tête de colonne.)

First	Last	Email
first (content) first	last (content) last	email (content) email

Cliquez sur la commande **éditer le traitement conditionnel** ouvre le [dialogue Éditer l'expression XPath](#)⁴²⁶, dans lequel vous saisissez l'expression XPath de la condition. Ici, vous trouverez quelques manières dont le traitement conditionnel peut être utilisé.

- Dans une colonne, ligne ou table, saisissez l'expression XPath `false()` pour masquer la colonne, `true()` pour l'afficher.
- Une colonne est la sortie uniquement si la somme de toutes les valeurs dans la colonne dépasse une certaine valeur entière.
- Une colonne ou une ligne est la sortie uniquement si aucune cellule dans cette colonne ou ligne, respectivement, est vide.
- Une colonne ou une ligne est la sortie uniquement si une certaine valeur de cellule existe dans cette colonne ou ligne, respectivement.

Pour éditer une condition déjà créée, cliquez avec la touche de droite sur le composant design respectif et sélectionnez **Éditer traitement conditionnel**. Dans le [dialogue Éditer expression XPath](#)⁴²⁶ qui apparaît, éditez l'expression XPath qui teste la vérité de la condition.

Supprimer le traitement conditionnel

Pour supprimer le traitement conditionnel d'une colonne, ligne ou en-tête, cliquez avec la touche de droite dans le composant design respectif et sélectionnez **Supprimer traitement conditionnel**.

5.6.4 Tables dans Aperçu de design

Les composants principaux des tables SPS statiques et dynamiques sont affichés dans les captures d'écran ci-dessous avec le markup de table (**Table | Aperçu balisage de table**) activé.

Header-C1	Header-C2
Footer-C1	Footer-C2

La capture d'écran ci-dessus affiche une table simple qui s'agrandit du haut vers le bas et qui a un en-tête et un bas de page.

- Une colonne est indiquée avec un rectangle contenant une flèche pointant vers le bas. Les indicateurs de colonnes sont situés au niveau supérieur des colonnes. Pour sélectionner une colonne entière — disons, pour attribuer une propriété de formatage à la colonne entière - cliquez sur l'indicateur de colonne de cette colonne.
- Une ligne est indiquée avec un rectangle contenant une flèche pointant vers le bas. Cliquez un indicateur de ligne pour sélectionner cette ligne entière.
- Dans les tables qui s'agrandissent du haut vers le bas (*capture d'écran ci-dessus*), les en-têtes et bas de page sont indiqués avec les icônes pointant vers le haut et le bas, respectivement. Dans les tables qui s'agrandissent de gauche vers la droite, les en-têtes et bas de page sont indiqués avec des icônes pointant vers la gauche et la droite, respectivement (*capture d'écran ci-dessous*).
- Pour sélectionner toute la table, cliquez dans le coin supérieur gauche de la table (dans les captures d'écran ci-dessus et ci-dessous, l'emplacement où pointe le curseur de la flèche).
- Quand une ligne ou colonne de table est sélectionnée, elle est mise en surbrillance avec un arrière-plan bleu foncé. Dans la capture d'écran ci-dessus, le bas de page est sélectionné.
- Dans les tables qui s'agrandissent du haut vers le bas, l'élément pour lequel la table a été créée est affiché tout à gauche, à l'extérieur de la grille de la ligne de colonne (*capture d'écran ci-dessus*). Dans les tables qui s'agrandissent de gauche à droite, l'élément pour lequel la table a été créée est affiché en haut, à l'extérieur de la grille de la ligne de colonne (*capture d'écran ci-dessous*).

Header-R1	Header-R2
Footer-R1	Footer-R2

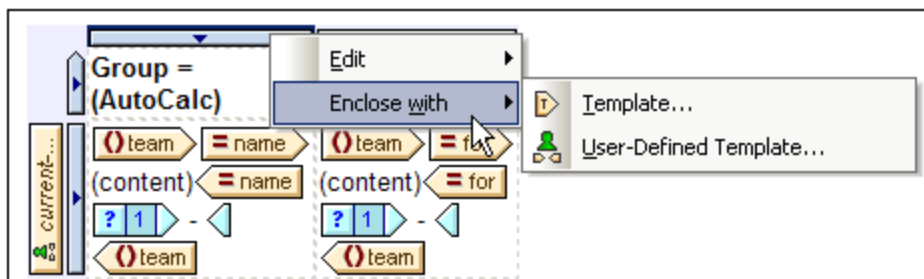
Une fois qu'une colonne ou une ligne ou une table a été sélectionnée, les styles et/ou propriétés peuvent être définis pour la sélection dans les fenêtres Styles et Propriétés.

Fonction glisser/déposer

Les colonnes et lignes d'une table SPS (statique ou dynamique) peuvent être glissées dans des emplacements alternatifs à l'intérieur de la même table et déposées ici.

Contenir et supprimer des modèles sur des lignes et colonnes

Une ligne ou une colonne peut être contenue dans un modèle en cliquant avec la touche droite de la souris sur la ligne ou l'indicateur de colonne et, depuis le menu contextuel qui s'ouvre (*capture d'écran ci-dessous*), sélectionnant **Contenant dans | Modèle** ou **Contenant dans | Modèle défini par l'utilisateur**. À l'étape suivante, vous pouvez sélectionner un nœud depuis l'arborescence de schéma ou saisir une expression XPath pour un [modèle défini par l'utilisateur](#)²³⁹. Un modèle sera créé autour de la ligne ou de la colonne.



Un modèle qui est autour de la ligne ou de la colonne peut aussi être supprimé tout en laissant la ligne ou la colonne intacte. À cette fin, sélectionnez la balise de modèle et appuyez sur la clé **Supprimer**.

La fonction contenir dans et supprimer des modèles est utile si vous souhaitez supprimer un modèle sans supprimer les contenus d'une ligne ou d'une colonne, et, si requis, contenir la ligne ou la colonne dans un autre modèle. Contenir dans un [modèle défini par l'utilisateur](#)²³⁹ permet aussi d'utiliser des résultats de correspondance de modèles intéressants à l'intérieur de la ligne ou de la colonne (via des calculs automatiques, par exemple).

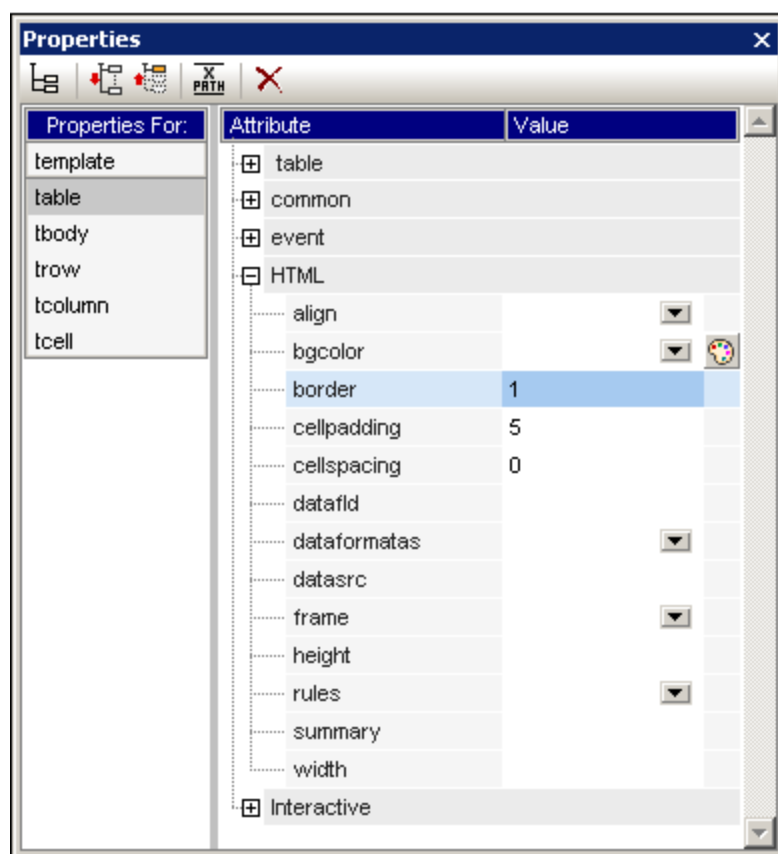
5.6.5 Formatage de table

Les tables statiques et dynamiques peuvent être formatées utilisant :

- Propriétés de formatage de la table HTML (dans la barre latérale des propriétés)
- Propriétés CSS (styling) (dans la barre latérale des Styles).

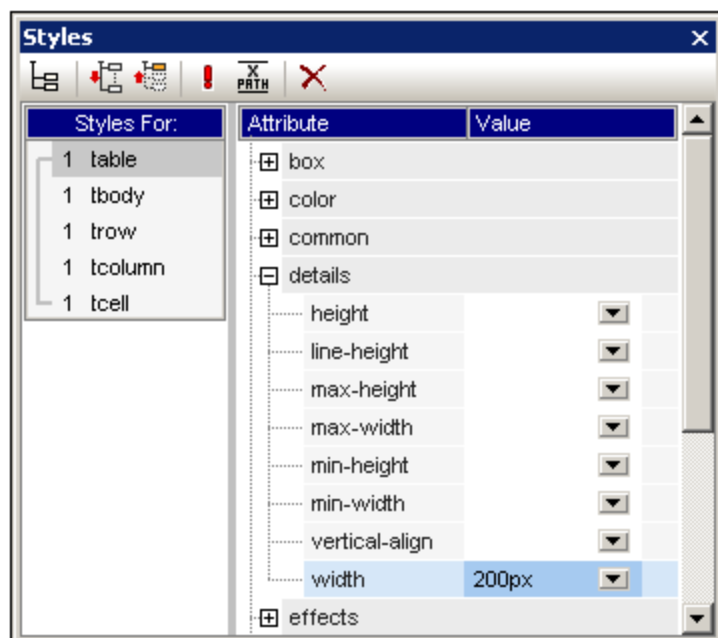
Barre latérale des propriétés

Les propriétés de formatage de la table HTML sont disponibles dans la barre latérale des propriétés (*capture d'écran ci-dessous*). Ces propriétés sont disponibles dans le groupe des propriétés *HTML* pour le composant de table et ses sous-composants (corps, ligne, colonne et cellule).



Barre latérale Styles

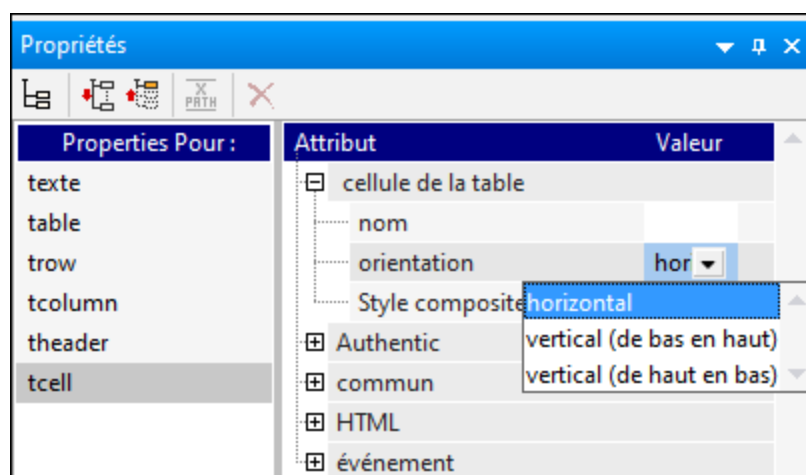
Les propriétés de formatage de la table CSS sont disponibles dans la barre latérale des propriétés (*capture d'écran ci-dessous*). Les propriétés CSS sont disponibles pour le composant de table et ses sous-composants (corps, ligne, colonne et cellule).



Note : si les cellules de table dans une ligne sont vides, Internet Explorer réduit la ligne et la ligne peut ne plus être visible. Dans ce cas, vous devriez utiliser la solution HTML qui est de mettre un espace « non-breaking » dans la/les cellule/s.

Texte vertical

Le texte dans les cellules de table peut être tourné à 90 degrés dans le sens de la montre ou dans le sens inverse de la montre pour que le texte soit vertical, se lisant du haut vers le bas et du bas vers le haut, respectivement. À cette fin, dans le design, sélectionnez le contenu dans la cellule de table qui doit être tourné et, dans la barre latérale des propriétés (*capture d'écran ci-dessous*), sélectionnez `tcell`. Dans le groupe de propriétés *Cellule de table*, sélectionnez la valeur requise pour la propriété *Orientation*.



Notez les points suivants :

- La rotation sera appliquée dans la sortie, mais ne sera pas affichée dans le design.
- Cette propriété est prévue être appliquée au texte et ne doit pas être utilisée pour un autre contenu.

- Outre le fait d'être appliquée au texte dans les cellules de table, la propriété peut aussi être appliquée au texte dans les [champs de texte](#) ¹⁷⁷.

Formatage de table via les Propriétés et Styles

Certaines propriétés de formatage sont disponibles dans la barre latérale des Propriétés ainsi que dans la barre latérale des Styles. La table ci-dessous recense quelques-unes des propriétés de table plus importantes disponibles dans les deux barres latérales.

Composant de table	Barre latérale des propriétés	Barre latérale Styles
Table	bord, cadre, règles ; cellpadding, cellspacing; bgcolor; hauteur, largeur (<i>écrasé par la hauteur, la largeur dans la barre latérale Styles, si cette dernière existe</i>); aligner	bords et padding dans les styles Box ; hauteur, largeur dans les groupes Details (<i>ils écrasent la hauteur et la largeur dans la barre latérale des Propriétés</i>); couleur, police, et styles de texte
Body	align, valign	height, vertical-align; color, font, and text styles
Colonne	align, valign	width, vertical-align; color, font, and text styles; box styles
Ligne	align, valign	height, vertical-align; color, font, and text styles; box styles
Cellule	align, valign	height, width, vertical-align; color, font, and text styles; box styles

Hauteur et largeur

La hauteur et la largeur des tables, lignes, colonnes et cellules doivent être définies dans la barre latérale Styles (dans le groupe de style Details). Quand une table, une colonne ou une ligne est redimensionnée dans l'affichage en utilisant la souris, les valeurs modifiées sont saisies automatiquement dans le style approprié dans la barre latérale Styles. Notez, toutefois, que les styles hauteur et largeur ne sont pas pris en charge pour les cellules qui sont couvertes (row-spanned ou column-spanned).

Centrer une table

Pour centrer une table, définissez la propriété `align` dans le groupe HTML des propriétés au centre. La propriété `align` peut être accédée en sélectionnant la table, puis en sélectionnant la commande de menu **Table | Propriétés de Table**. En alternative, la propriété est disponible dans le groupe de propriétés HTML dans la barre latérale des Propriétés.

Centrer la sortie PDF nécessitera des paramètres supplémentaires selon le processeur FOP que vous utilisez. Selon la spécification FO, la manière correcte de centrer une table est d'envelopper l'élément `fo:table` avec un élément `fo:table-and-caption` est de définir l'attribut `text-align` de l'élément `fo:table-and-caption` au centre. Stylevision ne crée pas automatiquement d'élément `fo:table-and-caption` quand une table est insérée dans le design, mais vous pouvez ajouter cet élément comme [Élément défini par l'utilisateur](#) ¹²². Si vous utilisez toutefois le processeur Apache FOP, vous devriez noter que l'élément `fo:table-and-caption` peut éventuellement ne pas être pris en charge, dépendant de la version FOP que vous utilisez. Dans ce cas, il

existe une solution de contournement : Transformez la table en table d'une largeur fixe. Faites ceci en spécifiant une valeur de longueur, telle que 4in ou 120mm, comme la valeur de la propriété `largeur` du groupe HTML des propriétés de table (accédée via la commande de menu **Table | Propriétés de Table**).

Donner différentes couleurs d'arrière-plan aux lignes alternantes

Si vous voulez alterner les couleurs d'arrière-plan pour les lignes de votre table dynamique, suivez les étapes suivantes :

1. Sélectionnez l'indicateur de ligne de la ligne pour laquelle les couleurs d'arrière-plan sont requises. N'oubliez pas qu'étant donné qu'il s'agit d'une table dynamique, un élément est créé comme ligne, et le design contient une seule ligne, qui correspond à l'élément étant créé comme table.
2. Avec l'indicateur de ligne sélectionné, dans la barre latérale des Propriétés, cliquez sur les *Propriétés pour:* `tr`.
3. Sélectionnez la propriété `bgcolor`.
4. Cliquez sur l'icône XPath dans la barre d'outils de la fenêtre des Propriétés, et, dans le [dialogue Éditer expression](#) ⁴²⁶ qui apparaît, saisissez une expression XPath semblable à ceci :

```
si ( position() mod 2 = 0 ), alors "white" sinon "gray"
```

Cette expression XPath spécifie une `bgcolor` de lignes blanches numérotées avec des chiffres pairs et une `bgcolor` en gris pour les lignes numérotées avec des chiffres impairs

Vous pouvez élargir le principe ci-dessus afin de fournir un formatage bien plus complexe.

Numéroter les lignes d'une table dynamique

Vous pouvez numéroter les lignes d'une table dynamique en utilisant la fonction `position()` de XPath. Pour ce faire, insérez d'abord une colonne dans le tableau pour contenir les nombres, puis insérez un calcul automatique dans la cellule de cette colonne avec un XPath de `:position()`. Puisque le nœud contextuel est l'élément qui correspond à la ligne d'une table dynamique, la fonction `position()` renvoie la position de chaque élément de ligne dans l'ensemble de tous les éléments de lignes.

En-têtes de table et de pied de page dans la sortie PDF

Si la table passe sur plus d'une page, alors l'en-tête et le pied de page apparaissent sur chaque page qui contient la table. Veuillez noter les points suivants :

- Si le pied de page contient des calculs automatiques, le pied de page qui apparaît à la fin du segment de table sur chaque page contient les calculs automatiques pour toute la table - et non ceux pour uniquement le segment de table sur cette page.
- L'en-tête et le pied de page ne seront pas désactivés pour des pages individuelles (par exemple, si vous voulez un pied de page uniquement à la fin de la table et non à la fin de chaque table).

Afin d'omettre que l'en-tête ou le pied de page soit affiché à chaque saut de page, utilisez les propriétés `table-omit-header-at-break` et/ou `table-omit-footer-at-break` (attributs) dans l'élément `table`. Ces propriétés sont disponibles dans la barre latérale Styles, dans le groupe XSL-FO des propriétés pour la table. Pour omettre l'en-tête ou le pied de page lors de sauts de page, spécifiez une valeur de `true` pour l'attribut respectif. (Notez que la valeur par défaut est `false`. Donc ne pas préciser ces propriétés a l'effet d'insérer les en-tête et bas de page quand il y a un saut.)

Mettre un trait d'union au contenu des cellules de table

Si vous désirez mettre un trait d'union à un texte dans les cellules de table de votre sortie PDF, notez que la spécification XSL:FO utilise l'attribut `mettre-un-trait-d-union` de l'élément `fo:block` à cette fin. Donc, pour appliquer la mise en place du trait d'union, vous devez définir explicitement, dans le document FO, la propriété `hyphenate` des éléments respectifs `fo:block` à `true`.

5.6.6 Affichage Ligne et Colonne

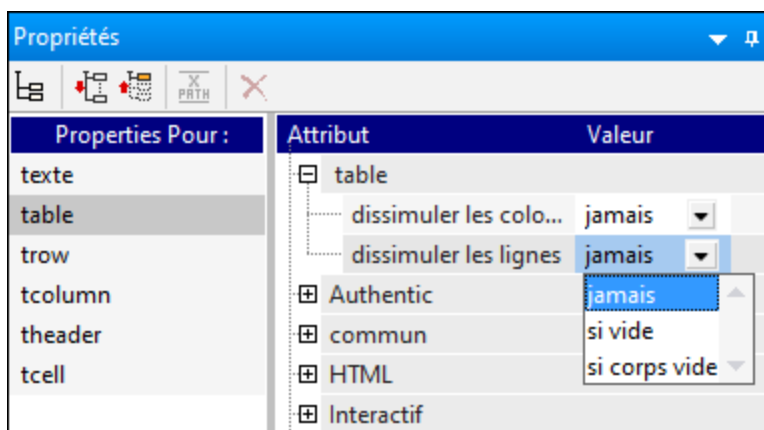
Pour les tables, les options d'affichage des lignes et colonnes sont disponibles dans la **sortie HTML uniquement**. Ces fonctions ne sont **pas prises en charge dans Authentic View** et exigent que XSLT 2.0 ou XSLT 3.0 soit sélectionné comme version XSLT de SPS.

- Des lignes et colonnes vides peuvent être masquées automatiquement.
- Chaque colonne a un bouton **Fermer**, qui permet à l'utilisateur de masquer des colonnes individuelles.
- Des éléments de ligne avec des relations descendantes peuvent être affichés avec les boutons agrandir/réduire.

Masquer des lignes et colonnes vides par défaut.

Pour masquer les lignes et/ou colonnes vides dans la sortie HTML, procédez comme suit :

1. Dans l'Aperçu de design, sélectionnez la table ou une partie de celle-ci (colonne, ligne, cellule).
2. Dans la barre latérale des Propriétés, sélectionnez les propriétés pour *Table*, et le groupe de propriétés *Table* (capture d'écran ci-dessous).



3. Sélectionnez la valeur requise pour les propriétés *Masquer colonnes* et *Masquer lignes*. Les options pour chacune de ces deux propriétés sont les mêmes : *Jamais*, *Si vide* et *Si corps vide*. L'option *Si vide* masque la colonne ou la ligne si l'entière colonne/ligne (y compris l'en-tête et le bas de page) est vide. *Si corps vide* requiert uniquement que le corps soit vide.

Note : si une table non XBRL a des lignes ou colonnes (où les cellules d'une ligne ou d'une colonne peuvent être jointes), masquer des lignes et colonnes vides pourrait ne pas marcher.

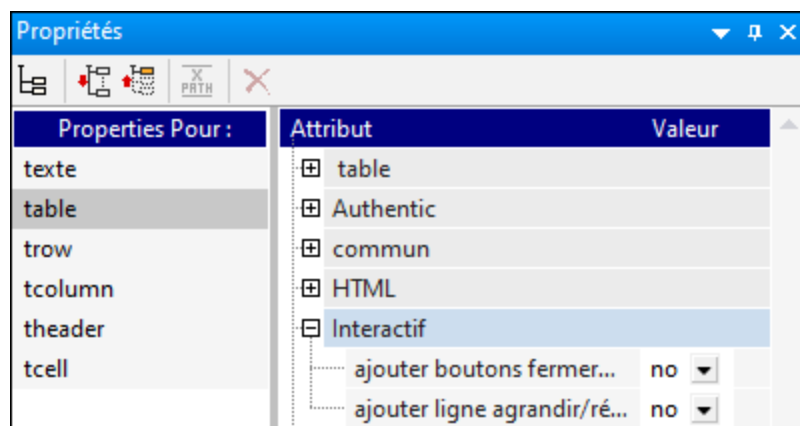
Une interaction d'utilisateur pour masquer des colonnes agrandir/réduire des lignes

Il peut être précisé dans le design que chaque colonne de table contient un bouton **Fermer** dans la sortie HTML (voir la capture d'écran ci-dessous). L'utilisateur peut ensuite masquer les colonnes individuelles en cliquant sur le bouton **Fermer**. Une fois que l'utilisateur dissimule une colonne, un symbole plus apparaît dans la première colonne (voir la capture d'écran ci-dessous). Cliquez sur ce symbole fait réapparaître les colonnes masquées.

Balance Sheet (in Millions)	2004-09-30	2004-07-01 - 2004-09-30	2003-12-31	2004-01-01
Assets, Total	€ 21.49		€ 24.02	
Current Assets, Total	€ 10.65		€ 12.32	
Non Current Assets, Total	€ 10.85		€ 11.7	
Liabilities and Equity, Total	€ 21.49		€ 24.02	
Liabilities, Total	€ 8.9		€ 10.79	
Minority Interests				
Equity, Total	€ 12.59		€ 13.23	
Issued Capital and Reserves	€ 12.59		€ 13.23	

Aussi, les éléments de ligne qui ont des éléments descendants peuvent être affichés dans la sortie HTML avec un symbole agrandir/réduire (plus/moins) à côté (voir capture d'écran ci-dessus). Cliquez sur ces symboles dans la sortie HTML élargit ou réduit cet élément de ligne. Dans le design, vous pouvez spécifier une indentation pour des lignes individuelles utilisant les propriétés CSS.

Les paramètres de ces deux fonctions sont créés dans le groupe des propriétés *Interactif* des propriétés de la Table (capture d'écran ci-dessous).



Ces options pour les deux propriétés sont *Oui* (pour ajouter la fonction) et *Non* (pour ne pas ajouter la fonction).

5.6.7 Tables CALS/HTML

Une table CALS/HTML est une structure XML hiérarchique dont les éléments : (i) définissent la structure d'une table CALS ou HTML, (ii) précisent le formatage de cette table, et (iii) contiennent des contenus de cellule de cette table. Cette structure XML doit correspondre exactement au modèle de table CALS ou HTML.

Pour créer une table CALS/HTML dans le design, suivez les étapes suivantes :

1. [Définir la structure XML comme structure de table CALS/HTML](#) ¹⁴²
2. [Spécifier les styles de formatage pour la table](#) ¹⁴⁴
3. [Insérer la table CALS/HTML dans le design SPS](#) ¹⁴⁵

Activer les structures de table CALS/HTML pour la sortie

Un document XML peut avoir une structure de données qui définit la structure et le contenu d'une table. Par exemple, la structure des données XML suivante correspond au modèle de table HTML et a, en fait, les mêmes noms d'élément que ceux dans le modèle de table HTML :

```
<table>
  <tbody>
    <tr>
      <td/>
    </tr>
  </tbody>
</table>
```

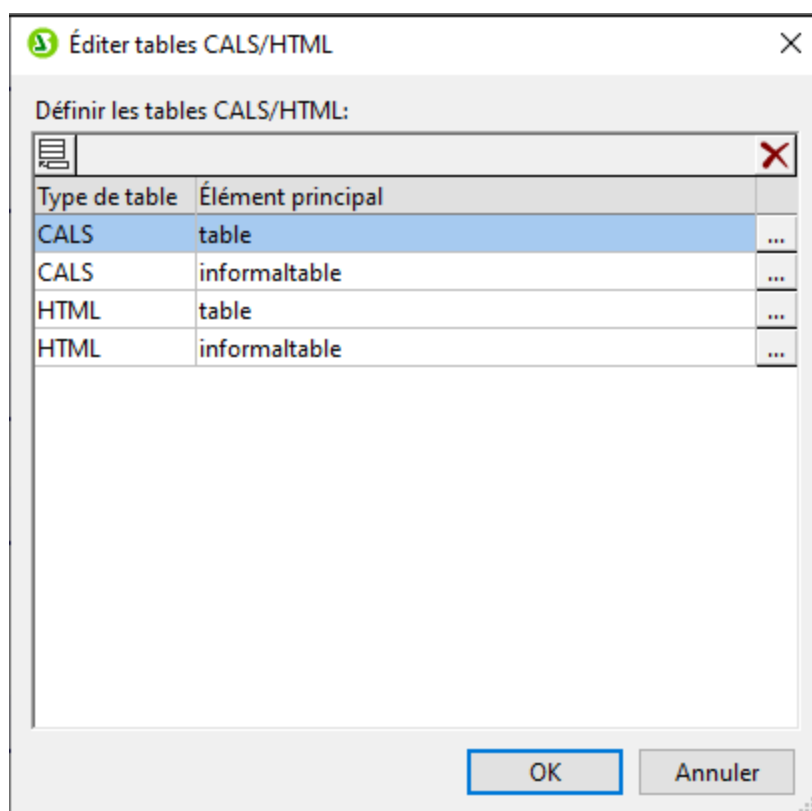
En alternative, la structure des données XML pourrait avoir une structure correspondant au modèle de table HTML, mais différents noms d'élément que dans le modèle de table HTML. Par exemple :

```
<semester>
  <subject>
    <class>
      <student/>
    </class>
  </subject>
</semester>
```

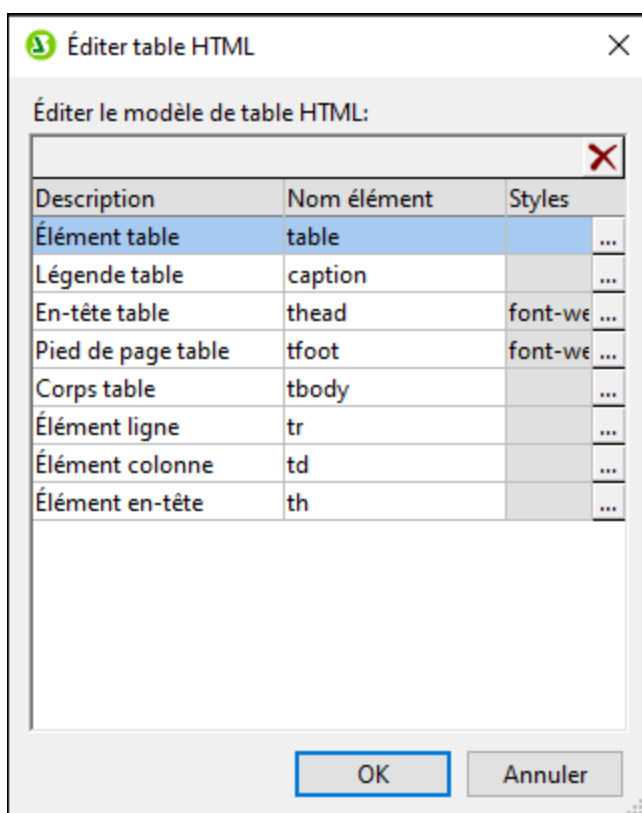
Cette structure de table, qui est définie dans le document XML, peut être utilisée directement pour générer une table dans les différents formats de sortie. À cette fin, vous devez définir cette structure de données XML comme table CALS ou HTML. Si la structure des données XML n'est pas définie comme CALS ou HTML (le défaut), les éléments dans la structure des données seront traités comme éléments « non-table » ordinaires et aucun markup de table sera ajouté dans le document de sortie.

Pour activer la table markup CALS/HTML dans la sortie, suivez les étapes suivantes :

1. Sélectionnez la commande **Table | Éditer Tables CALS/HTML**.
2. Dans le dialogue qui apparaît (*capture d'écran ci-dessous*), ajoutez une entrée pour la structure de données XML que vous voulez utiliser comme table CALS/HTML, selon si la structure de données suit un modèle de table CALS ou HTML. (Pour information sur le modèle de table CALS, voir le [modèle de table CALS à OASIS](#). Pour un exemple d'élément de table ayant une structure de table HTML, ouvrez `HTMLTable1.sps`, qui se trouve dans le dossier `Basics` du dossier de projet `Exemples` (dans la fenêtre `Projet de la GUI`.) Donc, si vous voulez activer un élément dans votre schéma CALS ou élément de table HTML, cliquez sur le bouton de table `Ajouter CALS/HTML` en haut du côté gauche du dialogue, puis sélectionnez soit la commande **Ajouter Table CALS** ou la commande **Ajouter Table HTML**. (Dans la capture d'écran ci-dessous, les éléments `table` et `informaltable` ont été activés comme tables CALS (de même que les tables HTML).) Cliquez sur **OK** pour confirmer.



- Un dialogue (Éditer Table CALS ou Éditer Table HTML) apparaît affichant les éléments du type de table que vous avez sélectionné (*capture d'écran ci-dessous*). Les noms d'élément qui sont recensés dans ce dialogue sont, par défaut, les noms d'élément dans le modèle de table sélectionné (CALS ou HTML). Si le schéma SPS contient des éléments avec les même noms que les noms du modèle de table CALS/HTML, alors les noms sont affichés en noir (*comme dans la capture d'écran ci-dessous*). Si un nom d'élément recensé n'est pas présent dans le schéma SPS, ce nom d'élément est recensé en rouge. Vous pouvez modifier un nom d'élément recensé pour correspondre à un nom de schéma en double-cliquant dans le champ pertinent *Nom d'élément* et éditant le nom.



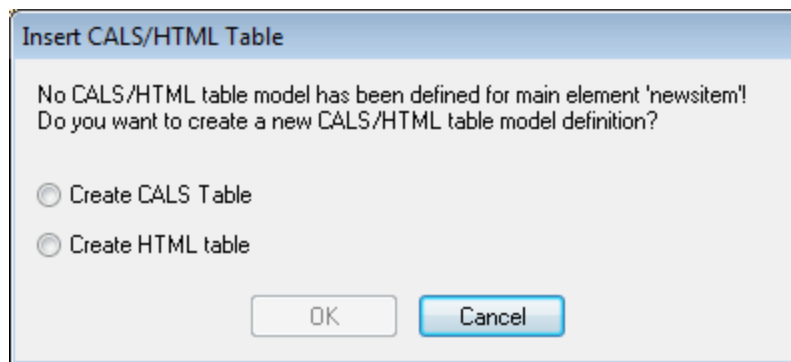
4. Cliquez sur **OK** pour définir cette structure des données XML comme table CALS ou HTML.
5. Vous pouvez ajouter des entrées pour autant de structures de données XML que vous le souhaitez (voir la capture d'écran à l'étape 2 ci-dessus). Le même élément principal peut être utilisé une fois pour chacun des types de table CALS et HTML.
6. Une fois que vous avez terminé de définir les structures des données XML que vous souhaitez activer comme tables CALS/HTML, cliquez sur **OK** pour terminer.

Si une table CALS/HTML a été définie et que la structure des données XML est [correctement insérée](#)¹⁴⁵ comme table CALS/HTML, alors la structure des données sera envoyée à la sortie comme table. Pour **supprimer une définition de table CALS/HTML**, dans le dialogue de table Éditer CALS/HTML, sélectionnez la définition que vous souhaitez supprimer et cliquez sur le bouton **Supprimer** en haut à droite du volet Définir Tables CALS/HTML.

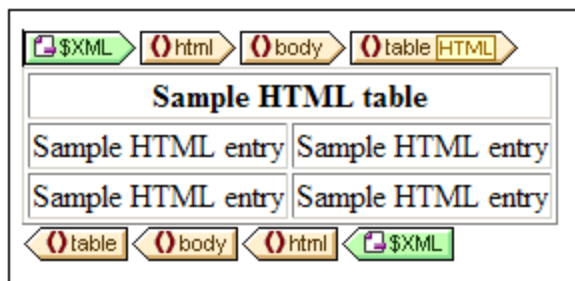
Formatage de table

Les tables CALS/HTML obtiennent leur formatage de deux manières :

1. En formatant les attributs dans le document XML source. Les modèles de table CALS et HTML permettent le formatage d'attributs. Si de tels attributs existent dans le document XML source, ils sont passés aux attributs de présentation du markup de table de la sortie.
2. Chaque élément individuel dans la table peut être formaté dans la colonne Styles du dialogue Éditer Table CALS ou du dialogue Éditer Table HTML (voir la capture d'écran ci-dessous).



Si l'élément a été créé dans le design comme table CALS/HTML, un espace réservé pour l'élément du design de table CALS/HTML est inséré à l'emplacement (*capture d'écran ci-dessous*).



Modèles globaux et éléments de table

Si les [modèles globaux](#) ²³⁴ des éléments de table suivants sont créés, ils seront utilisés dans la sortie de table CALS/HTML. Pour les tables CALS : `title` et `entry`. Pour les tables HTML : `caption`, `th` et `td`.

Exemples de fichiers

Les exemples de fichiers sont dans le dossier de projet `Exemples` (dans la fenêtre de Projet de la GUI).

5.7 Listes

Il existe deux types de listes qui peuvent être créées dans SPS:

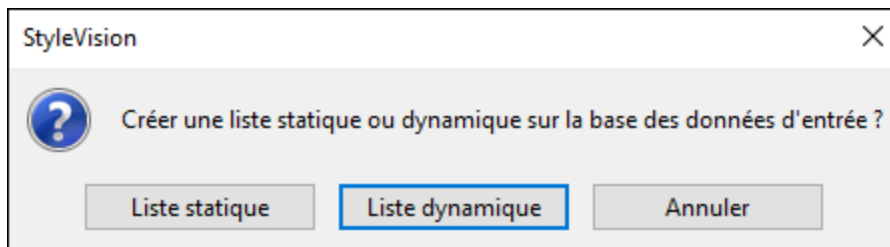
- Les [listes statiques](#)¹⁴⁷, qui sont des listes, dont les contenus sont saisis directement dans SPS. La structure de liste n'est pas dérivée de manière dynamique depuis la structure du document XML.
- Les [listes dynamiques](#)¹⁴⁹, qui sont des listes qui dérivent leurs structure et contenus de manière dynamique depuis le document XML.

Comment créer ces deux types de listes est décrit en détail dans les sous-sections de cette section.

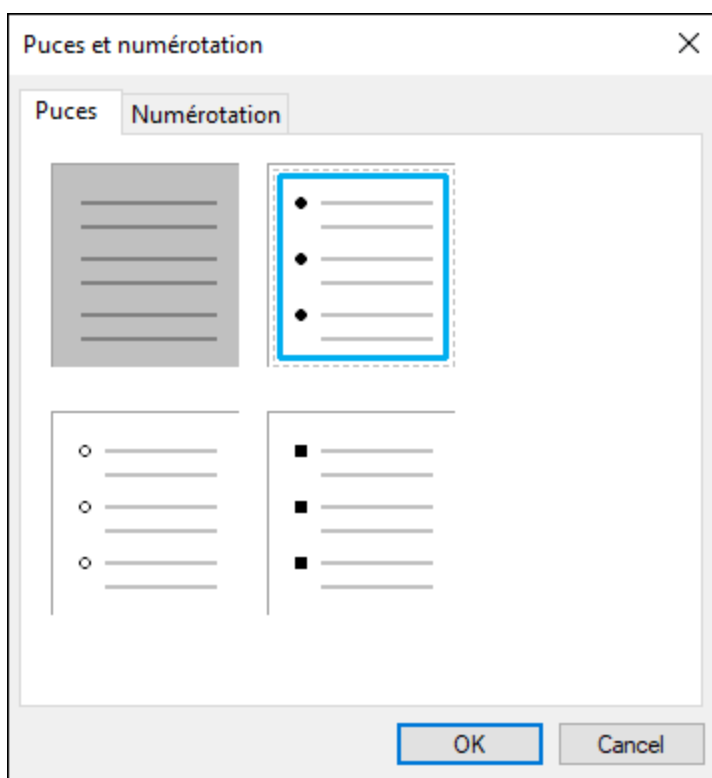
5.7.1 Listes statiques

Une liste statique est une liste dans laquelle les contenus d'item de la liste sont saisis directement dans SPS. Pour créer une liste statique, procédez comme suit :

1. Placez le curseur dans l'emplacement dans le design où vous souhaitez créer la liste statique et sélectionnez la commande de menu [Insérer | Puces et numérotation](#)⁴⁹⁶ (ou cliquez sur l'icône Puces et numérotation dans la [barre d'outils Insérer des Éléments Design](#)⁴⁴⁷). Ceci fait apparaître un dialogue vous demandant si vous souhaitez créer une liste statique ou une liste dynamique (*capture d'écran ci-dessous*).



2. Cliquez sur **Liste statique**. Ceci fait apparaître le dialogue Puces et numérotation (*capture d'écran ci-dessous*).



3. Sélectionnez le marqueur d'item de la liste désirée et cliquez sur **OK**. Un item de liste vide est créé.
4. Saisissez le texte du premier item de liste.
5. Appuyez sur **Enter** pour créer un nouvel item de liste.

Pour créer une liste imbriquée, placez le curseur à l'intérieur de l'item de liste qui doit contenir la liste imbriquée et cliquez sur la commande de menu [Insérer | Puces et numérotation](#)⁴⁹⁶. Utilisez ensuite la procédure décrite ci-dessus une fois de plus.

Note : vous pouvez aussi créer une liste statique en plaçant le curseur à l'emplacement où la liste doit être créée et cliquez soit sur l'icône Puces ou numérotation dans les icônes Puces et numérotation dans la [barre d'outils Formatage](#)⁴⁴⁵. Le premier item de liste sera créé au point de l'insertion du curseur.

Modifier le texte statique en liste

Pour modifier du texte statique, suivez les étapes suivantes :

Mettez le texte en surbrillance que vous désirez modifier en liste, sélectionnez la commande [Joindre à | Puces et numérotation](#)⁵¹¹, choisissez le type de marqueur désiré et cliquez sur **OK**. Si le texte contient un CR-LF, retour à la ligne et/ou retour chariot (inséré en appuyant sur la clé **Enter**), alors les items de liste séparés sont créés pour chaque fragment de texte séparé par un CR-LF. Si un fragment de texte à l'intérieur d'une ligne est mis en surbrillance, alors ce texte est créé comme « list-item » d'une liste « single-item » ; vous pouvez ajouter un nombre illimité d'items de liste supplémentaires en cliquant sur **Enter** autant de fois que requis. Notez que la commande [Joindre à | Puces et numérotation](#)⁵¹¹ peut être accédée par le biais du menu contextuel.

5.7.2 Listes dynamiques

Les listes dynamiques affichent le contenu d'un ensemble de nœuds frères du même nom, avec chaque nœud représenté comme item de liste unique dans la liste. L'élément, les instances qui doivent apparaître comme items de liste de la liste, est créé comme liste. Le mécanisme et l'usage sont expliqués ci-dessous.

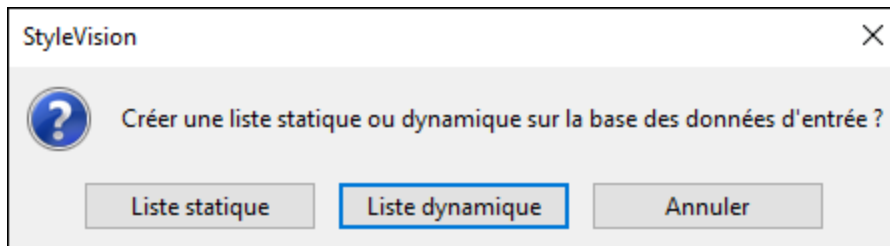
Mécanisme d'utilisation générale

- Tout élément peut être créé comme liste
- Lorsqu'un élément est créé comme item de liste ; les instances de ce élément sont créées comme items de la liste. Par exemple, si dans un élément `department`, il existe plusieurs éléments `person` (par ex., instances), et que vous voulez créer une liste de toutes les personnes dans le département, alors vous devez créer l'élément `person` comme liste.
- Une fois que la liste a été créée pour l'élément, vous pouvez modifier l'apparence ou le contenu de la liste ou de l'item de liste en insérant du contenu supplémentaire ou dynamique tel que du texte, des calculs automatiques, du contenu dynamique, etc.

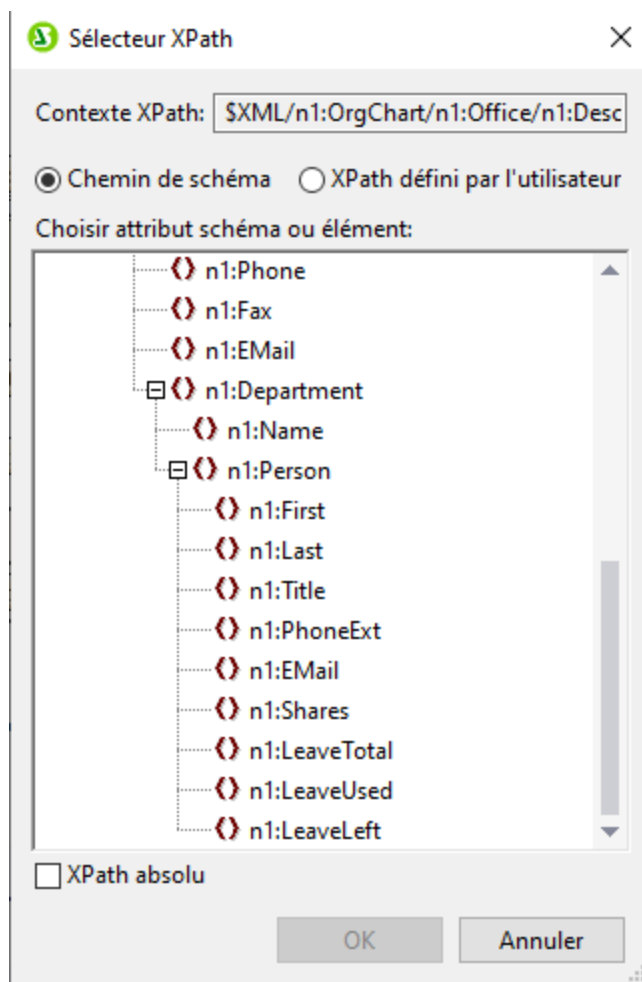
Créer une liste dynamique

Créer une liste dynamique comme suit :

1. Placez le curseur dans l'emplacement dans le design où vous souhaitez créer la liste dynamique et sélectionnez la commande de menu **Insérer | Puces et numérotation**⁴⁹⁶. Ceci fait apparaître un dialogue vous demandant si vous souhaitez créer une liste statique ou une liste dynamique (*capture d'écran ci-dessous*).

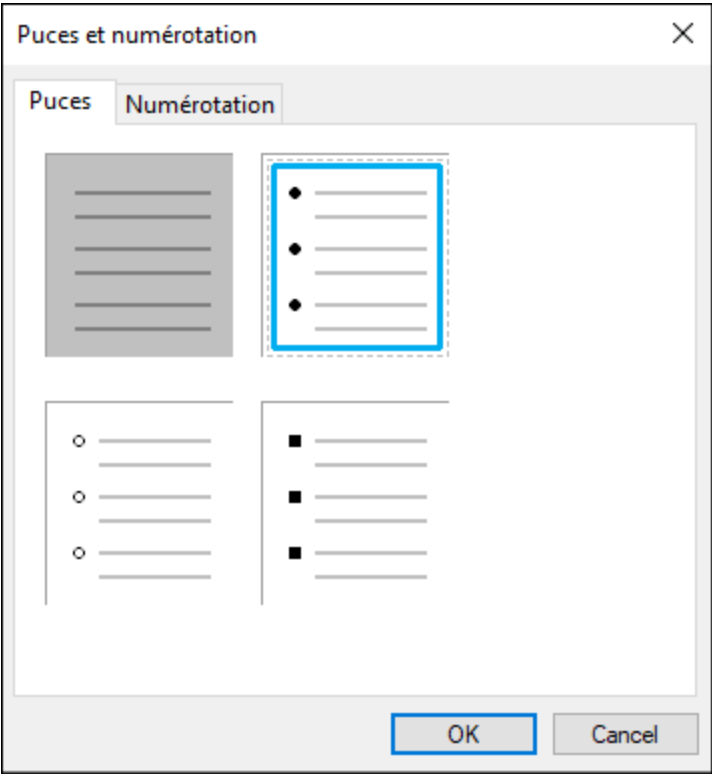


2. Cliquez sur **Liste dynamique**. Apparaîtra le dialogue Sélecteur XPath (*capture d'écran ci-dessous*).
3. Dans le dialogue Sélecteur XPath, notez que le Contexte XPath est le contexte de l'emplacement d'insertion, et qu'il ne peut pas être modifié dans le dialogue. Sélectionnez le nœud qui doit être créé en tant que liste dynamique. Dans la capture d'écran ci-dessous, le nœud contextuel est `n1:Department`, et le nœud `n1:Person` ont été sélectionnés comme nœud à créer comme liste. Ceci signifie que le contenu de chaque nœud `n1:Person` sera créé comme item dans la liste.



Si vous sélectionnez l'option XPath défini par l'utilisateur, vous pouvez alors saisir une expression XPath pour sélectionner le nœud à créer en tant que table dynamique. Cliquez sur **OK** fait apparaître le dialogue Puces et numérotation décrit dans la prochaine étape.

4. Dans le dialogue Puces et numérotation, sélectionnez le genre de liste que vous désirez créer. Vous pouvez choisir depuis une liste à puces (avec une puce, un cercle, ou un carré comme le marqueur de l'item d'une liste), ou une liste numérotée. Cliquez sur **OK** pour créer la liste avec le type de marqueur d'élément de liste que vous avez sélectionné.



5.8 Graphiques

Lorsque vous insérez des images dans le document de design, l'emplacement de l'image peut être spécifié directement dans la SPS (par le designer SPS) ou peut être pris ou dérivé d'un nœud dans le document XML. Comment spécifier l'emplacement de l'image est décrit dans la section [URI Image](#)¹⁵². Quels types d'images sont pris en charge dans les sorties sont recensés dans la section [Types et sortie d'image](#)¹⁵⁴.

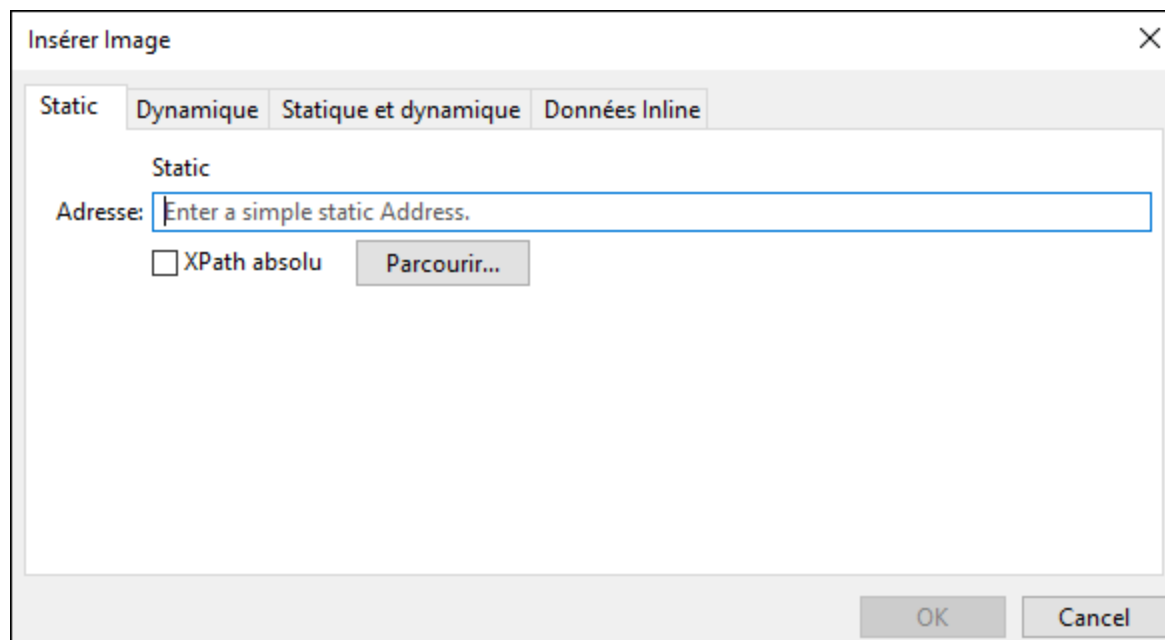
Propriétés d'image

Les images peuvent être définies dans la fenêtre des Propriétés. Procédez comme suit. Sélectionnez l'image dans le design. Puis, dans la fenêtre des Propriétés, (i) sélectionnez *image* dans les Propriétés pour la colonne, (ii) sélectionnez le groupe de propriété requis, et (iii) à l'intérieur du groupe de propriété sélectionné, sélectionnez la propriété requise. Par exemple, pour définir la hauteur et la largeur de l'image, définissez la *hauteur* et la *largeur* des propriétés dans le groupe des propriétés *HTML*.

5.8.1 Images : URI et données Inline

Les images peuvent être insérées sous tout emplacement dans le document de design. Ces images seront affichées dans les documents de sortie ; dans le Mode de design, les images insérées sont indiquées avec des vignettes ou espaces réservés.

Pour insérer une image, cliquez sur la commande de menu [Insérer | Image](#)⁴⁹², qui fait apparaître le dialogue Insérer Image (*capture d'écran ci-dessous*).



Les images peuvent être accédées de deux manières :

- L'image est un fichier, qui est accédé en saisissant son URI dans le dialogue Insérer Image.
- L'image est encodée comme texte Base-16 ou Base-64 dans le fichier XML.

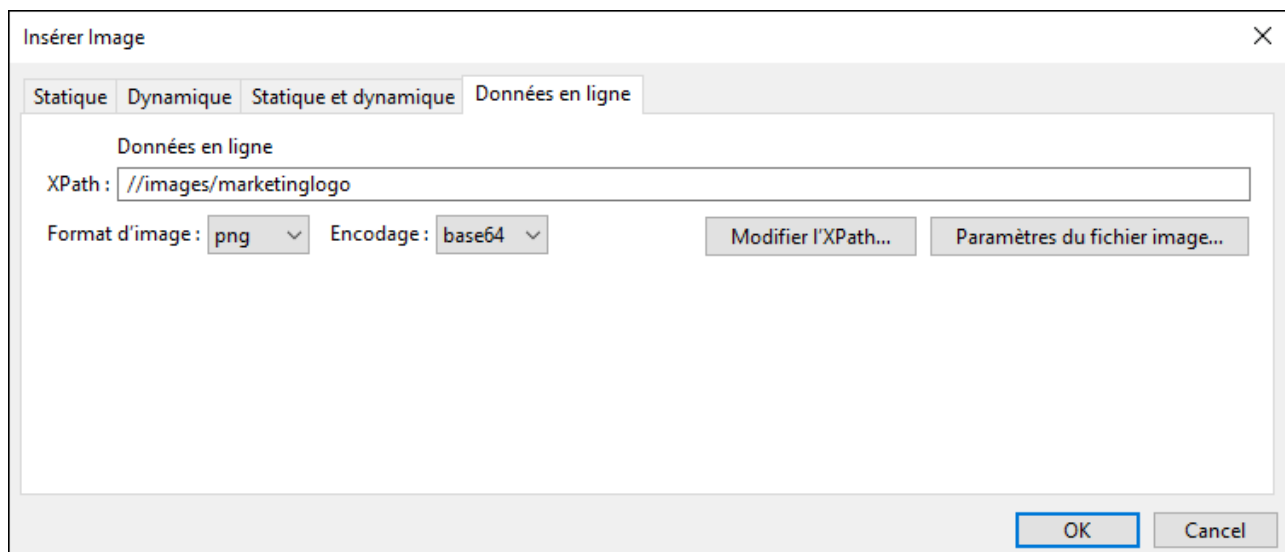
Insérer un fichier d'image

Un fichier d'image est inséré dans le design en spécifiant son URI. Ce fichier est accédé au moment de l'exécution et placé dans le document. Il existe trois manières de laquelle l'URI de l'image peut être saisie dans le dialogue Insérer Image (*capture d'écran ci-dessus*) :

- Dans l'onglet Static, l'URI est saisie directement comme URI absolu ou relatif. Par exemple, `nanonull.gif` (*URI relatif* ; voir [section ci-dessous](#)¹⁵⁴), et `C:/images/nanonull.gif` (*URI absolu*).
- Dans l'onglet dynamique, comme expression XPath qui sélectionne un nœud contenant soit (i) une URI (absolu ou relatif), ou (ii) un [nom d'entité non parsé](#)³⁶³. Par exemple, la saisie `image/@location` sélectionnerait l'attribut `emplacement` de l'élément `image` qui est l'enfant du nœud contextuel (c'est-à-dire, le nœud à l'intérieur duquel l'image est insérée). Le nœud de l'emplacement dans le document XML contiendrait l'URI de l'image. Pour savoir comment utiliser les entités non parsées, consultez la section [URI de l'entité non parsée](#)³⁶³.
- Dans l'onglet Static et Dynamic, une expression XPath dans la partie dynamique peut être préfixée et/ou avec suffixe avec les entrées statiques (texte). Par exemple, le préfixe statique pourrait être `C:/XYZCompany/Personnel/Photos/` ; la partie dynamique pourrait être `concat(First, Last)` ; et le suffixe statique pourrait être `.png`. Ceci résulterait en une URI absolue comme :
`C:/XYZCompany/Personnel/Photos/JohnDoe.png`.

Insérer une image qui est du texte encodé

Une image peut être stockée dans un fichier XML sous forme de texte encodé en Base-16 ou Base-64. L'avantage de ceci est que l'image ne doit pas être accédée depuis un fichier séparé (en lien avec celui-ci), mais est présente comme texte dans le fichier de source XML. Pour insérer une image qui est disponible comme texte encodé dans la source XML, utilisez l'onglet des données Inline du dialogue Insérer Image (*voir la capture d'écran ci-dessous*).



Utiliser une expression XPath pour localiser le nœud dans le document XML qui contient du texte encodé de l'image. Sélectionnez une option depuis la zone de liste déroulante Format de l'image pour indiquer quel format de fichier d'image doit être généré. (Un fichier d'image est généré à partir des données de texte encodées, puis ce fichier est utilisé dans le document de sortie.) Dans la zone de liste déroulante Encodage, sélectionnez

l'encodage qui a été utilisé dans la source XML. Ceci permet à StyleVision de lire correctement le texte encodé (en utilisant le format encodé que vous spécifiez).

Le dialogue Paramètres de fichier d'image (accessible en cliquant sur le bouton **Para. fichier image**) vous permet de donner un nom au fichier image qui sera créé. Vous pouvez choisir de ne pas fournir de nom, dans un tel cas StyleVision générera un nom par défaut.

Si vous souhaitez intégrer une image inline dans la sortie HTML, sélectionnez l'option *Imbrication de l'image avec schéma URI des données* dans le [dialogue des Propriétés](#) ⁴⁷² du SPS.

Accéder à l'image pour la sortie

L'image est accédée de différentes manières et à des moments différents dans les processus qui produisent différents documents de sortie. Veuillez noter les points suivants :

- Notez les formats de sortie disponibles pour votre édition : (i) HTML dans l'édition de base ; (ii) HTML et RTF dans Professional; (iii) HTML, RTF, PDF, et Word 2007+ dans l'édition Enterprise.
- Pour l'Aperçu Design, vous pouvez définir dans le [dialogue des Propriétés](#) ⁴⁷² si des chemins relatifs d'images devraient être relatifs à SPS ou le fichier XML.
- Pour la sortie HTML, l'URI de l'image est passée au fichier HTML et l'image est accédée par le navigateur. Donc, si le chemin de l'image est relatif, il doit être relatif dans l'emplacement du fichier HTML. Pour l'Aperçu HTML dans StyleVision, un fichier HTML temporaire est créé dans le même dossier que le fichier SPS, donc, pour le rendu dans l'Aperçu HTML, les chemins relatifs doivent être relatifs à cet emplacement.
- Si l'URI est relative ou absolue, l'image doit être physiquement accessible au processus qui la fournit.

Éditer les propriétés d'images

Pour éditer une image, cliquez avec la touche de droite dans l'espace réservé de l'image dans l'Aperçu Design et sélectionnez **Éditer URL** depuis le menu contextuel. Ceci fait apparaître le dialogue Éditer l'image, qui est le même que dans le dialogue Insérer Image (*capture d'écran ci-dessus*) et dans lequel vous pouvez réaliser les modifications requises. Le dialogue Éditer l'image peut aussi être accédé par le biais de la propriété `URL` du groupe de propriété *image* dans la fenêtre des Propriétés. Le groupe des Propriétés *image* inclut également la propriété `alt`, qui précise le texte alternatif pour l'image.

Supprimer des images

Pour supprimer une image, sélectionnez l'image et appuyez sur la touche **Supprimer**.

5.8.2 Types d'images et Sortie

La table ci-dessous affiche les types d'image pris en charge par StyleVision dans les différents formats de sortie pris en charge par StyleVision. Notez que les différentes éditions de StyleVision prennent en charge les différents ensembles de formats de sortie : *Enterprise Edition*, HTML, Authentic, RTF, PDF et Word 2007+; *Professional Edition*, HTML, Authentic, RTF; *Basic Edition*, HTML.

Type d'image	Authentic	HTML	RTF	PDF	Word 2007+
JPEG	Oui	Oui	Oui	Oui	Oui

GIF	Oui	Oui	Oui	Oui	Oui
PNG	Oui	Oui	Oui	Oui	Oui
BMP	Oui	Oui	Oui	Oui	Oui
TIFF	Oui*	Oui*	Oui	Oui	Oui
SVG	Oui*	Oui*	Non	Oui	Non
JPEG XR	Oui	Oui	Non	Non	Non

* Voir les notes immédiatement ci-dessous

Notez les points suivants :

- Dans le Mode Design, les images seront affichées uniquement si leur emplacement est une URL statique (par exemple, saisie directement dans le fichier SPS).
- Pour l'affichage d'images TIFF et SVG dans Authentic View et HTML View, Internet Explorer 9 ou plus récent est requis.
- Dans la sortie RTF, les images TIFF peuvent uniquement être mises en lien et non intégrées. Donc un redimensionnement n'est pas possible.
- Les documents SVG doivent être en format XML et se trouver dans l'espace de noms SVG.
- FOP rapporte une erreur si un fichier d'image ne peut pas être situé et ne génère pas de PDF.
- Si FOP est utilisé pour produire un PDF, rendre des images PNG requiert que la bibliothèque d'images JIMI soit installée et accessible à FOP.
- Pour plus de détails relatifs à la gestion des graphiques FOP, consultez le [site web FOP](#).

Fichier d'exemple

Un fichier d'exemple, `Images.sps`, est situé dans le dossier :

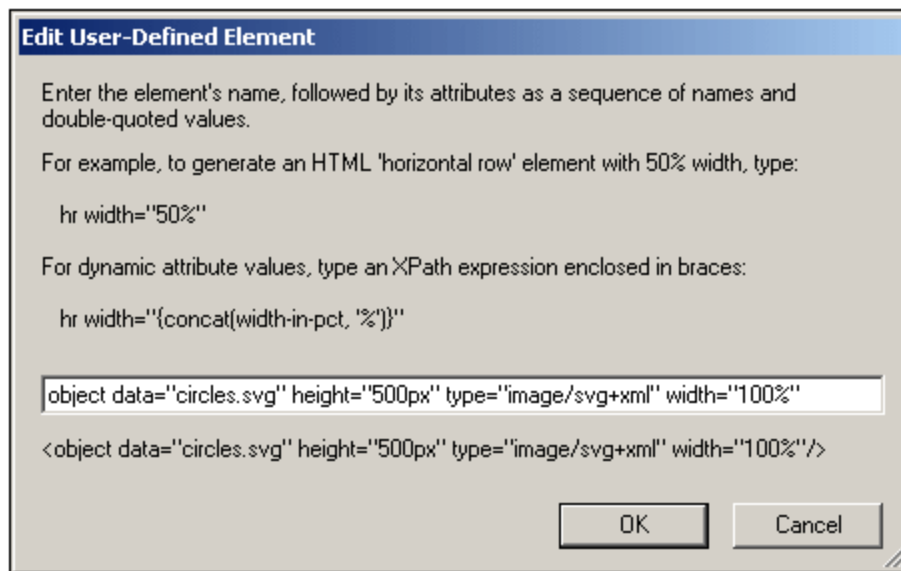
```
C:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\Images
```

Images SVG dans HTML

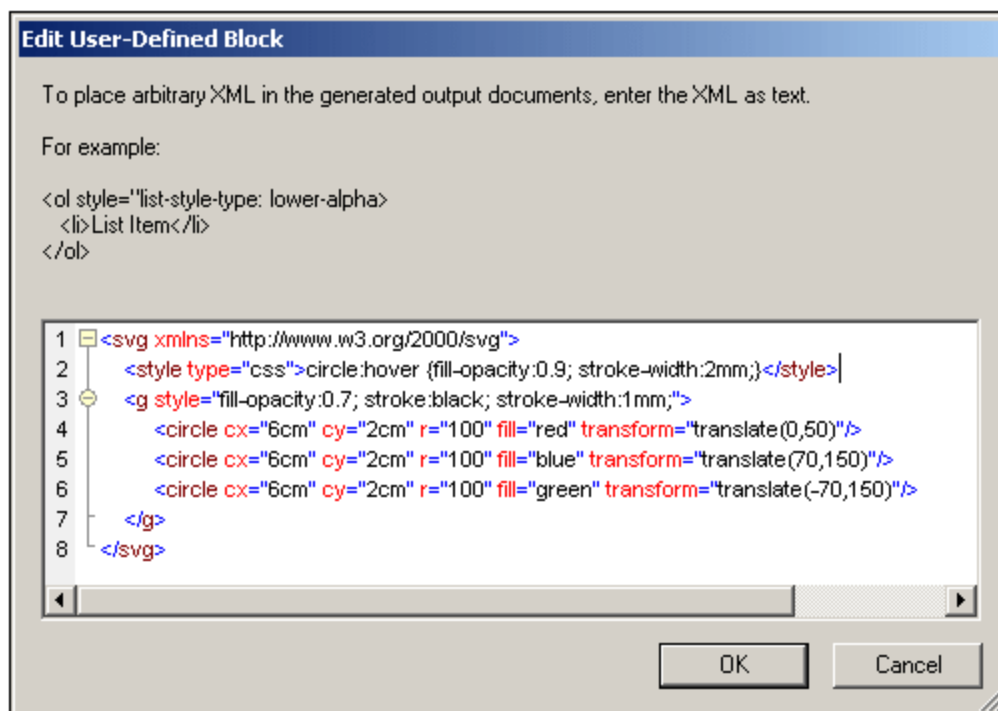
Quand un fichier SVG externe avec code pour des événements de souris est utilisé comme image, le fichier SVG est rendu à l'intérieur de l'image et ne sera plus interactif. Cette limitation peut être surmontée en utilisant le fichier d'image SVG externe comme objet ou en incluant le fragment de code SVG comme bloc XML défini par l'utilisateur.

Ci-dessous, vous trouverez trois manières desquelles les images SVG peuvent être incluses dans une page web.

1. SVG externe [insérée comme image](#)¹⁵² : Ceci génère une `<img>` dans le fichier HTML généré et l'interactivité sera perdue.
2. SVG externe insérée comme objet via la [fonction de l'élément défini par l'utilisateur](#)¹²² (voir la capture d'écran ci-dessous). Assurez-vous d'insérer l'attribut `type` correctement : ke `type="image/svg+xml"`. Lorsqu'inséré de cette manière, l'objet SVG est toujours interactif et la fonction « hover-functionality » de la souris fonctionnera.



3. Inline SVG inséré via un [bloc XML défini par l'utilisateur](#)¹²³. Voir la capture d'écran ci-dessous pour un exemple un fragment de code SVG. Dans ce cas, l'interactivité fonctionnera. Notez que l'élément `svg` n'a pas besoin d'être dans l'espace de noms SVG si la méthode de sortie est HTML 4.0 ou 5.0, mais l'espace de noms est requis si la méthode de sortie est XHTML.



5.8.3 Exemple : Un modèle pour les images

Le pack StyleVision contient un fichier SPS qui démontre l'utilisation d'images dans StyleVision. Ce fichier se trouve dans le [dossier \(Mes\) Documents](#)²⁴ : C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\Images\Images.sps. Le document d'images (Images.xml and Images.sps) consiste en trois parties :

- La deuxième partie contient une table affichant quels formats d'image sont pris en charge dans des formats de sortie StyleVision variés. Notez que les formats de sortie RTF, PDF et Word 2007+ sont disponibles uniquement dans les éditions Enterprise Edition et Professional (RTF uniquement) de StyleVision. Dans l'Aperçu Design, uniquement des images avec des URI statiques seront affichées. Tous les formats d'image recensés dans la table sont affichés dans la Partie 3 du document Images.
- Dans la Partie 3, tous les formats d'images populaires pris en charge par StyleVision sont affichés l'un en-dessous de l'autre. Après avoir ouvert le fichier `Images.sps` dans StyleVision, vous pouvez basculer entre les différents aperçus de StyleVision pour voir comment chaque image est affichée dans l'aperçu. Puisque l'emplacement de l'image se trouve dans un nœud XML, vous pouvez aussi saisir l'emplacement de vos propres images dans Authentic View et tester leur apparence dans les fenêtres d'aperçu.

5.9 Contrôles de formulaires

Les nœuds dans le document XML peuvent être créés comme appareils d'entrée des données (tels les champs d'entrée et les listes de zone déroulante). Dans la sortie HTML, l'appareil d'entrée des données est rendu comme objet qui est le même que celui affiché dans le Mode Design ou équivalent. Notez que les appareils d'entrée de saisie ne fonctionneront pas dans la sortie HTML.

Mécanisme général

Ci-dessous, vous trouverez une liste d'appareils d'entrée des données disponible dans StyleVision, ensemble avec (i) une explication sur la manière dont les données sont saisies dans le document XML pour chaque appareil et (ii) sur la sortie des appareils d'entrée des données comme format de texte brut.

Appareil d'entrée de données	Données dans le fichier XML	Sortie comme Format de texte brut
Champ d'entrée (case de texte)	Texte saisi par l'utilisateur	Texte saisi par l'utilisateur
Champ de saisie multiligne	Texte saisi par l'utilisateur	Texte saisi par l'utilisateur
Zone de liste	La sélection de l'utilisateur est mappée vers une valeur.	Valeur sélectionnée
Événement	La sélection de l'utilisateur est mappée vers une valeur.	[] (non sélectionnée); [x] (sélectionnée)
Bouton radio	La sélection de l'utilisateur est mappée vers une valeur.	() (non sélectionnée); /o (sélectionnée)
Bouton	La sélection de l'utilisateur est mappée vers une valeur.	Texte bouton

Les valeurs de texte saisies dans les champs d'entrée sont directement saisies dans le document XML comme contenu XML. Pour les autres appareils d'entrée des données, la sélection de l'utilisateur Authentic View est mappée vers une valeur. StyleVision vous permet de définir la liste des options que l'utilisateur verra et la valeur XML vers laquelle chaque option est mappée. Généralement, vous définirez les options et leurs valeurs correspondantes dans un dialogue.

Ci-dessous, vous trouverez une liste des appareils d'entrée des données disponible dans StyleVision.

Appareil d'entrée de données	Sortie comme Format de texte brut
Champ d'entrée (case de texte)	Texte saisi par l'utilisateur
Champ de saisie multiligne	Texte saisi par l'utilisateur
Zone de liste	Valeur sélectionnée

Événement	[] (non sélectionné); [x] (sélectionné)
Bouton radio	() (non sélectionné); /o (sélectionné)
Bouton	Texte bouton

Utilisation générale

Pour créer un appareil d'entrée des données, procédez comme suit :

1. Glissez un nœud depuis la fenêtre d'arborescence de schéma dans le Mode Design déposez-le à l'emplacement désiré.
2. Depuis le menu contextuel qui apparaît, sélectionnez un appareil d'entrée des données que vous souhaitez créer comme nœud.
3. Pour certains appareils d'entrée des données, un dialogue apparaît. Dans ces cas, saisissez l'information requise dans le dialogue et cliquez sur OK.

Pour **rouvrir et éditer** les propriétés d'un appareil d'entrée des données, sélectionnez l'appareil d'entrée des données (pas le nœud qui le contient) et éditez ses propriétés dans la barre latérale des Propriétés.

Note :

- Les données, ne peuvent pas être saisies dans des appareils d'entrée des données dans la sortie HTML. Dans la sortie HTML, les appareils d'entrée des données sont uniquement utilisés comme manière alternative de présenter le contenu.
- Les appareils d'entrée des données peuvent aussi être créés en modifiant le type de composant actuel d'un nœud vers un appareil d'entrée des données. À cette fin, cliquez avec la touche de droite dans le nœud et sélectionnez **Modifier en**.
- Dans la sortie HTML, l'entrée sélectionnée par l'utilisateur est affichée dans la sortie. Modifier une valeur de l'appareil d'entrée des données dans le document HTML ne change ni la valeur de texte dans le document XML, ni dans le document HTML.

5.9.1 Champs d'entrée, Champ de saisie multiligne

Vous pouvez insérer un Champ d'entrée ou un Champ de saisie multiligne dans votre SPS lorsque vous déposez un nœud depuis la fenêtre de sources de schéma dans le Mode Design. Le contenu de ce nœud est affiché dans le champ d'entrée ou dans le champ de saisie multiligne.

Éditer les propriétés des champs d'entrée

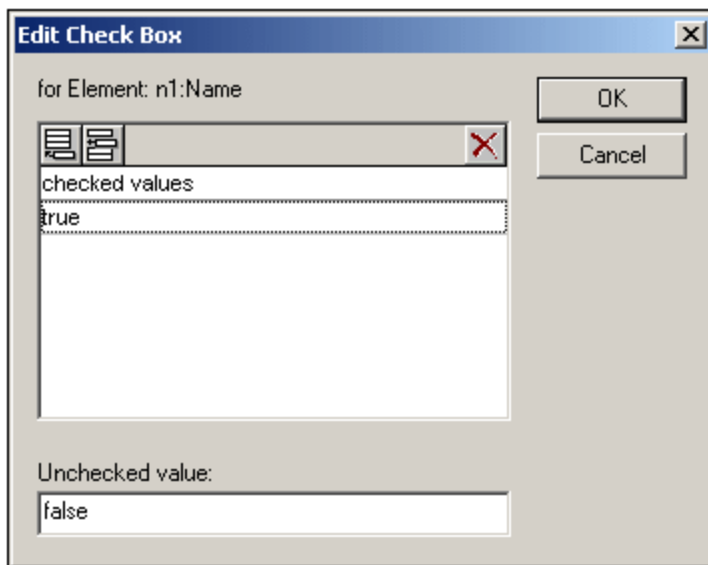
Vous pouvez modifier les propriétés HTML des champs d'entrée en sélectionnant le champ d'entrée et en modifiant ses propriétés *HTML* dans la barre latérale des Propriétés.

Par exemple, avec le champ d'entrée sélectionné, dans la fenêtre des Propriétés, sélectionnez `editfield`, sélectionnez le groupe de propriétés *HTML* et la propriété `maxlength`. Puis, double-cliquez dans le champ Valeur de `maxlength` et saisissez une valeur.

Note : les sections CDATA ne peuvent pas être insérées dans les champs de saisie (c'est-à-dire dans les zones de texte et les zones de texte multilignes). Les sections CDATA ne peuvent être saisies que dans les éléments affichés dans Authentic View en tant que composants de contenu textuel.

5.9.2 Cases à cocher

Vous pouvez créer une case à cocher comme appareil d'entrée des données. Dans l'édition Basic, vous pouvez laisser les paramètres dans le dialogue Éditer Case à cocher à leurs paramètres par défaut (puisque l'édition Basic ne prend pas en charge Authentic View, comme résultat dont aucune valeur ne peut être saisie dans le fichier XML.)



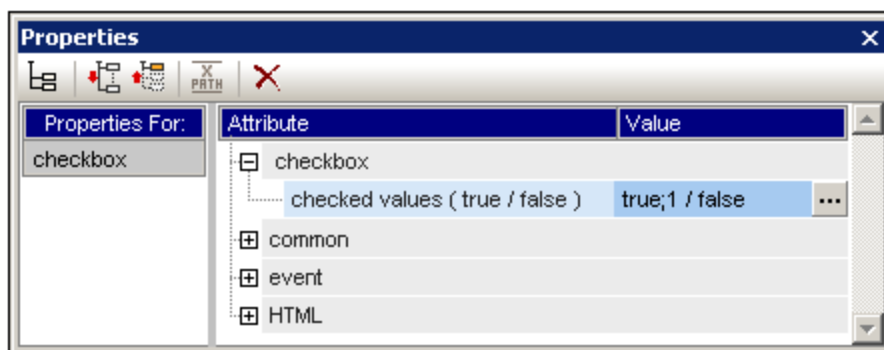
Dans la capture d'événement ci-dessus, un élément dénommé `Name` a été créé comme case à cocher. Si l'utilisateur Authentic View coche la case à cocher, une valeur `true` sera saisie comme valeur de l'élément `Name`. Si la valeur est décochée, alors la valeur `false` est saisie comme valeur XML de `Name` (telle que définie dans le dialogue).

Note : les cases à cocher dans la sortie de texte sont affichées comme des crochets : `[ ]` pour des cases à cocher non sélectionnées ; `[x]` pour des cases à cocher sélectionnées.

Accéder au dialogue Éditer Case à cocher

Si vous créez une nouvelle case à cocher, quand vous créez le nœud comme case à cocher, le dialogue Éditer case à cocher apparaît. Pour accéder au dialogue Éditer case à cocher par la suite, faites comme suit :

1. Sélectionnez la case à cocher dans le design.
2. Dans la barre latérale des Propriétés, sélectionnez l'item de la case à cocher, puis le groupe des Propriétés case à cocher (voir la capture d'écran ci-dessous).



3. Cliquez sur le bouton Éditer [...] de la propriété `checked values`. Ceci fera apparaître le dialogue Éditer Case à cocher.

Note : vous pouvez modifier les propriétés HTML d'une case à cocher en la sélectionnant et en modifiant ses propriétés HTML dans la barre latérale des Propriétés.

5.9.3 Listes de choix

une liste de choix déroulante présente les items dans une liste de choix. Les items peuvent être sélectionnés d'une des trois manières recensées ci-dessous. La liste peut être utilisée dans le document HTML généré à des fins requises ; par exemple, le HTML généré peut être traité ultérieurement pour que la liste de choix fournisse des entrées pour un formulaire HTML.

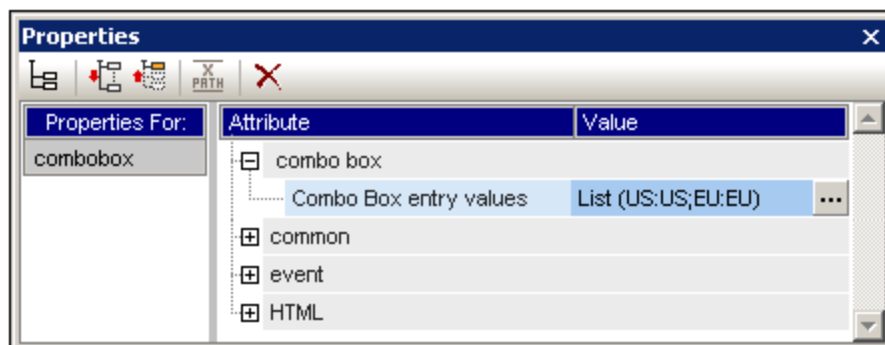
- Depuis des énumérations de schéma pour le nœud sélectionné.
- Depuis une liste définie dans le dialogue Éditer Liste de choix. Vous saisissez l'entrée visible et la valeur XML correspondante, qui peut être différente. La valeur XML s'applique aux éditions Enterprise et Professional, et fait référence à la valeur XML à laquelle la sélection de l'utilisateur Authentic View est mappée. L'édition Basic peut laisser cette colonne vide (puisque Authentic View n'est pas pris en charge dans l'édition Basic).
- Depuis une séquence de résultat d'une expression XPath relative au nœud actuel. Les items dans la séquence de résultat sont affichés comme entrées de la liste déroulante. Il s'agit d'une méthode puissante d'entrées dynamiques dans la liste de choix. Le nœud que vous créez comme liste de choix est important. Par exemple, disons que vous avez un élément `NameList` qui peut contenir un nombre illimité d'éléments `Name`, qui eux-mêmes ont des éléments enfant `First` et `Last`. Si vous créez l'élément `Name` comme liste de choix et sélectionnez l'élément enfant `Last` pour les valeurs de liste, alors vous recevrez autant de listes de choix qu'il y a d'éléments `Name` et chaque liste de choix aura le dernier enfant `Last` comme son entrée du menu déroulant. Afin d'avoir une seule liste de choix avec tous les éléments `Last` dans la liste déroulante du menu, vous devez créer un seul élément `NameList` comme liste de choix, et sélectionner l'élément `Last` dans l'expression XPath.

Accéder au dialogue Éditer Liste de choix

Si vous créez une nouvelle liste de choix, alors vous créez le nœud comme liste de choix, alors le dialogue Éditer liste de choix apparaît. Vous pouvez aussi insérer une liste de choix par la commande de menu (**Insérer | Insérer Contrôles du formulaire | Liste de choix**). Pour accéder au dialogue Éditer Liste de choix par la suite, procédez comme suit :

1. Sélectionnez la liste de choix dans le design.

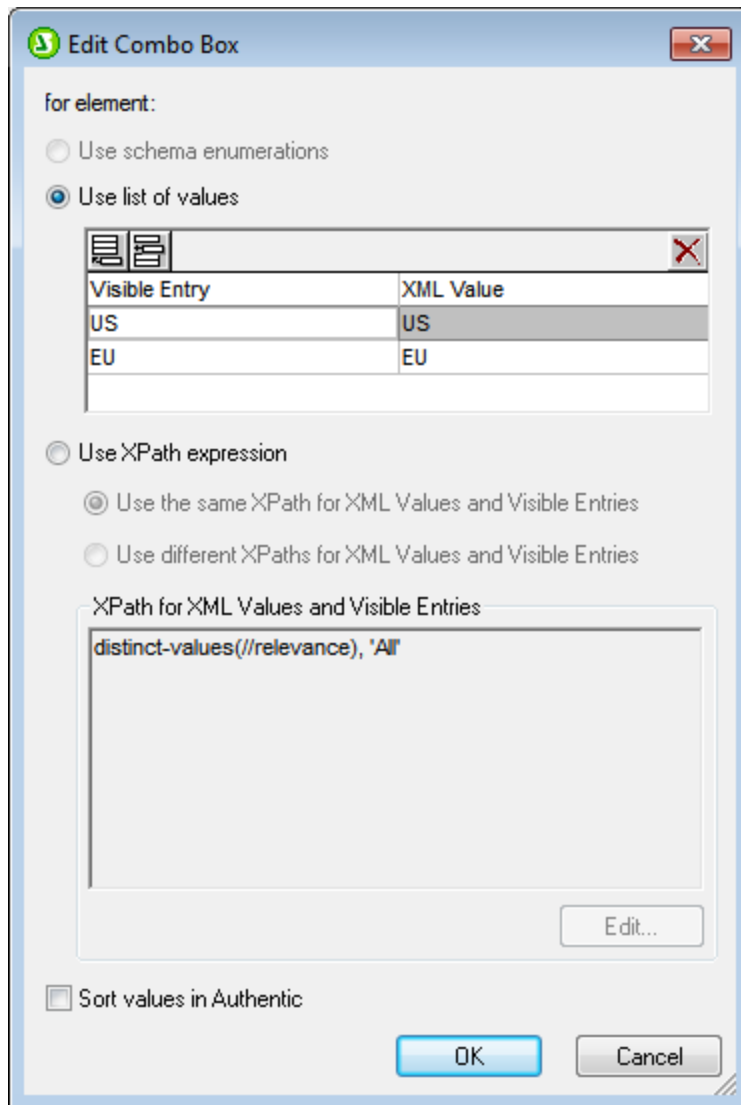
2. Dans la barre latérale des Propriétés, sélectionnez l'item de la case à cocher, puis le groupe des Propriétés *liste de choix* (voir la capture d'écran ci-dessous).



3. Cliquez sur le bouton Éditer [...] de la propriété `origin content`. Ceci fera apparaître le dialogue Éditer Liste de choix.

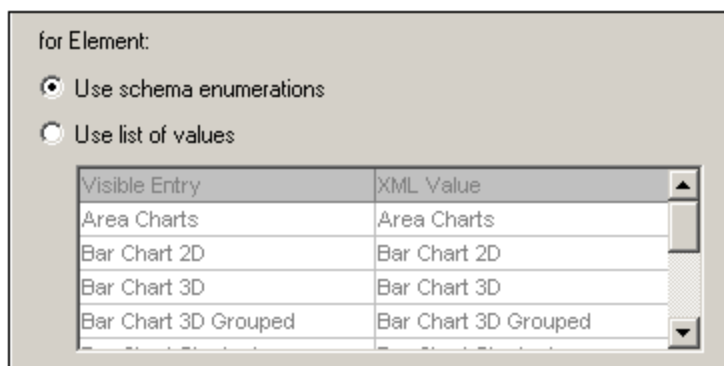
Utiliser le dialogue Éditer Liste de choix

Le dialogue Éditer Liste de choix est affiché ci-dessous.



Pour définir les entrées et valeurs pour la liste de choix, procédez comme suit :

1. Sélectionnez la méthode avec laquelle vous souhaitez définir les entrées et valeurs en cliquant sur le bouton radio approprié : (i) énumérations de schéma, (ii) liste de valeurs, ou (iii) les expressions XPath pour sélectionner les valeurs.
2. Si vous sélectionnez *Énumérations de schéma*, les énumérations attribuées à ce nœud dans le schéma sont saisies de manière automatique comme les (i) entrées visibles de la liste déroulante de la liste de choix, et (ii) les valeurs XML correspondantes (*capture d'écran ci-dessous*). Les entrées visibles sont des entrées dans la liste déroulante de la liste de choix. Chaque entrée de liste déroulante a une valeur XML correspondante. La valeur XML correspondante à l'entrée visible que l'utilisateur Authentic sélectionne sera la valeur XML qui est saisie dans le fichier XML. Les deux entrées visibles et les valeurs XML sont grisées dans la liste des valeurs car elles sont toutes deux obtenues depuis les énumérations de schéma et ne peuvent pas être éditées.



Si vous sélectionnez *Utiliser Liste des valeurs*, vous pouvez insérer, ajouter, éditer et supprimer tout nombre d'entrées pour la liste déroulante De la liste de choix de même que pour les valeurs XML correspondantes. Ces éditions sont reportées dans le volet en-dessous du bouton radio *Utiliser Liste des valeurs*. Vous pourrez également utiliser une expression XPath pour créer les entrées visibles et les valeurs XML. Les items dans la séquence renvoyée par l'expression XPath seront utilisés pour les entrées visibles et les valeurs XML. Vous pouvez spécifier : (i) que la même expression XPath est utilisée pour les entrées visibles et le valeurs XML, ou (ii) que différentes expressions XPath sont utilisées. Dans le dernier cas, un mappage d'index one-to-one entre les items des deux séquences détermine la correspondance de l'entrée visible à la valeur XML. Si le nombre d'items dans les deux séquences n'est pas égal, une erreur est rapportée.

3. Si vous souhaitez avoir des items qui apparaissent dans la liste déroulante de la liste de choix dans Authentic View triés, vérifiez la case à cocher *Trier les valeurs dans Authentic*.
4. Cliquez sur **OK** pour terminer.

Note

- Utiliser l'expression XPath pour sélectionner les items de la liste de choix déroulante vous permet de créer les listes de choix avec des entrées dynamiques depuis le fichier XML lui-même.
- Si les items dans la liste déroulante de la la liste de choix sont obtenus depuis les énumérations de schéma, il seront triés de manière alphabétique par défaut. Si les items sont obtenus depuis le fichier des données XML, ils apparaîtront dans l'ordre du document par défaut.
- Les listes de choix dans la sortie de texte ont affiché la valeur sélectionnée.

5.9.4 Boutons radio, boutons

Il existe deux types de bouton : les boutons radio et les boutons. Les boutons radio et les boutons peuvent être utiles pour l'entrée dans les formulaires ou pour déclencher des événements dans la sortie HTML. Ce dernier est fait en associant les scripts avec l'événement de bouton.

Note : vous pouvez modifier les propriétés HTML d'un bouton radio ou d'un bouton en le sélectionnant et en modifiant ses propriétés HTML dans la barre latérale des Propriétés.

Note : les boutons radio dans la sortie de texte sont affichés comme parenthèses : `( )` pour les boutons radio non sélectionnés ; `(o)` pour les boutons radio sélectionnés. Les boutons génèrent uniquement le texte de bouton dans la sortie de texte.

5.10 Liens

Les liens (ou hyperliens) peuvent être créés pour les signets situés dans le document ainsi que pour des ressources externes telles que les pages Web. Les liens peuvent aussi être créés pour des ancres générés de manière dynamique. StyleVision propose une flexibilité considérable en matière de création d'URI cibles pour les hyperliens.

La section, [Signets et Hyperliens](#)³²¹, décrit comment créer des signets statiques et dynamiques dans le document et comment les mettre en lien ainsi que créer les liens vers des documents externes.

Note : les liens ne sont pas affichés dans la sortie texte.

5.11 Codes-barres

L'élément de design du code-barres est pris en charge dans le **mode XSLT 2.0 ou XSLT 3.0** (pas XSLT 1.0) et permet la génération de codes-barres (*capture d'écran ci-dessous*) dans le document de sortie. À l'emplacement dans le document de design là où vous souhaitez saisir le code-barres, [insérez l'élément design du code-barres](#) ¹⁶⁶ et spécifiez ses [propriétés](#) ¹⁶⁸.



Important : Pour que les codes-barres fonctionnent, un environnement Java Runtime doit être installé. Ceci doit être la version 1.4 ou plus récente dans une version bit qui correspond à la version bit du pack de StyleVision installé sur votre système. 32-bit ou 64-bit

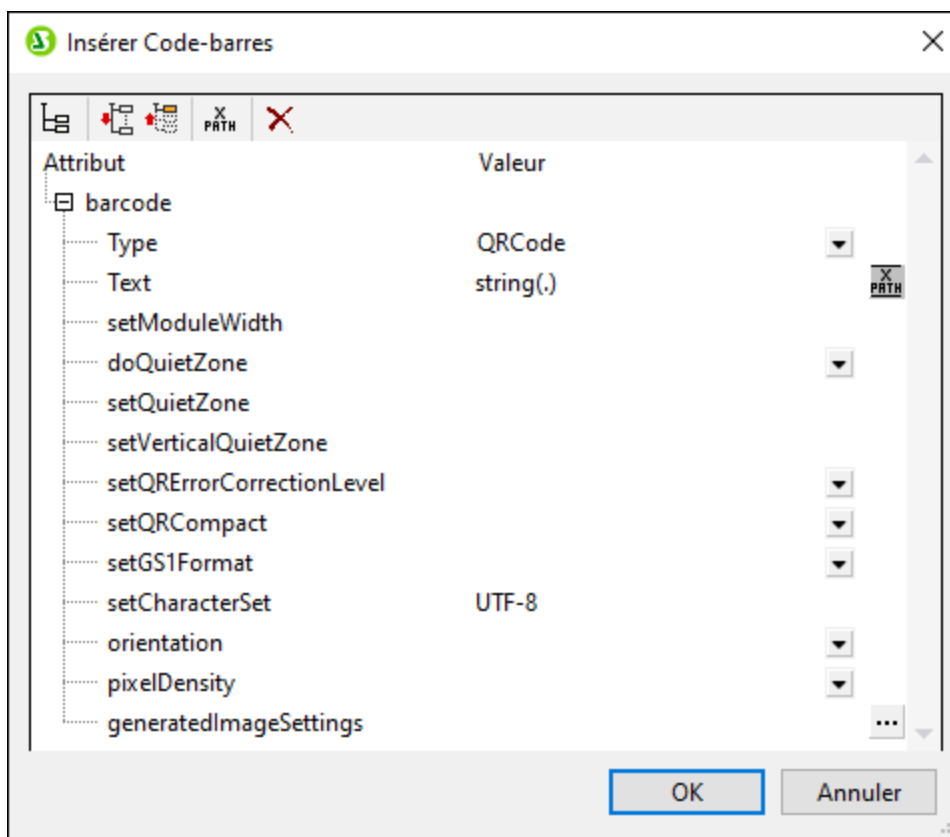
Important : Pour que les codes-barres soient générés dans la sortie, vous **devez utiliser le processeur XSLT d'Altova** pour générer la sortie. Ceci est dû au fait que les codes-barres dans un fichier SPS sont générés en appelant des fonctions d'extension Java spéciale qui ne font pas partie du standard XSLT. Les processeurs XSLT d'Altova prennent en charge ces fonctions d'extension spécifiques, tandis que d'autres processeurs XSLT ne le font probablement pas. En guise de résultat, les codes-barres ne seront pas générés si traités avec un processeur XSLT non-Altova. Le processeur XSLT d'Altova est packagé avec StyleVision et est appelé automatiquement quand vous générez la sortie via les commandes **Generate** dans le menu **Fichier**. De manière alternative, vous pouvez utiliser le [Serveur RaptorXML](#), qui est le processeur XSLT autonome d'Altova.

Note : les codes-barres ne sont pas rendus dans la sortie de texte.

Insérer un code-barres

Pour insérer un code-barres dans votre design, procédez comme suit :

1. À l'emplacement où vous voulez insérer le code-barres, cliquez avec la touche de droite et sélectionnez la commande **Insérer code-barres**. De manière alternative, sélectionnez la commande **Insérer | Insérer code-barres** ou cliquez sur l'icône du code-barres dans la barre d'outils et cliquez dans l'emplacement du design là où vous souhaitez insérer le code-barres. Vous pouvez aussi glisser-déposer un élément depuis l'arborescence de schéma dans le Mode Design, puis sélectionnez « Créer code-barres ». Le dialogue Insérer code-barres s'ouvre (voir la *capture d'écran ci-dessous*).



- Deux propriétés, *Type* et *Text*, sont obligatoires ; les autres sont optionnelles et/ou ont des valeurs appropriées par défaut. La propriété *Type*, la valeur qui peut être sélectionnée depuis une liste déroulante (voir la capture d'écran ci-dessus), spécifie le type de code-barres, par exemple EAN-13 (qui inclut les codes-barres ISBN) et UPC-A. La propriété *Text* spécifie la valeur qui générera le code-barres, par exemple, avec un numéro ISBN. Les différentes propriétés de code-barres sont [décrites ci-dessous](#)¹⁶⁸. Définissez les propriétés requises et toute autre propriété que vous désirez. Notez que, si vous souhaitez utiliser une valeur dans le fichier XML comme la valeur d'une propriété, vous pouvez [saisir une expression XPath](#)⁵¹ pour localiser le nœud XML que vous souhaitez accéder. Procédez comme suit : Sélectionnez la propriété, activez le bouton **XPath** dans la barre d'outils du dialogue des Propriétés, puis saisissez l'expression XPath dans le [dialogue Éditer expression XPath](#)⁴²⁶. L'expression XPath sera évaluée à l'intérieur du nœud contextuel actuel.
- Après avoir défini les propriétés, cliquez sur **OK**. L'image de code-barres sera insérée. Le code-barres généré (voir la capture d'écran ci-dessous) peut être immédiatement vu dans tout aperçu de sortie.



Note : Les images de code-barres sont générées comme fichiers PNG.

Propriétés de code-barres

Les propriétés de code-barres suivantes peuvent être spécifiées. Les propriétés *Type* et *Text* doivent être définies ; les autres propriétés sont optionnelles. Notez que les différentes propriétés sont disponibles pour différents types de code-barres.

- *Type* : Le système de code-barres sous lequel le message sera interprété, tel que EAN et UPC.
- *Texte* : La valeur qui sera utilisée pour générer le modèle de code-barres.
- *SetModuleWidth* : La largeur des barres dans le code.
- *SetBarHeight* : La hauteur des barres :
- *SetHeight* : La hauteur du graphique de code-barres.
- *DoQuietZone* : Les valeurs *Yes* ou *No* déterminent si la « zone silencieuse » (ou padding) autour du code-barres, qui est spécifié dans les propriétés *SetQuietZone* et *SetVerticalQuietZone* sera mis en œuvre.
- *SetQuietZone* : Définit la « zone silencieuse » (ou padding) autour du code-barres. Dans le cas des codes-barres unidimensionnels, la valeur spécifiée ici est appliquée dans la dimension horizontale. Dans le cas de codes-barres bidimensionnels, la valeur est appliquée aux deux dimensions horizontale et verticale. La valeur de la dimension verticale peut être écrasée par la valeur spécifiée dans la propriété *SetVerticalQuietZone*. Une unité de longueur en millimètres (mm) est requise. Exemple : 2mm.
- *SetVerticalQuietZone* : Définit la « zone silencieuse » (ou padding) pour la dimension verticale sur des codes-barres bidimensionnels. Une unité de longueur en millimètres (mm) est requise. Exemple : 2mm.
- *SetMsgPosition* : Spécifie où le texte du message apparaît relatif au code-barres. Les valeurs sont *top*, *bottom* et *none* (aucun message n'est généré).
- *SetPattern* : Définit un modèle pour le texte de message pour que le texte soit lisible. Un long string de nombres, par exemple, serait difficile à lire. La syntaxe des modèles est donnée ci-dessous.
- *SetFontName* : La police dans laquelle le texte devrait apparaître.
- *SetFontSize* : La taille de la police dans laquelle le texte devrait apparaître.
- *SetChecksumMode* : Les valeurs suivantes sont disponibles : (i) *Ajouter* : la checksum est ajoutée automatiquement au message ; (ii) *Check* : la checksum est cochée tout en rendant le code-barres (en patant du fait que la checksum est présente) ; (iii) *Ignore* : aucun traitement de la checksum n'est fait ; (iv) *Auto* : active le comportement par défaut du type de code-barres.
- *Orientation* : Si le code-barres doit être pivoté ou non. Les options sont en étapes de 90 degrés dans le sens contraire des aiguilles de la montre.
- *PixelDensity* : Spécifie la densité des pixels dans l'image du code-barres. Une densité de pixel plus élevée fournit des images plus nettes.
- *GeneratedImageSettings* : Vous permet de définir un nom pour le fichier d'image du code-barres généré. Si aucun nom n'est spécifié, un nom est généré automatiquement par StyleVision.

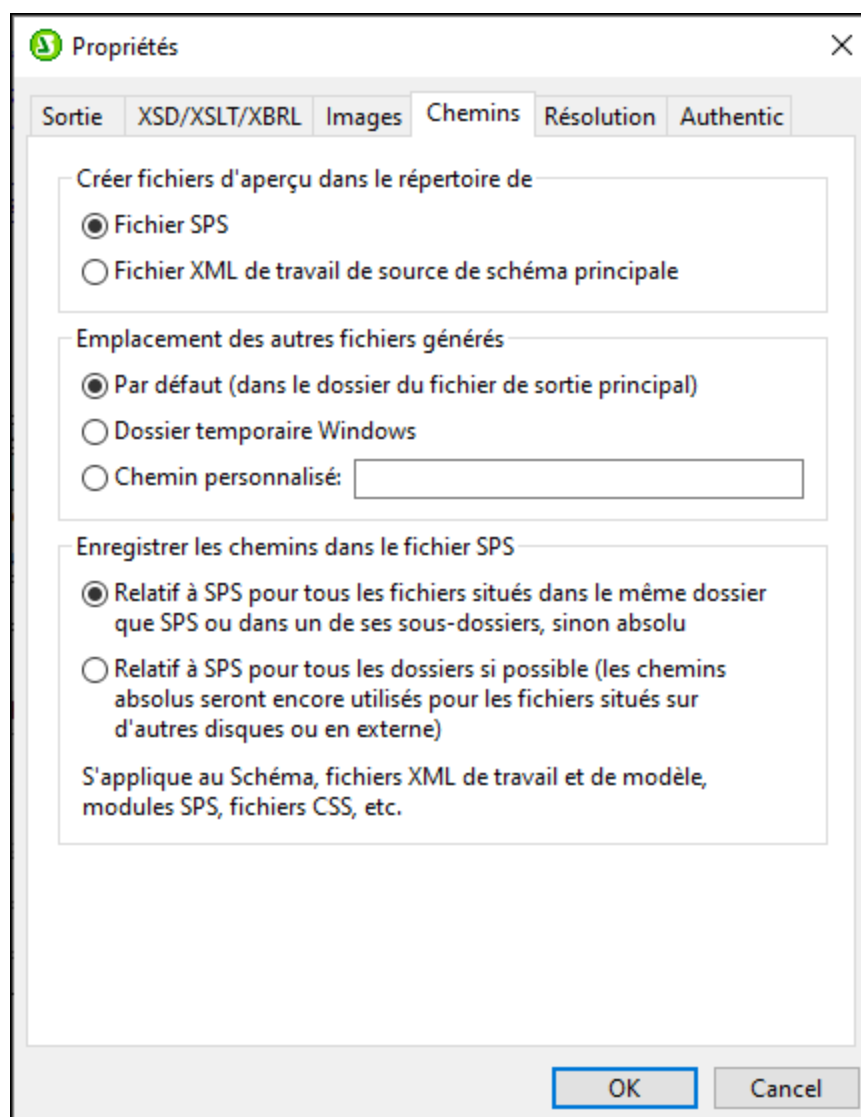
Syntaxe du modèle

Les modèles sont utilisés pour rendre le string du message d'entrée plus lisible dans le code-barres. Dans le modèle, chaque caractère de texte du message d'entrée est représenté par un trait de soulignement "_". Tout autre caractère inclus dans le modèle est inséré aux emplacements correspondants dans le texte du message d'entrée. La barre oblique inverse "\" est un symbole d'échappement. Donc, la combinaison de \"?' insérera le caractère '?' dans le texte de message de sortie où '?' peut être tout caractère. Le caractère '#' peut être utilisé pour supprimer un caractère depuis le message original. Ces points relatifs à la syntaxe du modèle sont illustrés avec les exemples ci-dessus.

Texte de message d'entrée	Pattern	Texte de message de sortie
123456	__ __ __	12 34 56
15032011094655	______ __:__:__ UTC	15\ 03\ 2011 09:46:55 UTC
15- 03- 2011	__#/_#/_	15/ 03/ 2011

Générer des fichiers de sortie

Les fichiers d'images du code-barres qui sont générés pour la sortie sont enregistrés aux emplacements spécifiés dans l'onglet Chemins du dialogue des Propriétés (*capture d'écran ci-dessous*), qui est accédé par la commande du menu **Fichier | Propriétés**.



Les fichiers d'image du code-barres pour les aperçus peuvent être créés dans le même répertoire que le fichier SPS ou le fichier de travail XML. Ils s'agit de fichiers temporaires, qui sont supprimés quand le fichier SPS est

fermé. Les fichiers d'images du code-barres qui sont créés lorsque la sortie est générée utilisant la commande **Fichier | Enregistrer fichier généré** peuvent être créés à tout emplacement. Leur emplacement cible est spécifié dans le volet, *Emplacement de fichiers générés supplémentaires* (voir la capture d'écran ci-dessus).

5.12 Modules de mise en page

Les modules de mise en page sont des objets contenant une mise en page. Le module en entier est inséré dans le design SPS et apparaît comme bloc à l'intérieur du flux du document. À l'intérieur d'un Module de mise en page, de multiples cases de mise en page, chacune contenant les éléments de design SPS standard, peuvent être placées conformément aux exigences du design. Pour cette raison, en utilisant les Modules de mise en page, les designers peuvent créer une mise en page juste comme ils utiliseraient une application design graphique basée sur un plan de travail.

Les étapes pour la création d'un module de mise en page sont comme suit :

1. Insérer un [Conteneur de mise en page](#)¹⁷¹. Le conteneur de mise en page peut occuper toute la largeur de la page ou peut avoir d'autres dimensions selon votre choix. Il peut contenir un blueprint du design pour servir comme guide de design et il peut être formaté (dans la barre latérale des Styles) utilisant les styles pour le conteneur de mise en page.
2. Insérer une ou plusieurs [cases de mise en page](#)¹⁷⁵ dans le conteneur de mise en page. Les cases de mise en page peuvent contenir de multiples éléments design (y compris du texte statique, des nœuds de schéma, des calculs automatiques, des images, listes, etc), et elles peuvent être formatées (dans la barre latérale des Styles) utilisant des styles pour la case de mise en page. Les cases de mise en page peuvent aussi être déplacées l'une par rapport à l'autre à l'intérieur du conteneur de mise en page et peuvent être positionnées devant ou derrière chacune.
3. Des [lignes](#)¹⁷⁸ peuvent être tirées, formatées, positionnées et déplacées vers l'avant ou l'arrière de la pile des objets de mise en page (cases de mise en page et autres lignes).

Note : les modules de mise en page ne sont pas rendus dans la sortie de texte.

Designs basés sur des formulaires

Quand vous [créez un nouveau fichier SPS](#)⁴⁵², vous avez le choix entre créer un design flottant librement et un design basé sur la forme. Un design « form-based » est essentiellement un design SPS consistant en un conteneur de mise en page.

Note : les modules de mise en page sont pris en charge dans Authentic View uniquement dans les éditions Entreprise des produits Altova.

5.12.1 Conteneurs de mise en page

Un conteneur de mise en page a les propriétés suivantes :

- Il peut être [inséré](#)¹⁷² à l'intérieur du flux d'un document, c'est-à-dire à l'intérieur d'un modèle. Ou il peut être inséré comme conteneur à l'intérieur du design du document dans lequel il est créé.
- Il peut avoir les mêmes dimensions que la dimension de la page définie pour cette section (la propriété « Auto-Fit to Page » des conteneurs de mise en page). Ou il peut avoir toute autre dimension que vous spécifiez. Voir la section [dimension du conteneur de mise en page](#)¹⁷² ci-dessous pour les détails.
- Une [grille de mise en page](#)¹⁷² et une [fonction zoom](#)¹⁷³ rendent le positionnement d'objets dans le conteneur de mise en page plus facile.
- Il peut contenir des [propriétés de style](#)¹⁷³, telles que les bords, les couleurs d'arrière-plan, les propriétés de police pour tout le conteneur, etc.

- Il peut [contenir des cases de mise en page et des lignes](#) ¹⁷⁴, mais aucun autre élément de design. (Tous les éléments de design doivent être placés à l'intérieur des cases de mise en page.)
- Il peut contenir un [blueprint](#) ¹⁷⁴, qui est une image placée sur le plan de travail pour servir de modèle de référence pour le designer. Le design peut ensuite être créé pour correspondre exactement au blueprint.

Note : les modules de mise en page sont pris en charge dans Authentic View uniquement dans les produits des Enterprise Editions d'Altova.

Insérer un conteneur de mise en page

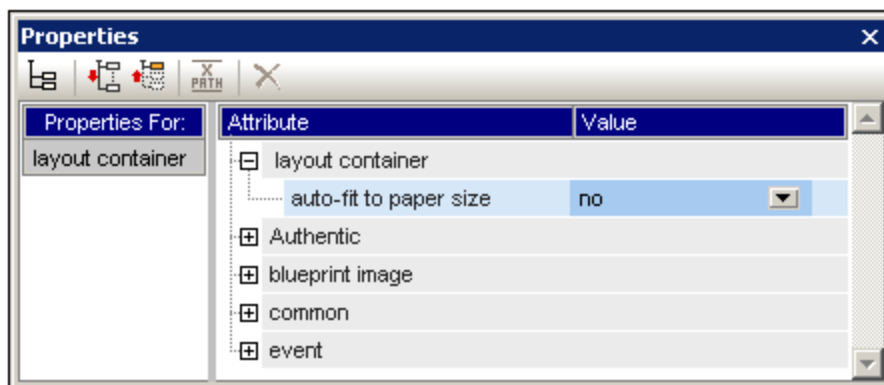
Pour insérer un conteneur de mise en page, cliquez sur l'icône **Insérer Conteneur de mise en page** dans la [barre d'outils Insérer Éléments de design](#) ⁴⁴⁷ et cliquez dans l'emplacement du design où le conteneur de mise en page doit être inséré. Un dialogue apparaît demandant si vous souhaitez ajuster automatiquement le conteneur de mise en page à la page. Si vous cliquez sur **Yes**, le conteneur de mise en page aura la même dimension que les dimensions de page définies dans les propriétés de mise en page de la page de la section du document particulier. Si vous cliquez sur **No**, alors un conteneur de mise en page avec une dimension par défaut de 3.5in x 5.0in est créé.

Notez qu'un conteneur de mise en page peut aussi être créé au moment où vous créez un fichier SPS.

Dimension du conteneur de mise en page

Il y a deux groupes de propriétés qui affectent la dimension du conteneur de mise en page :

- La propriété *Auto-Fit Page Size* (barre latérale des propriétés, *capture d'écran ci-dessous*) peut être définie comme *yes* pour créer un conteneur de mise en page ayant les mêmes dimensions que celles spécifiées pour les pages dans cette section du document. Une valeur *no* pour cette propriété crée un conteneur de mise en page avec une dimension qui se laisse personnaliser.



- Les propriétés *height* et *width* du groupe des détails des styles du conteneur de mise en page (dans la barre latérale des Styles) spécifient les dimensions du conteneur de mise en page. Les dimensions peuvent aussi être modifiées directement dans le design en glissant les marges droite et du bas du conteneur de mise en page. Notez que les propriétés *height* et *width* prendront effet uniquement lorsque la propriété *Auto-Fit Page Size* contient la valeur *no*.

Grille du conteneur de mise en page

Le conteneur de mise en page a une grille qui aide à résoudre des questions d'espacement dans la mise en page. Les paramètres suivants contrôlent l'utilisation de la grille :

- *Afficher/Dissimuler Grille* : Une commande bascule dans la barre d'outils Insérer Éléments de design active ou désactive l'affichage de la grille.
- *Dimension de la grille* : Dans l'onglet Design du dialogue des Options pour les longueurs horizontales et verticales peuvent être spécifiées. Notez que si les unités de longueur très volumineuses sont sélectionnées, la grille pourrait ne pas être clairement visible.
- *Aligner sur la grille* : Une commande bascule dans la barre d'outils Insérer des Éléments Design vous permet d'activer ou de désactiver la fonction Snap vers Grille. Lorsque la fonction Snap vers Grille est activée, les bords supérieur et gauche des cases de mises en page et les points de terminaison des lignes de mise en page s'alignent avec les lignes et points de grille, respectivement.

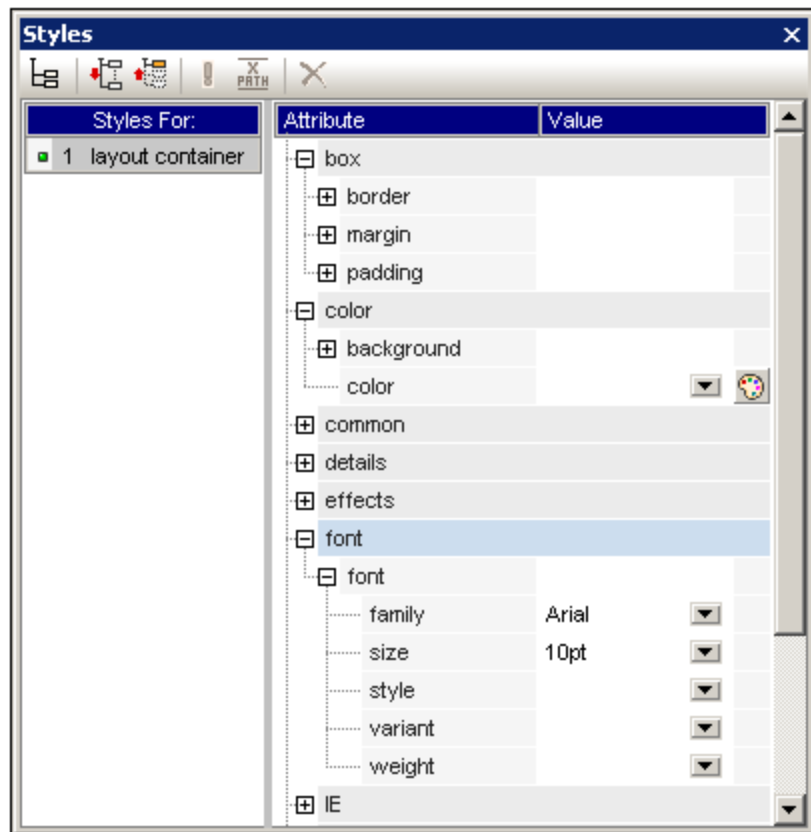
Zoom

Pour que les objets soient placés de manière plus précise, vous pouvez amplifier l'aperçu. Faites ceci en changeant le facteur Zoom dans la liste de choix (dans la barre d'outils autonome), ou en appuyant sur la clé **Ctrl** et en scrollant avec la souris.

Propriétés de style du conteneur de mise en page

Il existe deux types de propriétés de style qui peuvent être appliquées aux conteneurs de mise en page :

- Celles qui s'appliquent uniquement au conteneur de mise en page et qui ne peuvent pas être héritées, telles que les propriétés *border* et *background-color*.
- Celles qui peuvent être héritées par les cases de mise en page dans le conteneur de mise en page, telles que les propriétés de police.



Les propriétés de style d'un conteneur de mise en page sont définies dans les styles *Conteneur de mise en page* dans la barre latérale Styles (*capture d'écran ci-dessus*).

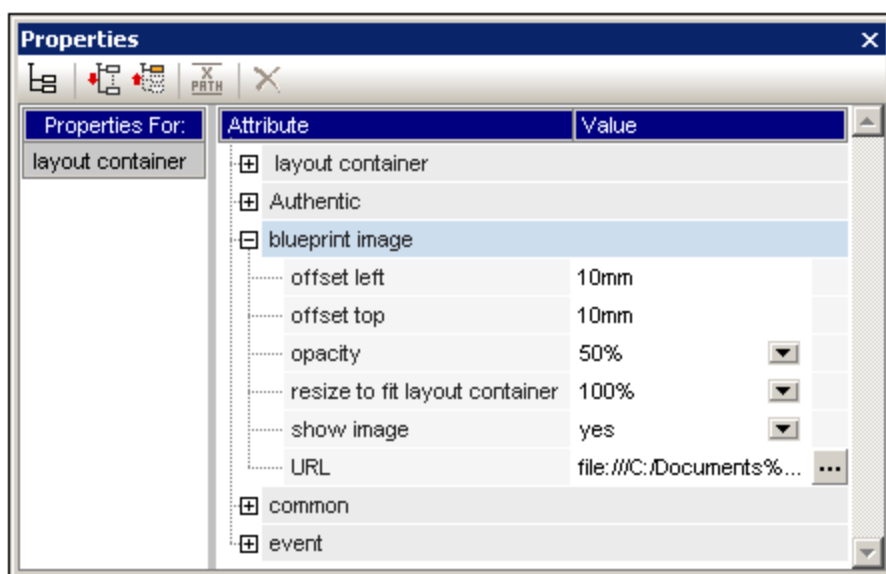
Contenus du Conteneur de mise en page

Les seuls items de design qui peuvent être contenus dans le conteneur de mise en page sont les cases de mise en page et les lignes. De plus, un blueprint (qui n'est pas un élément de design) peut être placé dans le Conteneur de mise en page comme aide au design. Tous les éléments de design doivent être placés dans la case de mise en page.

Blueprints

Un blueprint à la fois peut être placé dans le conteneur de mise en page pour aider le designer à créer le fichier SPS. Le blueprint est un fichier d'image qui peut être placé pour être ajusté exactement à la dimension du conteneur de mise en page. De manière alternative, si une image blueprint est plus petite que le conteneur de mise en page, elle peut être décalée vers l'emplacement désiré dans le design (*voir les propriétés de l'image blueprint ci-dessous*). Le designer peut utiliser le blueprint en reproduisant le design SPS par-dessus le design blueprint. De cette manière, le designer sera capable de concevoir les éléments dans la mise en page exactement comme dans le blueprint. Le blueprint apparaîtra uniquement dans l'Aperçu de design, mais dans **aucun** autre aperçu de sortie : la raison est que son seul objectif est d'assister lors de la création du design de SPS.

Les propriétés blueprint peuvent être contrôlées via le groupe de propriétés *Image blueprint* des propriétés du conteneur de mise en page (dans la barre latérale des propriétés, *capture d'écran ci-dessous*).



L'opacité du blueprint dans le conteneur de mise en page peut être spécifiée pour que le blueprint ne puisse pas interférer avec la consultation du design. L'affichage de l'image blueprint peut aussi être désactivé avec la propriété *Afficher image*, le cas échéant.

5.12.2 Cases de mise en page

Chaque élément de design dans une mise en page (tel que le texte statique, les nœuds de schéma, les calculs automatiques, les images, les listes, etc) doivent être placées dans la case de mise en page. Les cases de mise en page contenant les éléments de design sont présentées telles que requises dans le Conteneur de mise en page. Notez qu'un élément de design ne peut pas être placé directement dans un conteneur de mise en page ; il peut être placé dans une case de mise en page.

Cette section décrit comment les cases de mise en page sont utilisées et organisées en les sous-sections suivantes :

- [Insérer Case de mise en page](#) ¹⁷⁵
- [Sélectionner et déplacer les Cases de mise en page](#) ¹⁷⁵
- [Modifier la taille de la case de mise en page](#) ¹⁷⁵
- [Définir les propriétés de style de la case de mise en page](#) ¹⁷⁶
- [Insérer le contenu dans la case de mise en page](#) ¹⁷⁷
- [Ordre d'empilement des cases de mise en page](#) ¹⁷⁷

Insérer une Case de mise en page

Une case de mise en page peut être insérée uniquement dans le [Conteneur de mise en page](#) ¹⁷¹. Pour ajouter une case de mise en page, cliquez d'abord sur l'icône de la Case de mise en page dans la barre d'outils [Insérer Éléments de design](#) ⁴⁴⁷, puis cliquez dans l'emplacement à l'intérieur du Conteneur de mise en page où vous souhaitez insérer la Case de mise en page. Une Case de mise en page sera insérée, avec le bord gauche supérieur positionné là où vous avez cliqué. La case sera transparente, n'aura pas de bords, et n'aura pas de texte par défaut.

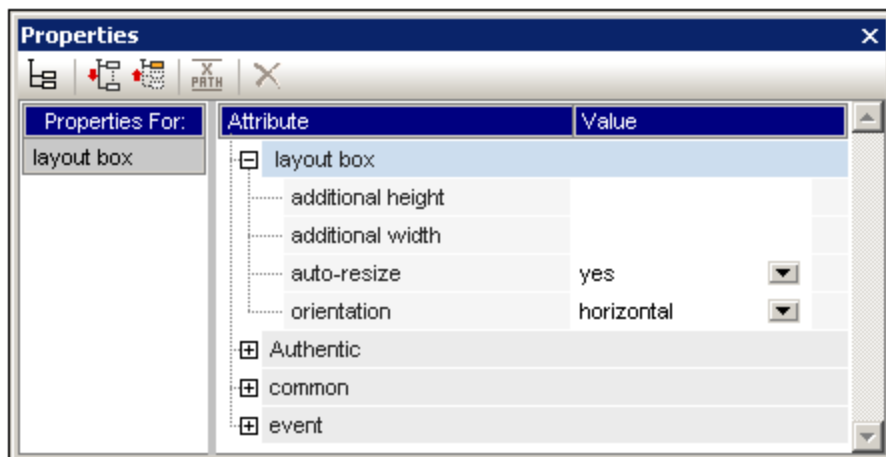
Sélectionner et déplacer les Cases de mise en page

Pour sélectionner une Case de mise en page, placez le curseur au-dessus du bord gauche ou sur le bord supérieur de la Case de mise en page pour que le curseur devienne une double flèche croisée. Lorsque ceci arrive, cliquez pour sélectionner la Case de mise en page. Si vous gardez le bouton de la souris enfoncé, vous pouvez déplacer la Case de mise en page vers un autre emplacement à l'intérieur de son Conteneur de mise en page. Vous pouvez aussi déplacer une Case de mise en page à gauche, à droite, vers le haut ou vers le bas en la sélectionnant et ensuite en appuyant sur la clé du curseur pour la direction requise. Quand la Case à cocher est sélectionnée, ses propriétés et styles sont affichés dans les barres latérales respectives.

Taille de la Case de mise en page

Chaque Case de mise en page a une propriété appelée *Auto-Resize* (voir la capture d'écran ci-dessous). Quand la valeur de cette propriété est définie comme *oui*, la Case de mise en page se redimensionne automatiquement pour accommoder exactement tout contenu statique (y compris le markup) qui est inséré dans le Mode Design. Lorsque la valeur du redimensionnement automatique est définie comme *non*, la taille de la Case de mise en page ne change pas automatiquement quand le contenu y est inséré.

Pour changer la dimension de la Case de mise en page manuellement, glissez son bord droit et le bord inférieur. Vous pouvez aussi modifier la taille de la Case de mise en page en utilisant les clés du curseur pour déplacer les bords droits et du bas de la case. À cette fin, [sélectionnez la Case de mise en page](#) ¹⁷⁵. Ensuite, procédez comme suit : (i) pour déplacer le bord droit, maintenez la clé **Shift** appuyée et appuyez sur la clé du curseur à droite ou à gauche jusqu'à ce que la taille requise soit obtenue ; (ii) pour déplacer le bord inférieur, maintenez la clé **Shift** appuyée et sur la clé du curseur supérieure ou inférieure.

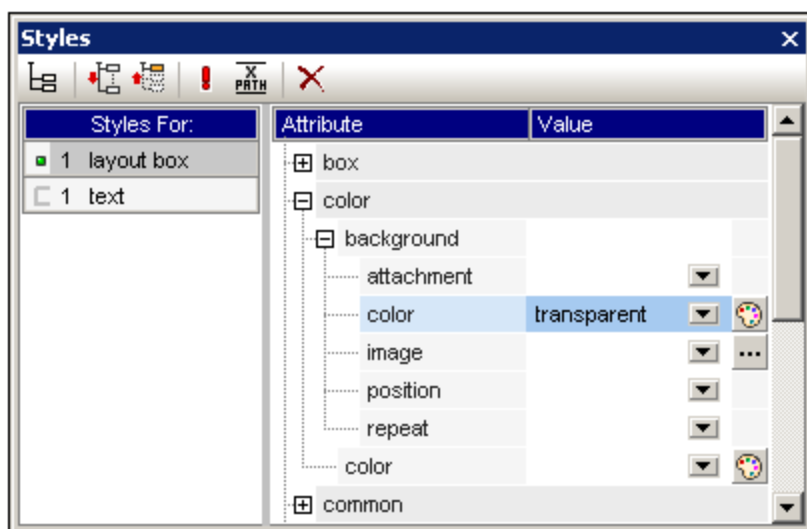


Les propriétés *Additional Height* et *Additional Width* confèrent les longueurs qui sont supplémentaires aux dimensions optimales comme déterminées par le redimensionnement automatique. Les longueurs supplémentaires sont obtenues quand une Case de mise en page est redimensionnée manuellement. À l'inverse, en changeant les valeurs de ces deux propriétés, la taille de la Case de mise en page peut être changée.

Note : dans une case de mise en page, un saut de ligne est obtenu en appuyant sur la clé **Enter**. Ceci est important car si du contenu est ajouté qui ne contient pas de saut de ligne, alors la longueur de la ligne actuelle augmente, augmentant ainsi la largeur optimale de la Case de mise en page et—par ailleurs—affectant la valeur *Additional Width*, qui est calculée avec la référence vers la largeur optimale.

Propriétés de style de la case de mise en page

Les propriétés de style d'une Case de mise en page sont définies dans les styles *Case de mise en page* dans la barre latérale Styles (*capture d'écran ci-dessous*). Les styles sont affichés quand la Case de mise en page est sélectionnée et peuvent être édités.

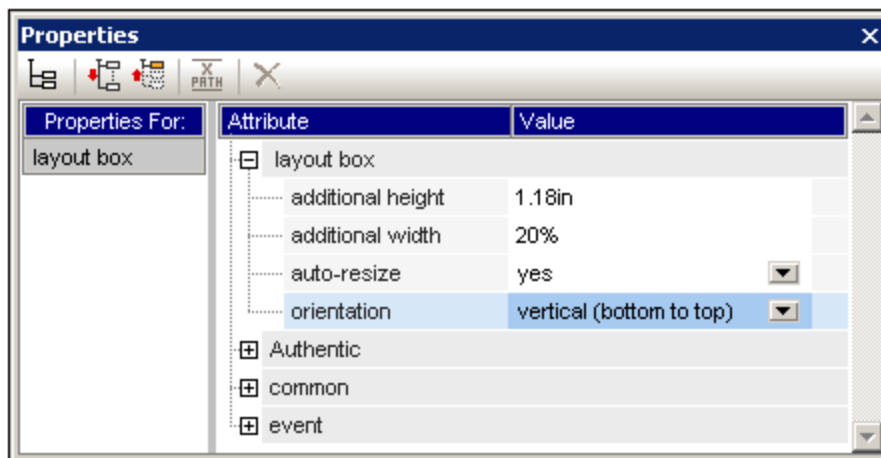


Note : la valeur de la *couleur d'arrière-plan* de `transparent` peut être sélectionnée dans la liste déroulante de la liste de choix de la propriété (elle n'est pas disponible dans la palette de couleur). La signification de cette valeur dans une situation où la Case de mise page fait partie d'une pile est expliquée ci-dessous.

Insérer le contenu dans la Case de mise en page

Tout type d'élément de design peut être inséré dans la Case de mise en page, et est inséré comme normalement dans SPS. Notez, toutefois, que ni le [Conteneur de mise en page](#)¹⁷¹ ni la [Ligne de mise en page](#)¹⁷⁸ ne peuvent être insérés dans la Case de mise en page. Veuillez noter les points suivants :

- Quand les éléments de design sont insérés qui requièrent un nœud contextuel, le nœud actuel sera pris comme nœud contextuel. Le nœud actuel est le nœud au sein duquel le module de mise en page a été créé.
- Le texte dans les cellules 90 degrés dans le sens de la montre ou dans le sens inverse de la montre pour que le texte soit vertical, se lisant du haut vers le bas et du bas vers le haut, respectivement. À cette fin, dans le design, sélectionnez le texte qui doit être pivoté et, dans la barre latérale des Propriétés (*capture d'écran ci-dessous*), sélectionnez `LayoutBox`. Dans le groupe des propriétés *Case de mise en page*, sélectionnez la valeur requise pour la propriété *Orientation*.

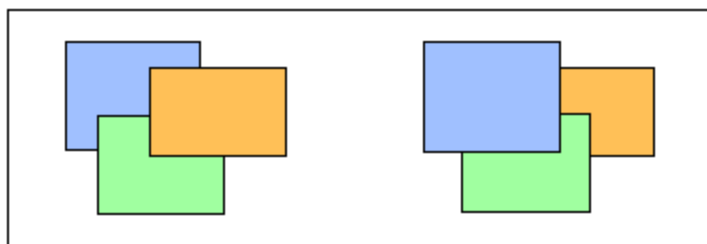


Notez les points suivants :

- La rotation sera appliquée dans la sortie, mais ne sera pas affichée dans le design.
- Cette propriété peut aussi être appliquée au texte dans les [cellules de table](#)¹³⁵.

Ordre d'empilement des Cases de mise en page

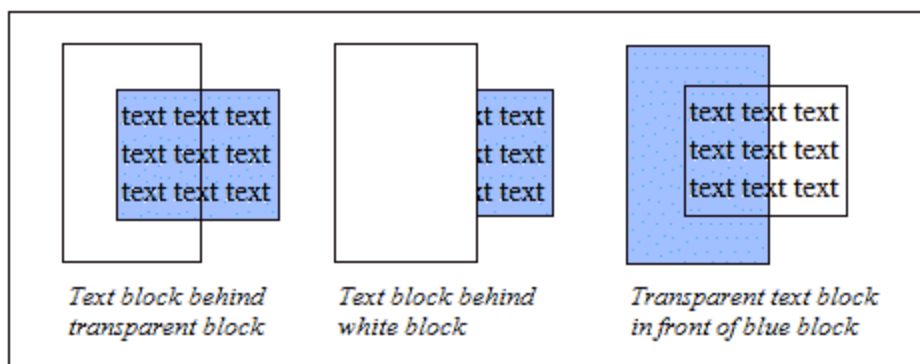
Les cases de mise en page peuvent être placées l'une sur l'autre. Lorsqu'une Case de mise en page est placée au-dessus de l'autre, alors, si elle est opaque, elle masque cette partie de la Case de mise en page qu'elle recouvre. Ce comportement peut être élargi à une pile de plusieurs cases de mise en page. Dans une telle pile, seule la case de mise en page la plus élevée sera entièrement visible ; les autres seront partiellement ou entièrement couvertes.



Les cases de mise en page peuvent être reculées ou avancées en utilisant les commandes de menu **Ordre** dans le menu contextuel de la case de mise en page sélectionnée. En utilisant ces commandes, une case de mise en page peut être triée comme suit : (i) relative à son voisin le plus proche sur la pile (les commandes **Avancer** et **Reculer**), ou (ii) relative à la pile entière (les commandes **Bring to Front** et **Send to Back**). Dans la capture d'écran ci-dessus, l'ordre d'empilement de l'avant vers l'arrière est comme suit :

- *Pile gauche* : orange, vert, bleu
- *Pile droite* : bleu, vert, orange

Notez que les cases de mise en page avec des arrière-plans transparents (arrière-plan par défaut des cases de mise en page) peuvent avoir l'air de ne pas bouger l'une par rapport à l'autre, en particulier si plus d'une case dans la pile est transparente et si les cases n'ont pas de bords. La capture d'écran ci-dessous présente plusieurs manières dont la transparence affecte l'empilement.



Note : les [lignes de mise en page](#) ¹⁷⁸ peuvent aussi être ajoutées à une pile de cases de mise en page, et chaque ligne peut être déplacée par rapport aux autres items dans la pile.

5.12.3 Lignes

Les lignes peuvent être [insérées dans un conteneur de mise en page](#) ¹⁷⁸ (mais pas dans des cases de mise en page), puis [sélectionnées, redimensionnées et déplacées](#) ¹⁷⁹ à l'intérieur d'un conteneur de mise en page, [propriétés assignées](#) ¹⁷⁹, et [déplacées vers l'arrière et vers l'avant d'une pile d'items de mise en page](#) ¹⁸⁰ consistant de cases de mise en page et d'autres lignes.

Insérer une ligne

Pour ajouter une ligne à un conteneur de mise en page, procédez comme suit :

1. Cliquez sur l'icône Insérer ligne dans la barre d'outils [Insérer des Éléments Design](#) ⁴⁴⁷.

2. Cliquez dans l'emplacement à l'intérieur du conteneur de mise en page là où vous voulez situer le point de départ de la ligne.
3. Sans lâcher le bouton de la souris, tirez la ligne du point de départ au point de terminaison souhaité. Puis, lâchez le bouton de la souris.

Une ligne noire sera insérée, avec un point à chaque fin de ligne indiquant respectivement les points de départ et de terminaison.

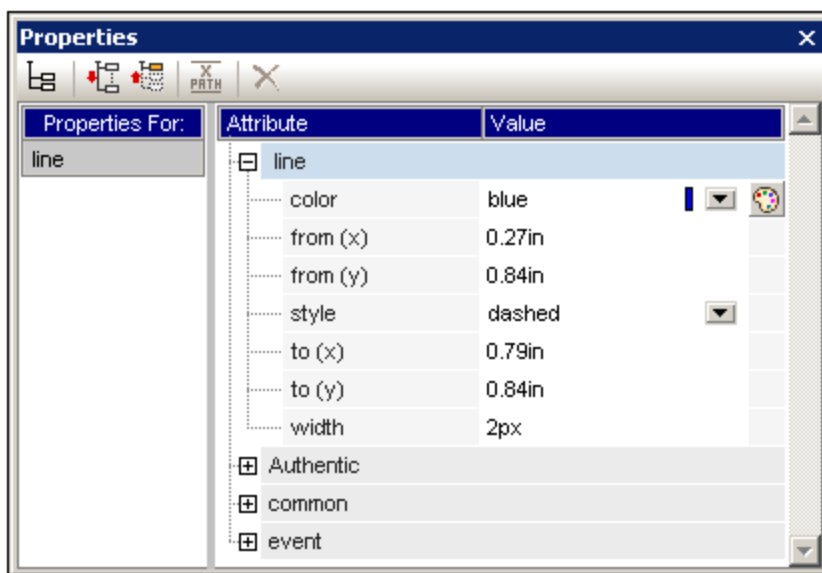
Sélectionner, déplacer et redimensionner une ligne

Dans la fenêtre principale, vous pouvez exécuter les fonctions glisser-et-déposer suivantes :

- Pour sélectionner une ligne, cliquez sur n'importe quelle partie de la ligne (le curseur devient une double flèche croisée lorsqu'il se trouve au-dessus de la ligne). Une fois qu'une ligne est sélectionnée, ses propriétés sont affichées dans la barre latérale des propriétés et peuvent y être éditées (*voir capture d'écran ci-dessus*).
- Pour déplacer une ligne, sélectionnez-la et glissez-la à l'emplacement souhaité. Vous pouvez aussi déplacer une ligne à gauche, à droite, vers le haut ou vers le bas en la sélectionnant, puis en appuyant sur la clé du curseur pour la direction requise.
- Pour redimensionner ou réorienter graphiquement une ligne, sélectionnez soit le point de départ ou le point de terminaison et repositionnez-le pour obtenir une nouvelle dimension et/ou orientation. Vous pouvez également redimensionner ou réorienter une ligne en appuyant sur **Shift** et les touches du curseur : les touches du curseur droite et gauche déplacent le point de terminaison droite à droite et à gauche, les touches du curseur haut et bas déplacent le point de terminaison droite respectivement vers le haut et vers le bas.

Propriétés de ligne

Quand une ligne est sélectionnée, ses propriétés sont affichées dans la barre latérale des Propriétés (*capture d'écran ci-dessous*), et les propriétés (recensées ci-dessous) peuvent être éditées dans la barre latérale. Vous pouvez aussi cliquer avec la touche de droite sur une ligne pour faire apparaître la barre latérale des Propriétés avec les propriétés de la ligne y incluses.

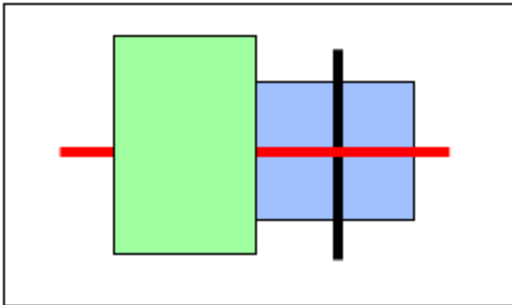


Les propriétés de ligne suivantes peuvent être éditées dans la barre latérale des Propriétés.

- *Couleur* : Spécifie une couleur pour la ligne. La valeur par défaut est noire.
- *Taille et position* : L'emplacement des points de départ et de terminaison de la ligne peut être spécifié utilisant un système de coordonnées x-y (horizontal-vertical). Le cadre de référence est créé avec le bord gauche supérieur du conteneur de mise en page ayant des coordonnées ($x=0$, $y=0$).
- *Largeur* : Spécifie l'épaisseur de la ligne.

Ordre des lignes et d'empilement

Quand une ligne est dans une pile consistant de cases de mises en page et autres lignes, elle peut être envoyée vers l'arrière ou vers l'avant utilisant les commandes de menu **Ordre** dans le menu contextuel de la ligne sélectionnée. En utilisant ces commandes, une ligne peut être triée comme suit : (i) relative à son voisin le plus proche sur la pile (les commandes **Avancer** et **Reculer**), ou (ii) relative à la pile entière (les commandes **Bring to Front** et **Send to Back**).



Dans la capture d'écran ci-dessus, l'ordre d'empilement de l'avant vers l'arrière est comme suit : case verte, ligne rouge, ligne noire, case bleue.

5.13 La fonction Change-To

La fonction **Change-To** est disponible quand un modèle ou les contenus d'un modèle sont sélectionnés et vous permettent de changer : (i) le nœud pour lequel le modèle s'applique, ou (ii) comment le nœud est créé dans le design.

Ce qui peut être changé avec la fonction Change-To

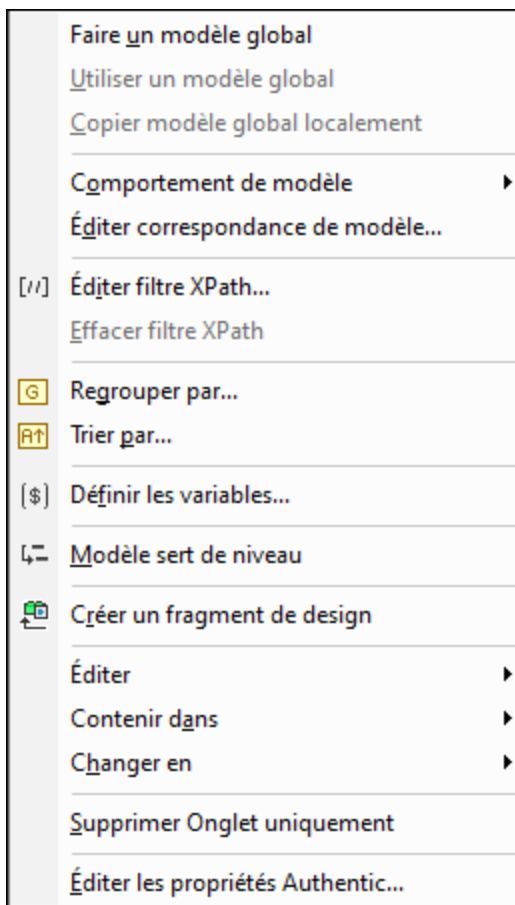
Soit son nœud ou son contenu peuvent être changés. Dans l'image ci-dessous, le nœud est sélectionné. Dans l'image à droite, les contenus du nœud sont sélectionnés.



L'élément `n1:Name` dans la capture d'écran ci-dessus a été créé comme `(contenus)`, donc les contenus du nœud sont représentés par l'espace réservé `(contenus)`. En alternative, le nœud peut avoir été créé comme autre type de contenu, par exemple, comme champ d'entrée ou zone de liste déroulante. D'autres types de contenus peuvent aussi être sélectionnés.

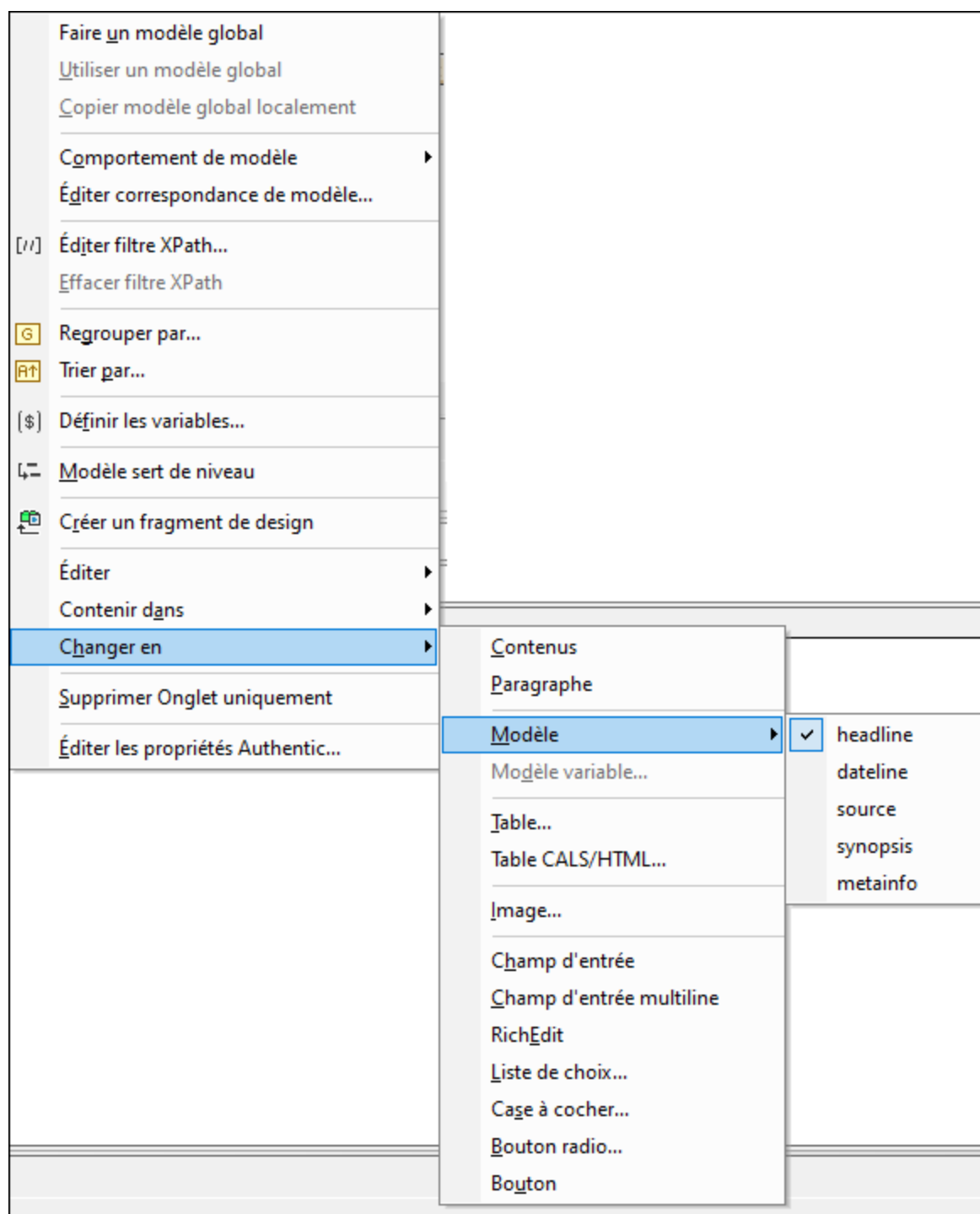
La commande Change-To

Accédez à la commande Change-To en cliquant avec la touche de droite sur votre sélection. Dans le menu contextuel qui apparaît, sélectionnez **Change-To** (capture d'écran ci-dessous).



Modifier les correspondances du modèle

Si un modèle est sélectionné, vous pouvez changer le nœud pour lequel le modèle s'applique. Ceci est utile si, par exemple, le nom d'un élément a été modifié dans le schéma. Lorsque vous passez au-dessus de la commande Change-To et sélectionnez le modèle depuis le sous-menu qui apparaît, vous êtes verrez une liste avec tous les nœuds qui peuvent être insérés comme enfant de l'élément parent du nœud sélectionné. Cliquez sur un de ces nœuds pour que le modèle soit appliqué à ce nœud.

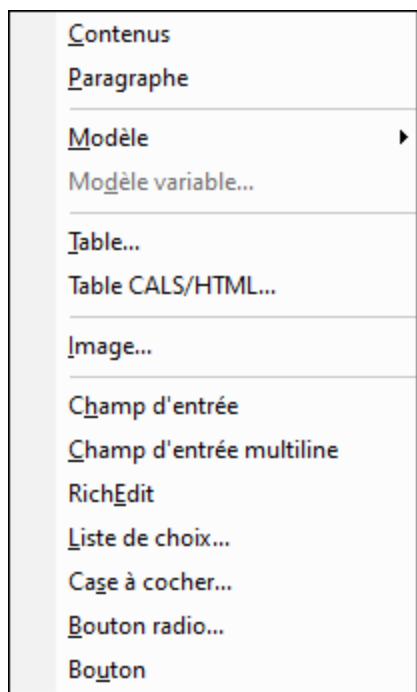


Si le nœud sélectionné a un modèle contenu qui ne correspond pas au modèle décrit, il y aura des inconsistances structurelles. De telles inconsistances sont des erreurs et elles sont indiquées avec des passages biffés en rouge dans les balises des nœuds qui sont invalides.

Vous pouvez aussi modifier la correspondance du modèle pour que celui corresponde à, non pas un nœud, mais à un [modèle de variable](#) ²⁴¹.

Changer le type de contenu du nœud

Si un modèle ou son contenu est sélectionné, alors vous pouvez changer le type de contenu du nœud tel qu'il est créé. En passant au-dessus de la commande Change-To dans le menu contextuel, le type de contenu vers lequel le nœud peut être modifié est affiché comme options dans le sous-menu qui apparaît (*voir la capture d'écran ci-dessous*).



La capture d'écran ci-dessus a été prise avec une zone de liste déroulante

6 Structure SPS

La structure d'un document SPS est pilotée par l'entrée ainsi que par la sortie, et elle est contrôlée par :

- [Sources de schéma](#)¹⁸⁷
- [SPS modulaire](#)²²⁰
- [Fragments de modèles et de design](#)²³⁴

Structure pilotée par l'entrée : fichiers SPS de schémas et modulaires

Par piloté par l'entrée, nous voulons dire que les schémas de source des fichiers SPS spécifient la structure du/des document/s d'entrée et que cette structure est la structure sur lequel le document SPS est basé. Par exemple, si un schéma de source spécifie une structure qui est la séquence des éléments `Office`, alors le design SPS pourrait avoir un modèle pour l'élément `Office`. Au moment du traitement, ce modèle sera appliqué à chaque élément `Office` dans le document des données source.

Un autre exemple afin de savoir comment la structure d'un document source pilote le design du fichier SPS peut être vu dans l'utilisation des tables. Disons qu'un élément `Office` contient de multiples enfants de l'élément `Person`, et que chaque élément `Person` contient un ensemble d'éléments enfant, tel que `Name`, `Address`, `Telephone`, etc., alors un modèle sous forme d'une table peut être créé pour l'élément `Person`. Chaque élément `Person` peut être présenté dans une ligne séparée de la table (*capture d'écran ci-dessous*), dans laquelle les colonnes sont le détails de `Person` (les éléments enfant de l'élément `Person`).

First	Last	Title (sorted by)
Loby	Matise	Accounting Manager
Frank	Further	Accounts Receivable
Vernon	Callaby	Office Manager

Un tel modèle est possible en raison de la structure de l'élément `Person` et parce que les éléments `Person` sont des frères. Dans le modèle de la table, une seule ligne est conçue pour l'élément `Person`, et ce traitement (le design de la ligne) est appliqué dans chaque élément `Person` dans le document source, créant une nouvelle ligne pour chaque élément `Person`, avec les éléments enfant formant des colonnes de table.

Vous trouverez une description des différentes sources de schéma dans la section [Sources de schéma](#)¹⁸⁷.

En plus, StyleVision permet aux SPS d'être réutilisés comme modules à l'intérieur d'autres SPS. De cette manière, les modules peuvent être inclus à l'intérieur d'une structure et peuvent la modifier. Toutefois, une structure de schéma contenue dans un module doit correspondre à la structure du schéma sous-jacent du SPS contenant. Vous trouverez dans la section [SPS modulaire](#)²²⁰ une description du travail avec le SPS modulaire.

Structure pilotée par la sortie : fragments de modèles et de design

Tandis que les sources de schéma fournissent la structure du document des données d'entrée, le design actuel du document de sortie est ce qui est spécifié dans le document SPS. Ce design est contenu dans un modèle de document appelé le modèle principal. Le modèle principal contient généralement plusieurs modèles

de composant et peut référencer des modèles globaux. Les modèles sont décrits dans la section, [Fragement de modèles et de design](#)²³⁴.

Cette composabilité (de modèles multiples) est davantage améliorée par la fonction de StyleVision nommée Fragments de design, qui permet qu'un traitement spécifique soit attribué à un fragment de document pouvant être réutilisé. Un fragment de design est différent d'un modèle global dans ce sens qu'il peut : (i) être composé de multiples modèles ; et (ii) un contenu identique avec un traitement différent peut être créé dans des fragments de design séparés, dont chacun peut être utilisé dans un modèle selon la situation. Par exemple, dans certaines situations de traitement, un nœud `Email` peut être requis comme lien qui ouvre un e-mail vide ; dans d'autres cas, l'élément `Email` peut être requis en gras et en rouge. Deux fragments de design séparés pourraient fournir le traitement respectif et les deux peuvent être réutilisés, tel que requis.

Les fragments de design sont décrits en détail dans la section [Fragments de design](#)²⁴⁵.

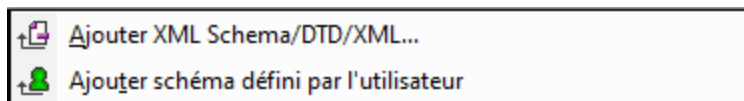
6.1 Sources de schéma

Les sources de schéma sont le point de démarrage du design, et la structure de design peut être influencée par : (i) les choix que vous faites au cours de la sélection de schéma, et (ii) les éléments racine que vous sélectionnez dans le schéma.

Sélection de schéma

La sélection du schéma pour un nouveau fichier SPS peut être faite des deux manières suivantes :

1. Cliquez sur **Fichier | Nouveau** et sélectionnez directement une source de schéma à ajouter via une des méthodes (à l'exception de **Nouveau (vide)**) disponible dans le menu qui apparaît.
2. Cliquez sur **Fichier | Nouveau**, sélectionnez Nouveau (vide) depuis le menu qui apparaît. Après avoir créé un nouveau SPS et l'avoir affiché dans la GUI, dans la [barre latérale de l'Aperçu de design](#)³⁵, sélectionnez la commande **Ajouter Nouveau schéma**. Ceci fait apparaître un menu qui recueille les méthodes que vous pouvez utiliser afin d'ajouter les différents types de schémas (*capture d'écran ci-dessous*). Chaque commande dans ce menu est décrite dans les sous-sections de cette section.



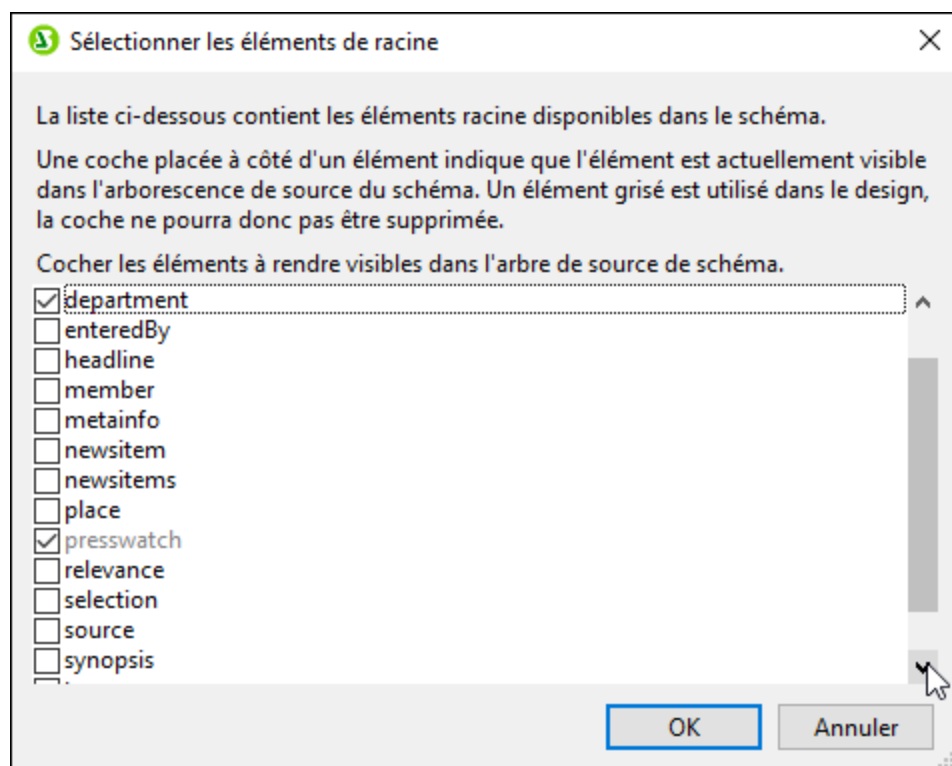
La source de schéma peut être sélectionnée depuis un fichier ou être une définie par l'utilisateur. Un point important à considérer sera si vous allez utiliser les modèles globaux, et si les éléments que vous souhaitez créer comme modèles globaux sont définis comme éléments globaux dans le schéma. Lorsque vous ajoutez un DTD du fichier, souvenez-vous que tous les éléments définis dans le DTD sont des éléments globaux. Lorsque vous ajoutez un schéma XML du fichier, il est bien de vérifier quels éléments sont définis comme éléments globaux et, si vous souhaitez apporter des modifications au schéma, si ceci est autorisé dans votre environnement XML.

Note : si vous souhaitez ajouter un espace de nom à une feuille de style SPS ou à une feuille de style XSLT générée depuis une SPS créée, l'espace de nom doit être ajouté à l'élément de niveau supérieur `schema` du schéma XML sur lequel la SPS se base.

Éléments racine

Si une source de schéma a de multiples [éléments globaux](#)²², alors les multiples éléments racine ([éléments du document](#)²²) peuvent être sélectionnés pour utilisation dans le design. Ceci permet au design SPS d'avoir des modèles qui correspondent aux multiples éléments du document. L'avantage est que si un SPS, disons `UniversalSPS.sps`, basé sur `UniversalSchema.xsd` a un modèle pour chacun de ses deux éléments racine, `Element-A` et `Element-B`, alors cet SPS peut être utilisé avec un document d'instance XML qui a l'`Element-A` comme son élément de document de même qu'avec un autre document d'instance XML qui a l'`Element-B` comme son élément de document. Pour chaque instance XML, le modèle pertinent est utilisé, tandis que l'autre n'est pas utilisé. Ceci est dû au fait que pour l'élément du document de chaque document d'instance XML, il n'existe qu'un modèle dans SPS qui correspond à cet élément de document. Par exemple, l'élément du document `/Element-A` sera complété par le modèle qui sélectionne `/Element-A` mais pas par celui qui sélectionne `/Element-B`. Dans cette connexion, il est important de se rappeler que si de multiples éléments globaux sont définis dans le schéma, un document XML avec un de ces éléments globaux est valide (assumant évidemment que sa sous-structure es valide selon le schéma).

Pour configurer un SPS pour qu'il puisse utiliser les multiples éléments racine ([éléments de document](#) ²²), cliquez sur le bouton  à droite de l'entrée /Root elements du schéma. Le dialogue suivant apparaît.



Le dialogue recense tous les éléments globaux dans le schéma. Sélectionnez les éléments globaux que vous souhaitez utiliser comme éléments racine ([éléments de document](#) ²²) et cliquez sur **OK**. L'/les élément/s sélectionné/s sera/-ont disponible/s comme éléments de document racine et sera/-ont affiché/s dans la liste des Éléments racine. Un modèle peut désormais être créé pour chacun de ces éléments de document. Chacun de ces modèles sert de modèle d'élément racine alternatif. Quand un document XML est traité avec SPS, seul un modèle d'élément racine alternatif sera utilisé : celui qui correspond à l'élément racine (ou de document) du document XML.

Donc, si un document XML ayant l'Element-A comme son élément de document est traité avec cet SPS, alors le modèle racine dans le SPS qui correspond à l'Element-A est déclenché, tous les autres modèles d'élément racine dans le SPS sont ignorés. Si un document XML ayant l'Element- B comme son élément de document est traité, alors le modèle racine dans le SPS qui correspond à l'Element- B est déclenché, tous les autres modèles d'élément racine dans le SPS sont ignorés. De cette manière, un seul SPS peut être utilisé pour traiter deux ou plus de documents XML, dont chacun a un élément racine (ou de document) différent.

6.1.1 DTD et schémas XML

Un fichier SPS peut être basé sur un schéma XML ou DTD. Un schéma XML ou DTD peut être créé comme source de schéma d'une des manières suivantes :

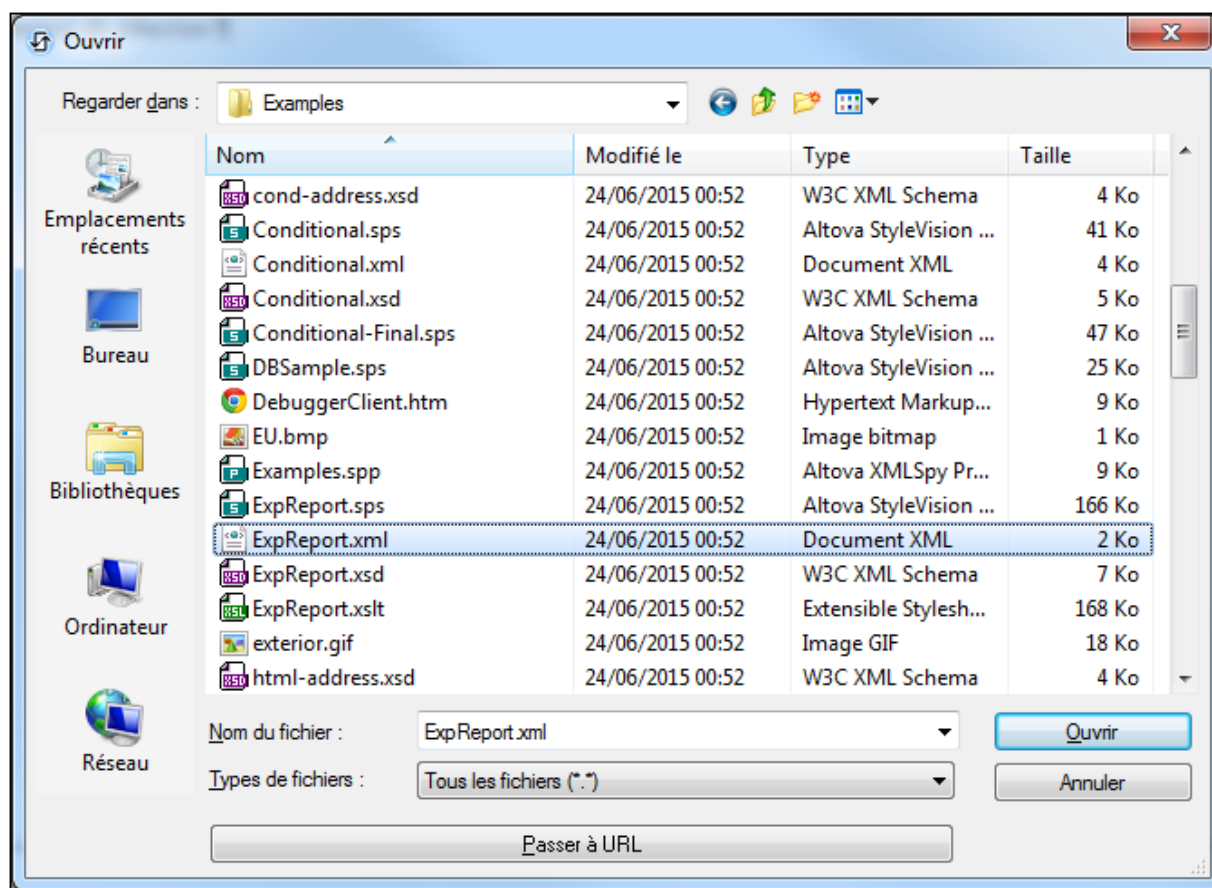
- Le Schéma XML ou DTD est créé comme source de schéma directement quand SPS est créé (**Fichier | Nouveau | Nouveau à partir de Schéma XML/DTD/XML**).

- Le Schéma XML ou DTD est ajouté au SPS (dans la [barre latérale de l'Aperçu du design](#) ³⁵).

Les commandes respectives vous invitent à naviguer depuis le Schéma XML ou DTD. Si le schéma est valide, il sera créé comme source de schéma dans l'arborescence des Sources de schéma de la barre latérale de l'arborescence de schéma. De manière alternative, un fichier XML peut être sélectionné. Si un fichier de schéma XML (.xsd) ou DTD est associé au fichier XML, ensuite le schéma XML ou fichier DTD est chargé comme schéma de source et le fichier XML est chargé comme fichier de travail XML. Si aucun schéma n'est associé au fichier XML, un dialogue apparaît demandant si vous voulez générer un schéma XML basé sur la structure et les contenus du fichier XML ou rechercher un schéma existant. Si vous choisissez de générer un schéma, le schéma généré sera chargé comme schéma de source, et le fichier XML sera chargé comme fichier de travail XML.

▼ Sélectionner et enregistrer les fichiers via des URL et des Ressources globales

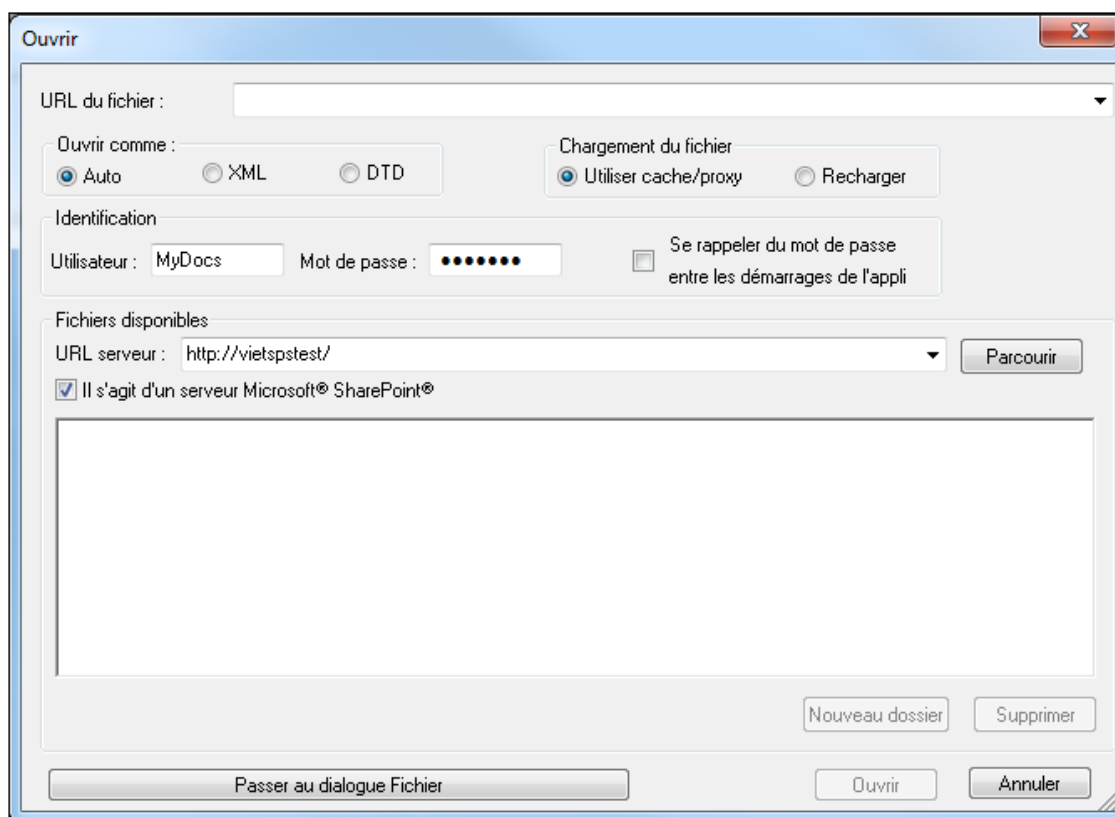
Dans plusieurs dialogues Fichiers Ouvrir et Fichier Enregistrer, vous pouvez choisir de sélectionner le fichier nécessaire ou d'enregistrer un fichier via une URL ou une Ressource globale (*voir capture d'écran ci-dessous*). Cliquer sur **Passer à URL** ou sur **Passer à Ressources globales** pour vous rendre sur un de ces processus de sélection.



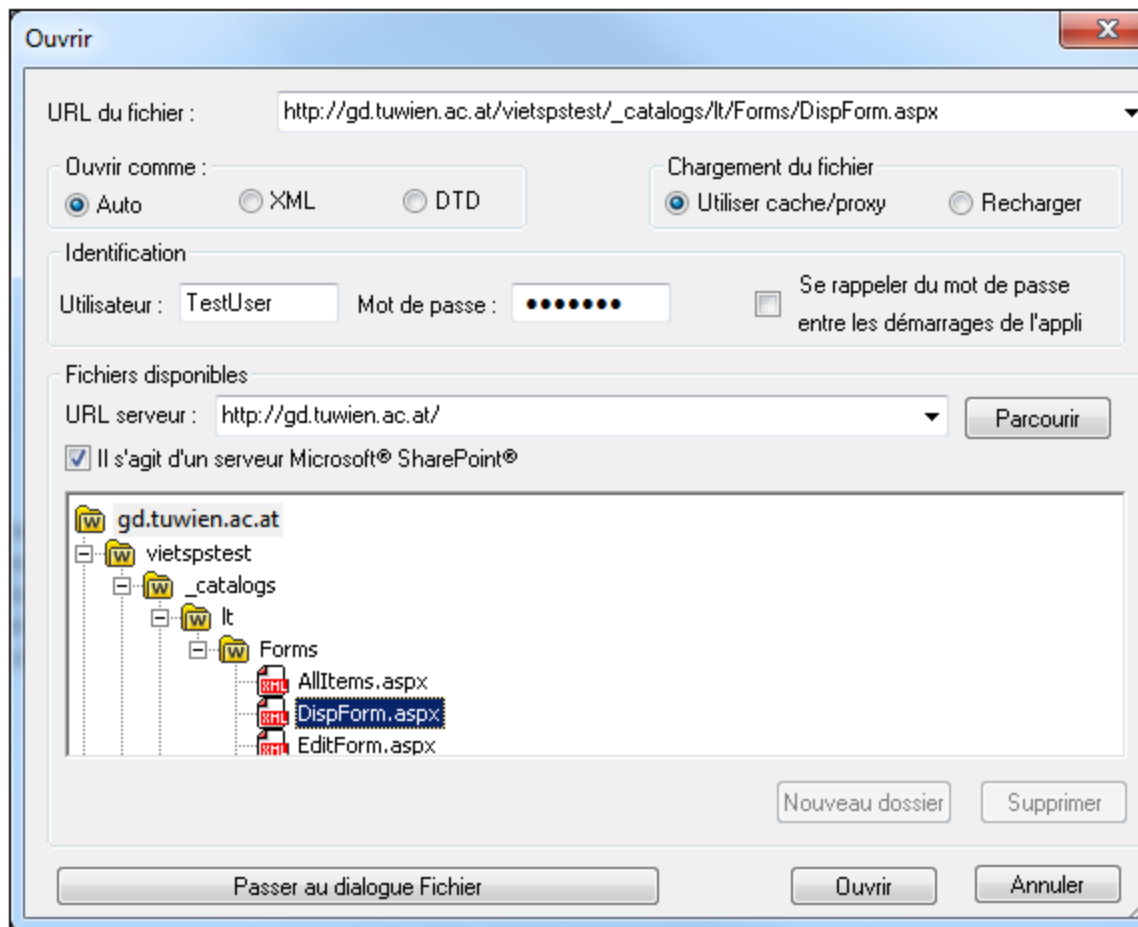
Sélectionner les fichiers via des URL

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*).



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.

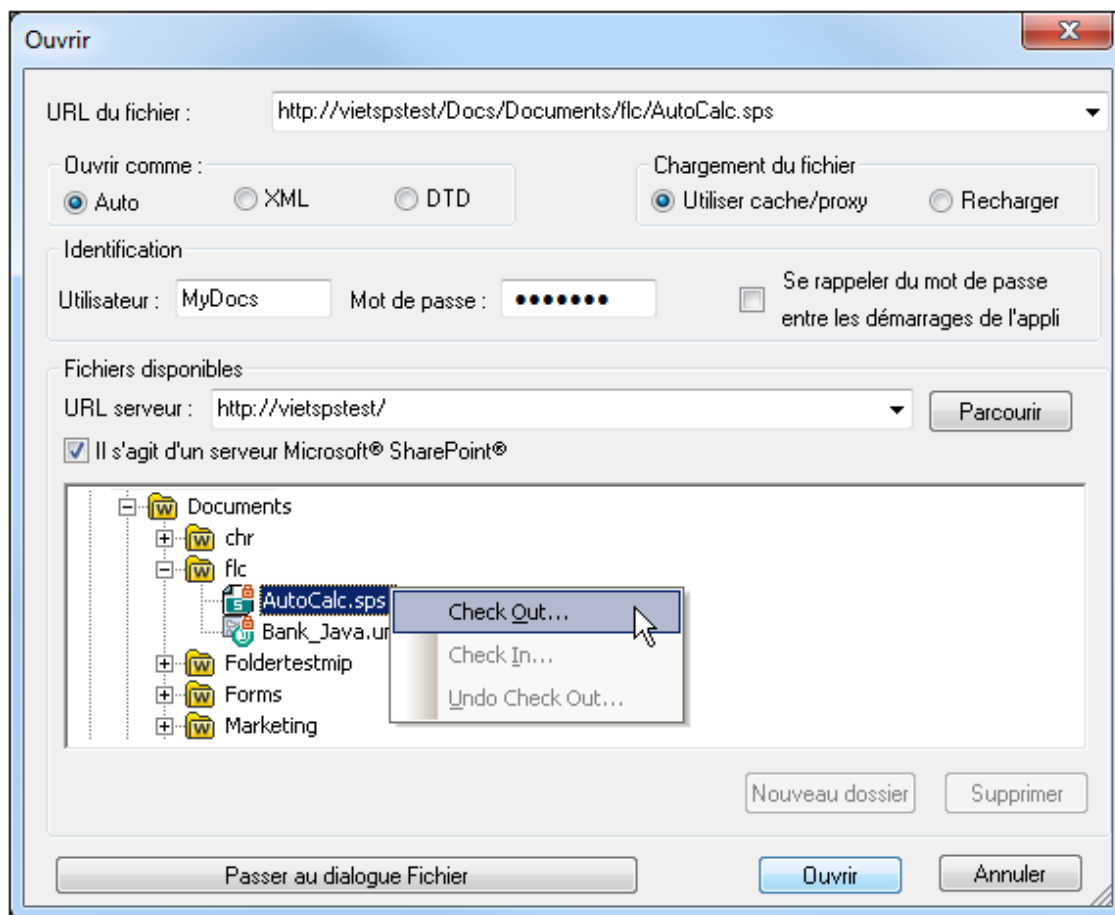
Notez les points suivants :

- 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

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.



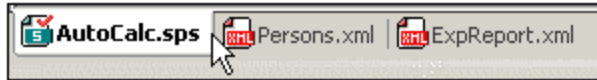
Cliquer avec la touche de droite sur un fichier, pour ouvrir un menu contextuel contenant les commandes disponibles pour ce fichier (*capture d'écran ci-dessus*).

- 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. Pour une description générale des Ressources globales, consulter la section dans cette documentation.

Le type de données anyType du Schéma XML

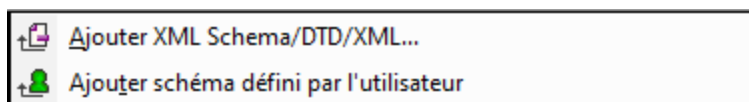
Si un élément dans le Schéma XML s'est vu attribué le type de données `anyType` du Schéma XML ou s'il n'a pas été attribué à un type de données, alors l'arborescence de schéma dans l'Arborescence de schéma montrera cet élément comme ayant tous les éléments globaux de ce schéma en tant qu'enfants possibles. Par exemple, si un élément appelé `email` ne s'est pas vu attribuer un type de données, alors il sera affiché dans l'arborescence de schéma avec tous les éléments globaux en tant qu'enfants possibles, comme : `person`, `address`, `city`, `tel`, etc. Pour éviter ceci, attribuez un type de données à l'élément `email` tel que `xs:string`.

6.1.2 Schémas définis par l'utilisateur

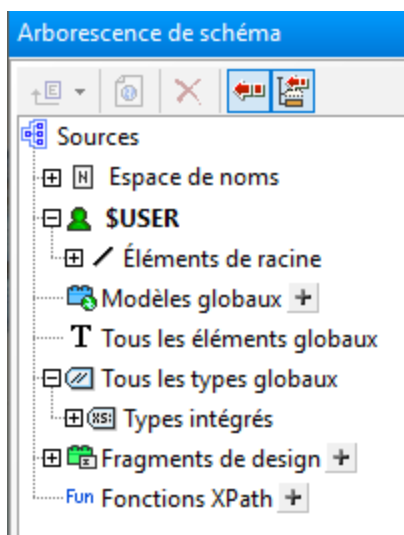
Vous pouvez créer rapidement un schéma défini par l'utilisateur dans la [barre latérale de l'Arborescence de schéma](#) ³⁷. Ceci est utile si vous avez un document XML qui n'est pas basé sur un schéma et que vous souhaitez créer un SPS pour ce document XML.

Pour ajouter et créer un schéma défini par l'utilisateur, suivez les étapes suivantes dans la barre latérale de l'arborescence de schéma :

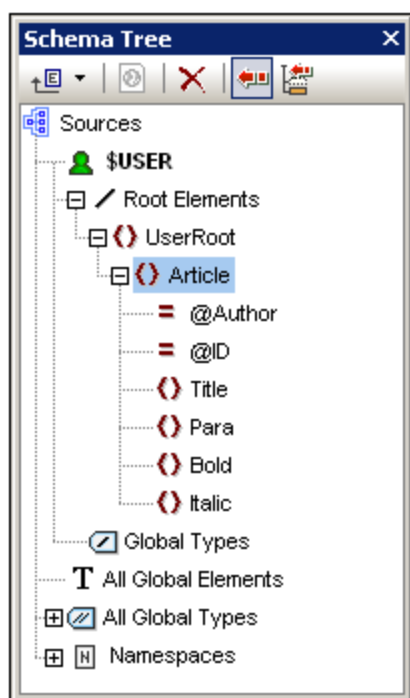
1. Cliquez sur **Fichier | Nouveau | Nouveau (vide)**. Dans la [barre latérale de l'Aperçu de design](#) ³⁵, cliquez sur la commande **Ajouter Nouvelle Source** (sous l'en-tête Sources), et sélectionnez **Ajouter le Schéma défini par l'utilisateur** (*capture d'écran ci-dessous*).



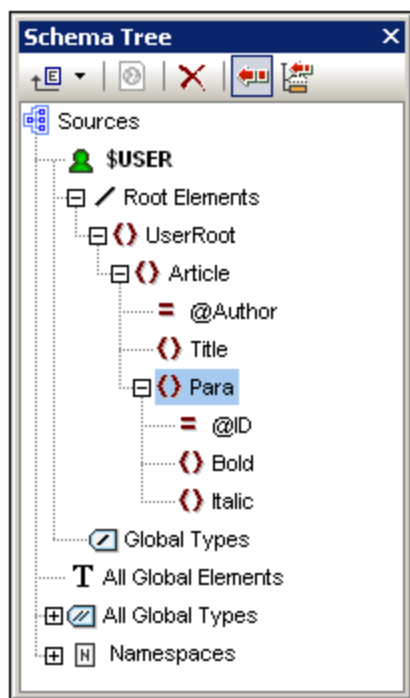
Le nouveau schéma est créé et est indiqué par les paramètres `$USER` (capture d'écran ci-dessous).



2. Dans l'arborescence des Éléments racine, il existe un seul [élément racine \(élément de document\)](#) ²² appelé `UserRoot`.
3. Double-cliquez sur `UserRoot` et renommez-le pour qu'il corresponde à l'[élément de document](#) ²² du document XML pour lequel vous êtes en train de créer ce schéma.
4. Pour attribuer un élément enfant ou un attribut à l'élément document, sélectionnez l'élément document (`UserRoot`), puis cliquez respectivement sur (i) l'icône déroulante [Ajouter nouvel élément](#) ³⁷ dans la barre d'outils de la [barre latérale Arborescence de schéma](#) ³⁷ ; et (ii) la flèche déroulante de l'icône [Ajouter nouvel élément](#) | la commande [Ajouter nouvel attribut](#). De manière alternative, vous pouvez cliquer avec la touche de droite et sélectionner la commande requise depuis le menu contextuel. Quand un élément est sélectionné, ajouter et insérer un élément ajoute le nouvel élément comme élément frère, respectivement après et avant l'élément sélectionné. Vous pouvez aussi ajouter un élément enfant et un attribut enfant. Quand un attribut est sélectionné, vous pouvez ajouter ou insérer un autre attribut, respectivement, après et avant l'attribut sélectionné. Une fois que le nouvel élément ou attribut est ajouté à l'arborescence, saisissez le nom désiré. Vous pouvez aussi glisser les nœuds à l'emplacement désiré (décrit dans la prochaine étape). Dans la capture d'écran ci-dessous, l'élément `Article` est l'élément du document. Les éléments `Title`, `Para`, `Bold`, et `Italic`, et les attributs `ID` et `Author` ont été ajoutés au niveau enfant de l'`Article`.



5. Pour déplacer les éléments **Bold** et **Italic**, et l'attribut **ID** au niveau enfants de **Para**, sélectionnez chacun individuellement et glissez l'élément **Para**. Lorsque la flèche pointant vers le bas  apparaît, déposez le nœud glissé. Il sera créé comme « enfant » de **Para** (capture d'écran ci-dessous).



6. Lorsque tout élément autre que l'élément de document est sélectionné, ajouter un nouvel élément ou attribut ajoute le nouveau nœud au même niveau que l'élément sélectionné. Glissez un nœud (élément ou attribut) dans un nœud d'élément pour le créer comme « enfant » du nœud d'élément.

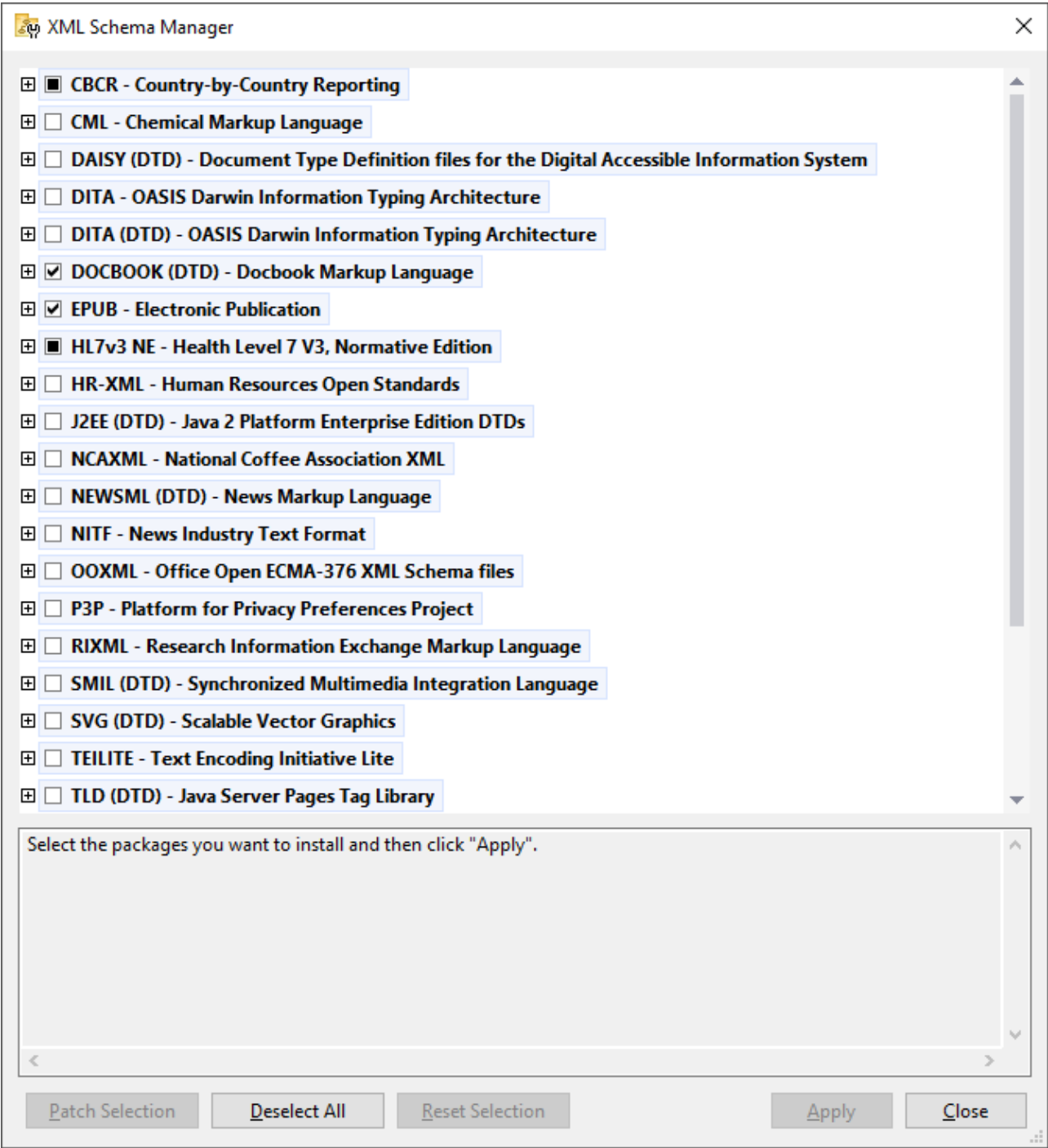
Modification des noms de nœuds et suppression de nœuds

Pour modifier le nom d'un élément ou d'un attribut, double-cliquez sur le nom et modifiez-le. Pour supprimer un nœud, sélectionnez-le et cliquez sur l'icône Supprimer  dans la barre d'outils. Vous pouvez également sélectionner **Supprimer** dans le menu contextuel.

6.1.3 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 « XML-Schema-aware » d'Altova, y compris StyleVision

- 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, Gestionnaire de schéma est disponible uniquement dans la ligne de commande. (Les applications serveur d'Altova sont disponibles sur Windows et Linux ; *voir la liste ci-dessous*.)



Application d'Altova qui fonctionnent avec Schema Manager

Applications desktop (Windows uniquement)	Applications serveur (Windows, Linux)
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 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.
- Dans un produit Altova, si vous essayez de valider un schéma qui n'est pas installé, mais qui est disponible via Gestionnaire de schéma, l'installation est automatiquement déclenchée. Toutefois, si le pack de schéma contient des mappages d'espace de noms, il n'y aura pas d'installation automatique ; dans ce cas, vous devez lancer Gestionnaire de schéma, sélectionner le/s pack/s que vous voulez installer et exécuter l'installation. Si, après l'installation, votre application Altova ouverte ne redémarre pas automatiquement, vous devez la redémarrer manuellement.

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/fr/schema-manager>), vous pouvez sélectionner un schéma et les 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`²¹², Gestionnaire de schéma installera les schémas que vous avez sélectionnés.

Cache local : suivre vos schémas

Toutes les informations sur les schémas installés sont suivies dans un répertoire cache centralisé sur votre ordinateur, situé ici :

<i>Windows</i>	C:\ProgramData\Altova\pkgs\cache
<i>Linux</i>	/var/opt/Altova/pkgs/cache

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 StyleVision pour la première fois dans un jour donné du calendrier.
- Si StyleVision 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`²¹⁵ 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`²¹⁴, et (ii) exécutez la commande `initialize`²¹². (En alternative, exécutez la commande `reset` avec l'option `--i.`)

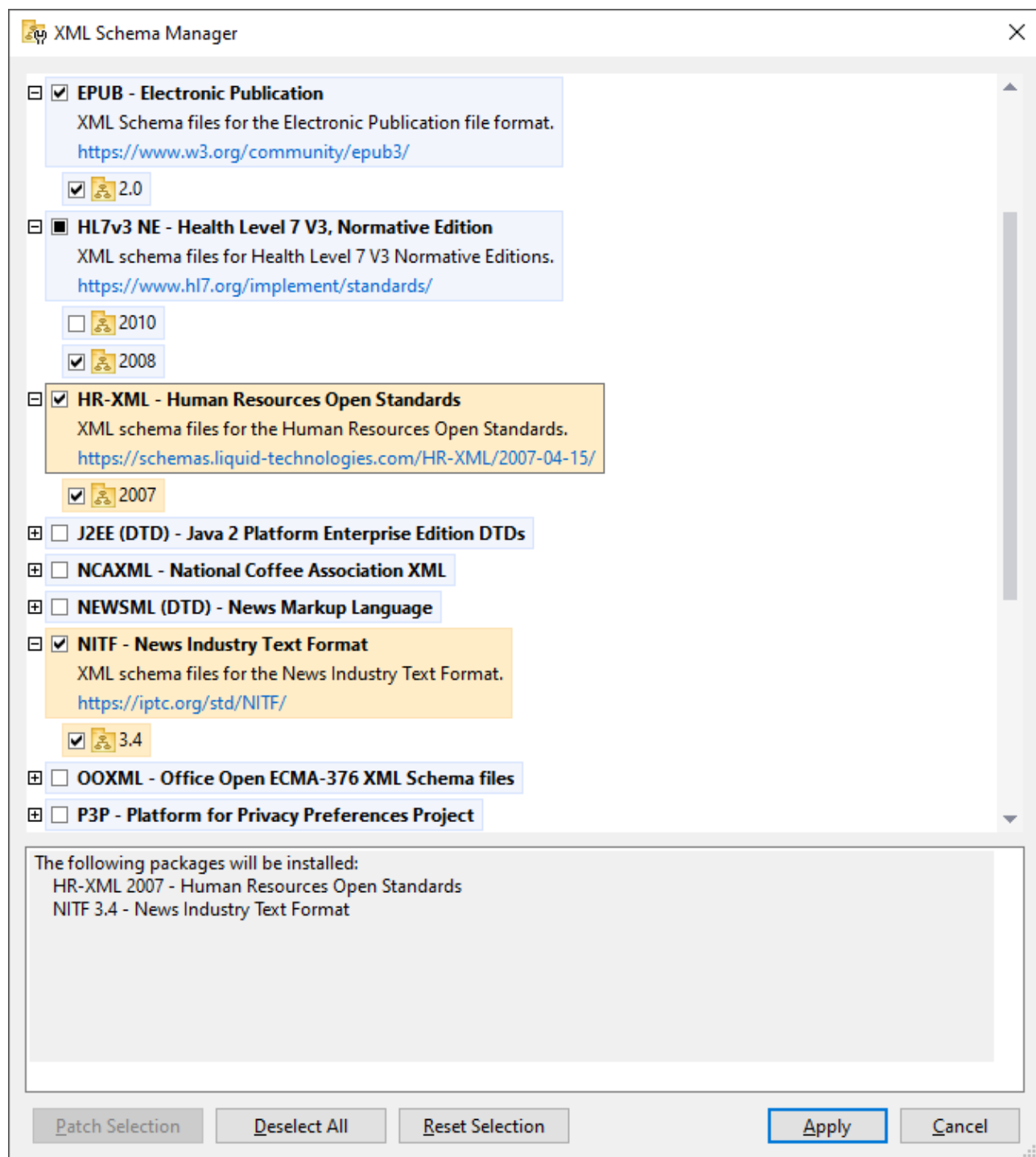
6.1.3.1 Exécuter Schema Manager

Interface utilisateur graphique

Vous pouvez accéder à la GUI de Gestionnaire de schéma des manières suivantes :

- *Durant l'installation de StyleVision:* 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 StyleVision:* 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*).



Lorsque vous cliquez sur **Appliquer**, la progression de l'installation s'affiche. Si une erreur survient (un problème de connexion, par exemple), un message d'erreur apparaît. Dans ce cas, cliquez sur le bouton **Avancé** dans le dialogue, vérifiez le schéma sélectionné, puis réessayez en cliquant sur **Appliquer**.

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 (applications 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](#) ²¹⁰.

Pour afficher l'aide pour la commande, exécutez l'étape suivante :

- *sur Windows* : `xmlschemamanager.exe --help`
- *sur Linux (applications serveur uniquement)* : `sudo ./ xmlschemamanager --help`

6.1.3.2 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**.



- 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`²¹³ est utilisée avec différentes options pour recenser les différentes catégories de schémas :

<code>xmlschemamanager.exe list</code>	Recense tous les schémas installés et disponibles ; ceux pouvant être mis à niveau sont également indiqués.
<code>xmlschemamanager.exe list -i</code>	Recense les schémas installés uniquement ; ceux pouvant être mis à niveau sont également indiqués
<code>xmlschemamanager.exe list -u</code>	Recense les schémas pouvant être mis à niveau

Note : Sur Linux, utiliser `sudo ./xmlschemamanager list`

6.1.3.3 Retoucher ou installer un schéma

Retoucher un schéma installé

Occasionnellement, des schémas XML peuvent recevoir des patches (mises à niveau ou révisions) de leurs émetteurs. Lorsque Gestionnaire de schéma détecte que des patches sont disponibles, ceux-ci sont indiqués dans les listes de schéma de Gestionnaire de schéma et vous pouvez installer les patches rapidement.

Dans la GUI

Les patches sont indiqués par l'icône . (Voir aussi la rubrique précédente sur les [catégories de statut](#)²⁰².) Si les patches sont disponibles, le bouton **Patch Selection** sera activé. Cliquez dessus pour sélectionner et préparer tous les patches 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 patches qui doivent être appliqués. Lorsque vous êtes prêt pour installer des patches 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`²¹³ . Cela liste tout schéma lorsque des mises à niveau sont disponibles.
2. Exécutez la commande `upgrade`²¹⁶ pour installer les patches.

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`²¹² :

```
xmlschemamanager.exe install [options] Schema+
```

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](#)²¹³.) Vous pouvez saisir autant de schémas que vous le souhaitez. Pour plus de détails, voir la description de la commande [install](#)²¹².

Note : sur Linux, utilisez la commande `sudo ./ xmlschemamanager`.

Installer un schéma requis

Lorsque vous exécutez une commande activée par XML dans StyleVision, et que StyleVision 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**.

6.1.3.4 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 le schéma que vous voulez désinstaller référence d'autres Schémas, les schémas référencés sont également désinstallés.

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](#)²¹⁴ :

```
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](#)²¹³.) Vous pouvez saisir autant de schémas que vous le souhaitez. Pour plus de détails, voir la description de la commande [désinstaller](#)²¹⁴.

Note : sur Linux, utilisez la commande `sudo ./ xmlschemamanager`.

Réinitialiser Gestionnaire de schéma

Sur le CLI

La commande [réinitialiser](#)²¹⁴ supprime tous les schémas installés et le répertoire de mise sous cache. Une fois avoir exécuté cette commande, vous devrez exécuter la commande [initialize](#)²¹², pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécutez la commande [reset](#)²¹⁴ avec l'option `-i`. Notez que [reset -i](#)²¹⁴ restaure l'installation originale du produit, il est recommandé d'exécuter la commande [update](#)²¹⁵ après avoir réalisé la réinitialisation. En alternative, exécutez la commande [reset](#)²¹⁴ avec les options `-i` and `-u`.

Dans la GUI

Le bouton **Réinitialiser sélection** réinitialise l'affichage dans la GUI pour afficher ce qui est actuellement installé. Les sélections ou désélections effectuées dans la session en cours ne sont pas affichées.

6.1.3.5 Options

Vous pouvez définir les options d'accès au réseau, de proxy réseau et de gestion des packs. Pour ouvrir le dialogue **Options**, cliquez sur le bouton Options en bas à gauche de l'interface graphique Gestionnaire de schéma. Accédez à la section souhaitée (*voir les descriptions ci-dessous*) et définissez vos options. Une fois terminé, cliquez sur **OK** pour appliquer les paramètres.

Réseau

La section **Réseau** (*voir la capture d'écran ci-dessous*) vous permet de configurer les paramètres de proxy de réseau.

Réseau

Adresse IP

☐ Préférer adresses IPv6

Délai d'expiration

☒ Délai d'expiration du transfert: 40 s

☐ Délai d'expiration de la phase de connexion: 300 s

Certificat

☒ Vérifier le certificat serveur

☒ Vérifier l'identité du serveur TLS/SSL

Adresses IP

Lorsque les noms d'hôte résolvent plus d'une adresse dans les réseaux mixtes IPv4/IPv6, sélectionner cette option fait que les adresses IPv6 sont utilisées. Si l'option n'est pas sélectionnée dans de tels environnements,

alors les adresses IPv4 sont utilisées.

Délai d'expiration

- *Délai d'expiration du transfert* : Si cette limite est atteinte pour deux packs de données consécutifs d'un transfert (envoyés ou reçus), alors le transfert entier est abandonné. Les valeurs peuvent être spécifiées en secondes [s] ou millisecondes [ms], avec le défaut étant 40 secondes. Si l'option n'est pas sélectionnée, alors il n'y a pas de limite de temps pour abandonner un transfert.
- *Délai d'expiration de la phase de connexion* : Il s'agit de la limite de temps au sein de laquelle la connexion doit être établie, y compris le temps pris pour les négociations de sécurité. Les valeurs peuvent être spécifiées en secondes [s] ou millisecondes [ms], avec le défaut étant 300 secondes. Ce délai d'expiration ne peut être désactivé :

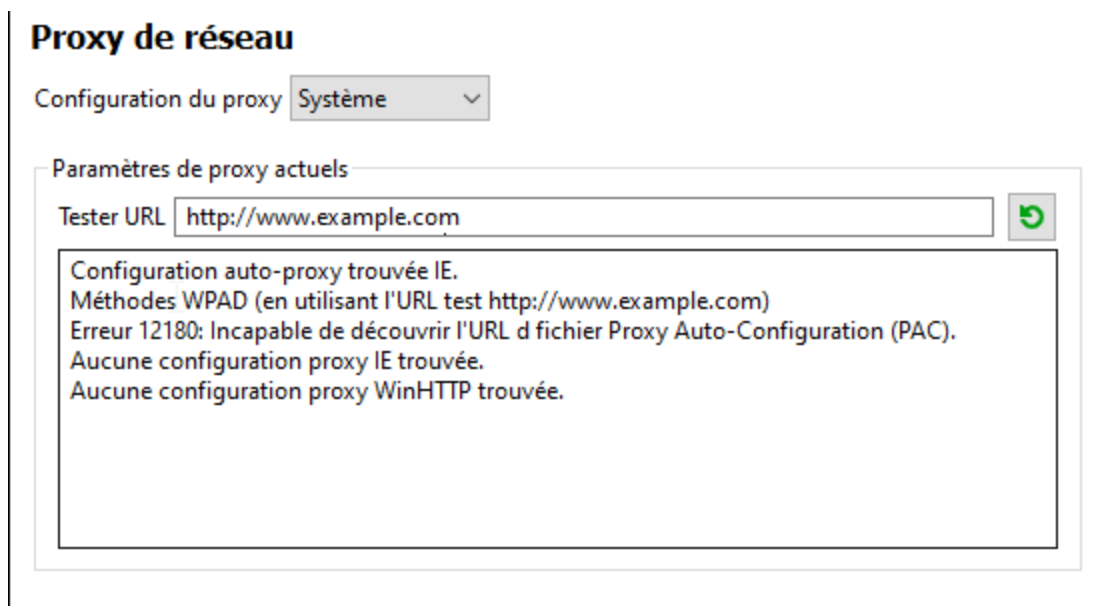
Certificat

- *Vérifier le certificat de serveur TLS/SSL* : Si sélectionné, alors l'authenticité du certificat de serveur est vérifiée en contrôlant la chaîne de signatures numériques jusqu'à ce qu'un certificat racine de confiance est atteint. Cette option est activée par défaut. Si cette option n'est pas sélectionnée, alors la communication n'est pas sécurisée, et les attaques (par exemple, une attaque « man-in-the-middle ») ne serait pas détectée. Notez que cette option ne vérifie pas que le certificat est en fait pour le serveur avec lequel il communique. Pour permettre une sécurité entière, les deux certificats et l'identité doivent être vérifiés (*voir la prochaine option*).
- *Vérifier l'identité de serveur TLS/SSL* : Si sélectionné, alors le certificat de serveur est vérifié pour appartenir au serveur avec lequel nous voulons communiquer. Ceci est fait en vérifiant que le nom de serveur dans l'URL est le même que le nom dans le certificat. Cette option est activée par défaut. Si cette option n'est pas sélectionnée, alors l'identité du serveur n'est pas vérifiée. Notez que cette option n'active pas la vérification du certificat du serveur. Pour permettre une sécurité entière, les deux certificats et l'identité doivent être vérifiés (*voir l'option précédente*).

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 schéma 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.

Proxy de réseau

Configuration du proxy Manuel

Proxy HTTP Port

☐ Utiliser ce serveur proxy pour tous les protocoles

Proxy SSL Port

Pas de Proxy pour

☐ Ne pas utiliser le serveur de proxy pour les adresses locales

Paramètres de proxy actuels

Tester URL 

(en utilisant l'URL test http://www.example.com)
Sans utiliser de proxy.

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 (.).

Note : Si un serveur proxy a été défini et que vous voulez déployer une de mappage sur [Altova FlowForce Server](#), vous devez sélectionner l'option *Ne pas utiliser le serveur de proxy pour les adresses locales*.

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*).

Gestion des packs

Ce paramètre vous permet de spécifier si les packs de schéma téléchargés doivent être stockés dans le dossier `ProgramData` (dans `C:\ProgramData\Altova\pkgs`) ou dans le dossier utilisateur `<USER-HOME>\Documents\Altova\pkgs`. Sélectionnez le dossier utilisateur si vous n'avez pas l'autorisation d'écrire dans le dossier `ProgramData`.

	Dossier de packs système	Dossier de packs utilisateur
Windows	<code>C:\ProgramData\Altova\pkgs</code>	<code>C:\Users\<user>\Documents\Altova\pkgs</code>
Linux	<code>/var/opt/Altova/pkgs</code>	<code>/home/<user>/Altova/pkgs</code>

Notez que rien n'est partagé entre le dossier `ProgramData` et le dossier utilisateur. Par conséquent, si vous décidez de passer d'un dossier à l'autre et que vous avez besoin des mêmes packs dans le second dossier, vous devez réinstaller ces packs.

6.1.3.6 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, une fois que vous avez changé de répertoire pour accéder à celui contenant l'exécutable, vous pouvez appeler ce dernier 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.

6.1.3.6.1 help

Cette commande propose une aide contextuelle pour les commandes liées à l'exécutable Gestionnaire de schéma.

Syntaxe

```
<exec> help [command]
```

Où [command] est un argument optionnel qui spécifie un nom de commande valide.

Veillez 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
```

6.1.3.6.2 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](#)²¹³.)
- 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
```

6.1.3.6.3 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 :

<code>--silent, --s</code>	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux .
<code>--verbose, --v</code>	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux .
<code>--help, --h</code>	Afficher l'aide pour la commande.

Exemple

La commande suivante initialise Gestionnaire de schéma:

```
xmlschemamanager initialize
```

6.1.3.6.4 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](#)²¹³. 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 ?](#)¹⁹⁶.

Options

La commande **install** accepte les options suivantes :

<code>--silent, --s</code>	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux .
<code>--verbose, --v</code>	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux .
<code>--help, --h</code>	Afficher l'aide pour la commande.

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
```

6.1.3.6.5 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 :

<code>--installed, --i</code>	Recenser uniquement la liste des schémas installés. Le réglage par défaut est faux .
<code>--upgradeable, --u</code>	Recenser uniquement les schémas lorsque des mises à niveau (patches) sont disponibles. Le réglage par défaut est faux .
<code>--help, --h</code>	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` :

6.1.3.6.6 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](#)²¹², pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécuter la commande `reset` avec l'option `-i`. Puisque `reset -i` restaure l'installation originale du produit, nous vous recommandons que vous exécutiez la commande [update](#)²¹⁵ 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 :

<code>--init, --i</code>	Initialiser Gestionnaire de schéma après le reset. Le réglage par défaut est faux .
<code>--update, --u</code>	Mettre à jour la liste de schémas disponibles dans le cache. Le réglage par défaut est faux .
<code>--silent, --s</code>	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux .
<code>--verbose, --v</code>	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux .
<code>--help, --h</code>	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`

6.1.3.6.7 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.

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` ²¹³. 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 ?](#) ¹⁹⁶.

Options

La commande **désinstaller** accepte les options suivantes :

<code>--keep-references, --k</code>	Définir cette option pour garder les schémas référencés. Le réglage par défaut est faux .
<code>--silent, --s</code>	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux .
<code>--verbose, --v</code>	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux .
<code>--help, --h</code>	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
```

6.1.3.6.8 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` ²¹⁴ et `initialize` ²¹².

Syntaxe

```
<exec> update [options]
```

Options

La commande **mise à jour** accepte les options suivantes :

<code>--silent, --s</code>	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux .
<code>--verbose, --v</code>	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux .
<code>--help, --h</code>	Afficher l'aide pour la commande.

Exemple

La commande suivante met à jour le cache local avec la liste des derniers schémas :

```
xmlschemamanager update
```

6.1.3.6.9 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](#) ²¹³.

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 :

<code>--silent, --s</code>	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux .
<code>--verbose, --v</code>	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux .
<code>--help, --h</code>	Afficher l'aide pour la commande.

6.2 Fusionner des données XML depuis des sources multiples

Les données XML depuis de multiples fichiers source XML peuvent être fusionnées quand XSLT 2.0 ou 3.0 est utilisé comme version XSLT du fichier SPS.

Généralement, fusionner des données sera basé sur un morceau de données commun, tel qu'une ID. Par exemple, un employé d'une entreprise, qui est identifié par un numéro d'ID personnel, peut avoir ses données personnelles stockées dans de multiples fichiers XML gérés par le département du personnel : (i) détails personnels, (ii) liste de paie, (iii) travail et congé, (iv) cours suivis, etc. Les données de ces fichiers différents peuvent être fusionnées en un seul document de sortie utilisant le numéro ID personnel comme clé.

Note : l'édition Enterprise vous permet d'inclure des sources de schéma multiples, donc les nœuds XML depuis d'autres schémas peuvent être sélectionnés utilisant le nom de paramètre pour le schéma correspondant (comme c'est le cas dans l'exemple ci-dessous). Dans les éditions Professional et Basic, la fonction `doc()` de XPath 2.0 peut être utilisée pour trouver le fichier XML requis et le nœud XML à l'intérieur de ce fichier. La fonction `doc()` de XPath 2.0 fournit l'accès à la racine du document des documents XML externes, permettant donc d'insérer du contenu de nœud des documents XML externes à insérer dans la sortie. Un [calcul automatique](#) ²⁶¹ qui utilise la fonction `doc()` peut, pour cette raison, aussi être utilisé afin de fusionner les données XML (*voir exemple ci-dessous*).

Exemple

Le [dossier \(Mes\) Documents](#) ²⁴, C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples, contient un fichier d'exemple SPS (`MergeData_2_Files.sps`) qui affiche comment les données de différents fichiers source XML peuvent être fusionnées. Le fichier SPS sélectionne les données selon un ordre (`MergeOrder.xml`, *recensé ci-dessous*) placé par un client fictif.

```
<?xml version="1.0" encoding="UTF-8"?>
<Order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeOrder.xsd">
  <Item partNum="238-KK" quantity="3" shipDate="2000-01-07" comment="With no inclusions,
please."/>
  <Item partNum="748-OT" quantity="1" shipDate="2000-02-14" comment="Valentine's day
packaging."/>
  <Item partNum="229-OB" quantity="1" shipDate="1999-12-05"/>
  <Item partNum="833-AA" quantity="2" shipDate="1999-12-05" comment="Need this for the
holidays!"/>
</Order>
```

La valeur de l'attribut `/Order/Item/@partNum` dans ce fichier (*voir ci-dessus*) est utilisée pour sélectionner les produits commandés du catalogue d'articles stocké dans un autre fichier, `MergeArticles.xml` (*voir la liste ci-dessous*).

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeArticles.xsd">
  <Article PartNum="833-AA">
    <ProductName>Lapis necklace</ProductName>
    <Price>99.95</Price>
  </Article>
```

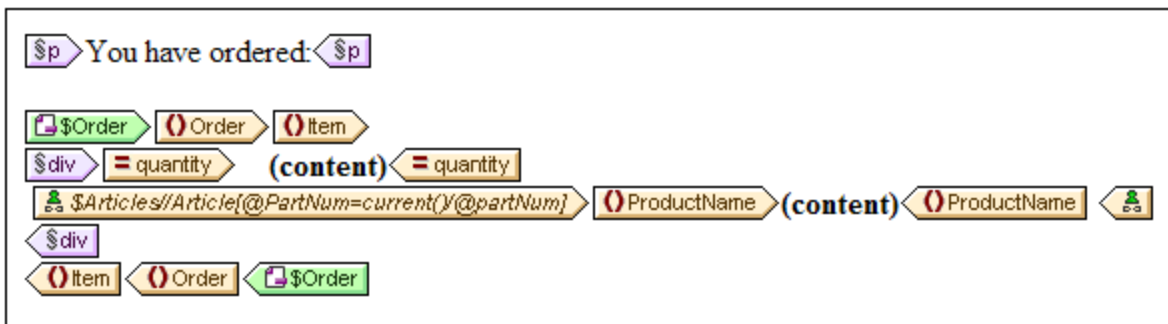
```

<Article PartNum="748-OT">
  <ProductName>Diamond heart</ProductName>
  <Price>248.90</Price>
</Article>
<Article PartNum="783-KL">
  <ProductName>Uncut diamond</ProductName>
  <Price>79.90</Price>
</Article>
<Article PartNum="238-KK">
  <ProductName>Amber ring</ProductName>
  <Price>89.90</Price>
</Article>
<Article PartNum="229-OB">
  <ProductName>Pearl necklace</ProductName>
  <Price>4879.00</Price>
</Article>
<Article PartNum="128-UL">
  <ProductName>Jade earring</ProductName>
  <Price>179.90</Price>
</Article>
...
</Article>

```

La manière dont les données sont fusionnées est la suivante : un [modèle défini par l'utilisateur](#) ²³⁹ au sein du modèle `/Order/Item` (voir la *capture d'écran ci-dessous*) qui sélectionne l'élément correspondant `Article` dans le fichier `MergeArticles.xml` en utilisant le numéro de partie de l'item commandé afin d'identifier l'article. L'expression XPath (qui est dans le contexte `/Order/Item`) est :

```
$Articles//Article[@PartNum=current()/@partNum]
```



Ce modèle produira une sortie comme celle affichée dans la capture d'écran ci-dessous.

```

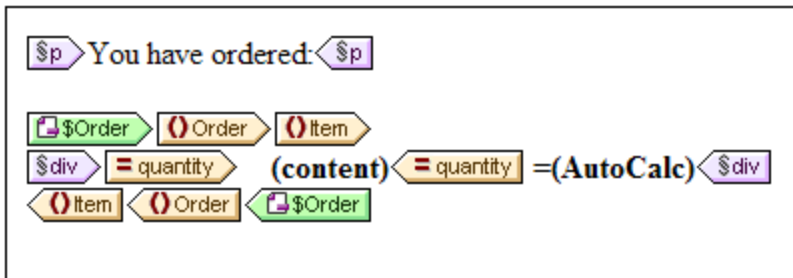
You have ordered:

3 Amber ring
1 Diamond heart
1 Pearl necklace
2 Lapis necklace

```

Notez que tandis que la quantité commandée de chaque item est prise du fichier `MergeOrder.xml`, le nom de l'article commandé est pris du fichier `MergeArticles.xml`. Notez également comment le nœud `ProductName` est sélectionné dans le contexte du modèle `/Articles/Article`.

Le même résultat tel que celui obtenu ci-dessus peut être atteint également en utilisant le [calcul automatique](#) ²⁶¹ (voir la capture d'écran ci-dessous). Glissez l'attribut `quantity` de la fenêtre de l'arborescence de schéma et créez-le comme son contenu. Ajoutez ensuite un calcul automatique tel qu'affiché dans la capture d'écran et attribuez une expression XPath au calcul automatique tel que décrit ci-dessous.



L'expression XPath du calcul automatique pourrait cibler le nœud requis utilisant soit le paramètre d'une autre source de schéma ou de la fonction `doc()` :

```
$Articles//Article[@PartNum=current()/@partNum]/ProductName
```

ou

```
doc('MergeArticles.xml')//Article[@PartNum=current()/@partNum]/ProductName
```

Notez que, tandis que la première expression XPath utilise un paramètre pour faire référence à un autre schéma XML (une fonction disponible uniquement dans l'édition Enterprise), la deuxième expression utilise la fonction `doc()` de XPath 2.0 (une fonction disponible dans les éditions Professional et Basic également).

6.3 SPS modulaires

Les modèles globaux d'un fichier SPS, ainsi que des fragments de design, des fonctions JavaScript et des items de mise en page de pages peuvent être utilisés dans le design d'un autre SPS. Ceci permet :

1. De réutiliser les modèles globaux et autres composants dans les multiples SPS, dont les principaux avantages sont l'édition de la seule source et la cohérence de la sortie.
2. Les SPS à modulariser, et donc la structuration plus flexible.

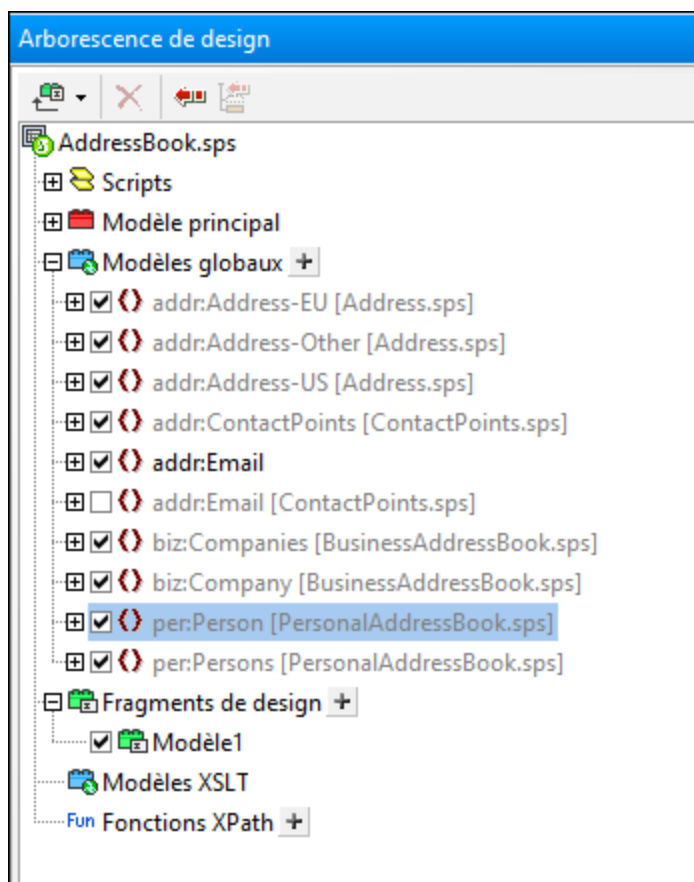
Dans tout SPS donné, un ou plusieurs SPS peuvent être ajouté comme module. Quelques types de composants (ou objets) dans ces modules sont ensuite disponibles dans le fichier SPS à importer (ou de référence).

Objets modulaires disponibles

La section, [Objets modulaires disponibles](#) ²²¹, ne décrit pas uniquement l'étendue vers laquelle et les conditions sous lesquelles les différents composants d'un fichier SPS sont disponibles pour un SPS à importer. Elle recense également ces composants qui ne sont pas disponibles dans le SPS à importer. Vous devriez noter que si un module ajouté contient lui-même des modules, alors ceux-ci sont ajoutés de manière récursive au fichier SPS de référence. De cette manière, une modularisation peut être élargie à différents niveaux et dans une vaste structure de design.

Créer un fichier SPS modulaire

Pour construire un SPS modularisé, [ajoutez d'abord le fichier SPS requis](#) ²²⁴ à l'SPS principal comme module. Toutes les fonctions JavaScript, modèles globaux, fragments de design, et fonctions XPath dans le module ajouté sont disponibles pour le SPS de référence. Chacun des objets disponibles est recensé dans l'arborescence de design, sous son en-tête respectif (*capture d'écran ci-dessous*), et peut être activé ou désactivé, respectivement, en cochant ou décochant sa case.



Ces objets peuvent ensuite être réutilisés dans un SPS de référence conformément à leurs mécanismes d'inclusion respectifs. Les modèles globaux ne doivent généralement être uniquement activés pour être activés dans l'SPS de référence. Les fragments de design doivent être glissés de l'arborescence de design à l'emplacement requis. Les fonctions de JavaScript sont assignées via la fenêtre de propriétés comme gestionnaires d'événement pour les composants de design sélectionnés. Et les fonctions XPath disponibles (activées) peuvent être utilisées dans les expressions XPath.

Comment créer et travailler avec un SPS modulaire est décrit dans la section, [Créer un SPS modulaire](#)²²⁴.

Terminologie

Quand un SPS est utilisé avec un autre module, on dit qu'il est ajouté au dernier, et nous appelons le processus **adding**. Les deux SPS font respectivement référence au module **added SPS** et au module **référant SPS**. Quand un module SPS est ajouté, ses **objets** sont ajoutés au module SPS de référence. Ces objets sont appelés **objets de module**, et sont de types suivants : modèles globaux ; fragments de design ; fonctions JavaScript et items de mise en page des pages.

6.3.1 Objets modulaires disponibles

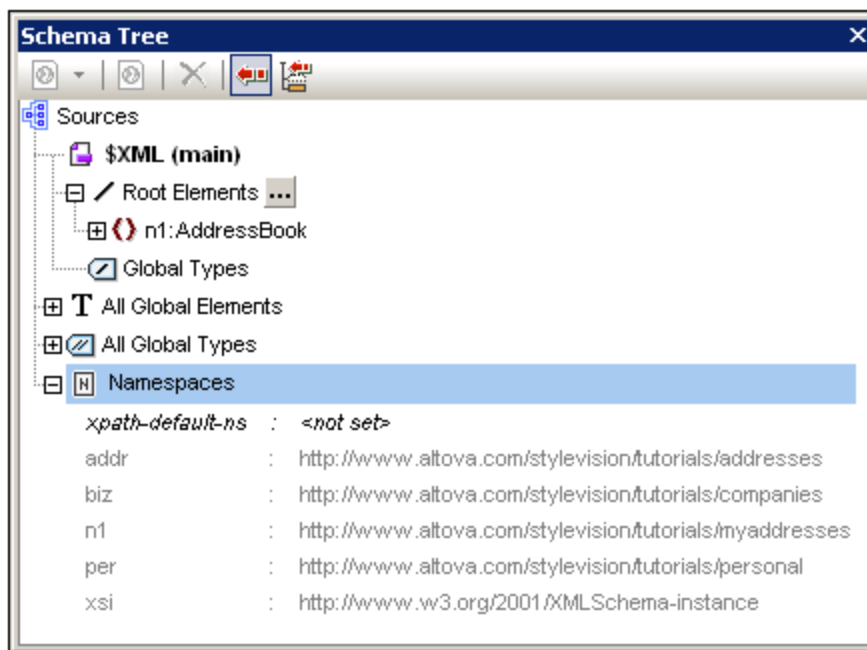
Cette section recense les objets dans les [modules SPS ajoutés](#)²²¹ qui sont disponibles dans le [module SPS de référence](#)²²¹. La liste explique de quelle manière chaque objet est disponible dans le module SPS de référence et comment il peut être utilisé. Pour une approche étape par étape en ce qui concerne la création

d'un SPS modulaire, voir la prochaine section, [Créer un fichier SPS modulaire](#)²²⁴. La section se termine par une liste d'objets dans le fichier SPS ajouté qui n'est pas disponible dans le module SPS de référence ; ceci vous aidera à mieux comprendre comment le travail SPS modulaire fonctionne.

- [Déclarations d'espace de nom](#)²²²
- [Modèles globaux](#)²²²
- [Fragments de design](#)²²³
- [Modules ajoutés](#)²²⁴
- [Scripts](#)²²⁴
- [styles CSS](#)²²⁴
- [Mises en page de pages](#)²²⁴
- [Objets de module indisponibles](#)²²⁴

Déclarations d'espace de nom

Chaque SPS stocke une liste d'URI d'espaces de noms et leurs préfixes. Quand une SPS est ajoutée comme module, les espaces de noms intégrés sont comparés aux espaces de noms dans la/les source/s de schéma du fichier SPS de référence. Si un URI d'espace de noms dans le fichier SPS ajouté correspond à l'URI d'espace de noms dans la/les source/s de schéma du fichier SPS, alors le préfixe utilisé dans la source de schéma de l'SPS de référence est adopté comme préfixe pour cet espace de noms dans l'SPS ajouté. Si un URI d'espace de noms dans l'SPS ajouté ne peut pas correspondre à tout URI d'espace de noms dans la/les source/s de schéma de l'SPS de référence, alors le message d'erreur indiquant ceci est affiché.

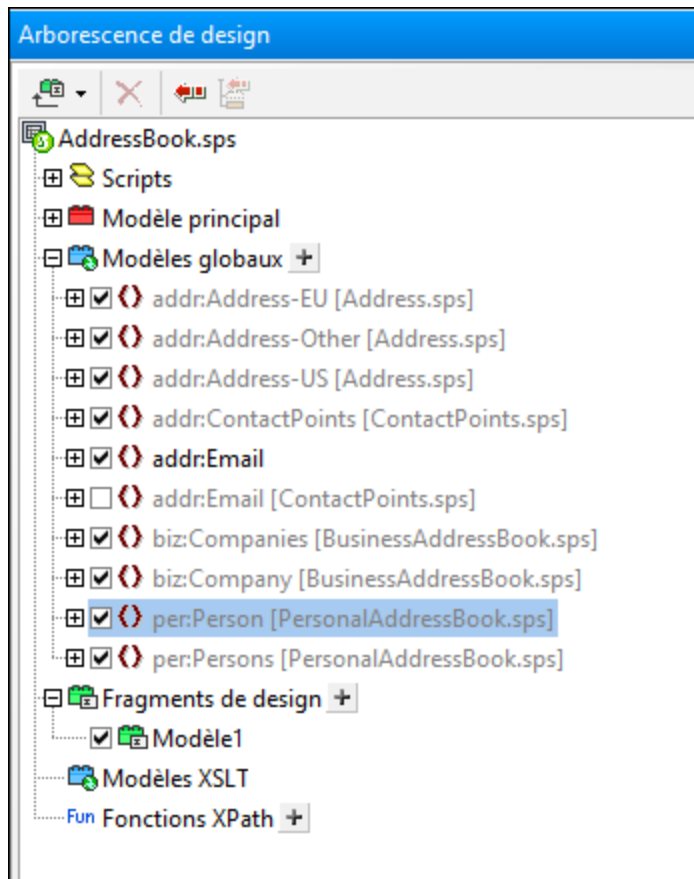


La capture d'écran ci-dessus affiche les différents espaces de noms dans SPS, ensemble avec leurs préfixes, dans la barre latérale de l'arborescence de schéma. Ces espaces de noms viennent du/des schéma/s de source et ne peuvent pas être édités.

Modèles globaux

Les [modèles globaux](#)²³⁴ du module SPS ajouté sont disponibles dans le module SPS de référence et affichés dans la [barre latérale de l'arborescence de design](#)⁴⁰ (capture d'écran ci-dessous). Ils sont activés ou

désactivés (cochés ou décochés) par défaut, selon le statut d#activation respectif dans le module ajouté. Si vous désirez créer un modèle global pour écraser un modèle global depuis le module ajouté, créez le nouveau modèle global en cliquant sur l'icône  à côté de l'entrée des Modèles globaux. Dans Ajouter Nouveau Modèle global qui apparaît, sélectionnez un élément ou attribut pour lequel vous souhaitez créer un modèle global. En alternative, saisissez une expression XPath qui sélectionne le nœud requis dans le schéma. En cliquant sur **OK**, on vous demandera si le nouveau modèle global créé devrait être activé en préférence au modèle global dans le module ajouté. La réponse que vous sélectionnez active soit le nouveau modèle global créé, soit le modèle global dans le module ajouté. Vous pouvez changer votre sélection à tout moment en vérifiant l'autre des deux modèles globaux.



Notez que le modèle principal des modules ajoutés ne sont pas disponibles. Ceci signifie que si vous planifiez de réutiliser un modèle par le biais d'une approche modulaire, vous devez le créer comme modèle global. Si aucun modèle global n'est défini pour un élément particulier et que le traitement est invoqué pour cet élément, alors le traitement par défaut pour cet élément (modèles XSLT intégrés) sera utilisé.

Fragments de design

Les [modèles globaux](#) ²⁴⁵ du module SPS ajouté sont disponibles dans le module SPS de référence et affichés dans la [barre latérale de l'arborescence de design](#) ⁴⁰ (capture d'écran ci-dessous). Lorsque vous insérez un fragment de design dans le design, faites bien attention de placer le fragment de design à l'intérieur du nœud contextuel correct dans le design.

Modules ajoutés

Chaque module SPS ajouté rend également ses propres modules ajoutés disponibles dans le SPS de référence et leurs modules ajoutés, etc. De cette manière, ajouter un module de manière récursive rend tous les modules disponibles qui y ont été ajoutés, à plusieurs niveaux vers le bas. Il va sans dire que ces modules doivent construire ensemble un modèle de contenu valide conformément au/x schéma/s de source du module SPS de référence. Les modules sont affichés et peuvent être gérés dans la [barre latérale de l'Aperçu du design](#) ³⁵.

Scripts

Les modèles globaux du module SPS ajouté sont disponibles dans le module SPS de référence et affichés dans la [barre latérale de l'arborescence de design](#) ³⁸⁹ (capture d'écran ci-dessous). En effet, les scripts de tous les modules ajoutés sont recueillis dans une bibliothèque qui se trouve maintenant —dans le SPS— de référence pour la sélection dans le dialogue des Propriétés.

Styles CSS

Les styles globaux présents dans les modules SPS ajoutés sont différés dans le SPS de référence comme styles globaux et les règles de style sont affichées dans la [barre latérale du Référentiel de style](#) ³⁴⁴. Les fichiers CSS sont aussi recensés dans la [barre latérale de l'Aperçu du design](#) ³⁵. De manière semblable, des fichiers CSS externes qui ont été disponibles dans le module SPS ajouté, sont disponibles dans le module SPS de référence.

Mises en page de pages

Les mises en page de pages d'un module ajouté sont disponibles dans le module SPS de référence et affichés dans la [barre latérale de l'arborescence de design](#) ⁴⁰.

Les objets de module qui ne sont pas disponibles dans le SPS de référence

Les objets suivants du module ajouté ne sont pas disponibles dans le SPS de référence :

- **Définitions de paramètres** : sont ignorées.
- **Sources de schéma** : La source de schéma sur laquelle le SPS est basé est ignorée. N'oubliez pas que le modèle de contenu de l'élément du document du SPS ajouté doit se trouver à l'intérieur du modèle de contenu du SPS de référence ; autrement, il ne serait pas possible d'utiliser correctement le SPS ajouté comme module. Si vous le souhaitez, vous pouvez toujours ajouter un schéma défini par l'utilisateur dans le SPS de référence. Le schéma supplémentaire pourrait accommoder le modèle de contenu du/des modèle/s global/-aux ajouté/s.
- **Le fichier de travail XML et le fichier modèle XML** : Les références à ces fichiers sont ignorées. Le SPS de référence utilise ses propres fichiers de travail XML et de modèle XML.
- **Espaces de noms XPath par défaut** : S'ils ont été définis sur un module qui est importé, alors ils ne seront pas transportés jusqu'au SPS importé.

6.3.2 Créer un fichier SPS modulaire

Créer un fichier SPS modulaire consiste en quatre grandes parties :

1. Concevoir et enregistrer le [module SPS à ajouter](#) ²²⁵.
2. [Ajouter le module](#) ²²⁵ à l'SPS dans lequel il doit être utilisé (c'est-à-dire, au module SPS de référence).
3. [Activer ou désactiver l'/les objet/s ajouté/s](#) ²²⁷ tel que requis.
4. Appliquer l'objet requis, là où requis.

Le module SPS à ajouter

Il existe deux points dont il faut se rappeler en créant un fichier SPS qui sera ajouté à un autre :

1. Les modèles qui peuvent être utilisés dans le [module SPS de référence](#) ²²¹ peuvent uniquement être des [modèles globaux](#) ²³⁴. Ceci signifie que les modèles que vous voulez réutiliser doivent être créés comme modèles globaux dans le [module SPS qui doit être ajouté](#) ²²¹.
2. La structure du document définie dans le module SPS à ajouter doit être valide à l'intérieur du modèle de contenu défini par le/les [schéma/s de source de l'SPS de référence](#) ²²⁴. Si un modèle ajouté n'est pas contenu dans le modèle de contenu défini par le schéma principal de l'SPS, son modèle de contenu, peut toutefois être défini dans un schéma défini par l'utilisateur.

En créant le module SPS à ajouter, le schéma sur lequel vous basez le SPS pourrait être le suivant :

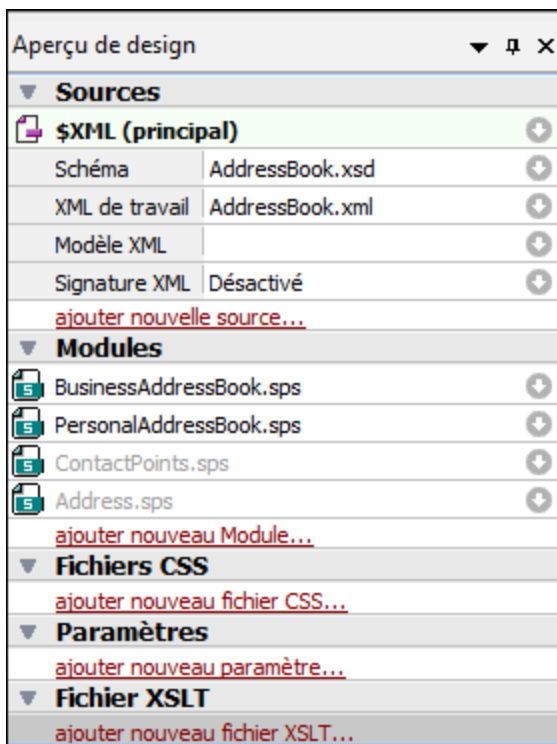
- Le schéma de source principal du SPS de référence. Dans ce cas, quand le SPS est ajouté, les modèles globaux ajoutés feront partie du modèle de contenu du schéma principal du SPS principal. La sortie de ces modèles globaux dans Authentic View est donc éditable.
- Un schéma qui définit un modèle de contenu qui fait partie du modèle de contenu défini par le schéma principal du SPS de référence. Dans ce cas, lorsque des modèles globaux sont ajoutés, ils s'inscriront dans un modèle de contenu du schéma principal du SPS de référence. La sortie de ces modèles globaux dans Authentic View est donc éditable.
- Un schéma qui définit un modèle de contenu qui **ne fait pas** partie du modèle de contenu défini par le schéma principal du SPS. Quand ce module SPS est ajouté, ses modèles globaux ne feront pas partie du modèle de contenu du schéma principal de l'SPS de référence. Ils peuvent, toutefois, être utilisés pour produire une sortie si un schéma défini par l'utilisateur est utilisé qui définit un modèle de contenu du/des modèle/s global/-aux. Dans Authentic View, toutefois, la sortie de ces modèles globaux ne peut pas être éditée.

Lorsque vous définissez les modèles de contenu dans vos schémas, vous devriez prêter une attention particulière aux [espaces de noms](#) ²²² utilisés étant donné que ceux-ci déterminent les noms élargis des nœuds.

Vous pourriez utiliser le [fichier de travail XML](#) ²³ pour tester la sortie du module SPS à ajouter. La référence de ce fichier de travail XML sera [ignorée par le SPS de référence](#) ²²⁴.

Ajouter le module SPS

Pour ajouter un module à un SPS, dans l'[Aperçu de design](#) ³⁵ (*capture d'écran ci-dessous*), cliquez sur une commande Ajouter Nouveau module, recherchez le fichier SPS requis dans le dialogue qui apparaît, sélectionnez-le et cliquez sur **Ouvrir**.



Le module est ajouté dans SPS et est recensé sous l'en-tête des Modules dans l'Aperçu du design. Dans la capture d'écran ci-dessus, les modules `BusinessAddressBook.sps` et `PersonalAddressBook.sps` ont été ajoutés au module `AddressBook.sps` (le SPS actif). Tous les objets de module ajoutés sont répertoriés dans la barre latérale de l'arborescence de design ; les fichiers CSS ajoutés sont également répertoriés dans l'Aperçu de design. Si les modules ajoutés font eux-mêmes référence à des modules, ces derniers, indirectement importés, sont répertoriés sous l'en-tête Modules, mais en gris. Des informations sur les modules qui importent un module indirectement importé sont disponibles dans une fenêtre contextuelle qui s'affiche lorsque vous passez la souris sur le module indirectement importé.

Pour ouvrir un des modules ajoutés ou des modules indirectement importés rapidement dans StyleVision, cliquez avec la touche de droite sur ce module et sélectionnez **Ouvrir module de définition** depuis le menu contextuel qui apparaît.

Ordre de modules ajoutés

L'ordre dans lequel les modules sont ajoutés et recensés est significatif pour les styles CSS de priorisation.. Conformément à l'ordre de cascade CSS, les règles de style CSS d'un module relativement récent (plus bas dans la liste) ont la priorité sur les règles de style définies dans un module relativement antérieur (plus haut dans la liste). Les styles CSS dans le module SPS de référence ont la priorité sur ceux dans tout module ajouté. Pour changer la position relative à un module ajouté, cliquez avec la touche de droite dans l'Aperçu du design et cliquez, tel que requis, sur la commande **Move Up** ou **Move Down** dans le menu contextuel.

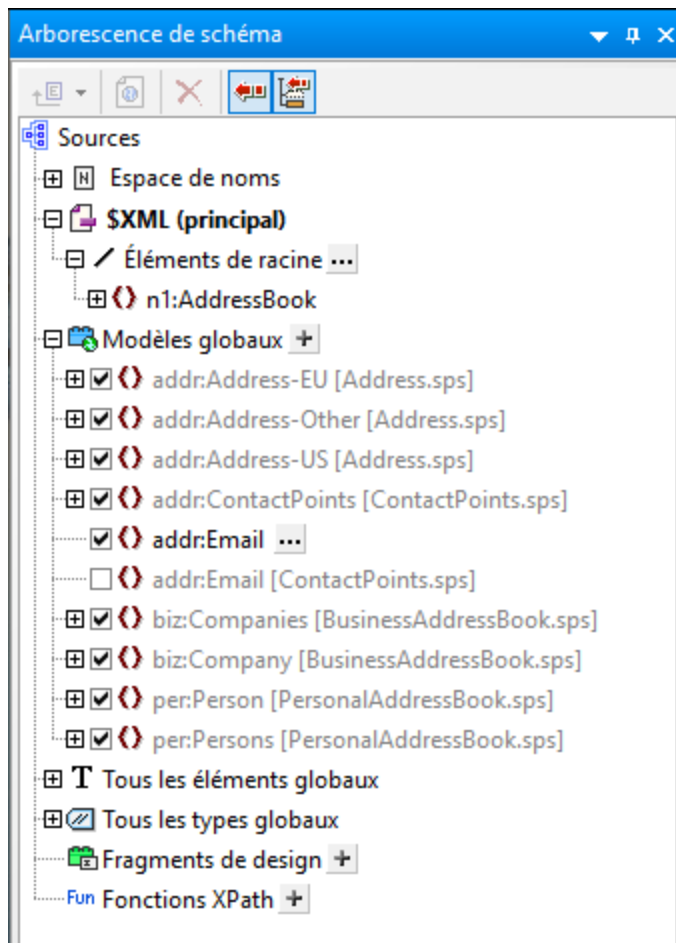
L'ordre du module n'est pas significatif pour résoudre les conflits parmi les scripts, les modèles globaux, les fragments de design et les items de mise en page de pages.

Alertes de modification du fichier

Si un fichier ajouté (que ce soit un module SPS, schéma ou fichier de travail XML) est modifié après avoir ouvert un module SPS de référence, alors une fenêtre pop up de modification du fichier vous avertira du changement et vous demandera si le module SPS de référence doit être actualisé avec ces modifications.

Activer/Désactiver l'objet ajouté

Tous les objets de module dans tous les modules ajoutés (qu'ils soient ajoutés directement ou indirectement) sont ajoutés au SPS de référence et recensés sous les en-têtes correspondants dans l'arborescence de design : Scripts ; Modèles globaux ; fragments de design ; modèles XSLT ; et fonctions XPath. À côté de chacun de ces objets, il se trouve une case à cocher (voir la capture d'écran ci-dessous), que vous pouvez cocher ou décocher, respectivement, activer ou désactiver cet objet. Quand un objet est désactivé, il est supprimé efficacement du SPS.



Dans la capture d'écran ci-dessus, tous les modèles globaux utilisés dans le module `AddressBook.sps` sont répertoriés sous l'en-tête Modèles globaux. Ceux qui ont été ajoutés via d'autres modules (que ce soit directement ou indirectement) sont affichés en gris. Ceux qui ont été créés directement dans `AddressBook.sps` sont affichés en noir. La capture d'écran affiche que seul un modèle global, `addr:Email`, a été créé dans `AddressBook.sps` lui-même. Tous les autres modèles globaux ont été ajoutés via d'autres modules, et le fichier dans lequel chacun de ces modules est défini est recensé à côté de son nom.

Notez qu'il existe deux modèles globaux pour `addr:Email`, l'un créé dans le SPS de référence (`AddressBook.sps`) lui-même, et l'autre créé dans le module ajouté `ContactPoints.sps`. Si plus d'un modèle global a le même nom élargi (espace de nom-), alors seul un de ces modèles sera actif à la fois. Vous pouvez sélectionner lequel en cochant sa case à cocher. (En alternative, vous activez le modèle global depuis son menu contextuel dans l'Aperçu de design.) Ce mécanisme est utile si : (i) vous souhaitez écraser un modèle global ajouté avec un que vous créez dans le module SPS de référence, ou (ii) souhaitez résoudre une situation où un modèle global pour un élément est défini dans plus d'un module ajouté.

Un modèle global qui a été défini dans le SPS actuel peut être supprimé en le sélectionnant et en cliquant sur le bouton **Supprimer**. Toutefois, des modèles globaux qui ont été définis dans un module ajouté ne peuvent pas être supprimés du SPS de référence. Ils doivent être supprimés en ouvrant le SPS ajouté et en supprimant le modèle global ici.

Des scripts individuels, des fragments de design et des items de mise en page peuvent être activés ou désactivés de la même manière.

Appliquer ou utiliser des objets modulaires

Dans le [module SPS de référence](#) ²²¹, vous concevez vos designs comme d'habitude. Chaque type différent d'objet ajouté est utilisé ou appliqué différemment. Vous devriez, bien évidemment, vous assurer que chaque objet de module que vous souhaitez appliquer a [été activé](#) ²²⁷.

Modèles globaux

Lorsque vous souhaitez utiliser un [modèle global](#) ²³⁴ d'un des modules SPS ajoutés, vous devez vous assurer que ce modèle global est réellement appliqué. Cela peut se faire de deux manières, selon celle qui convient le mieux à votre design :

- Dans le modèle principal, spécifiez que le modèle d'élément utilise le modèle global pour cet élément ou copie ce modèle global localement. Ces deux commandes sont disponibles dans le menu contextuel qui apparaît lorsque vous cliquez avec le bouton droit de la souris sur la balise de l'élément dans le design.
- Dans le modèle principal, les espaces réservés du contenu ou du reste du contenu entraînent l'application de modèles, ce qui conduit au traitement des modèles globaux correspondants.

Fragments de design

Pour utiliser un fragment de design, faites-le glisser de l'arborescence de design vers l'emplacement souhaité dans le modèle principal ou un modèle global. Assurez-vous que l'emplacement du fragment de design est déposé dans le nœud correct du contexte pour ce fragment de design. Pour les détails, voir les [fragments de design](#) ²⁴⁵.

Scripts

Toutes les fonctions JavaScript (qu'elles soient dans un module ajouté ou créé dans le SPS de référence) sont disponibles comme gestionnaires d'événement, et peuvent être [définies pour un événement particulier via la barre latérale des propriétés](#) ³⁹¹.

6.3.3 Exemple : Un Carnet d'adresse

Le [dossier \(Mes\) Documents](#) ²⁴, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\ModularSPS`, contient des

exemples de SPS modulaires. Ces exemples de fichiers dans ce dossier comprennent un projet dans lequel un carnet d'adresse contient des contacts professionnels et personnels est modularisé. Cet exemple ne démontre non seulement les mécanismes dans lequel la modularisation est mise en œuvre, mais illustre également les principales raisons pour lesquelles un serait modularisé.

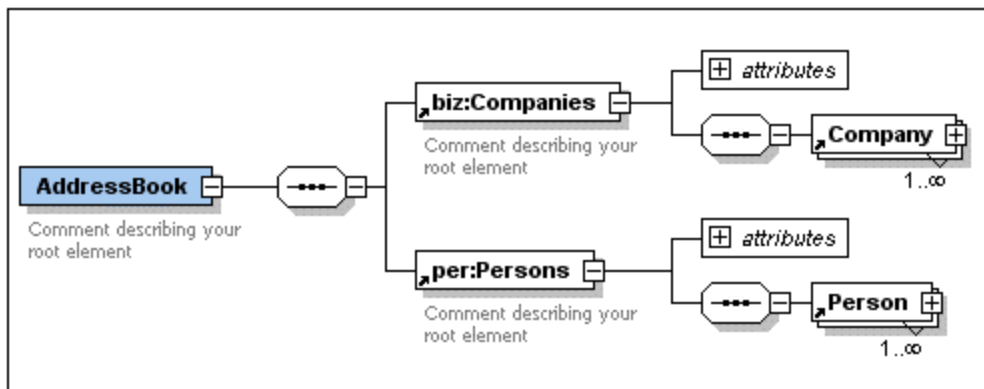
- Tout le carnet d'adresse est composé de deux modules : (i) un carnet d'adresse professionnel, et (ii) un carnet d'adresse personnel, dont chacun a un SPS séparé définissant différents designs. Les deux modules ensemble constituent le carnet d'adresse composite. La modularisation dans ce cas est utilisé pour composer : les modules sont les composants de l'unité plus élargie.
- Bien que le modèle de contenu de chaque module (carnets d'adresses professionnels et personnels) diffère légèrement de l'autre, les deux ont un module commun, qui est le module ContactPoints, consistant en les détails de contact : adresse, téléphone, fax et e-mail. Le module ContactPoints peut, pour cette raison, être partagée entre les deux carnets d'adresse (professionnel et personnel). La modularisation dans ce cas permet à un seul module d'être utilisé comme unité commune à l'intérieur des autres multiples unités.
- De plus, le module ContactPoints peut être modularisé pour fournir plus de flexibilité. Dans cet exemple de projet, nous avons créé un module d'adresse qui contient l'adresse postale pouvant contenir un des trois modèles de contenu, dépendant si l'adresse est dans l'UE, aux États-Unis d'Amérique ou ailleurs. La sortie de tous les trois modèles de contenu est définie dans un seul SPS. Toutefois, ils auraient pu être définis dans des SPS séparés, qui auraient fourni une granularité plus fine. Dans ce cas, la modularisation fournirait plus de flexibilité puisque les modules pourrait être réutilisés plus facilement.

La description de ce projet est organisée comme suit :

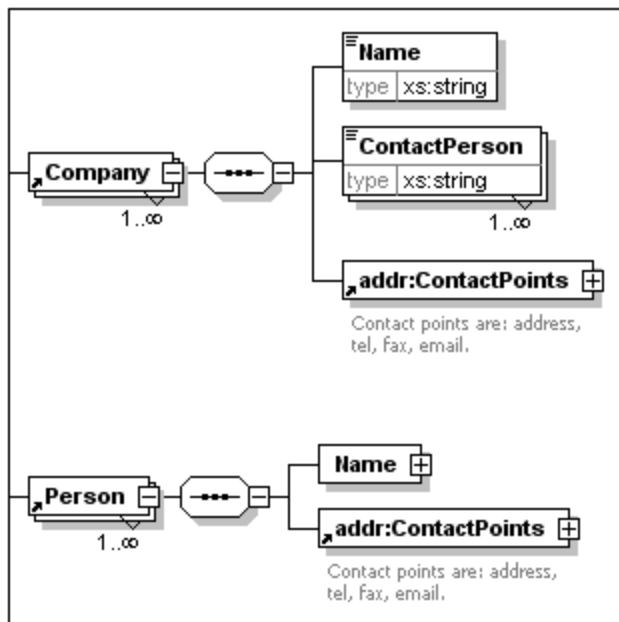
- [Les fichiers de schéma](#) ²²⁹
- [Les sources de données XML](#) ²³¹
- [Les fichiers SPS](#) ²³¹

Les fichiers de schéma

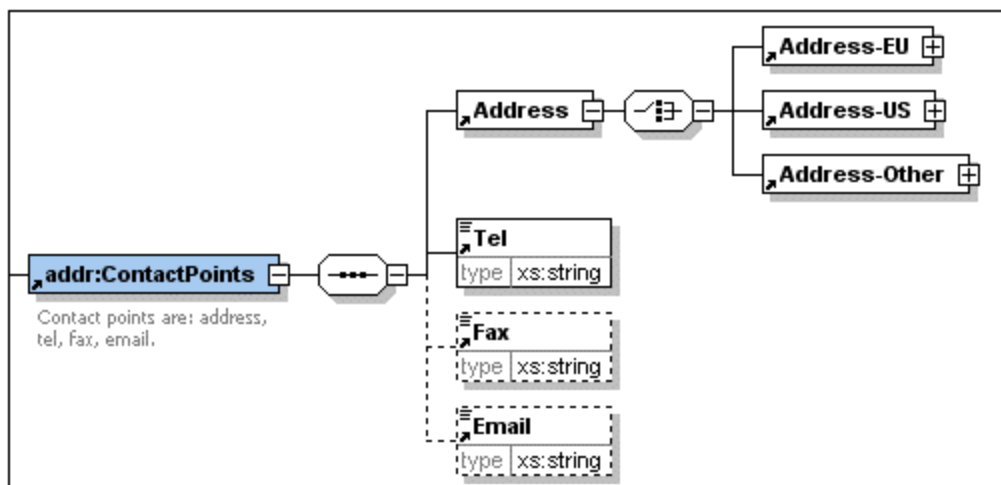
Lorsque vous créez des schémas pour les SPS modulaires, n'oubliez pas de créer les éléments que vous souhaitez réutiliser comme éléments globaux. Le schéma pour le carnet d'adresse est `AddressBook.xsd`. Ce schéma a été construit en important les schémas pour le carnet d'adresse professionnel (`BusinessAddressBook.xsd`) et le carnet d'adresse personnel (`PersonalAddressBook.xsd`). Le schéma `BusinessAddressBook.xsd` fournit un modèle de contenu pour les entreprises, tandis que le schéma `PersonalAddressBook.xsd` fournit un modèle de contenu pour les personnes (see screenshot below).



Les deux schémas importent le schéma `ContactPoints.xsd` (voir capture d'écran ci-dessous), qui définit un modèle de contenu pour les détails du contact.



Finalement, le schéma `ContactPoints.xsd` (capture d'écran ci-dessous) inclut le schéma `Address.xsd`, qui définit les trois modèles de contenu d'address-type : pour l'UE, les États-Unis d'Amérique, et d'autres adresses.



Les importations sont utilisées quand le schéma importé appartient à un espace de nom différent que le schéma importateur. Les Inclus sont utilisés quand le schéma inclus appartient au même espace de nom tel que le schéma inclus.

Note : les captures d'écran ci-dessus sont celles du schéma dans le l'Aperçu de schéma de XMLSpy d'Altova.

Les sources de données XML

Les données XML sont contenues dans le fichier `AddressBook.xml` : Ce fichier est structuré pour que l'élément `AddressBook` contienne les éléments `companies` et `persons` comme ses enfants. Les modèles de contenu de ces deux éléments sont définis dans les fichiers de schéma, `BusinessAddressBook.xsd` et `PersonalAddressBook.xsd`, respectivement.

Il s'agit de deux fichiers de données XML supplémentaires, qui correspondent aux schémas `BusinessAddressBook.xsd` et `PersonalAddressBook.xsd`. Ces deux fichiers XML, `BusinessAddressBook.xml` et `PersonalAddressBook.xml`, sont utilisés comme fichiers de travail XML des fichiers SPS correspondants.

Les trois fichiers XML sont les [fichiers de travail XML](#) ²³ des modules SPS suivants :

- `AddressBook.xml` => `AddressBook.sps`, `ContactPoints.sps`, `Address.sps`
- `BusinessAddressBook.xml` => `BusinessAddressBook.sps`
- `PersonalAddressBook.xml` => `PersonalAddressBook.sps`

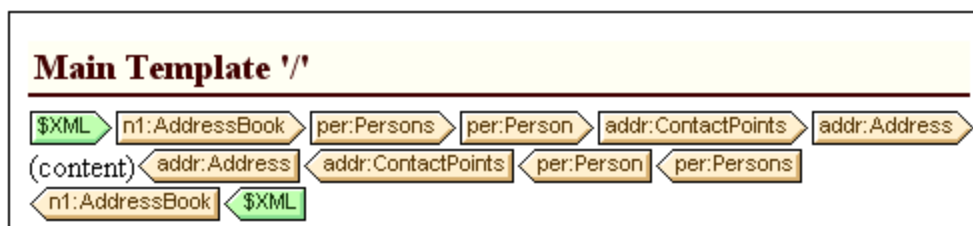
Les modules SPS

La description des modules SPS commence par le module le plus basique (`Address.sps`) et les progrès dans des étapes progressives compositionnelles du carnet d'adresse entier (`AddressBook.sps`). Tous les modules SPS utilisent `AddressBook.xsd` comme leur schéma.

Address.sps

Les points clés à noter sont l'utilisation du schéma et du fichier de travail XML.

- `Address.sps` utilise `AddressBook.xsd` comme son schéma, mais le schéma pourrait également avoir été `Address.xsd`, `ContactPoints.xsd`, `BusinessAddressBook.xsd` ou `PersonalAddressBook.xsd` — puisque l'élément `Address` est présent dans tous ces schémas et serait disponible comme élément global. Lorsque le module SPS est ajouté dans un autre module SPS, le schéma du module importé est ignoré, donc il n'est pas important lequel est utilisé quand SPS est ajouté comme module.
- Le fichier de travail XML est `AddressBook.xml`. Notez que le modèle principal dans `Address.sps` spécifie que l'élément `Address` devrait être traité, et que les modèles globaux pour `Address-EU`, `Address-US` et `Address-Other` ont été définis.

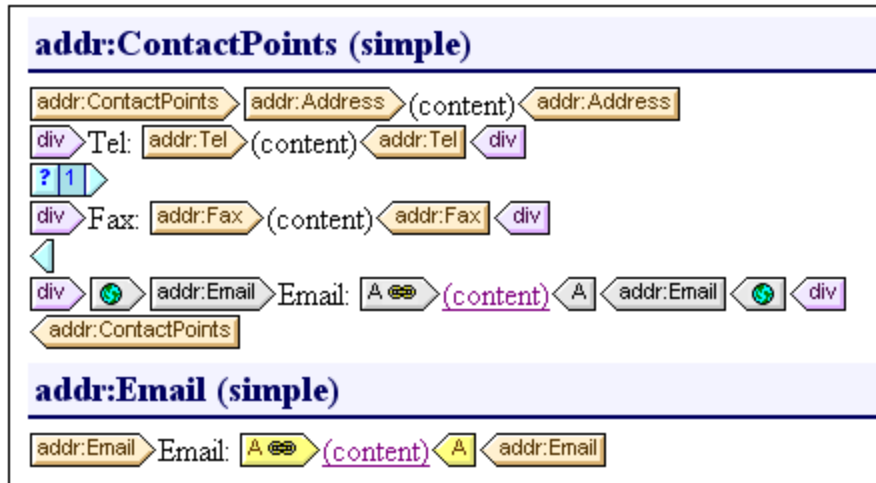


Car ce n'est que l'élément `Address` qui est traité, les aperçus de sortie affichent uniquement la sortie de `Address`. Quand `Address.sps` est utilisé comme module, les modèles globaux sont ajoutés et le modèle principal est ignoré.

ContactPoints.sps

Cet SPS importe un module. Notez l'utilisation de modèles globaux à l'intérieur d'autres modèles globaux et le modèle principal.

- `ContactPoints.sps` utilise `AddressBook.xsd` comme son schéma et comme son fichier de travail XML `AddressBook.xml`.
- `Address.sps` est ajouté comme module, rendant ainsi les modèles globaux des éléments `Address-EU`, `Address-US` et `Address-Other` disponibles.
- Les modèles globaux pour les éléments `ContactPoints` et `Email` sont définis. Notez que la définition `ContactPoints` utilise le modèle global de `Email` (capture d'écran ci-dessous).



- Le modèle principal — requis pour les aperçus — utilise le modèle global de l'élément `ContactPoints`, permettant ainsi des aperçus de la sortie `ContactPoints`.

BusinessAddressBook.sps et PersonalAddressBook.sps

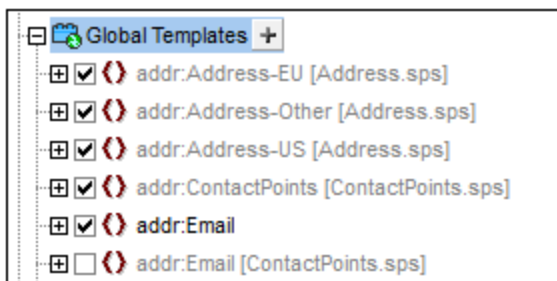
Ces SPS importent chacun un module, qui, à son tour, en importe un autre. Notez que le modèle principal applique simplement des modèles globaux.

- Chacun de ces deux modules utilise `AddressBook.xsd` comme son schéma. Les fichiers de travail XML sont, respectivement, `BusinessAddressBook.xml` et `PersonalAddressBook.xml`.
- `ContactPoints.sps` est ajouté comme un module. Ceci fait que `Address.sps` est importé indirectement. Tous les modèles globaux dans ces deux modules sont disponibles au module SPS de référence.
- Dans `BusinessAddressBook.sps`, les modèles globaux sont définis pour les éléments `Companies` et `Company`. Notez que la définition `Company` utilise le modèle global de `ContactPoints`.
- Dans `PersonalAddressBook.sps`, les modèles globaux sont définis pour les éléments `Person` et `Persons`. La définition `Person` utilise le modèle global de `ContactPoints`.

AddressBook.sps

Il existe deux modèles globaux pour l'élément `Email` ; les deux peuvent être activés au choix.

- `AddressBook.sps` utilise `AddressBook.xsd` comme son schéma. Le fichier de travail XML est `AddressBook.xml`.
- `BusinessAddressBook.sps` et `PersonalAddressBook.sps` sont ajoutés comme modules, et ceci fait que `ContactPoints.sps` et `Address.sps` sont importés indirectement.
- Un modèle global est défini pour l'élément `Email`. Ceci signifie qu'il existe désormais deux modèles globaux pour `Email`, un dans `ContactPoints.sps` et l'autre dans `AddressBook.sps` (voir la capture d'écran ci-dessous).



- Dans la liste des modèles globaux dans l'Arborescence de design (*capture d'écran ci-dessus*), vous pouvez sélectionner lequel des deux modèles globaux doit être activé. StyleVision ne permet qu'un module à la fois. Celui qui est actif est utilisé à l'intérieur du modèle global `ContactPoints`.
- Le modèle global contient du contenu statique pour l'en-tête de sortie.

6.4 Fragments de modèles et de design

Le document de design est composé de modèles, et il est important de reconnaître les différents types de modèles qui peuvent être utilisés.

- *Modèles principaux et modèles globaux* : Le document de design consiste en un [modèle principal](#)²³⁴ et, en option, un ou plusieurs [modèles globaux](#)²³⁴. Les modèles globaux peuvent être référencés par le biais du modèle principal.
- *Modèles de nœud et itérateurs variables* : Il s'agit des modèles qui constituent le modèle principal et les modèles globaux. Un [modèle de nœud](#)²⁴² correspond au nœud dans une source de schéma.
- *Fragments de design* : Il s'agit de modèles qui sont conçus séparément et réutilisés dans différentes parties du design (modèle principal ou modèles globaux).

Dans cette section, nous décrivons le rôle que jouent les modèles et fragments de design dans la structure du design. Nous ne nous penchons pas sur les [propriétés de présentation](#)³²⁸ ici dans le design, uniquement sur la structure.

Note : Dans l'Aperçu de design, la SPS peut avoir plusieurs modèles : le modèle principal, les modèles globaux, et les fragments de design. Vous pouvez contrôler lesquels des types de modèles sont affichés dans l'Aperçu de design en utilisant les [filtres d'affichage du modèle](#)⁴⁴⁹, qui sont disponibles comme les [icônes de la barre d'outils](#)⁴⁴⁹. Ces filtres d'affichage vous aideront à optimiser et basculer entre les différents affichages de votre SPS.

6.4.1 Modèle principal

Le [modèle principal](#)²³ détermine la structure de la sortie. Ceci signifie que la séquence dans laquelle le modèle principal est présenté dans le design est la séquence dans laquelle sont présentés. Dans le jargon de programmation, il s'agit de traitement procédural. Le traitement commence au début du modèle et se poursuit en séquence jusqu'à la fin. Au cours de l'opération, les nœuds du document XML sont traités. Les modèles qui traitent ces nœuds sont appelés [modèles locaux](#)²². Une fois qu'un modèle local est traité, le processeur se déplace vers le prochain composant dans le modèle principal, etc. Occasionnellement, un nœud peut faire référence au [modèle global](#)²² pour son traitement. Dans de tels cas, une fois que le modèle global est exécuté pour ce nœud, le processeur retourne à la position du modèle principal de laquelle il a été élargi continue en séquence depuis le prochain composant.

Le point d'entrée pour le modèle principal est le [nœud du document](#)²² du schéma. StyleVision offre l'option de sélectionner les multiples éléments racine ([éléments de document](#)²²). Ceci signifie qu'à l'intérieur du modèle principal, il peut y avoir des [modèles locaux](#)²² pour chacun des éléments du document actifs. Celui qui est exécuté au cours du traitement sera celui pour l'élément qui est l'élément de document du document d'instance XML étant traité.

6.4.2 Modèles globaux

Un [modèle global](#)²² peut être défini pour tout nœud ou type dans le schéma, ou pour un nœud spécifié dans un modèle XPath.

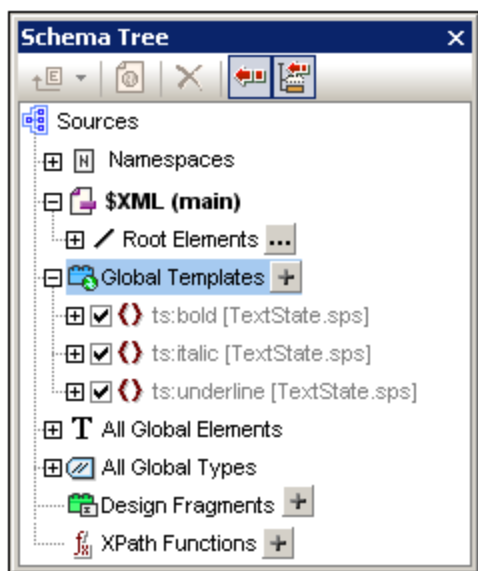
Un modèle global spécifie les instructions pour un nœud sélectionné ou type, et est invoqué par un appel du [modèle principal](#) ²³, [fragments de design](#) ²⁴⁵, ou autres modèles globaux. Le modèle de traitement est semblable à celui des langages de programmation déclaratifs, dans ce sens qu'un seul modèle est défini et invoqué plusieurs fois. De cette manière, une seule définition peut être réutilisée de multiples fois. Les modèles globaux sont invoqués lors de deux situations :

- Lorsqu'un nœud ou type dans le [modèle principal](#) ²³ a été défini pour référencer son modèle global (fait en cliquant avec la touche de droite de la souris sur le composant dans le design et sélectionnant Créer Modèle global).
- Quand un [\(contenus\)](#) ¹¹⁰ ou [\(reste-de-contenus\)](#) ¹¹³ est inséré à l'intérieur d'un élément ou type dans un [modèle local](#) ²² et le reste du contenu de cet élément ou type inclut un nœud ou type pour lequel le [modèle global](#) ²² existe.

Des modèles globaux sont utiles si un nœud (ou type) apparaît à l'intérieur d'éléments variés ou emplacements variés, et un seul ensemble d'instructions est requis pour toutes les occurrences. Par exemple, partons du fait qu'un élément `para` doit être formaté de la même manière peu importe s'il apparaît dans un élément `chapter`, `section`, `appendix` ou `blockquote`. Une approche efficace serait de définir un modèle global pour `para` et puis s'assurer que dans le [modèle principal](#) ²³, le modèle global pour l'élément `para` est traité là où requis (par exemple, en incluant `//chapter/para` dans le modèle principal et spécifiant que `para` référence son modèle global ; ou en incluant `//chapter/title`, puis en incluant [\(contenus\)](#) ¹¹⁰ ou [\(rest-of-contents\)](#) ¹¹³ pour que le reste du contenu de l'élément `chapter` est traité avec les modèles globaux et modèles par défaut). Aussi, un modèle global peut être défini pour un type complexe (par exemple, un qui définit un modèle d'adresse) ou même un type simple (par exemple, `xs:decimal`). Dans de tels cas, toutes les occurrences du type (complexe ou simple) qui invoquent le modèle global pour ce type seront traitées selon les règles dans le modèle global.

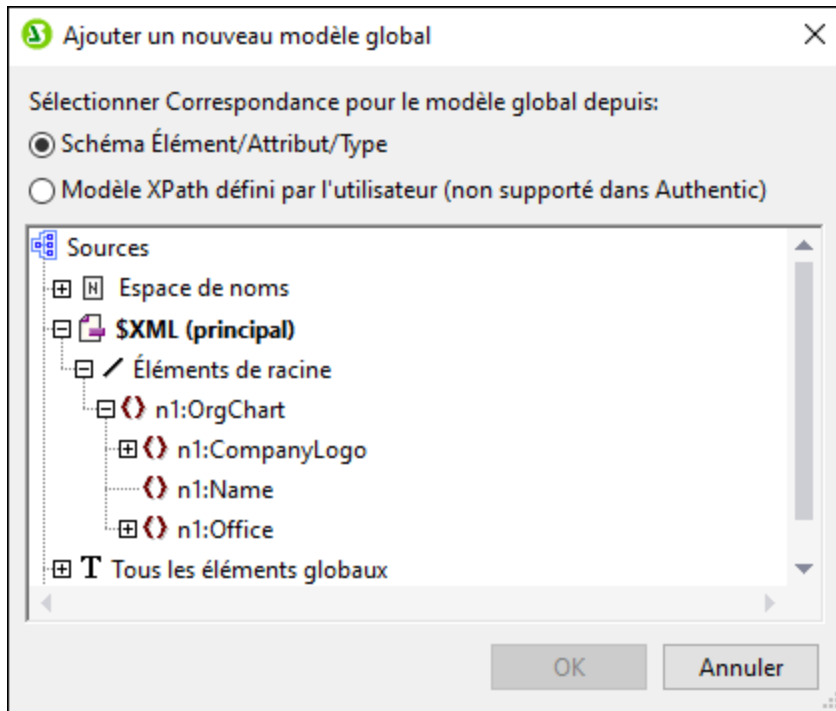
Créer un modèle global

Les modèles globaux peuvent être créés pour tout nœud dans le schéma, ou pour un nœud spécifié dans un modèle XPath et sont créés depuis la barre latérale de l'Arborescence de schéma (*capture d'écran ci-dessous*).



Un modèle global peut être créé d'une des deux manières :

- Cliquez sur le bouton **Ajouter Nouveau modèle global** à droite de l'item Modèles globaux dans l'Arborescence de schéma (voir la capture d'écran ci-dessous). Le dialogue Ajouter Nouveau modèle global apparaît (capture d'écran ci-dessous). Vous pouvez sélectionner un élément, un attribut ou un type depuis l'arborescence de schéma dans le dialogue, ou vous pouvez saisir un modèle XPath. Ceci sélectionne le nœud à créer comme modèle global. Cliquez sur **OK** pour terminer. Le modèle sera créé et ajouté aux modèles déjà existants dans l'Aperçu de design et peut ensuite être édité. Dans l'Arborescence de schéma, le nœud de schéma ou type sera marqué avec l'icône du symbole plus placé juste devant.



- Cliquez avec la touche de droite sur le nœud de schéma ou le composant type dans l'Arborescence de schéma (sous les Éléments racine, Tous les Éléments globaux, ou Tous les Types globaux, le cas échéant), et sélectionnez la commande **Créer/Supprimer Modèle global**. Le modèle sera créé et ajouté aux modèles déjà existants dans l'Aperçu de design et peut ensuite être édité. Dans l'Arborescence de schéma, le nœud de schéma ou type sera marqué avec l'icône du symbole plus placé juste devant.
- Les modèles globaux peuvent aussi être créés depuis les modèles dans le modèle principal de l'Aperçu de design. Cliquez avec la touche de droite sur le modèle (soit dans l'Aperçu de design, soit dans la barre latérale de l'Arborescence de schéma) et sélectionnez la commande **Créer Modèle global**. Un modèle global est créé depuis le modèle sélectionné (il est ajouté aux modèles dans l'Aperçu de design) et le modèle dans le modèle principal est automatiquement défini pour **utiliser** ce modèle global (voir ci-dessous pour une explication sur comment les modèles globaux sont *utilisés*).

Un modèle global est situé dans l'Aperçu de design en dessous du modèle principal. Il est indiqué par une barre mauve contenant le nom du nœud pour lequel le modèle global a été créé, suivi de son type : (*simple*) ou (*complex*). Un modèle global est indiqué dans la capture d'écran ci-dessous.



Notez que le traitement du modèle global est défini par l'utilisateur et pourrait inclure les deux composants statiques et dynamiques, ainsi que toute l'étendue des options de traitement disponibles pour traiter le modèle principal.

Utiliser le modèle global

Une fois que le modèle global a été créé, il peut être utilisé quand un nœud ayant le même nom qualifié est inséré dans le document (lorsque le nœud est abandonné dans le design, sélectionnez la commande **Utiliser le Modèle global** depuis le menu qui apparaît) en le déposant. En alternative, si un modèle local est présent dans le design et qu'un modèle global existe pour un nœud ayant le même nom qualifié, alors le modèle global peut être utilisé à la place du modèle local. Pour utiliser un modèle global pour un modèle local, cliquez avec la touche de droite sur le modèle local dans l'Aperçu de design et sélectionnez la commande **Utiliser le modèle global**. Lorsqu'un modèle global est utilisé, ses instructions de traitement sont appelées et utilisées par le modèle local en cours d'exécution.

À chaque moment qu'un modèle global est utilisé dans le design, un modèle XPath peut être créé dans le modèle global pour filtrer l'ensemble de nœud en question. Pour créer un tel filtre, cliquez avec la touche de droite sur la balise du modèle global dans le design et sélectionnez **Éditer filtre XPath**²⁴³ dans le menu contextuel qui apparaît. Ceci ouvre le **dialogue Éditer expression XPath**⁴²⁶, dans lequel l'expression requise peu être saisie.

Modèles globaux récursifs

Les modèles globaux peuvent être récursifs, c'est-à-dire qu'un modèle global peut s'appeler lui-même. Toutefois, pour éviter une boucle sans fin dans Authentic View, une propriété pour limiter la call-depth peut être définie. Cette propriété, la propriété *Maximum Template-Call-Depth*, est disponible dans l'onglet Authentic du dialogue des Propriétés du SPS (**Fichier | Propriétés**⁴⁷²). Elle spécifie un nombre maximum d'appels de modèles qui peuvent être traités de manière récursive lors du traitement de la sortie Authentic View. Si le nombre d'appels de modèle excède le nombre spécifié dans la propriété *Maximum Template-Call-Depth*, une erreur est renvoyée.

Copier un modèle global localement

Une fois qu'un modèle global a été créé, ses instructions de traitement peuvent être copiées directement dans un modèle du même nom qualifié dans le modèle principal. À cette fin, cliquez avec la touche de droite sur le modèle local et sélectionnez la commande **Copy Global Template Locally**. Copier le modèle global est différent qu'utiliser le modèle global (en cours d'exécution) en ce sens que les instructions de traitement sont simplement copiées de manière ponctuelle. Le modèle global n'a pas d'autre influence sur le modèle local. Soit l'un des modèles global et local ou les deux peuvent être modifiés par la suite de manière indépendante sans affecter l'un ou l'autre. D'autre part, s'il est spécifié qu'un modèle global doit être *utilisé* (en cours d'exécution) par un modèle local, alors toute modification au modèle global sera reflétée dans le modèle local en cours d'exécution.

Activer et désactiver les modèles globaux

Un modèle global peut être activé en cochant son entrée dans les modèles globaux recensés dans la barre latérale de l'arborescence de schéma. Il peut être désactivé en décochant l'entrée. Si un modèle global a été

activé (le paramètre par défaut lorsque le modèle global a été créé), il est généré dans la feuille de style XSLT. S'il a été désactivé, il n'est pas généré dans la feuille de style XSLT mais est toujours enregistré dans le design SPS.

Tout modèle local qui utilise un modèle global désactivé sera — puisqu'il n'est pas en mesure de référencer le modèle global manquant — replié sur les modèles XSLT par défaut, qui ont l'effet collectif de sortir les contenus des nœuds de texte descendant.

Les avantages de la fonction activation/désactivation sont : (i) les modèles globaux ne doivent pas être supprimés s'ils ne sont pas requis temporairement ; ils peuvent être réactivés plus tard lorsqu'ils sont requis ; (ii) s'il existe des conflits de nom avec des modèles des feuilles de style importées, alors le modèle global qui n'est pas requis peut être désactivé temporairement.

Supprimer un modèle global

Pour supprimer un modèle global, cliquez avec la touche de droite sur le modèle global à supprimer, soit dans l'Aperçu de design ou la barre latérale de l'Arborescence de schéma, et sélectionnez la commande **Créer/Supprimer Modèle global**.

Modèles globaux simples et modèles globaux complexes

Les modèles globaux ont deux types : simple et complexe. Les modèles globaux complexes sont disponibles pour les raisons de rétro-compatibilité. Si un modèle global dans SPS créé avec une nouvelle version de StyleVision avant la version 2006 contient une table ou une liste, alors le modèle global sera généralement ouvert dans StyleVision 2006 et des versions ultérieures en tant que modèle global complexe.

Un modèle global complexe est différent qu'un simple modèle global dans la manière dont le nœud est traité pour lequel le modèle global a été créé. Lorsque la première instance du nœud apparaît dans le document, le modèle global complexe traite toutes les instances ultérieures de ce nœud immédiatement après. D'un autre côté, un modèle global simple, traite chaque instance de nœud uniquement si l'instance du nœud apparaît individuellement.

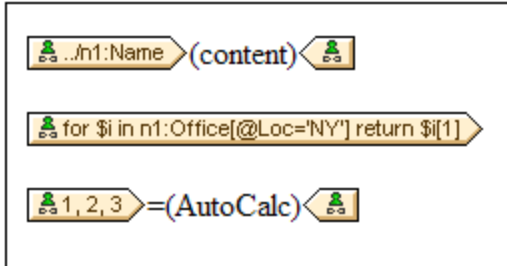
Il est important de noter qu'un simple modèle global sera automatiquement converti en un modèle global complexe si un [format prédéfini](#) ³²⁹ ou une newline est créé **autour** du nœud de l'élément pour lequel le modèle global a été créé. Le comportement de traitement en résultera pour des modèles globaux complexes (décrit dans l'item de la liste précédent). Pour revenir au modèle global simple, le [format prédéfini](#) ³²⁹ devrait être supprimé (en déplaçant le nœud à l'extérieur du format prédéfini, puis en supprimant le format prédéfini), ou le newline devrait être supprimé (en supprimant l'item dans la [barre latérale de l'Arborescence de design](#) ⁴⁰), comme cela pourrait être le cas. Pour éviter une conversion automatique du modèle global simple au modèle global complexe, assurez-vous que le [format prédéfini](#) ³²⁹ ou la newline est ajouté à l'intérieur des balises de nœuds de l'élément pour lequel le modèle global simple a été créé.

Les modèles globaux dans les SPS modulaires

Quand un [module SPS est ajouté dans un autre SPS](#) ²²⁰, les modèles globaux dans le module ajouté sont disponibles pour l'utilisation à l'intérieur du SPS de référence. Pour plus d'information sur l'utilisation des SPS modulaires, voir la section [SPS modulaires](#) ²²⁰.

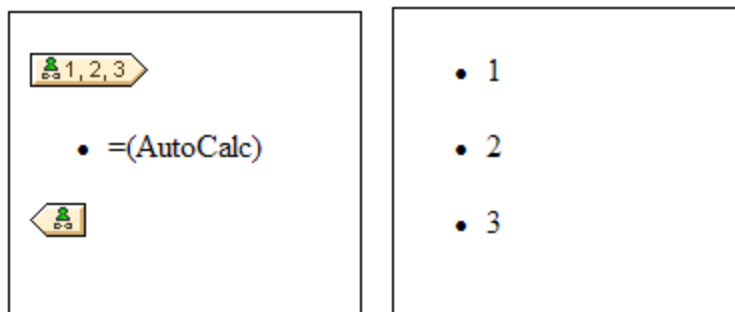
6.4.3 Modèles définis par l'utilisateur

Les modèles définis par l'utilisateur sont des modèles pour des items générés par une expression XPath que vous spécifiez. Ces items peuvent être des valeurs atomiques ou des nœuds. Dans la capture d'écran ci-dessous, qui affiche trois modèles définis par l'utilisateur, notez l'icône du modèle défini par l'utilisateur à gauche des balises (un symbole de personne en vert). Les modèles définis par l'utilisateur sont très utiles car ils fournissent une flexibilité extraordinaire pour créer les modèles.



L'expression XPath des trois modèles définis par l'utilisateur affichés dans la capture d'écran ci-dessus fait les choses suivantes :

- Sélectionne un nœud dans le schéma de source. En utilisant une expression XPath, tout nœud dans les sources de schéma peut être atteint de l'intérieur de chaque nœud contextuel. Si StyleVision peut cibler sans ambiguïté le nœud spécifié, le modèle sera modifié automatiquement depuis un modèle défini par l'utilisateur vers un modèle normal. S'il s'agit d'un modèle défini par l'utilisateur, ceci sera indiqué par une icône de modèle défini par l'utilisateur en vert du côté gauche des balises de modèle.
- Sélectionne un nœud qui remplit une condition spécifiée par la construction `pour` de XPath 2.0 et XPath 3.0. De tels modèles ne peuvent jamais être résolus en modèles normaux (mais resteront des modèles définis par l'utilisateur) car la construction `for` n'autorise pas StyleVision à résoudre sans ambiguïtés la cible de l'information de schéma uniquement de l'information de schéma dont elle dispose actuellement.
- Sélectionne une séquence de valeurs atomiques {1, 2, 3}. Tandis qu'il est permis de créer un modèle pour une valeur atomique, vous ne pouvez pas utiliser l'espace réservé `contents` à l'intérieur d'un tel modèle. Ceci est dû au fait que l'instruction `xsl:apply-templates` (c'est ce que l'espace réservé `contents` génère) ne peut être appliquée qu'aux items de nœud (pas les valeurs atomiques). Vous devriez, toutefois, utiliser un calcul automatique en combinaison avec un élément de design tel qu'une liste. Par exemple, le modèle défini par l'utilisateur à gauche générerait la sortie à droite.



Note : si le fichier SPS utilise XSLT 1.0, alors l'expression XPath que vous saisissez doit retourner un ensemble de nœud. Autrement une erreur est reportée.

L'avantage d'utiliser XPath pour sélectionner un nœud modèle

L'avantage de sélectionner un nœud de schéma via une expression XPath (modèles définis par l'utilisateur) est que le mécanisme du sélecteur de chemin XPath peut être utilisé pour sélectionner tout nœud ou séquence d'items, de même que filtrer ou définir les conditions pour la sélection de nœud. Le résultat étant que des nœuds de document XML spécifiques peuvent être ciblés pour tout modèle donné. Par exemple, l'expression XPath `//Office/Department[@Location="NY"]` sélectionnera uniquement ces nœuds `Department` qui ont un attribut `Location` avec une valeur de `NY`. Voir également d'autres exemples dans cette section.

Note : Si une expression XPath contient plusieurs étapes de chemin d'accès, il est important — en particulier pour le regroupement et le tri — de déterminer si les parenthèses doivent être placées autour des différentes étapes du chemin d'accès ou non. Par exemple, l'expression XPath `/Org/Office/Dept` sera traitée différemment que `(/Org/Office/Dept)`. Pour l'expression précédente (sans crochets pointus), le processeur transite à travers l'étape de l'emplacement. Pour la dernière expression (avec des crochets pointus), tous les éléments `Dept` de tous les éléments `Office` sont renvoyés en un ensemble de nœud non différencié.

Crochets pointus	Mécanisme XSLT sous-jacent	Effet
Non	<pre><xsl:for-each select="Org"> <xsl:for-each select="Office"> <xsl:for-each select="Dept"> ... </xsl:for-each> </xsl:for-each> </xsl:for-each></pre>	Chaque élément <code>Office</code> a sa propre population <code>Dept</code> . Donc regrouper et trier peut être effectué à l'intérieur de chaque <code>Office</code> .
Oui	<pre><xsl:for-each select="/Org/Office/Dept"> ... </xsl:for-each></pre>	La population <code>Dept</code> s'élargit sur tous les éléments <code>Office</code> et à travers <code>Org</code> .

La différence d'évaluation des expressions XPath peut être significative pour regrouper et trier.

Insérer un modèle défini par l'utilisateur.

Pour insérer un modèle défini par l'utilisateur, faites comme suit :

1. Cliquez sur l'icône **Insérer modèle défini par l'utilisateur** dans la barre d'outils Insérer éléments de design, puis cliquez dans l'emplacement du design là où vous souhaitez insérer le modèle. En alternative, cliquez avec la touche de droite dans l'emplacement de design où vous souhaitez insérer le modèle et, depuis le menu contextuel qui apparaît, sélectionnez la commande **Insérer modèle défini par l'utilisateur**.
2. Dans le dialogue [Éditer expression XPath](#) ⁴²⁶ qui s'ouvre, saisissez l'expression XPath que vous souhaitez et cliquez sur **OK**. Notez que le nœud contextuel de l'expression XPath sera le nœud dans lequel vous avez cliqué. Un modèle de nœud vide sera créé. Dès fois, un nœud joint est créé. Lorsqu'un nœud est joint, les nœuds d'instance ciblés sont sélectionnés comme s'ils étaient à un seul niveau, tandis que si un nœud n'est pas joint (c'est-à-dire s'il est fractionné en de multiples niveaux hiérarchiques), alors la sélection du nœud est effectuée en passant à travers chaque instance à chaque niveau de fractionnement. L'ensemble de nœud dans les deux cas de la sélection (joint et fractionné) est le même sauf si le critère de groupement ou de tri est spécifié. Pour une discussion de l'effet qu'ont des nœuds joints sur le groupement et le tri des mécanismes, voir [Opérations Nœud-](#)

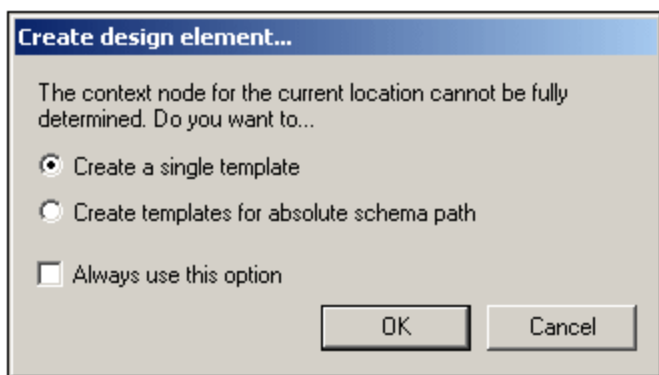
Modèle ²⁴²

Éditer une correspondance de modèle

La sélection du nœud de tout modèle de nœud (défini par l'utilisateur ou normal) peut être modifiée en utilisant une expression XPath pour sélectionner la nouvelle expression de correspondance. Pour éditer la correspondance de modèle d'un modèle de nœud, cliquez avec la touche de droite sur le modèle de nœud, puis sélectionnez la commande **Éditer correspondance de modèle**. Ceci ouvre le dialogue Éditer expression XPath, dans lequel vous saisissez l'expression XPath peut être saisie. Ensuite, cliquez sur **OK**.

Ajouter des nœuds aux modèles définis par l'utilisateur

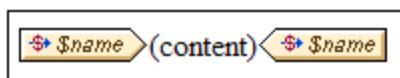
Si un nœud de l'arborescence de schéma est ajouté à un modèle défini par l'utilisateur, le contexte pour un nouveau nœud ne sera pas connu si le modèle défini par l'utilisateur a été créé pour un nœud ou une séquence qui ne peut pas être placé dans le contexte de la source de schéma du SPS. Vous serez donc invité pour savoir comment le nouveau nœud (*capture d'écran ci-dessous*) doit être référencé: (i) par son nom (essentiellement, un chemin relatif), ou (ii) par un chemin entier de la racine à la source de schéma.



Le comportement par défaut est de demander un avis afin de voir comment procéder. Ce comportement par défaut peut être changé dans l'onglet de design du [dialogue Outils | Options](#) ⁵³⁵.

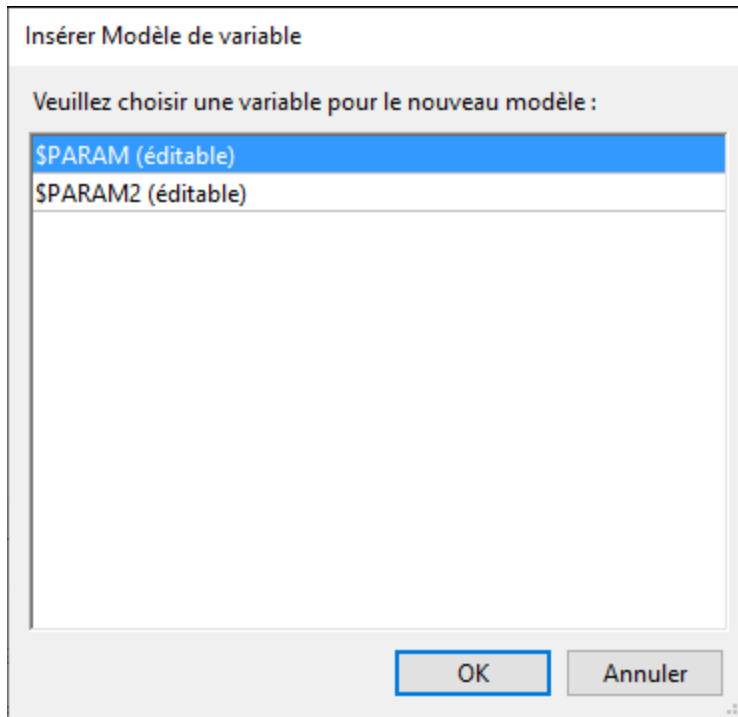
6.4.4 Modèles variables

Un **Modèle de variable** est un modèle qui cible une variable et, par défaut, affiche son contenu. Il est inséré avec la commande **Insérer | Modèle de variable** ou **Contenir dans | Variable**, qui insère, au point d'insertion du curseur, un modèle pour une variable définie dans SPS. Le modèle variable (*capture d'écran ci-dessous*) contient par défaut un espace réservé `contenu`, qui sert à afficher le contenu de la variable. Vous pouvez insérer du contenu supplémentaire (statique ou dynamique) dans le modèle variable selon vos besoins, ou le modifier comme vous le feriez avec tout autre modèle. Un modèle variable est indiqué par le symbole dollar dans ses balises de début et de fin.



Pour insérer un modèle variable, procédez comme suit :

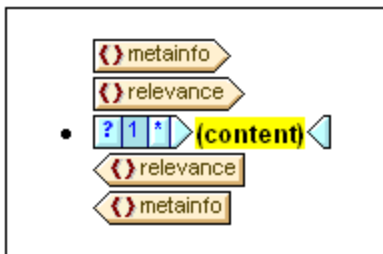
1. Placez le curseur à l'emplacement où vous souhaitez insérer le modèle dans le design.
2. Cliquez avec le bouton droit de la souris et sélectionnez la commande **Insérer | Modèle de variable**. Le dialogue Insérer Modèle de variable s'affiche (*capture d'écran ci-dessous*).



3. Le dialogue contient une liste de tous les [paramètres et variables déclarés par l'utilisateur](#)²⁸⁵ et définis dans le SPS. Sélectionnez la variable à laquelle vous souhaitez ajouter un modèle.
4. Cliquez sur **OK** pour terminer.

6.4.5 Opérations Nœud-Modèle

Un nœud-modèle est un modèle dans le design qui spécifie le traitement pour un nœud. Dans le design, des nœud-modèles sont affichés des balises de début et de fin en beige (*capture d'écran ci-dessous*). Le type de nœud est indiqué par un symbole à l'intérieur des balises (par exemple : crochets angulaires pour les nœuds d'élément égaux aux signes pour les nœuds d'attribut). La capture d'écran ci-dessous contient deux modèles de nœuds, tous deux destinés aux éléments : `metainfo` et `relevance`. Voir également, [Nœuds dans le document XML](#)⁴²¹.



Les opérations qui peuvent être exécutées sur un nœud-modèle sont accessibles via le menu contextuel de ce nœud-modèle (accédé en cliquant avec la touche de droite soit sur la balise de début ou de fin d'un nœud-modèle).

Les commandes dans ce menu contextuel sont décrites ci-dessous :

- [Modèles globaux](#) ²⁴³
- [Correspondance de modèle](#) ²⁴³
- [Filtres XPath](#) ²⁴³
- [Grouper par, Trier par, définir variables, modèle sert de niveau](#) ²⁴⁴
- [Créer fragment de design](#) ²⁴⁴
- [Supprimer uniquement balise](#) ²⁴⁵
- [Éditer, Joindre à, Changer en](#) ²⁴⁵

Ces commandes de menu sont décrites ci-dessous. Notez que pour un nœud-modèle donné, quelques commandes pourraient ne pas être disponibles ; celles-ci sont grisées dans le menu contextuel.

Modèles globaux : créer, utiliser, copier localement

Un nœud-modèle dans le modèle principal peut être modifié en ou associé au modèle global par le biais des commandes suivantes :

- *Créer un Modèle global* : Cette option est disponible si le nœud-modèle représente un élément qui est défini comme élément global dans le schéma. Un modèle global sera créé depuis le nœud-modèle. Le nœud-modèle dans le modèle principal utilisera ce modèle global et ses balises seront ensuite affichées en gris (indiquant son utilisation du modèle global).
- *Utiliser le modèle global* : Si un modèle global du même nom qualifié que celui du nœud-modèle a été défini, le nœud-modèle utilisera le traitement du modèle global. Les balises du nœud-modèle deviendront grises.
- *Copier modèle global localement* : Les instructions de traitement du modèle global du même nom qualifié que le nœud-modèle sont copiées physiquement dans le nœud-modèle. Le nœud-modèle est indépendant du modèle global. Par la suite, le nœud-modèle et le modèle global peuvent être modifiés indépendamment l'un de l'autre. Puisque le nœud-modèle ne référence pas de modèle global, il garde a couleur beige.

Pour plus d'informations, voir la section [Modèles globaux](#) ²³⁴.

Éditer la correspondance de modèle

Le nœud pour lequel un modèle a été créé peut être modifié en utilisant cette commande. La commande Éditer correspondance de modèle ouvre le [dialogue Éditer expression XPath](#) ⁴²⁶, dans lequel vous pouvez saisir une expression XPath qui sélectionne un autre nœud dans le schéma. Vous pouvez aussi saisir toute expression XPath pour modifier le modèle en un [modèle défini par l'utilisateur](#) ¹¹⁹.

Éditer/Effacer le filtre XPath

Un filtre XPath vous permet de filtrer un ensemble de nœud sur lequel un nœud-modèle est appliqué. Les filtres XPath peuvent aussi être appliqués dans les [modèles globaux](#) ²³⁴.

Par défaut, un nœud-modèle sera appliqué aux nœuds (éléments ou attributs) correspondant aux nœuds pour lequel le nœud-modèle a été créé (ayant le même nom et apparaissant à ce point dans la hiérarchie de schéma). Par exemple, un nœud-modèle pour le nœud `/Personnel/Office` sélectionnera tous les éléments `/Personnel/Office`. Si un filtre XPath avec l'expression `1` est désormais créé sur l'élément `Office`

(en cliquant avec la touche de droite sur l'élément *Office* et éditant son filtre XPath), ceci a pour effet d'ajouter une expression de prédicat à l'élément *Office*, pour que toute l'expression XPath ait l'air de ceci : `/Personnel/Office[1]`. Cette expression XPath sélectionne le premier enfant *Office* de l'élément *Personnel*, filtrant de manière efficace les autres éléments *Office*.

Un filtre peut être ajouté à tout nœud-modèle et à des nœud-modèles multiples dans le design. Ceci vous permet d'avoir des sélections correspondant à des expressions XPath telles que `/Personnel/Office[@country='US']/Person[Title='Manager']` pour sélectionner tous les managers dans les bureaux américains de l'entreprise. Dans cet exemple, un filtre a été créé sur les nœud-modèles de *Office* ainsi que sur *Person* respectivement.

Partout où un modèle global est utilisé—c'est-à-dire appelé—, un filtre XPath peut y être appliqué. Donc, pour chaque instance d'un modèle global qui est utilisé, un filtre XPath peut être appliqué au modèle global afin de restreindre l'ensemble du nœud ciblé.

Pour ajouter un filtre XPath à un nœud-modèle, cliquez avec la touche de droite sur le nœud-modèle et sélectionnez **Éditer filtre XPath**. Saisissez une expression de filtre XPath sans parenthèse, crochets ou délimiteurs de tout genre. Toute expression XPath valide peut être saisie. Par exemple :

- 1
- @country='US'
- Title='Manager'

Une fois qu'un filtre XPath a été créé pour un nœud-modèle, ceci est indiqué par un symbole de filtre dans la balise de début du nœud-modèle. Dans la capture d'écran ci-dessous, le nœud-modèle *synopsis* a un filtre.



Note : chaque modèle de nœud prend en charge un filtre XPath.

Grouper par, Trier par, définir variables, modèle sert de niveau

Les mécanismes derrière ces commandes sont décrits en détail dans leurs sections respectives :

- La commande **Group by** permet aux instances du nœud représenté par le nœud-modèle sélectionné d'être groupé. Le mécanisme de groupement est décrit dans la section [Grouping](#) ²⁷¹.
- La commande **Sort by** permet aux instances du nœud représenté par le nœud-modèle sélectionné d'être trié. Le mécanisme de tri est décrit dans la section [Sorting](#) ²⁷⁹.
- La commande **Define Variables** vous permet de définir les variables de portée sur le nœud-modèle sélectionné. Comment travailler avec des variables est décrit dans la section [Variables](#) ²⁹⁰.
- La commande **modèle sert de niveau** est une commande toggle qui crée/supprime un niveau sur le nœud-modèle. Les niveaux peuvent être spécifiés à différents niveaux afin de structurer le document dans la hiérarchie. Cette structure peut ensuite être utilisée afin de générer une table de contenus (TOC), une numérotation automatique et des références de texte. Ces fonctions sont décrites en détail dans la section [Table de contenus \(TOC\) et Référencement](#) ²⁹³.

Créer fragment de design

Crée un fragment de design depuis le modèle sélectionné. Le modèle de fragment de design est ajouté aux modèles de fragments de design en bas du design, puis ajouté à l'arborescence de design et à l'arborescence de schéma. Le fragment de design est aussi appliqué à ce point dans le design où il a été créé.

Supprimer (Modèle ou Formatage) balise uniquement

Cette commande supprime la balise du modèle ou du formatage sélectionné uniquement. Elle ne supprime aucun nœud descendant ou balise de formatage. Cette commande est utile pour supprimer une balise de formatage ou une balise d'élément parent sans supprimer tout ce qui est contenu à l'intérieur de la balise (ce qui arriverait si l'opération **Supprimer** est exécutée avec une balise sélectionnée). Notez, toutefois, que supprimer un élément parent pourrait rendre des nœuds descendants de l'élément supprimé invalide. Dans un tel cas, les nœuds invalides sont indiqués avec une rayure rouge.

Éditer, Joindre à, Changer en

Ces commandes sont décrites ci-dessous :

- **Éditer** : Ouvre un sous-menu avec des commandes Windows connues : couper, copier, coller et supprimer.
- **Joindre à** : Le nœud-modèle peut être joint aux composants de design suivants, dont chacun est décrit dans une section séparée de cette documentation : [paragraphe](#)¹¹⁰, [paragraphe spécial](#)¹¹², [Puces et numérotation](#)¹⁴⁷, [Hyperlien](#)³²¹, [Condition](#)²⁶⁶, [Signet TOC et niveau](#)²⁹³.
- **Changer en** : La fonction Changer en vous permet de changer : (i) le nœud pour lequel le modèle s'applique, ou (ii) comment le nœud est créé dans le design. Ceci est décrit en détail dans la section [La fonction Changer en](#)¹⁸¹.

6.4.6 Fragments de design

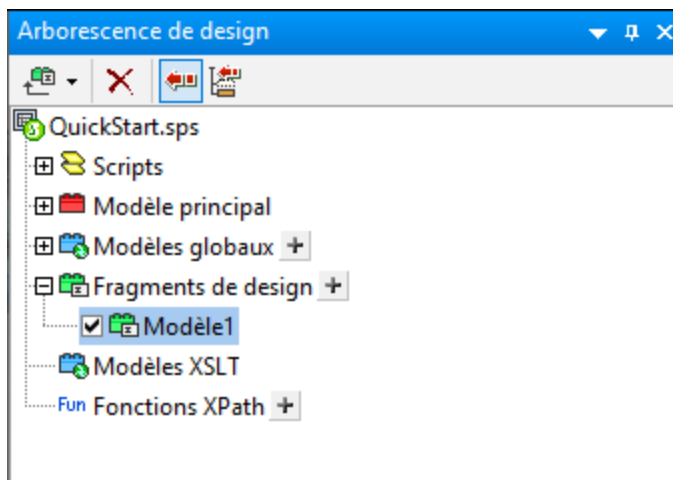
Les fragments de design sont utiles pour créer des parties qui peuvent être réutilisées à différents emplacements dans le document, semblable à la manière dont les fonctions sont réutilisées. Le mécanisme d'utilisation est comme suit :

1. [Créer le fragment de design dans le design](#)²⁴⁵
2. [Remplir les contenus du fragment de design](#)²⁴⁶
3. [Insérer le fragment de design à un emplacement dans un modèle](#)²⁴⁷.

Créer un fragment de design

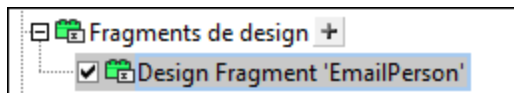
Pour créer un fragment de design, procédez comme suit :

1. Dans l'arborescence de design ou l'arborescence de schéma, cliquez sur l'icône Ajouter fragment de design , qui se situe à droite de l'élément des fragments de design dans l'arborescence (*voir capture d'écran ci-dessous*). Ceci ajoute un item de fragment de design dans la liste des fragments de design de l'arborescence. (*Voir aussi note ci-dessous.*)



Notez que le fragment de design est créé dans le design SPS. Ce modèle est ajouté aux modèles qui se trouvent déjà dans le design et est indiqué par un en-tête vert. (Si vous souhaitez uniquement voir les fragments de design qui se trouvent dans le design, masquez le modèle principal et les modèles globaux en cliquant soit sur leurs icônes [Afficher/Masquer](#) ⁴⁴⁹ dans la barre d'outils de StyleVision, à savoir le [filtre de design](#) ⁴⁴⁹.) De plus, les modèles de fragment de design sont également recensés dans l'arborescence de schéma pour y accéder d'ici.

2. Double-cliquez dans l'item du fragment de design (soit dans l'arborescence de design ou dans l'arborescence de schéma) afin d'éditer son nom. Nommez le fragment de design tel que requis et appuyez sur **Enter**. Le nom édité est saisi dans l'arborescence de design (*voir la capture d'écran ci-dessous*) et dans le modèle dans le design.



3. Dans le design, créez les contenus du modèle du fragment de design. Pour ce faire, consultez la prochaine section.

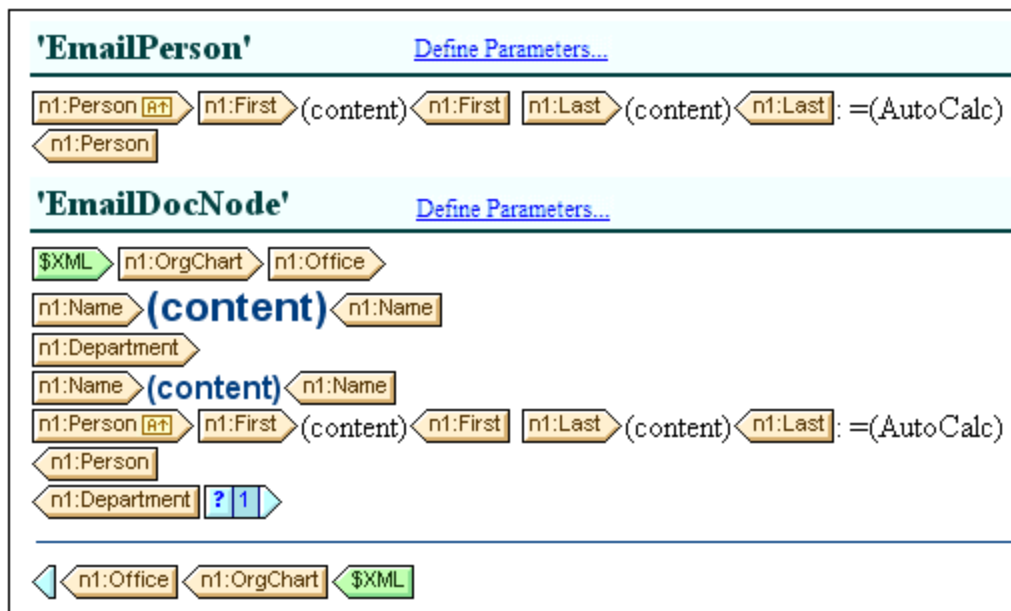
Note : si vous souhaitez créer un formulaire de fragment de design depuis un modèle déjà existant, cliquez avec la touche de droite sur le modèle et sélectionnez la commande **Créer fragment de design** depuis le menu contextuel qui s'ouvre. Ceci crée un modèle de fragment de design depuis le modèle sélectionné à ce point précis dans le design. Le modèle de fragment de design est également ajouté aux modèles de fragment du design en bas du design, puis ajouté à l'arborescence de design et à l'arborescence de schéma. Créer un fragment de design de cette manière l'applique aussi directement au point où il a été créé, il n'y a pas besoin de [l'insérer depuis l'arborescence de design ou de l'arborescence de schéma](#) ²⁴⁷.

Remplir les contenus du fragment de design

Les contenus du modèle de fragment de design sont créés [comme pour tout autre modèle](#) ¹⁰⁹. Pour insérer le contenu statique, placez le curseur dans le modèle de fragment du design et insérez le contenu statique requis. Pour insérer le contenu dynamique, glissez le nœud de schéma requis dans le modèle de fragment du design.

Lorsque vous glissez un nœud depuis la source de schéma, vous pouvez glisser le nœud soit : (i) de l'arborescence des Éléments globaux, ou (ii) de l'arborescence des Éléments racine. La différence est significative. Si un nœud est glissé de l'arborescence des Éléments globaux, il est créé sans ses éléments

ancêtre (dans la capture d'écran ci-dessous, voir le fragment de design `EmailPerson`) et, pour cette raison, lorsqu'il est utilisé dans un modèle, il devra être utilisé à l'intérieur du contexte de son parent. D'autre part, si un nœud est glissé depuis l'arborescence des Éléments racine, il est créé à l'intérieur de la structure en commençant par le nœud du document (dans la capture d'écran d'écran ci-dessous, voir le fragment de design `EmailDocNode`) et peut pour cette raison être utilisé partout dans le modèle.



La capture d'écran ci-dessus affiche deux modèles de fragment du design qui produisent une sortie identique pour l'élément `Person`. Dans le modèle de fragment du design `EmailPerson`, le nœud `Person` a été créé en glissant l'élément global `Person` dans le modèle `EmailPerson`. Dans le modèle de fragment du design `EmailDocNode`, le nœud `Person` a été glissé depuis l'arborescence des Éléments racine, et est créé avec un chemin absolu (de `$XML`, le nœud du document).

Quand ces modèles de fragment du design sont insérés dans le modèle principal, il faut faire attention que le modèle `EmailPerson` est appelé de l'intérieur du contexte qui est le parent du nœud `Person`. Vous pouvez expérimenter avec ces fragments de design. Ils se trouvent dans le fichier d'exemple `Email.sps`, qui sont dans le [dossier \(Mes\) Documents](#) ⁽²⁴⁾, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\DesignFragments`.

Vous pouvez aussi définir un paramètre avec une valeur par défaut dans le fragment du design. Le paramètre peut se voir assigner une valeur différente dans toute instance de fragment du design. Voir [Paramètres des fragments du design](#) ⁽²⁸⁷⁾ pour les détails.

Une fois que vous avez terminé le design, notez que les composants du design sont représentés graphiquement dans l'arborescence de design.

Insérer un fragment de design dans un modèle

Pour insérer un fragment de design, glissez le fragment de design de l'arborescence de design ou de l'arborescence de schéma à l'emplacement requis. L'emplacement dans lequel le fragment de design est déposé devrait être tel qu'il fournit un contexte correct. Si les contenus du fragment de design ont été créés depuis un élément global, alors le contexte correct dans le modèle principal serait le parent du nœud glissé dans le fragment de design. Voir [Créer les contenus du fragment de design](#) ⁽²⁴⁶⁾ ci-dessus.

En alternative, cliquez avec la touche de droite dans l'emplacement où le fragment de design doit être inséré et sélectionnez **Insérer fragment de design** depuis le menu contextuel.

Note : si un fragment de design est référencé dans le modèle principal et si le nom du fragment de design est modifié par la suite, alors la référence dans le modèle principal ne sera plus correcte et une erreur XSLT en résultera. Pour corriger ceci, supprimez la référence originale dans le modèle principal et créez une nouvelle référence dans le fragment de design nouvellement dénommé.

Fragment de design récursif

Les fragments de design peuvent être récursifs, c'est-à-dire qu'un fragment de design peut s'appeler lui-même. Toutefois, pour éviter une boucle sans fin dans Authentic View, une propriété pour limiter la call-depth peut être définie. Cette propriété, la propriété *Maximum Call-Depth*, est disponible dans l'onglet Authentic du dialogue des Propriétés du SPS ([Fichier | Propriétés](#) ⁴⁷²). Elle spécifie un nombre maximum d'appels de modèles qui peuvent être traités de manière récursive lors du traitement de la sortie Authentic View. Si le nombre d'appels de modèles excède le nombre spécifié dans la propriété *Maximum Call-Depth*, une erreur est renvoyée.

Supprimer un fragment de design

Pour supprimer un fragment de design, sélectionnez-le dans l'arborescence de design et cliquez sur l'icône **Supprimer** de la barre d'outils de l'arborescence de design .

Les fragments de design dans les SPS modulaires

Quand un [module SPS est ajouté dans un autre SPS](#) ²²⁰, les fragments de design dans le module ajouté sont disponibles pour l'utilisation à l'intérieur du SPS de référence. Pour plus d'information sur l'utilisation des SPS modulaires, voir la section [SPS modulaires](#) ²²⁰.

Fichier d'exemple

Pour un exemple SPS, allez au [dossier \(Mes\) Documents](#) ²⁴, C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\DesignFragments.

6.5 Modèles XSLT

Les fichiers XSLT peuvent être importés dans SPS et des modèles XSLT intégrés seront disponibles dans la feuille de style comme des modèles globaux. Si, au cours du traitement du document XML, un des nœuds XML correspond au nœud dans un modèle XSLT importé, puis le modèle XSLT est appliqué à ce nœud. Si le fichier XSLT importé contient des modèles dénommés, ceux-ci sont disponibles pour le placement dans le design.

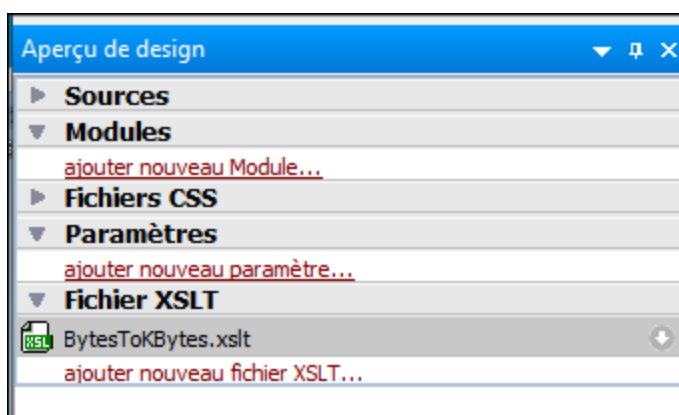
Notez les points suivants :

- Les modèles XSLT importés ne peuvent pas être modifiés dans StyleVision.

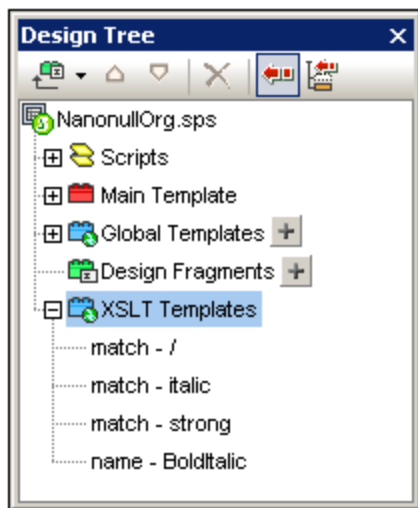
Importer le fichier XSLT

Pour importer un fichier XSLT, procédez comme suit :

1. Dans la barre latérale de l'Aoerçu de Design (*capture d'écran ci-dessous*), cliquez sur le lien **Ajouter nouveau fichier XSLT**.



2. Dans le dialogue Ouvrir qui apparaît, naviguez pour le fichier XSLT requis, sélectionnez-le et cliquez sur **Ouvrir**. Le fichier XSLT est importé. Une instruction `xsl:import` est ajoutée dans la feuille de style XSLT, et dans la barre latérale de l'arborescence de design (*capture d'écran ci-dessous*), les modèles XSLT contenus dans le fichier XSLT importé sont affichés sous les en-têtes de modèles XSLT.



Il existe deux types de modèles XSLT importés : (i) modèles correspondants (indiqués par *Match*), et (ii) modèles nommés (indiqués par *Name*). Dans l'arborescence de design, ces deux types sont recensés avec la valeur de l'attribut `select` des modèles correspondants, et (ii) par la valeur de l'attribut `name` des modèles nommés, respectivement.

Faire correspondre les modèles

Faire correspondre les modèles sera utilisé lorsqu'un modèle, au cours du traitement, applique des modèles à un nœud dans l'instance du document XML, et le modèle correspondant est sélectionné pour être appliqué. Ceci arrivera quand un nom qualifié du nœud XML correspond au nom qualifié du modèle correspondant importé. Si un modèle global a été créé dans SPS qui a le même nom qualifié, alors il a préséance sur le modèle importé et sera utilisé. S'il existe plusieurs fichiers XSLT le fichier importé en premier (et recensé d'abord dans le code XSLT) a la préséance la plus basse, suivie de la deuxième préséance la plus basse pour le fichier importé, etc.

Modèles nommés

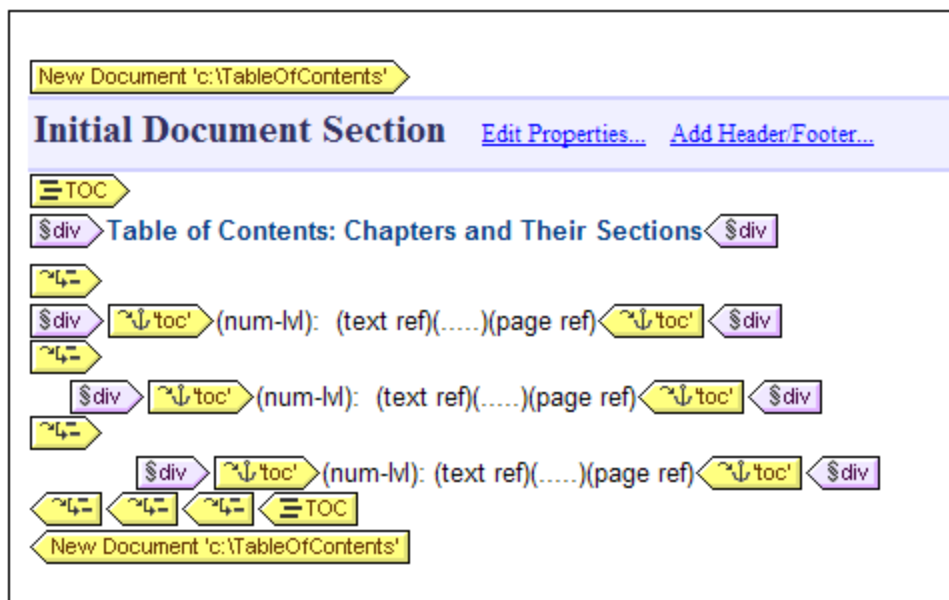
Un modèle nommé peut être glissé depuis l'arborescence de design vers tout emplacement du design. À cet emplacement, il sera créé comme un élément `xsl:call-template` (*capture d'élément ci-dessous*) qui appelle le modèle nommé.

```
<xsl:call-template name="BoldItalic" />
```

L'effet de ceci dans la sortie est de mettre en œuvre le modèle nommé à cet emplacement dans le design. Ceci peut être utile pour insérer du contenu indépendant du document d'instance XML ainsi que de la feuille de style XSLT.

6.6 Sortie multiple de document

Vous pouvez concevoir un SPS pour produire de multiples documents de sortie : un document de sortie principal et un ou plusieurs documents supplémentaires. Ceci est particulièrement utile si vous souhaitez modulariser la sortie. Les documents de sortie sont créés dans le design en insérant un nouveau modèle de document (*voir la capture d'écran ci-dessous*). Le contenu de chaque document de sortie est placé à l'intérieur du nouveau modèle de document.



Les nouveaux modèles de document peuvent être créés partout dans le design du document, permettant ainsi à la sortie de modulariser à tout niveau. Donc, par exemple, un rapport sur les différents bureaux des filiales d'une organisation globale peut avoir des documents de sortie à chacun des niveaux différents : (i) monde, (ii) continent, (iii) pays, (iv) état, et/ou (v) filiales. Chaque filiale, par exemple, peut être présentée dans un document de sortie ou alors toutes les filiales dans un pays peuvent apparaître ensemble dans un seul rapport de pays. Dans le design, un nouveau modèle devra être créé à chacun des niveaux hiérarchiques pour lequel les documents de sortie séparés son requis. Comment définir la structure correcte du document est décrit dans la section, [Nouveaux modèles de documents et Structure de design](#) ²⁵³.

Cette description de multiples documents de sortie est organisée en sous-sections comme suit :

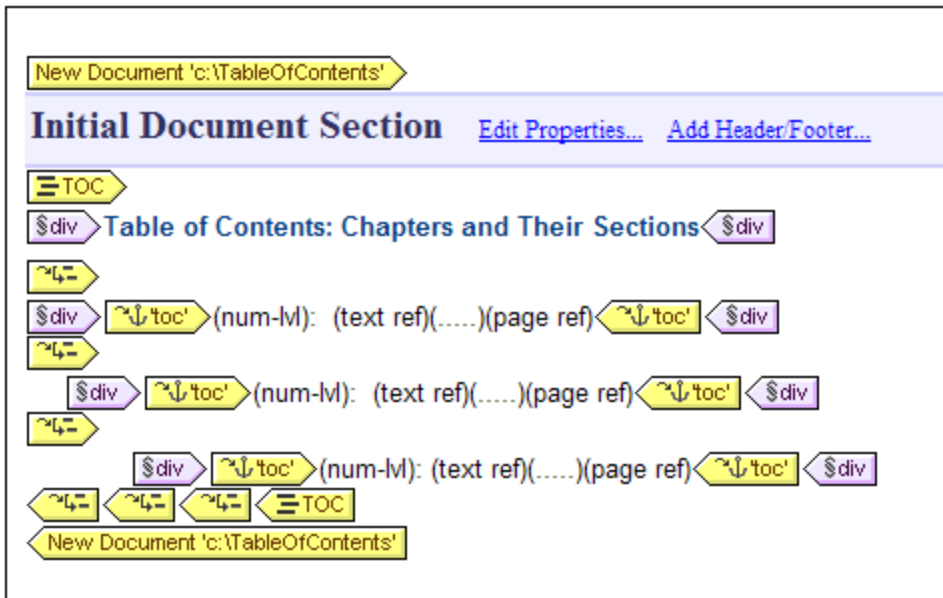
- [Insérer un nouveau document modèle](#) ²⁵²
- [Nouveaux modèles de document et Structure de design](#) ²⁵³
- [URL de nouveaux modèles de document](#) ²⁵³
- [Fichiers d'aperçu et de documents de sortie](#) ²⁵⁶
- [Propriétés du document et Styles](#) ²⁵⁸

6.6.1 Insérer un nouveau modèle de document

Un nouveau modèle de document peut être placé dans un design d'une des deux manières :

- Un nouveau modèle de document de sortie peut être **inséré** à tout emplacement dans le design. Dans ce cas, le contenu du Nouveau document est ajouté au modèle après avoir inséré le modèle. Pour insérer un Nouveau modèle de document, placez le curseur à l'emplacement désiré dans le design et sélectionnez la commande **Insérer | Insérer Nouveau document** ou cliquez avec la touche de droite à l'emplacement et, depuis le menu contextuel qui s'ouvre, sélectionnez **Insérer Nouveau document**.
- Un nouveau document de sortie peut être placé dans le design en **enveloppant le contenu** avec un Nouveau modèle de document. Le nouveau modèle de document contiendra, dans ce cas, le contenu enveloppé lorsqu'il est créé. Vous pouvez ajouter ou modifier ce contenu dans le design. Pour placer le Nouveau modèle de document pour qu'il enveloppe le contenu, mettez en lumière le contenu à envelopper, puis sélectionnez la commande **Envelopper avec | Nouveau document**. En alternative, vous pouvez sélectionner le contenu à envelopper, puis cliquez avec la touche de droite et, depuis le menu contextuel, qui apparaît, sélectionnez la commande **Envelopper avec | Nouveau document**.

Un modèle Nouveau document avec du contenu est indiqué dans la capture d'écran ci-dessous.



Notez la chose suivante de la capture d'écran ci-dessus :

1. Les balises du nouveau modèle de document contiennent l'URL (chemin et nom) du document de sortie qu'il générera. Le suffixe du filename sera généré automatiquement selon le type du fichier du format de sortie. Par exemple, pour le format de sortie HTML, le suffixe du filename `.html` sera ajouté au filename dans l'URL. Les questions pertinentes à l'attribution des URL sont discutées dans la section, [URL de nouveaux modèles de document](#) ²⁵³.
2. Le nouveau Modèle de document contient une section de document initiale.

6.6.2 Nouveaux modèles de document et Structure de design

Lorsque vous créez de multiples documents de sortie, vous devez créer les nouveaux modèles de document sur les nœuds du document source. Pour cela, vous devez prendre en considération la [structure de sortie](#) ²⁵³ de même que la [structure d'entrée \(document XML source\)](#) ²⁵³ lorsque vous concevez de multiples documents de sortie.

Document de sortie principal et documents de sortie supplémentaires (structure de sortie)

Lorsque le premier Nouveau modèle de document est ajouté au design, tout le contenu de design à l'extérieur de ce nouveau modèle de contenu est automatiquement attribué à un document séparé. Ce document séparé est considéré être le document de sortie principal, et, dans les aperçus de sortie de StyleVision, il est fait référence au document de sortie principal.

Dans les documents de sortie générés (créés en utilisant la commande **Fichier | Enregistrer Fichiers générés**), le nom du document de sortie principal sera le nom que vous attribuez lorsque vous générez des fichiers de document de sortie utilisant la commande **Enregistrer fichiers générés**. Les noms des fichiers du document de sortie supplémentaire seront les noms attribués dans les URL des Nouveaux modèles de document.

Nouveaux modèles de document et structure du document source

Lorsqu'un Nouveau modèle de document est créé, l'emplacement hiérarchique où il est créé est important. Deux possibilités existent :

1. Le nœud à l'intérieur duquel le Nouveau modèle de document est créé est traité une seule fois. Dans ce cas, le Nouveau modèle de document est également traité une seule fois. Le filename dans la propriété de l'URL du Nouveau modèle de document peut être un nom statique pour cette raison.
2. Le nœud à l'intérieur duquel le Nouveau modèle de document est créé est traité plusieurs fois. En guise de résultat, le Nouveau modèle de document sera traité autant de fois que le nœud est traité. Un exemple d'une telle situation serait la suivante. Un élément `Office` a de multiples enfants d'élément `Department` (pour ses différents départements). Si un Nouveau modèle de document est créé à l'intérieur du nœud `Department` dans le design, alors, puisque le nœud `Department` sera traité de multiples fois (pour tous les différents éléments `Department` dans cet élément `Office`), le Nouveau modèle de document sera aussi traité de multiples fois, une fois pour chaque élément `Department` dans le document XML source. Le filename dans la propriété de l'URL du Nouveau modèle de document peut être un nom dynamiques pour cette raison. Autrement, les documents de sortie créés pour les éléments `Department` auront chacun le même filename.

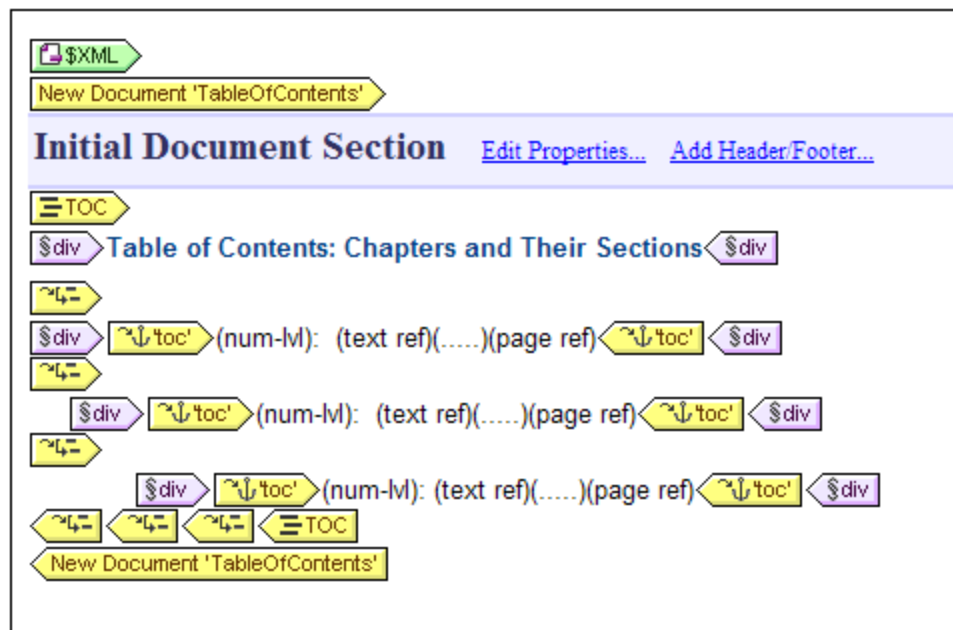
6.6.3 URL de nouveaux modèles de document

Dans cette section, nous décrivons comment les [URL des Nouveaux modèles de document portent sur la structure de design](#) ²⁵³, comment les [URL sont éditées](#) ²⁵⁵ et [comment de multiples documents de sortie peuvent être liés](#) ²⁵⁵ entre eux.

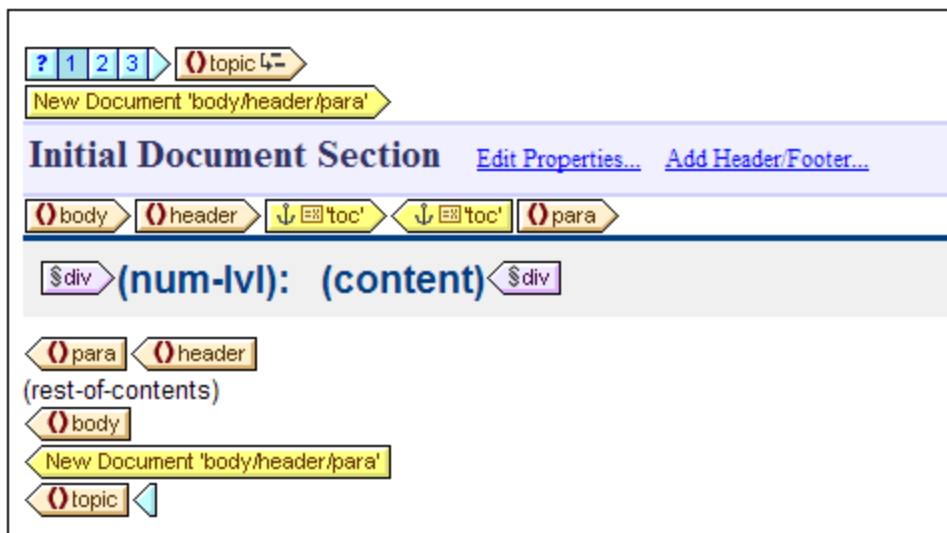
URL de Nouveaux modèles de document

Si le Nouveau modèle de document est traité une seule fois (voir [section précédente](#) ²⁵³), alors la propriété du modèle `URL` peut être un URL statique. Dans la capture d'écran ci-dessous, puisque le Nouveau modèle de

document est immédiatement à l'intérieur de l'élément du document (`$XML`), il sera traité une seule fois. L'URL a reçu une valeur statique de `TableOfContents`. Cette valeur sera donc le filename du document de sortie. Puisqu'aucun chemin n'a été préfixé au nom du fichier, le fichier sera généré dans le même répertoire que le fichier du document principal (voir [multiples sorties et aperçus de document](#)²⁵⁶ pour les détails). En alternative, si l'URL contenait un chemin, le document de sortie sera enregistré à l'emplacement spécifié dans le chemin.

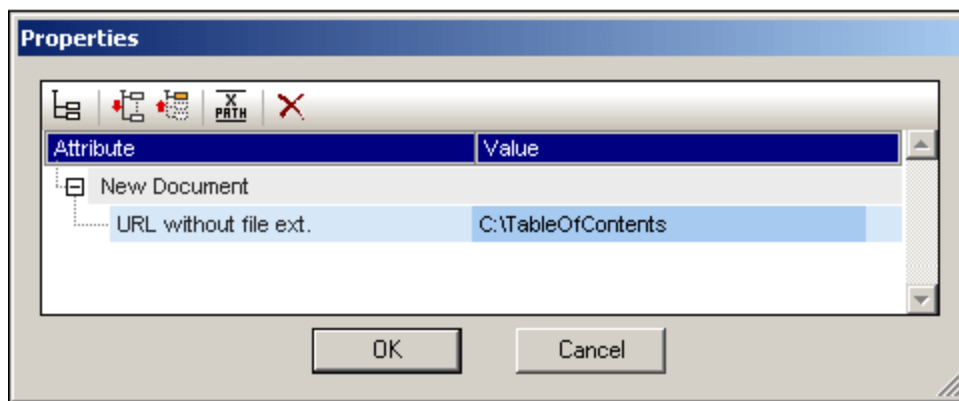


Si, d'un autre côté, un nouveau modèle de document sera traité de multiples fois pour générer des documents de sortie multiples (voir [section précédente](#) ²⁵³), alors la propriété `URL` doit être une URL dynamique qui est sélectionnée avec une expression XPath. Dans la capture d'écran ci-dessous, l'URL du nouveau modèle de document est l'expression XPath : `body/header/para`. Le nouveau modèle de document est à l'intérieur de l'élément `topic`, donc il sera traité à chaque fois que l'élément `topic` est traité. À chaque itération par le biais de l'élément `topic`, le contenu de l'élément `topic` de cet élément `body/header/para` sera attribué comme URL du nouveau modèle de document. Ceci crée un nouveau document pour chaque élément `topic`. Chacun de ces documents a un nom différent, celui de son élément `body/header/para` (qui est le texte de l'en-tête du chapitre).



Éditer l'URL

Lorsqu'un nouveau modèle de document est ajouté au design, il est créé avec un URL par défaut. Il s'agit d'un string de texte statique : `DocumentX` (là où `X` est un entier). Si vous souhaitez éditer l'URL, cliquez avec la touche de droite sur le nouveau modèle de document et sélectionnez la commande **Éditer l'URL**. Ceci ouvre le dialogue des Propriétés (*capture d'écran ci-dessous*), dans lequel vous pouvez éditer le champ de Valeur de la propriété *URL sans texte de fichier*.



Si vous souhaitez saisir une URL statique, éditez le champ de valeur pour contenir le texte URL requis. Si vous souhaitez saisir une URL dynamique, cliquez dans le champ Valeur, cliquez sur le bouton XPath dans la barre d'outils du dialogue des Propriétés et saisissez l'expression XPath que vous voulez. Veuillez noter les points suivants : (i) le nœud du contenu pour l'expression XPath est le nœud au sein duquel le nouveau modèle de document a été inséré. (ii) pour préfixer le chemin à l'expression de l'emplacement XPath, utilisez la fonction `concat()` de XPath. Par exemple : `concat('C:\MyOutput\ ', body\header\para)`. Cette expression d'exemple générera le string URL : `C:\MyOutput\filename`. L'extension du fichier approprié sera généré automatiquement selon le format de sortie.

Lier les documents

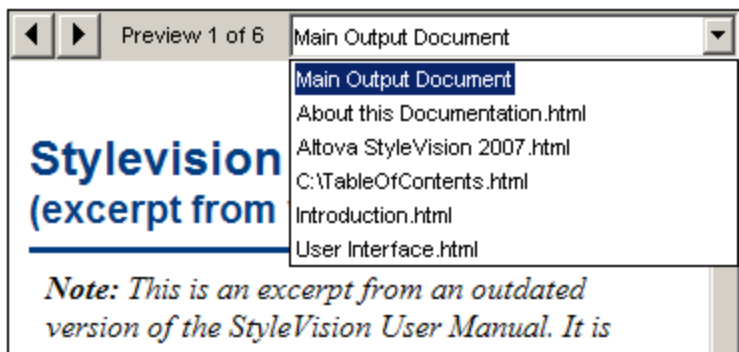
Les multiples documents de sortie peuvent être liés à un autre utilisant les [signets et hyperliens](#)³²¹. Un [signet](#)³²¹ peut être placé en haut du nouveau modèle de document ou partout à l'intérieur du nouveau modèle de document. Les [hyperliens](#)³²³ peuvent ensuite être créés dans d'autres documents pour établir le lien de

retour au signet. Si les signets sont requis sur un nœud qui est traité de multiples fois, assurez-vous que le signet est généré de manière dynamique. Autrement, (si un nom de signet statique est donné), de multiples nœuds dans la sortie auront le même nom de signet.

Une [Table des matières \(TOC\)](#)²⁹³ peut également être utilisée pour lier les documents. La TOC pourrait être dans un document séparé (par exemple, le document principal) et être liée à différents documents de sortie, tandis que les documents de sortie pourraient être liés de retour à la TOC.

6.6.4 Fichiers d'aperçu et fichiers de documents de sortie

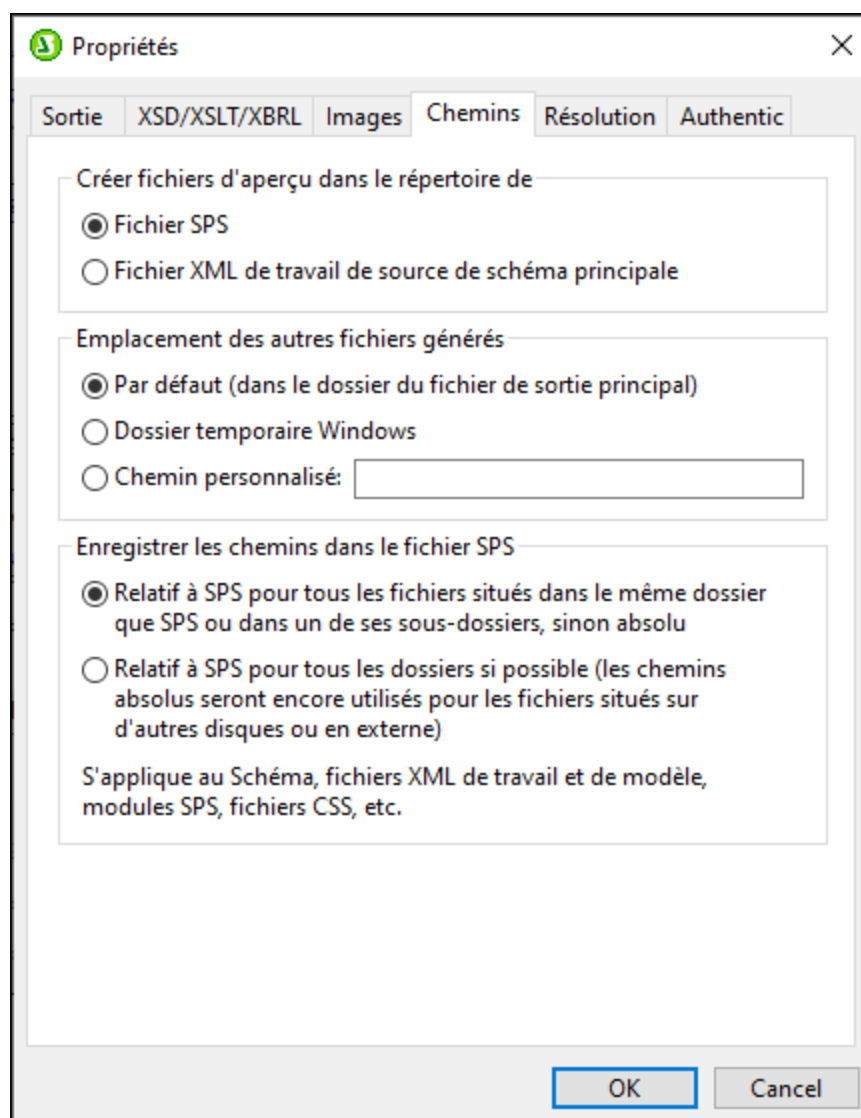
Les aperçus de sortie d'un document de design affichent chacun de multiples documents de sortie qui ont été spécifiés dans le design de documents séparés (*voir la capture d'écran ci-dessous*).



La capture d'écran ci-dessus affiche l'Aperçu HTML d'un document SPS qui a été conçu pour générer de multiples documents de sortie. Chaque document de sortie peut être appelé dans le fenêtre d'aperçu soit en : (i) naviguant à travers le document disponible utilisant les boutons de flèches en haut à gauche, ou en (ii) sélectionnant le document requis depuis la liste déroulante de la zone de liste déroulante (*voir la capture d'écran ci-dessus*). Notez que les items de la liste déroulante affichent l'URL entier (chemin plus filename).

Emplacement des fichiers d'aperçu

Les fichiers d'aperçu sont créés par défaut dans le répertoire dans lequel le fichier SPS est créé. Ce paramètre par défaut peut être changé dans l'onglet Chemins du dialogue des Propriétés du fichier SPS (*capture d'écran ci-dessous*), qui est accédé par la commande **Fichier | Propriétés**. Dans cet onglet, vous pouvez spécifier le répertoire du fichier de travail XML comme emplacement alternatif. Si l'URL d'un Nouveau modèle de document contient un chemin, l'emplacement spécifié dans ce chemin sera utilisé comme emplacement des fichiers d'aperçu respectifs. Si le serveur ne peut pas être trouvé, un message d'erreur s'affiche. Vous devriez savoir où les documents de sortie seront enregistrés si vous définissez les documents de sortie pour les mettre en lien.



Dans l'onglet Chemin du dialogue des Propriétés (voir la capture d'écran ci-dessus), vous pouvez aussi préciser à quel endroit les fichiers temporaires supplémentaires pour aperçus, tels que les fichiers du document de sortie, et les fichiers d'images et d'images de graphique seront enregistrés. Notez que si l'URL d'un Nouveau modèle de document contient un chemin, alors l'emplacement dans ce chemin sera utilisé.

Générer la sortie (chemins, etc.)

Pour générer les fichiers du document de sortie, procédez comme suit :

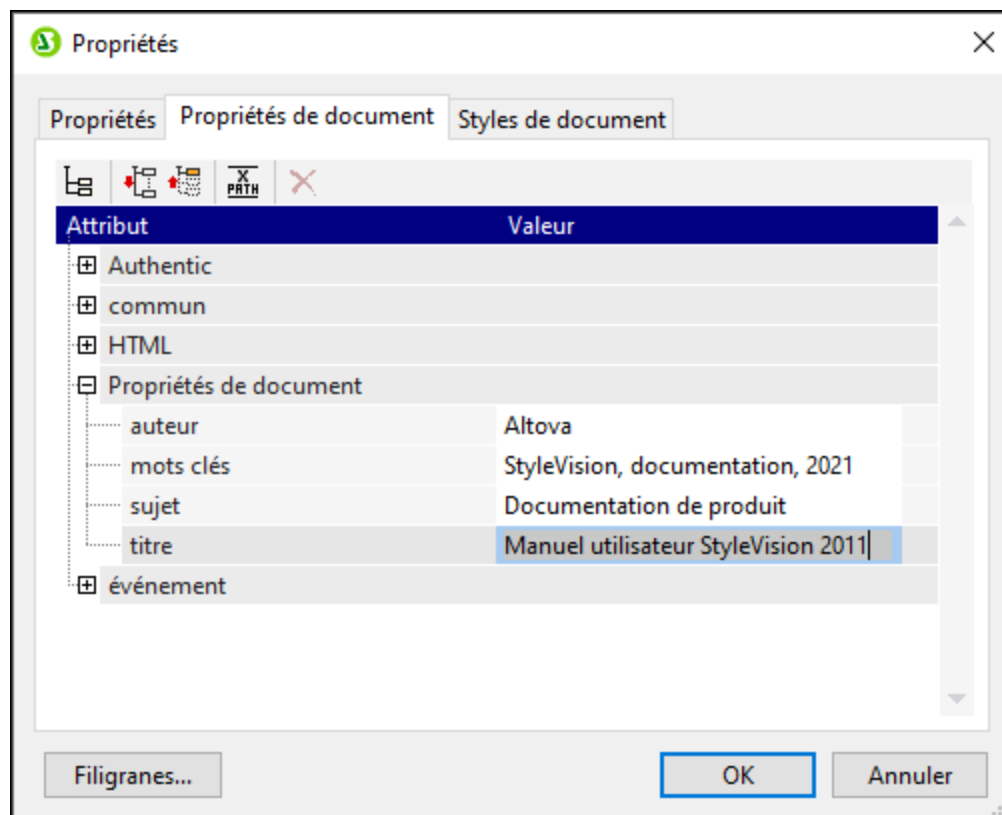
1. Passez au-dessus de la commande de menu **Fichier | Enregistrer les fichiers générés** et cliquez sur le format de sortie requis.
2. Dans le dialogue Enregistrer le fichier généré qui s'ouvre, recherchez le dossier dans lequel vous souhaitez enregistrer le fichier généré.
3. Saisissez le nom du fichier de document principal et cliquez sur **Enregistrer**.

L'emplacement de tous les fichiers du document de sortie ainsi que tous les autres fichiers générés de manière supplémentaire, tels que les fichiers d'images graphique, sera affiché dans une fenêtre pop-up pour votre information.

Le fichier du document principal sera enregistré à l'emplacement du dossier principal que vous avez sélectionné dans le dialogue Enregistrer fichier généré. Tous les multiples documents de sortie qui ont été créés avec les Nouveaux modèles de document et dont leurs URL n'ont pas d'information de chemin seront enregistrés dans le même dossier que le fichier du document principal. Si un chemin a été préfixé au filename de l'URL du Nouveau modèle de document, le document de sortie sera enregistré à l'emplacement précisé dans l'URL. Si l'emplacement du dossier n'existe pas, une erreur sera générée.

6.6.5 Propriétés du document et Styles

Dans un design SPS, vous pouvez partager la sortie en multiples documents. Chacun de ces documents peut se voir attribuer des propriétés de document et des styles de document séparés. Celles-ci sont spécifiées dans les onglets *Propriétés de document* et *Styles de document*, respectivement (voir *capture d'écran ci-dessous*), du dialogue des Propriétés de la section initiale du document. Pour accéder le dialogue des Propriétés, cliquez sur le lien *Éditer propriétés* dans la barre de titre de la section initiale du document pour laquelle vous souhaitez définir ces propriétés. Les propriétés du document et les styles du document s'appliquent à tout le document de sortie entier.



Dans l'onglet des Propriétés du document, le groupe des propriétés des Propriétés du document permettent qu'une méta-information soit saisie pour le document. Cette méta-information sera enregistrée dans le

document de sortie respectif et selon le format de sortie. Par exemple, dans le format de sortie HTML, les propriétés sont stockées dans les balises `META` respectives de l'élément `HEAD`.

Les styles du document sont décrits dans la section [Configurer des valeurs de propriété CSS](#)³⁵².

7 Fonctions avancées

Comment créer le contenu basique et la structure du design SPS est décrit dans les sections [Contenu du fichier SPS](#)¹⁰⁹ et [Structure du fichier SPS](#)¹⁸⁵. Très souvent vous serez toutefois obligé de modifier ou manipuler le contenu et/ou la structure des données source de manière particulière. Par exemple, vous allez éventuellement trier un groupe de nœuds, disons des nœuds contenant de l'information sur le personnel, selon un critère particulier, disons par ordre alphabétique des noms de famille des employés. Ou vous allez vouloir grouper tous les clients dans une base de données par ville. Ou ajouter le chiffre d'affaires d'une ville particulière. Une telle fonction est proposée dans les fonctions avancées de StyleVision, et elles sont dans cette section.

Ci-dessous, veuillez trouver une liste des fonctions avancées du fichier SPS de StyleVision :

- [Calculs automatiques](#)²⁶¹. Les calculs automatiques sont un mécanisme puissant basé sur XPath pour manipuler les données et (i) présenter les données manipulées dans la sortie ainsi que (ii) mettre à jour les nœuds dans le document XML avec le résultat du calcul automatique.
- [Conditions](#)²⁶⁶. Il s'agit du traitement des modèles et le contenu des modèles peut être conditionnel aux structures des données ou valeurs dans l'XML, ou au résultat d'une expression XPath
- [Groupement](#)²⁷¹. Le traitement peut être défini pour un groupe d'éléments qui sont sélectionnées dans une expression XPath.
- [Trier](#)²⁷⁹. Un ensemble d'éléments XML peut être trié sur de multiples clés de tri.
- [Paramètres et variables](#)²⁸⁵. Les paramètres sont déclarés au niveau SPS global avec une valeur par défaut. Ces valeurs peuvent être écrasées lors de l'exécution par les valeurs transmises à la feuille de style depuis la ligne de commande. Les variables peuvent être définies dans le SPS et ces variables peuvent être référencées pour utilisation dans le SPS.
- [Table des matières \(TOC\) et référencement](#)²⁹³. Les Tables des matières (TOC) peuvent être construites à divers emplacements dans la sortie du document, pour tous les formats de sortie. Le mécanisme TOC fonctionne en sélectionnant d'abord les items qui doivent être référencés dans la TOC, puis en référençant ces items marqués dans la table des matières. D'autres fonctions qui utilisent le référencement sont : (i) la [numérotation automatique](#)³¹⁶ (les nœuds répétitifs dans le document peuvent être numérotés automatiquement et les numéros formatés) ; (ii) les [références de texte](#)³¹⁹ (le texte dans le document peut être marqué pour référencement, puis être référencé d'ailleurs dans le document) ; et (iii) les [signets et hyperliens](#)³²¹ (les signets marquent les points clés dans le document de sortie, qui peuvent être ciblés par les hyperliens. Les hyperliens peuvent être mis en lien vers des ressources externes utilisant des méthodes pour déterminer l'URI cible (statique, dynamique, une combinaison des deux, et une entité non parsée d'URI).) Tous ces mécanismes de référencement sont décrits dans cette section.

7.1 Calcul automatique

La fonction **Auto-Calcul** (i) affiche le résultat d'une évaluation XPath à tout emplacement désiré dans le document de sortie, et (ii) en option, met à jour un nœud dans le document principal XML (le document XML étant édité dans Authentic View) avec le résultat de l'évaluation XPath.

La fonction du calcul automatique est un mécanisme utile pour :

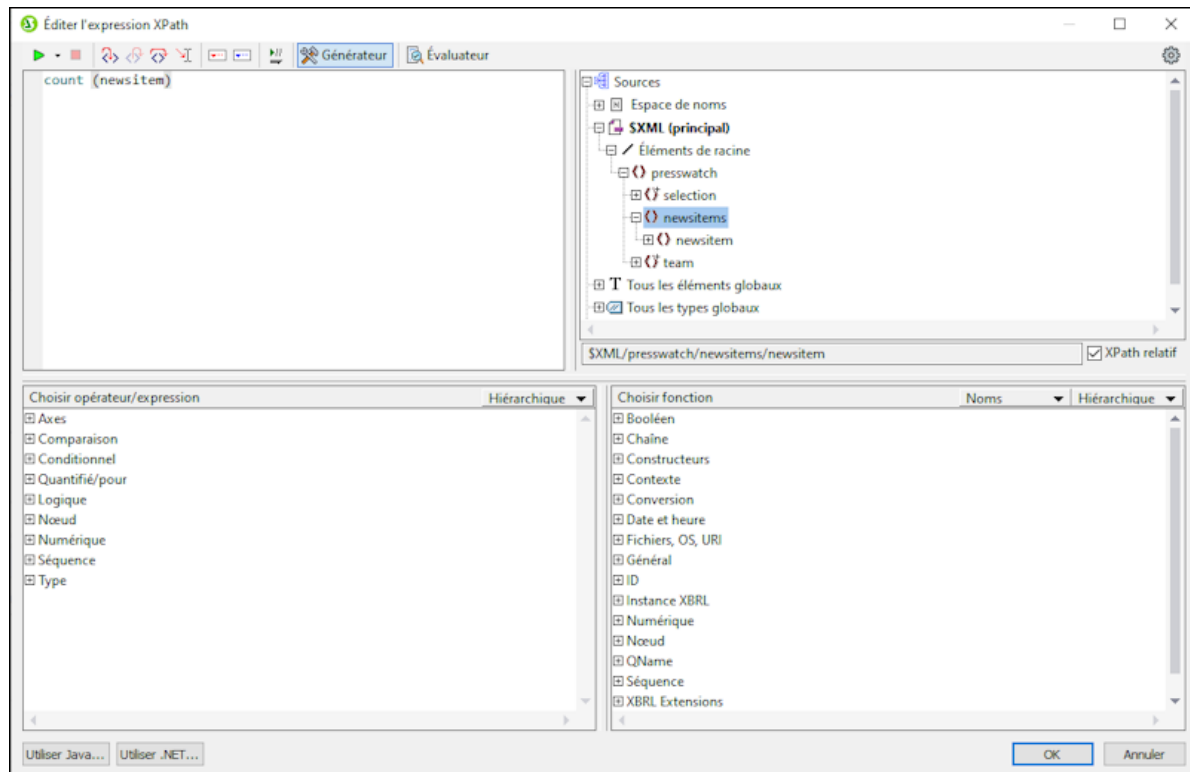
- Insérer les calculs impliquant les opérations sur les valeurs de données dynamiques. Par exemple, vous pouvez compter le nombre d'éléments `Employee` dans un élément `Office` (avec `count(Employee)`), ou la somme des valeurs de tous les éléments `Price` dans chaque élément `Invoice` (avec `sum(Price)`), ou joindre les éléments `FirstName` et `LastName` d'un élément `Person` (avec `concat(FirstName, ' ', LastName)`). De cette manière, vous pouvez générer de nouvelles données depuis des données changeant de manière dynamique dans le document XML, et envoyer les données générées à la sortie.
- Afficher l'information dérivée de la structure du document. Par exemple, vous pouvez utiliser la fonction `position()` de XPath pour insérer de manière dynamique des nombres de lignes dans une table dynamique, ou pour numéroté de manière dynamique les sections d'un document. Ceci a l'avantage de générer automatiquement l'information basée sur les structures du document changeant.
- Insérer des données depuis des documents XML externes. La fonction `doc()` de XPath 2.0 fournit l'accès à la racine du document des documents XML externes, permettant donc d'insérer du contenu de nœud des documents XML à insérer dans la sortie.
- Présentant les contenus d'un nœud à tout emplacement dans le design.

7.1.1 Éditer et déplacer des calculs automatiques

Créer des calculs automatiques

Pour créer un calcul automatique, procédez comme suit :

1. Placez le curseur comme un **point d'insertion** à l'emplacement où le résultat du calcul automatique est d'afficher et cliquez sur **Insérer | Calcul automatique**. Dans le sous-menu qui apparaît, sélectionnez Valeur si le résultat apparaît comme texte clair, sélectionnez Champ d'entrée s'il doit apparaître dans le champ d'entrée (par ex., dans la case de texte), ou sélectionnez Champ entrée multiligne s'il doit apparaître dans une case de texte multiligne. (Notez que la sortie du calcul automatique est affichée comme une valeur, ou un champ d'entrée. Il s'agit d'une sortie dans Authentic View, et ne peut pas être édité ici.) Le dialogue Éditer expression XPath s'ouvre (*capture d'écran ci-dessous*).

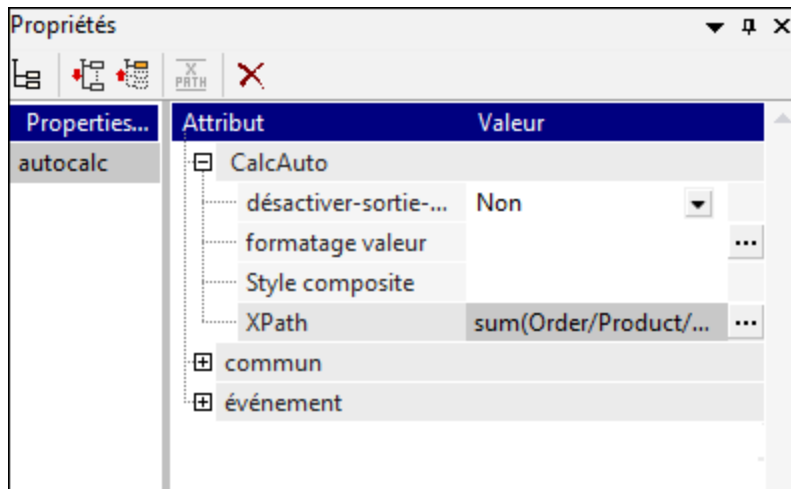


2. Dans le volet *Expression*, saisissez l'expression XPath pour le calcul automatique par le biais du clavier. De manière alternative, saisissez l'expression en double-cliquant sur les nœuds, opérateurs, et/ou fonctions dans les volets respectifs du dialogue. Il est important d'être conscient du nœud contextuel au point d'insertion ; le nœuds contextuels en surbrillance dans l'arborescence source de schéma (dans la capture d'écran ci-dessus, le nœud contextuel, `newsitems`, est en surbrillance). Si vous avez sélectionné XSLT 1.0 comme version du langage XSLT pour votre SPS, alors vous devez utiliser les expressions XPath 1.0 ; si vous avez sélectionné XSLT 2.0 ou XSLT 3.0, alors vous devez utiliser, respectivement, les expressions XPath 2.0 ou XPath 3.0. Pour une description détaillée du dialogue Éditer l'expression XPath, voir la section [Éditer l'expression XPath](#) ⁴²⁶.

Cliquez sur le bouton **OK** pour terminer. Dans l'onglet de Design, le symbole du calcul automatique est affiché. Pour voir le résultat du calcul automatique, changez à HTML.

Éditer des calculs automatiques

Pour éditer l'expression XPath de calcul automatique, sélectionnez le calcul automatique et, dans la barre latérale des Propriétés, cliquez sur le bouton **Éditer** de la propriété XPath dans le groupe des propriétés *AutoCalc* (capture d'écran ci-dessous). Ceci ouvre le dialogue [Éditer le dialogue de l'expression XPath](#) ⁴²⁶ (capture d'écran ci-dessus), dans lequel vous pouvez éditer l'expression XPath.



Formater des calculs automatiques

Vous pouvez appliquer des formats prédéfinis et des styles CSS aux calculs automatiques tout comme vous le feriez pour le texte normal : sélectionnez le calcul automatique et appliquez le formatage. De plus, le [formatage d'entrée](#) ³³⁴ d'un calcul automatique qui est de datatype numérique ou de date peut être précisé par la propriété de formatage de l'entrée dans le groupe des propriétés AutoCalc dans la fenêtre des propriétés.

Notez aussi que vous pouvez inclure les retours chariots et/ou les linefeeds (CR/LFs) dans l'expression XPath. Si le calcul automatique est enveloppé dans le type de paragraphe spécial `pre`, la sortie d'un CR/LF produira une nouvelle ligne dans la sortie. Un exemple d'une telle expression XPath est :

```
translate('a;b;c', ';', codepoints-to-string(13))
```

Déplacer des calculs automatiques

Vous pouvez déplacer un calcul automatique vers un autre emplacement en cliquant sur le calcul automatique (pour le sélectionner) et le déplacer vers le nouvel emplacement. Vous pouvez aussi utiliser couper/copier-et-coller vers déplacer/copier un calcul automatique. Notez, toutefois, que l'expression XPath aura besoin de changer si le nœud contextuel dans le nouvel emplacement n'est pas le même que dans l'emplacement précédent.

Résumé des points importants

Notez les points suivants :

- Un calcul automatique peut être inséré partout dans le document de design.
- Le point où vous insérez le calcul automatique détermine le nœud contextuel pour l'évaluation XPath.

7.1.2 Exemple : Un Invoice (une facture).

L'exemple SimpleInvoice.sps dans le [dossier \(Mes\) Documents](#) ²⁴, C:\Documents and Settings\\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\Auto-Calculations\, démontre comment les calculs automatiques peuvent être utilisés aux fins suivantes :

- Compter les nœuds
- Sélectionner un nœud basé sur l'entrée de l'utilisateur de Authentic View
- Créer des calculs complexes

Dans le fichier d'exemple, les calculs automatiques ont été mis en surbrillance ayant une couleur d'arrière-fond jaune (*voir la capture d'écran ci-dessous*).

Compter les nœuds

Dans l'exemple SimpleInvoice, chaque produit dans la liste est numéroté selon sa position dans la liste des produits qu'un client a trié (*Product 1, Product 2, etc.*). Cette numérotation est obtenue avec un calcul automatique (*capture d'écran ci-dessous*).

Product 1:	Learning XMLSpy
Net price:	€ 35.00
Category:	Book
VAT:	10%
Price including VAT:	€ 38.5

Product 2:	Scooby Doo's Greatest Hits
------------	----------------------------

Dans ce cas particulier, il suffirait que l'expression XPath `position()` obtienne la numérotation correcte. Une autre manière utile d'obtenir la position d'un nœud est de compter le nombre de frères qui précèdent et en ajouter un. L'expression XPath serait : `count(preceding-sibling::Product)+1`. Cette approche pourrait s'avérer utile dans des contextes où la fonction `position()` est utilisée est difficile à employer ou ne peut pas être utilisée. Vous pouvez tester le calcul automatique dans le fichier d'exemple en supprimant les produits, et/ou en ajoutant et supprimant les nouveaux produits.

Sélectionner un nœud basé sur l'entrée de l'utilisateur

Dans l'exemple SimpleInvoice, la catégorie de produit (*Book, CD, DVD ou Electronics*) est contenue dans le nœud `//Product/Category` et affichée dans une zone de liste. Cette sélection est saisie dans le nœud `//Product/Category` dans le document XML. Un calcul automatique utilise ensuite cette valeur pour référencer une « table lookup » dans le document XML et identifier le nœud ayant le pourcentage T.V.A. pour cette catégorie de produit. L'expression XPath du calcul automatique est :

```
for $i in Category return /Invoice/Categories/Category[. = $i]/@rate.
```

Le pourcentage T.V.A. est affiché dans l'emplacement du calcul automatique dans la sortie. Dans l'exemple Invoice, la table lookup est stockée dans le même document XML comme données de la facture. Toutefois, une telle table peut aussi être stockée dans un document séparé, dans ce cas, elle serait accédée utilisant la fonction `doc()` de XPath 2.0. Notez que la valeur T.V.A. de différents produits est différente (*Book=10%; CD=15%; DVD=15%; Electronics=20%*) ; elle a été calculée par le calcul automatique.

Créer des calculs automatiques complexes

Le pourcentage de T.V.A., obtenu dans le calcul automatique décrit ci-dessous, est requis pour calculer le prix brut (prix net + montant T.V.A.) de chaque produit. La formule à utiliser pourrait être calculée comme suit :

Prix brut = prix net + *montant T.V.A.*

Puisque *montant T.V.A.* = *Prix net * pourcentage T.V.A. div 100*

Prix brut = prix net + (*prix net * pourcentage T.V.A. div 100*)

Le prix net d'un produit est obtenu du nœud `PriceNet`. Le pourcentage T.V.A. est calculé par un calcul automatique tel que décrit ci-dessus, il n'est pas contenu dans un nœud. Depuis que cette valeur ne peut pas être obtenue directement d'un nœud, elle doit être recalculée dans le prix brut du calcul automatique. L'expression XPath qu'il vous faut est la suivante :

```
for $i in Category return PriceNet + (PriceNet * /Invoice/Categories/Category[. =
    $i]/@rate) div 100)
```

L'expression XPath peut être [consultée et éditée dans la fenêtre des Propriétés](#) ²⁶¹. Vous pouvez tester le calcul automatique pour le prix brut en changeant, dans le fichier XML, puis en rechargeant le SPS, soit le prix ou la catégorie de produit de tout produit. Notez que le prix brut (prix incluant la T.V.A.) du produit change également.

Product 6:	A Short History of the American Century
Net price:	€ 20.00
Category:	<input type="text" value="DVD"/>
VAT:	15%
Price including VAT:	€ 23

7.2 Conditions

Vous pouvez insérer les conditions partout dans le design, dans le modèle principal et les modèles globaux. Une condition est un composant SPS qui est constitué d'une ou plusieurs branches, chaque branche étant définie par une expression XPath. Par exemple, considérez une condition composée de deux branches. L'expression XPath de la première branche teste si la valeur de l'attribut de l'emplacement du nœud contextuel est « US ». L'expression XPath de la deuxième branche teste si la valeur de l'attribut Emplacement est « EU ». Chaque branche contient un modèle —un modèle de condition. Quand un nœud est traité avec une condition, la première branche avec un test qui évalue à true est exécutée, à savoir, son modèle de condition est traité, et la condition est quittée ; aucune autre branche de cette condition ne sera évaluée. De cette manière, vous pouvez utiliser différents modèles dépendant de la valeur du nœud. Dans cet exemple juste cité, différents modèles pourraient être utilisés pour les emplacements US et UE.

Cette section est constituée des rubriques suivantes :

- [Définir des conditions](#)²⁶⁶, qui décrivent comment créer une condition et ses branches.
- [Éditer les conditions](#)²⁶⁹, à savoir comment éditer les expressions XPath des branches de condition après avoir été créées.
- [Conditions et calculs automatiques](#)²⁷⁰, explique les questions d'utilisation quand les conditions et les calculs automatiques sont utilisés en combinaison.

7.2.1 Définir les conditions

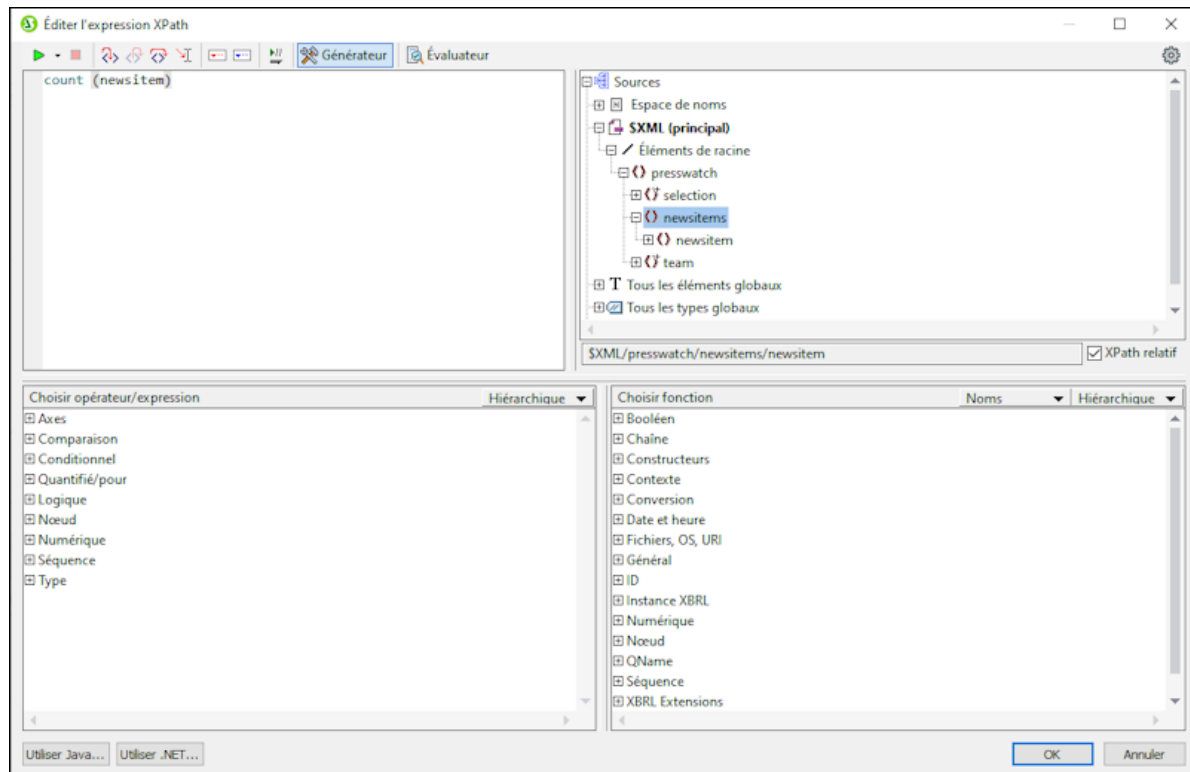
Définir la condition consiste en les étapes suivantes :

1. Créer la condition avec sa première branche.
2. Créer les branches supplémentaires pour le traitement alternatif.
3. Créer et éditer les modèles au sein des différentes branches de la condition.

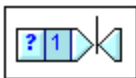
Créer la condition avec sa première branche

Configurer une condition comme suit :

1. Placez le curseur partout dans le design ou sélectionnez un composant, puis sélectionnez la commande de menu **Insérer | Condition**. Le [dialogue Éditer l'expression XPath](#)⁴²⁶ s'ouvre (*capture d'écran ci-dessous*).



2. Dans le volet Expression, saisissez l'expression XPath pour le « Condition Branch » par le biais du clavier. De manière alternative, saisissez l'expression en double-cliquant sur les nœuds, opérateurs, et/ou fonctions dans les volets du dialogue. Il est important d'être conscient du nœud contextuel au point d'insertion ; le nœud contextuel en surbrillance dans l'arborescence source de schéma s'ouvre.
3. Cliquez sur **OK** pour terminer. La condition est créée avec sa première branche ; l'expression XPath que vous avez saisie est l'expression XPath de la première branche. Si la condition a été insérée au point d'insertion du texte, la première branche est vide (il n'existe pas de modèle à l'intérieur ; voir la capture d'écran ci-dessous). Si la condition a été insérée avec un composant sélectionné, la condition est créée autour du composant, et le composant devient le modèle de la première branche.

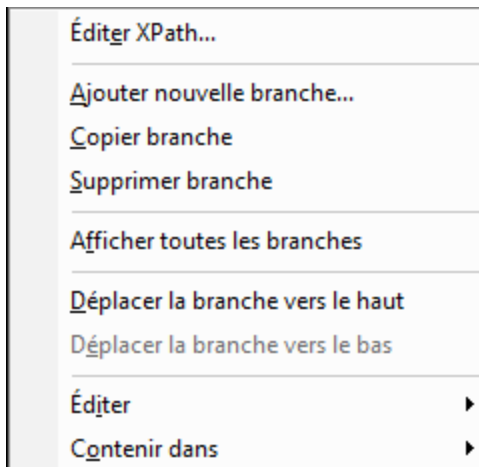


Pour sélectionner toute la condition, cliquez sur la cellule avec le point d'interrogation. Pour sélectionner toute la condition, cliquez sur la cellule avec le numéro un.

Après avoir créé une condition avec une branche (qui peut ou ne peut pas avoir un modèle à l'intérieur), vous pouvez créer autant de branches supplémentaires que requis.

Créer des branches supplémentaires

Les branches supplémentaires sont créées une à la fois. Une branche supplémentaire est créée via le menu contextuel (capture d'écran ci-dessous) et peut être créée de deux manières : (i) sans modèle à l'intérieur (**Ajouter nouvelle branche**); et (ii) avec une copie du modèle existant à l'intérieur de la nouvelle branche (**Copier branche**).



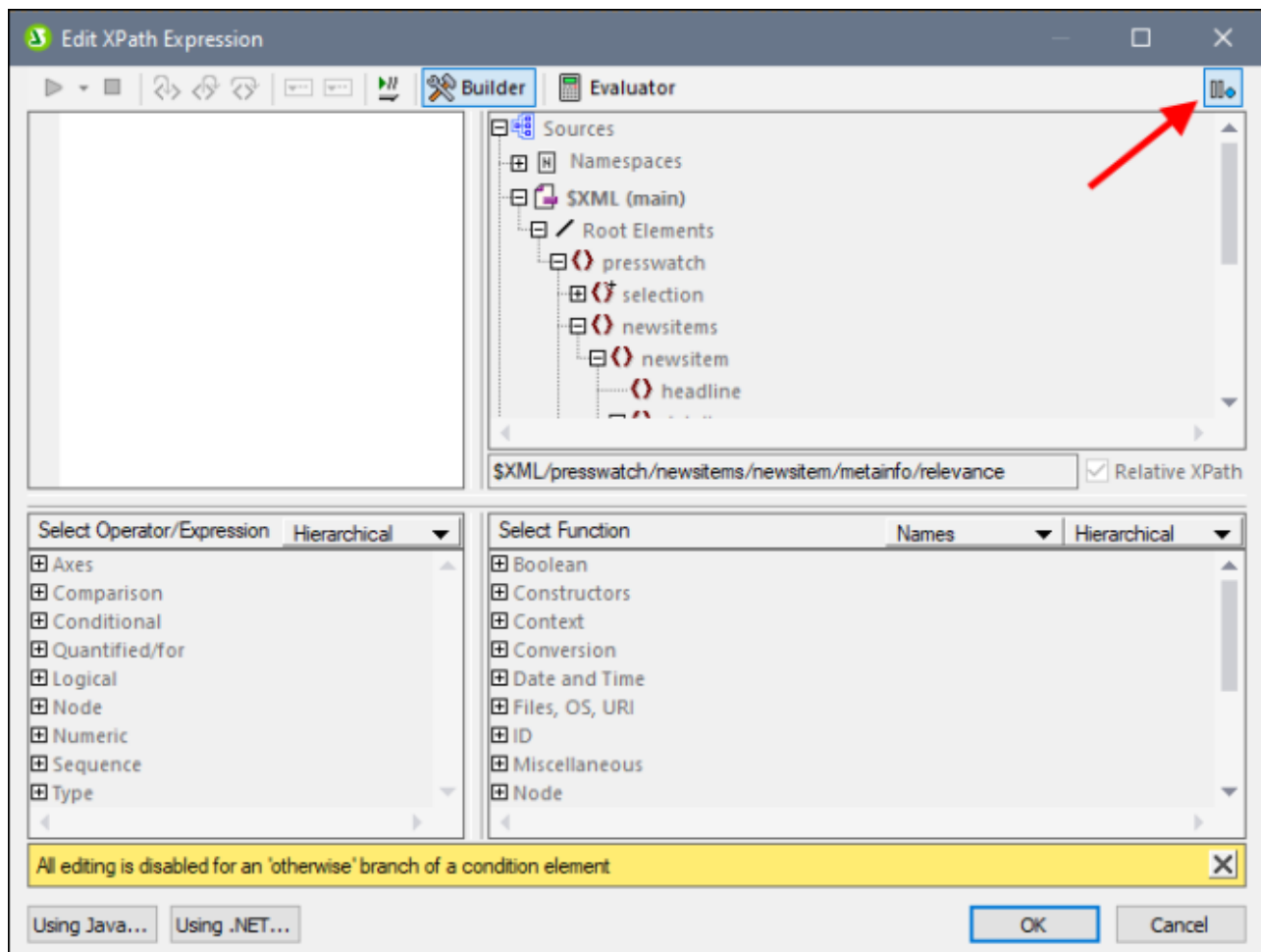
Pour créer une nouvelle branche, cliquez avec la touche de droite dans toute branche de la condition et sélectionnez **Ajouter nouvelle branche** depuis le menu contextuel. Le [dialogue Éditer l'expression XPath](#)⁴²⁶ s'ouvrira. Après avoir saisi une expression XPath et cliqué sur **OK**, une nouvelle branche vide est ajoutée à la condition. Ceci est indiqué dans le design par une nouvelle cellule étant ajoutée à la condition ; la nouvelle cellule a un nombre incrémenté par un sur la dernière branche avant l'ajout.

Pour créer une copie d'une branche existante, cliquez avec la touche de droite sur la branche de la condition que vous souhaitez copier et sélectionner **Copier branche**. Le [dialogue Éditer l'expression XPath](#)⁴²⁶ s'ouvrira, contenant l'expression XPath de la branche étant copiée. Après avoir modifier l'expression XPath et cliqué sur **OK**, une nouvelle branche est ajoutée à la condition. La nouvelle branche contient une copie du modèle de la branche qui a été copiée. La nouvelle branche est indiquée dans le design par une nouvelle cellule avec un nombre incrémenté par un sur la dernière branche avant l'ajout.

La branche Otherwise

La branche *Otherwise* est une branche alternative catch-all pour spécifier un certain type de traitement (modèle) dans le cas où aucune des branches ne sont définies à `true`. Sans la branche *Otherwise*, vous allez soit devoir créer des branches pour toutes les éventualités possibles, soit être préparées à la possibilité que le modèle conditionnel est quitté sans qu'une branche ne soit exécutée.

Pour insérer une branche *Otherwise*, utilisez soit les commandes **Ajouter nouvelle branche** ou **Copy Branch**, tel que décrit ci-dessus, et dans le [dialogue Éditer expression XPath](#)⁴²⁶, cliquez sur la case à cocher *Otherwise* (voir la capture d'écran ci-dessous).



Déplacer les branches vers le haut et vers le bas

L'ordre des branches dans la condition est important, car la première branche à évaluer à true est exécutée et la condition est ensuite quittée. Pour déplacer les branches vers le haut et vers le bas relatives les unes par rapport aux autres, sélectionnez la branche à déplacer, puis cliquez avec la touche de droite et sélectionnez **Déplacer vers le haut** ou **Déplacer vers le bas**.

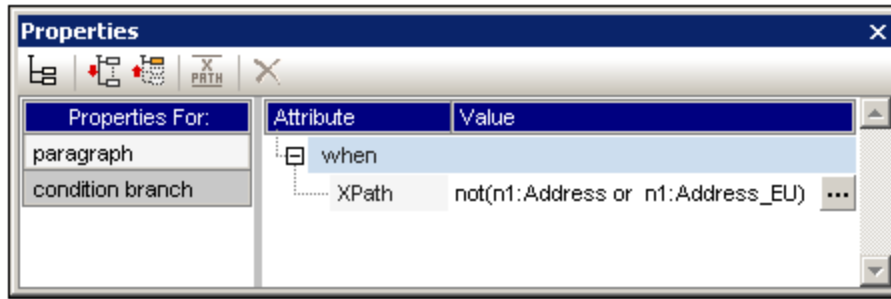
Supprimer une branche

Pour supprimer une branche, sélectionnez la branche à supprimer, puis cliquez avec la touche de droite et sélectionnez **Supprimer branche**.

7.2.2 Éditer les conditions

Pour éditer l'expression XPath d'une branche de condition, suivez les étapes suivantes :

1. Sélectionner la branche de condition (pas la condition).
2. Dans la barre latérale des Propriétés, sélectionnez `branche de condition` dans les Propriétés Pour colonne (*capture d'écran ci-dessous*).



3. Cliquez sur le bouton **Éditer**  de la propriété `XPath` dans le groupe des propriétés *When*. Ceci fait que le [dialogue Éditer l'expression XPath](#) ⁴²⁶ s'ouvre, dans lequel vous pouvez éditer l'expression XPath pour cette branche de la condition.

7.2.3 Conditions et calculs automatiques

Lorsque vous utilisez les conditions et calculs automatiques, il y a quelques questions dont il faut tenir compte. Les deux points fondamentaux à retenir sont :

- Seuls les calculs automatiques **dans des conditions visibles**—qui sont la branche sélectionnée comme true—qui sont évalués.
- Les calculs automatiques sont évalués avant les conditions.

Ici, vous trouverez quelques orientations pour résumer ces questions.

1. Si un calcul automatique met à jour un nœud, et si le nœud est impliqué dans une condition (soit en étant une expression XPath d'une branche ou dans le contenu du modèle conditionnel), ensuite gardez le calcul automatique à l'extérieur de la condition, si possible. Ceci permet au calcul automatique de toujours rester visible—peu importe quelle branche de la condition est visible. Si le calcul automatique était à l'intérieur d'une branche qui n'est pas visible, il ne serait pas déclenché.
2. Si un calcul automatique doit être placé à l'intérieur d'une condition, assurez-vous (i) qu'il soit placé dans une branche de la condition, et (ii) que les diverses branches de la condition couvrent toutes les conditions possibles. Il ne devrait pas y avoir d'éventualité qui ne soit pas couverte par une condition dans le modèle conditionnel ; autrement, il y a un risque (si le calcul automatique ne se trouve pas dans un modèle visible) que le calcul automatique ne soit pas déclenché.
3. Si vous avez besoin de différents calculs automatiques pour différentes conditions, assurez-vous que toutes les éventualités possibles pour chaque calcul automatique soit couvertes.
4. Souvenez-vous que l'ordre selon lequel les conditions sont définies dans un modèle conditionnel est significatif. La première condition à évaluer comme true est exécutée. La condition `otherwise` est une catch-all pratique pour les éventualités non spécifiques.

7.3 Regroupement

La fonctionnalité du regroupement est disponible dans les SPS **XSLT 2.0 et 3.0** et pour la sortie HTML.

Le regroupement permet aux items (généralement des nœuds) d'être traités en groupes. Par exemple, considérez un inventaire de voitures, dans lequel les détails de chaque voiture se trouvent sous un élément `car`. Si, par exemple, l'élément `car` a un attribut `brand`, alors les voitures peuvent être regroupées par marque. Ceci peut être utile pour une multitude de raisons. Par exemple :

- Toutes les voitures d'une seule marque peuvent être présentées ensemble dans la sortie, sous l'en-tête de son nom de marque.
- Les opérations peuvent être effectuées à l'intérieur d'un groupe et les résultats de cette opération présentés séparément pour chaque voiture. Par exemple, le nombre de modèles disponibles pour chaque marque peut être recensé.

De plus, un groupe peut ensuite être traité davantage dans les sous-groupes. Par exemple, à l'intérieur de chaque marque, les voitures peuvent être groupées par modèle et par année.

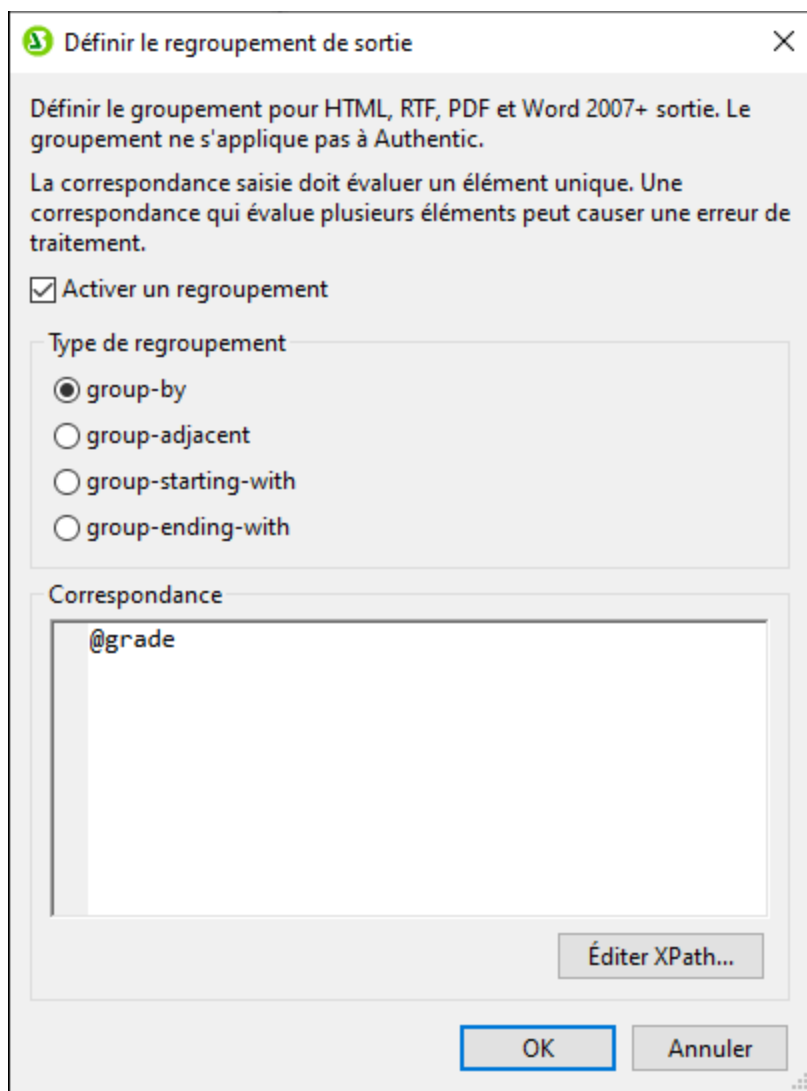
Critères de regroupement

Les items peuvent être regroupés utilisant deux critères généraux : (i) une clé de regroupement, qui teste généralement la valeur d'un nœud, et (ii) la position relative des items. Les critères de regroupement spécifiques suivants sont disponibles :

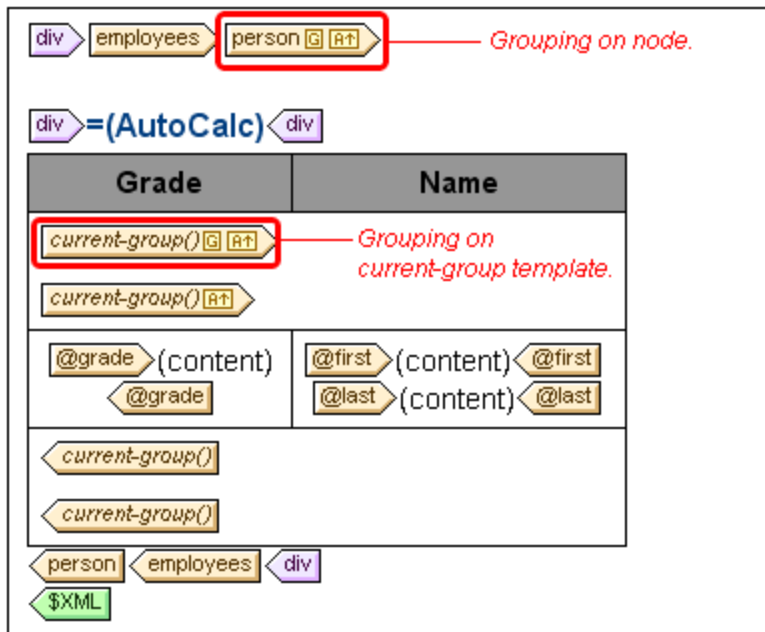
- **group-by**, qui regroupe les items sur la base d'une clé XPath définie. Par exemple, les éléments `car` peuvent être regroupés sur la base de leurs attributs `brand`. Le regroupement est défini sur l'élément `car`, et une expression XPath sélectionne l'attribut `brand`.
- **group-adjacent** utilise une combinaison de `grouping-key` et de critères de position. Tous les items adjacents qui ont la même valeur pour la clé de groupement sont inclus dans un groupe. Si la valeur de `grouping-key` d'un item est différent de l'item précédent, alors cet item commence un nouveau groupe.
- **group-starting-with** lance un nouveau groupe quand un nœud correspond à un modèle XPath défini. Si un nœud ne correspond pas à un modèle XPath défini, alors il est attribué au groupe actuel.
- **group-ending-with** termine un groupe quand un nœud correspond à un modèle XPath défini ; le nœud correspondant est le dernier dans ce groupe. Le prochain nœud lance un nouveau groupe. Si un nœud suivant à celui qui lance un groupe ne correspond pas à un modèle XPath défini, il est attribué au groupe actuel.

Créer des groupes

Les groupes peuvent être créés soit sur un nœud, soit sur un modèle de `current-group` via le menu contextuel. Pour créer un groupe, cliquez avec la touche de droite sur le nœud ou modèle de `current-group`, et dans le menu contextuel qui apparaît, sélectionnez la commande **Group by**. Ceci fait apparaître le dialogue Définir regroupement de sortie (*capture d'écran ci-dessous*).



Dans le dialogue, vérifiez la case à cocher Activer la case à cocher du regroupement, sélectionnez ensuite le Type de regroupement et, dans la case de texte Correspondre, saisissez l'expression XPath qui définit la clé de regroupement (pour les options *group-by* et *group-adjacent*) ou le modèle de correspondance désiré (pour les options *group-starting-with* and *group-ending-with*). Quand vous cliquez sur **OK**, un dialogue s'ouvre vous demandant si vous souhaitez trier le group-set par ordre alphabétique (par ordre ascendant). Vous pouvez toujours trier les group-sets par la suite ou supprimer un tel tri par la suite. La capture d'écran ci-dessous affiche les nœuds et les modèles du current-group qui ont eu des regroupements qui ont été ajoutés à ceux-ci.



Dans la capture d'écran ci-dessus, le nœud `person` a été regroupé et les groupes résultant triés. Par exemple, si les éléments `person` ont été regroupés par département, alors les divers départements peuvent être triés par ordre alphabétique ascendant. Les groupes ainsi créés ont été regroupés ainsi davantage en créant le regroupement dans le modèle `current-group()`. De cette manière, les éléments `person` peuvent être groupés, disons, d'abord par le département, et ensuite par le niveau d'emploi.

Trier les groupes

Après avoir défini une définition de regroupement, une fenêtre pop-up vous demandera de confirmer si les groupes doivent être triés par ordre ascendant ou non. Vous pouvez définir le tri par la suite à tout moment, ou modifier ou supprimer, à tout moment, l'ensemble de tri à cette étape.

Pour définir, modifier ou supprimer le tri par la suite, cliquez avec la touche de droite sur le modèle de regroupement requis et sélectionnez **Sort by**. Ceci ouvre le [dialogue Définir l'ordre du tri de sortie](#)²⁷⁹. Pour savoir comment utiliser ce dialogue, veuillez consulter la section [Utilisateurs](#)²⁷⁹. Le point important à noter est que pour trier les groupes sur la base de leur grouping-key, vous devez sélectionner la fonction XPath `current-grouping-key()` comme clé de tri. Par exemple, voir les fichiers décrits dans les sections suivantes.

Consulter et éditer les paramètres de regroupement et de tri

Pour consulter et éditer les paramètres de regroupement et de tri sur un modèle, cliquez avec la touche de droite sur le modèle et sélectionnez **Group-by** ou **Sort by**, respectivement. Ceci ouvre le dialogue respectif, dans lequel les paramètres peuvent être consultés ou modifiés.

Modèles définis par l'utilisateur

Les [modèles définis par l'utilisateur](#)²³⁹ sont des modèles qui sont appliqués à des items sélectionnés par une expression XPath que vous spécifiez. Les nœuds sélectionnés par l'expression XPath d'un modèle utilisé par l'utilisateur peut aussi être regroupé. Dans ce cas, le regroupement est appliqué dans le modèle défini par l'utilisateur.

7.3.1 Exemple: Group-By (Persons.sps)

L'exemple `Persons.sps` est basé sur le schéma `Persons.xsd` et utilise `Persons.xml` comme son fichier de travail XML. Il est situé dans le [dossier \(Mes\) Documents](#)²⁴, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\Grouping\Persons\`. La structure de document XML est comme suit : un élément de document `employees` peut contenir un nombre illimité d'employés `person`. Chaque employé `person` est structuré selon cet exemple :

```
<person first="Vernon" last="Callaby" department="Administration" grade="C"/>
```

Dans le design, nous regroupons des personnes selon le département. Chaque département est représenté par une table séparée et les départements sont triés par ordre alphabétique ascendant. À l'intérieur de chaque table de département, les personnes sont regroupées selon le grade (trié par ordre alphabétique ascendant) et, à l'intérieur de chaque grade, les personnes sont recensées par ordre alphabétique ascendant de leurs noms de famille.

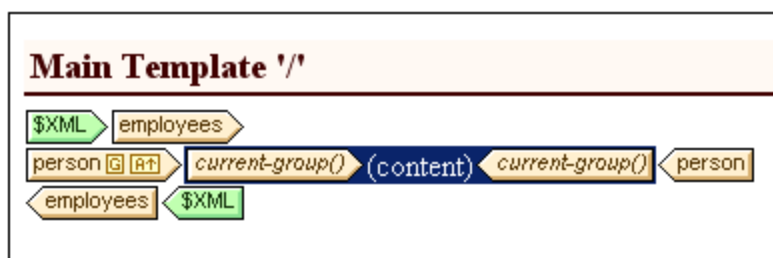
Stratégie

La stratégie pour créer les groupes est la suivante. Le regroupement est créé dans l'élément `person` avec l'attribut du département étant le `grouping-key`. Ceci fait que l'élément `person` à trier dans les groupes basés sur la valeur de l'attribut du département. (Si le tri est précisé, alors les groupes de départements peuvent être organisés par ordre alphabétique, par exemple, d'abord l'Administration, etc.) Puisque les départements doivent être créés dans des tables séparées, le `current-grouping` (qui est basé sur le clé `grouping-key` du département) est créé comme table. Maintenant, à l'intérieur de cet ordre groupé des éléments `Person`, nous précisons que chaque groupe doit être trié davantage avec l'attribut `grade` comme `grouping-key`.

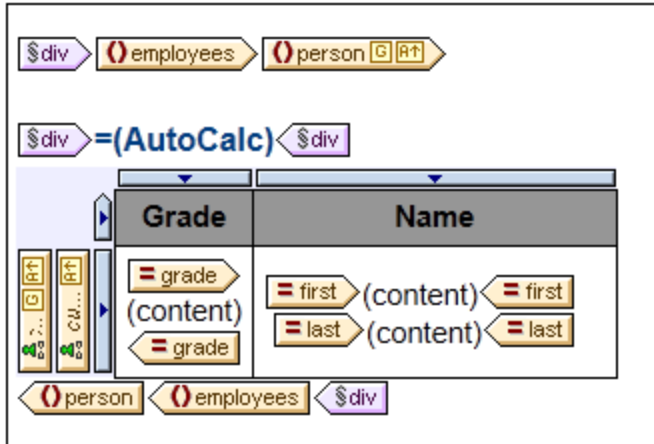
Créer SPS

Le design a été créé comme suit :

1. Glissez l'élément `person` depuis l'arborescence de schéma et créez-le comme son contenu.
2. Cliquez avec la touche de droite sur la balise de l'élément `person` et, dans le menu contextuel, sélectionnez `Group by`.
3. Dans le dialogue Définir regroupement de sortie, sélectionnez `group-by`, configurez l'expression XPath dans la case de texte Correspondre `@department`, puis cliquez sur **Yes**.
4. Un dialogue apparaît demandant si les groupes devraient être triés. Puisque nous souhaitons que les groupes soient triés selon l'ordre alphabétique ascendant par défaut, cliquez sur **OK**. (Le tri peut toujours être défini, modifié ou supprimé par la suite.)
5. Puisque chaque groupe (qui est un département) doit être créé dans une table séparée, créez le groupe actuel comme table. Faites ceci avec clic droit de la balise `current-group()` (*capture d'écran ci-dessous*), et sélectionnez **Change to | Table**, sélectionnant les attributs enfant `@last` and `@grade` comme colonnes de la table.



6. Réorganiser les contenus des colonnes et cellules de la table pour que la première colonne contienne @grade et la deuxième colonne contienne les nœuds @first et @last (voir la capture d'écran ci-dessous).
7. À l'intérieur du groupe actuel, qui est groupé par département, nous souhaitons regrouper par grade. Donc dans le modèle `current-group()`, créez un regroupement pour l'attribut `grade`. Confirmer le tri par défaut. Un nouveau modèle `current-group()` est créé (voir capture d'écran ci-dessous).
8. Trier le groupe actuel (qui est le sous-groupe de persons et trié par grade), dans le dernier attribut `last`.



9. Définir le formatage pour la table.
10. Au-dessus, la table fournir une en-tête pour la table. Puisque chaque table représente un département, le nom du département peut être obtenu de manière dynamique depuis le contexte actuel en utilisant un calcul automatique avec l'expression XPath qui appelle la fonction `current-grouping-key()` de XPath 2.0/3.0.
11. Répétez tout le processus, pour créer la sortie semblable, mais cette fois-ci, les grouping persons par grade, puis par département.

Pour consulter ou modifier le regroupement ou le tri d'un modèle, cliquez avec la touche de droite dans le modèle et sélectionnez **Group by** ou **Sort by** depuis le menu contextuel. Ceci ouvre le dialogue respectif, dans lequel les paramètres peuvent être consultés ou modifiés.

7.3.2 Exemple: Group-By (Scores.sps)

L'exemple `Scores.sps` est basé sur le schéma `Scores.xsd` et utilise `Scores.xml` comme son fichier de travail XML. Il est situé dans le [dossier \(Mes\) Documents](#) ²⁴, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\Grouping\Scores\`. La structure du document XML est comme suit : l'élément de document `results` contient un ou plusieurs éléments `group` et un ou plusieurs éléments `match`. Un élément `group` contient un ou plusieurs éléments `team`, et un élément `match` est structuré selon cet exemple :

```
<match group="A" date="2007-10-12">
  <team name="Brazil" for="2" points="3"/>
  <team name="Germany" for="1" points="0"/>
</match>
```

Le design consiste en trois parties (*capture d'écran ci-dessous*) : (i) les résultats match présentés par jour ((groupés sur //match/@date); (ii) les résultats match présentés par groupe (groupés sur //match/@group); et (iii) tables dde groupes fournissant un aperçu des positions par groupe (une table dynamique de l'élément du groupe, avec des calculs automatiques pour calculer les données requises).

Match Results: Day-by-Day

2007-10-12

Brazil - Germany 2 - 1
Italy - Holland 2 - 2

2007-10-13

Argentina - France 2 - 0
England - Spain 0 - 0

Match Results: By Group

Group A

Brazil - Germany 2 - 1
Italy - Holland 2 - 2
Brazil - Italy 1 - 2
Germany - Holland 2 - 2
Brazil - Holland 1 - 0
Germany - Italy 1 - 1

Group Tables

Group A

Team	P	W	D	L	F	A	Pts
Brazil	3	2	0	1	4	3	6
Italy	3	1	2	0	5	4	5
Germany	3	0	2	1	4	5	2
Holland	3	0	2	1	4	5	2

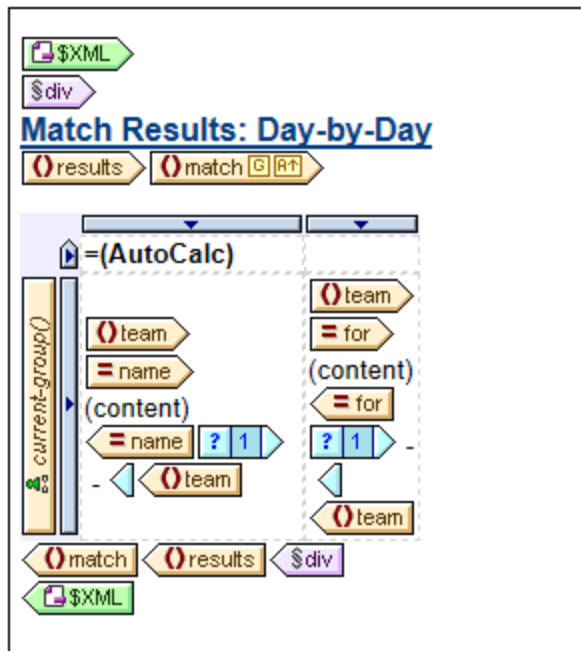
Stratégie

Pour les deux sections contenant les résultats match, nous regroupons les correspondances par date et tournament-group. Pour les membres de chaque groupe (groupe date et tournament), nous créons des tables sans bords (à des fins d'alignement). Donc les correspondances lues sur une seule date seront dans une table séparée, et les résultats d'une correspondance d'un seul groupe tournament seront dans une table séparée (par exemple, correspondances du Groupe A). Pour la section des group-tables, l'élément `group` est créé comme table dynamique, avec des calculs automatiques fournissant la valeur des données requises.

Créer SPS

Le design a été créé comme suit :

1. Glissez l'élément `/results/match` depuis l'arborescence de schéma et créez-le comme son contenu.
2. Cliquez avec la touche de droite sur la balise de l'élément `match` et, dans le menu contextuel, sélectionnez **Group by**.
3. Dans le dialogue Définir regroupement de sortie, sélectionnez *group-by*, configurez l'expression XPath dans la case de texte Match à `@ date`, puis cliquez sur **OK**.
4. Un dialogue apparaît demandant si les groupes devraient être triés. Puisque nous souhaitons que les groupes soient triés selon l'ordre alphabétique ascendant par défaut, cliquez sur **Oui**. (Le tri peut toujours être défini, modifié ou supprimé par la suite.)
5. Puisque chaque groupe (qui est une date) doit être créé dans une table séparée, créez le groupe actuel comme table. À cette fin, cliquez avec la touche de droite sur la balise `current-group()`, sélectionnant **Change to | Table**, puis en sélectionnant les nœuds descendants `team/@name` et `team/@for` comme colonnes de la table (voir la capture d'écran ci-dessous).



6. Définir un tiret dans chaque cellule qui sera la sortie si la correspondance n'est pas la dernière dans le groupe actuel. Faites-le en utilisant le modèle conditionnel avec une condition définie comme `position() != last()`. Ceci fournit une sortie telle que : Brazil - Germany ou 2 - 1.

7. Mettez un calcul automatique dans l'en-tête qui sort la grouping-clé pour le groupe respectif pour le groupe respectif (expression XPath: `current-grouping-key()`).
8. Formater la table telle que requise.
9. Pour grouper les correspondances par groupe tournament, répétez le processus entier, mais grouper les correspondances cette fois-ci dans l'attribut `group de match`.
10. Pour les tables du groupe (dans la troisième section du design), qui contiennent les niveaux de chaque équipe dans le groupe, créez l'élément `/results/group` comme table dynamique. Ajoutez des colonnes tel que requis (utilisant les commandes **Table | Ajouter Colonne** ou **Table | Insérer Colonne**). Définir les calculs automatiques dans chaque colonne pour calculer la sortie requise (3 points pour un win; 1 point pour un draw; 0 point pour un loss). Puis, finalement, triez la table par ordre descendant de points totaux obtenus. Pour voir les expressions XPath utilisées pour obtenir ces résultats, cliquez avec la touche de droite sur le calcul automatique ou sur le modèle trié, et sélectionnez, respectivement, les commandes **Éditer XPath** et **Trier par**.

7.4 Trier

La fonction de tri est disponible pour HTML.

Un ensemble de nœuds d'élément frère du même nom qualifié peut être trié sur une ou plusieurs sort-keys que vous sélectionnez. Par exemple, tous les éléments `Person` (à l'intérieur, disons, l'élément `Company`) peuvent être triés sur l'élément enfant `LastName` de l'élément `Person`. La sort-key doit être un nœud dans le document, et est typiquement un nœud descendant (élément ou attribut) du nœud de l'élément étant trié. Dans l'exemple mentionné, `LastName` est la sort-key.

S'il existe deux éléments dans l'ensemble soumis pour le tri qui ont des nœuds sort-key avec la même valeur, alors la sort-key supplémentaire pourrait un tri supplémentaire. Dans l'exemple `Person` juste cité, en plus de la première sort-key de `LastName`, une deuxième sort-key de `FirstName` pourrait être spécifiée. Donc, pour les éléments `Person` avec la même valeur `LastName`, un tri supplémentaire pourrait être fait sur `FirstName`. De cette manière, dans SPS, les instructions de tri multiples (chacun utilisant une sort-key) peuvent être définies pour une seule action de tri.

Le modèle est appliqué à l'ensemble trié et les résultats sont envoyés à la sortie dans l'ordre trié. La fonction tri est disponible pour HTML.

Modèles définis par l'utilisateur

Les [modèles définis par l'utilisateur](#)²³⁹ sont des modèles qui sont appliqués à des items sélectionnés par une expression XPath que vous spécifiez. Les nœuds sélectionnés par l'expression XPath d'un modèle utilisé par l'utilisateur peut aussi être trié. Dans ce cas, le tri est appliqué dans le modèle défini par l'utilisateur.

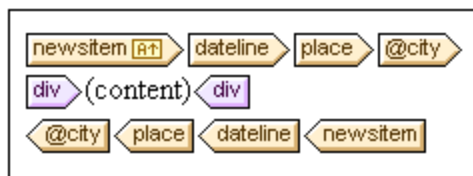
Dans cette section

- Le [mécanisme de tri](#)²⁷⁹ est décrit.
- Un [exemple](#)²⁸¹ démontre comment le tri est utilisé.

7.4.1 Le mécanisme de tri

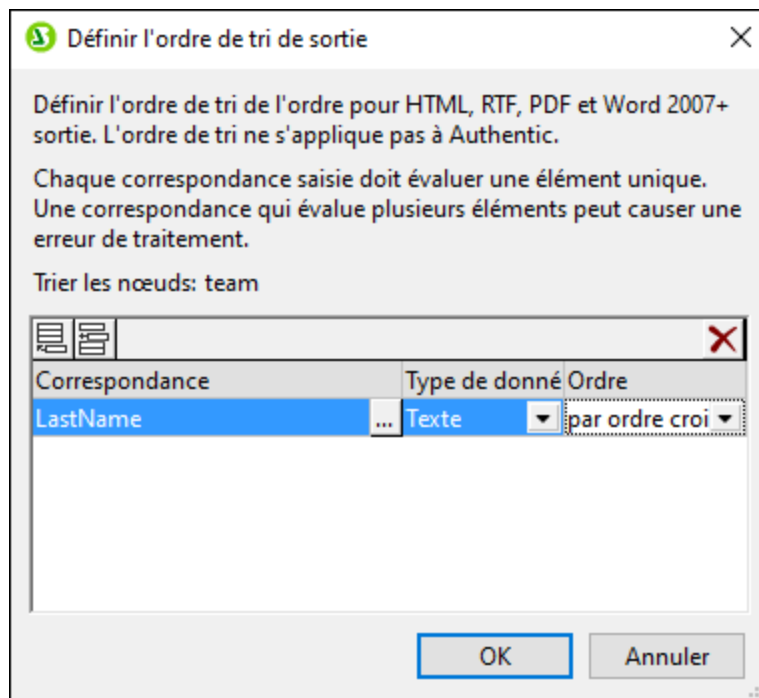
Configurer un nœud d'élément de schéma pour trier consiste en deux étapes :

1. Dans le Mode Design, sélectionnez un nœud d'élément de schéma qui doit être trié. Notez qu'il s'agit des instances de l'élément **this** dans le document XML qui sera trié. Souvent, il n'est pas tout de suite apparent quel élément doit être trié. Par exemple, si vous prenez la structure affichée dans la capture d'écran ci-dessous.



Chaque `newsitem` a une `dateline` contenant un élément `place` avec un attribut `city`. Les nœuds `@city` de tous les éléments `newsitem` sont la sortie par ordre alphabétique. Dans le design, le nœud `@city` devrait-il être sélectionné pour le tri, ou les éléments `place`, `dateline`, ou `newsitem` ? Avec `@city` sélectionné, il n'existera que le nœud `city` qui sera trié. Avec `place` ou `dateline` sélectionné, il n'y aura que l'élément respectif à trier, puisqu'à l'intérieur de leurs parents, ils apparaissent seuls. Avec `newsitem` sélectionné, toutefois, il y aura de multiples éléments `newsitem` à l'intérieur de l'élément `newsitems` parent. Dans ce cas, il s'agit de l'élément `newsitem` qui doit être trié, utilisant une `sort-key` de `dateline/place/@city`.

- Après avoir sélectionné l'élément à trier, dans le menu contextuel (obtenu en cliquant sur la touche de droite de la sélection de l'élément), cliquez sur la commande **Sort Output**. Ceci ouvre le dialogue Définir l'ordre de tri de la sortie (*capture d'écran ci-dessous*), dans lequel vous insérez ou ajoutez une ou plusieurs instructions.



Chaque instruction de tri contient : (i) une `sort-key` (saisie dans la colonne Match); (ii) le datatype que le nœud `sort-key` devrait être (texte ou nombre); (iii) et l'ordre de tri (ascendant ou descendant). L'ordre dans lequel les instructions de tri sont listées est important. Le tri est exécuté utilisant chaque instruction de tri à son tour, en commençant par le premier, et en procédant vers le bas de la liste lorsque de multiples items ont la même valeur. Tout nombre d'instructions de tri est autorisé.

Pour un exemple à savoir comment le tri est utilisé, voir [Exemple : Trier sur de multiples Sort-Keys](#) ²⁸¹.

Modèles définis par l'utilisateur

Les [modèles définis par l'utilisateur](#) ²³⁹ sont des modèles qui sont appliqués à des items sélectionnés par une expression XPath que vous spécifiez. Les nœuds sélectionnés par l'expression XPath d'un modèle utilisé par l'utilisateur peut aussi être trié. Dans ce cas, le tri est appliqué dans le modèle défini par l'utilisateur.

Une note sur les sort-keys

L'expression XPath que vous saisissez pour la sort-key doit sélectionner un **seul nœud** pour chaque instance d'élément—pas un nodeset (XPath 1.0) ou une séquence d'items (XPath 2.0 et XPath 3.0) ; la clé pour chaque élément devrait être résolue vers un string ou une valeur de nombre.

Dans un **XSLT 2.0 ou 3.0** SPS, si la sort-key renvoie une séquence de nœuds, une erreur de traitement XSLT sera retournée. Donc, dans l'exemple Person cité ci-dessus, avec le nœud contextuel de `Person`, une expression XPath telle que : `../Person/LastName` renverrait une erreur parce que cette expression renvoie tous les éléments `LastName` contenus dans le parent de `Person` (assumant qu'il y a plus qu'un élément `Person`). L'expression XPath correcte, avec `Person` dans le nœud contextuel, serait : `LastName` (puisque'il n'y a qu'un nœud `LastName` pour chaque élément `Person`).

Dans **XSLT 1.0**, la spécification requiert qu'un ensemble de nœuds est retourné par un sélecteur sort-key, la valeur de texte du premier nœud est utilisée. StyleVision ne renvoie pour cette raison aucune erreur si l'expression XPath sélectionne de multiples nœuds pour la sort-key ; le texte du premier nœud est utilisé et les autres nœuds sont ignorés. Toutefois, le premier nœud sélectionné pourrait ne pas être la sort-key désirée. Par exemple, l'expression XPath `../Person/LastName` de l'exemple décrit ci-dessus ne retournerait pas d'erreur. Mais elle ne trierait pas, car il s'agit de la même valeur pour chaque élément dans la boucle de tri entière (la valeur de texte du premier nœud `LastName`). Une expression telle que : `location/@*`, toutefois, trierait, utilisant le premier attribut de l'élément enfant `location` comme sort-key. Ce genre d'expression, toutefois, doit être évité, et une sélection plus précise de la sort-key (sélectionnant un nœud) est recommandée.

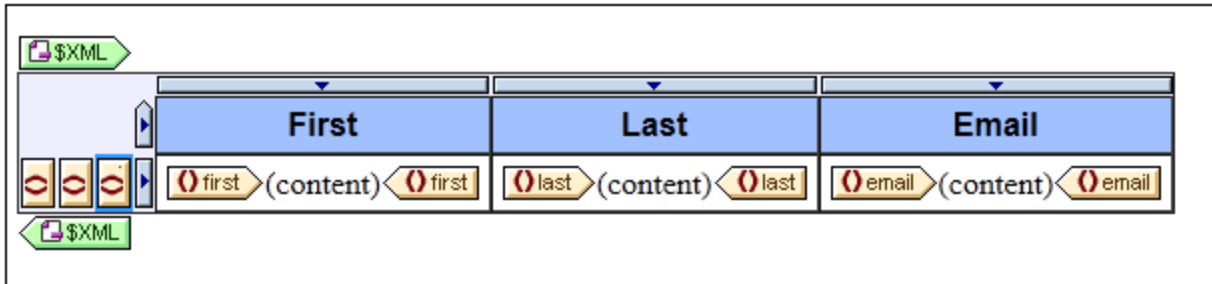
7.4.2 Exemple : Trier sur de multiples Sort-Keys

Dans l'exemple ci-dessous (disponible dans le [dossier \(Mes\) Documents](#) ²⁴, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\Sorting\SortingOnTwoTextKeys.sps`), les membres de l'équipe sont recensés dans une table. Chaque membre est recensé par son prénom, nom de famille et adresse e-mail dans une ligne de la table. Disons que nous voulons trier la liste des membres par ordre alphabétique, d'abord par leurs noms de famille, puis par leurs prénoms. Voici comment cela fonctionne.

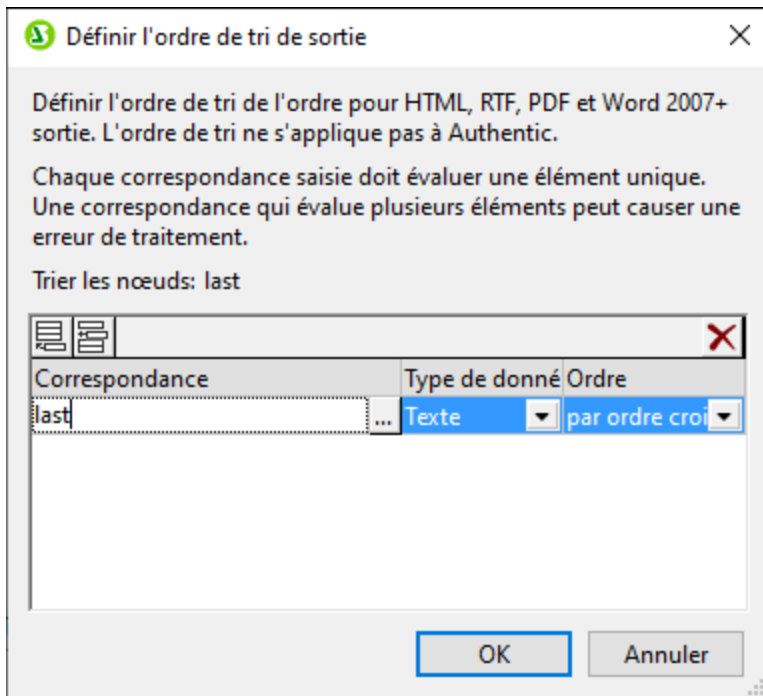
Lorsque la liste n'est pas triée, l'ordre de sortie est l'ordre dans lequel les éléments `member` sont recensés dans le document XML (*capture d'écran ci-dessous, qui est la sortie HTML*).

First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

Dans le Mode Design, cliquez avec la touche de droite dans l'élément `member` (mis en surbrillance bleue dans la capture d'écran ci-dessous), et depuis le menu contextuel qui apparaît, sélectionnez la commande **Tri de sortie**.



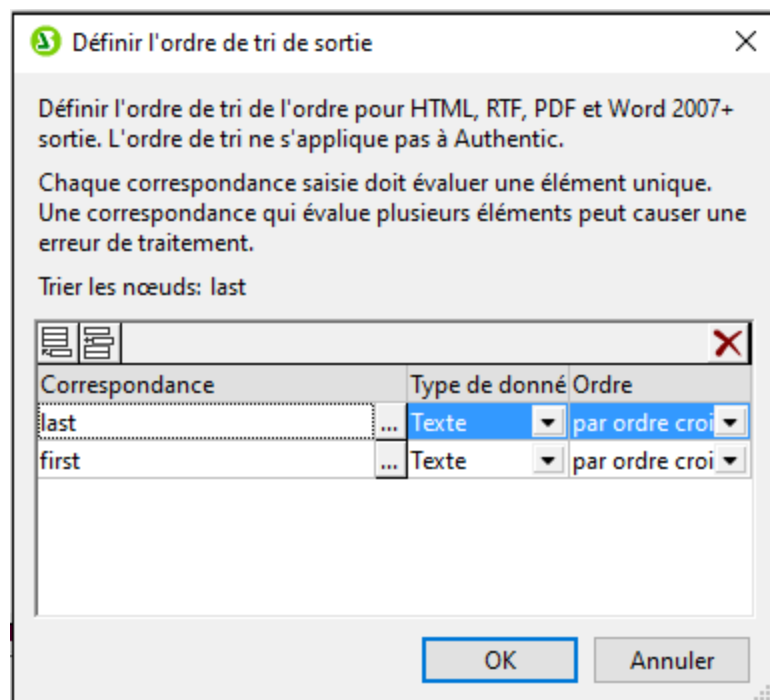
Ceci fait apparaître le dialogue Définir Tri de sortie (capture d'écran ci-dessous). Notez que l'élément sélectionné pour le tri, `member`, est nommé selon l'entrée des Nœuds de tri. Ce nœud est également le nœud contextuel pour l'expression XPath afin de sélectionner la « sort-key ». Cliquez sur le bouton Ajouter ligne (à gauche de la barre d'outils du volet) pour ajouter la première instruction de tri. Dans la ligne ajoutée, saisissez une expression XPath dans la colonne Match pour sélectionner le nœud `last`. Vous pouvez également cliquer sur le bouton Créer [...] pour générer l'expression XPath. La colonne Datatype vous permet de sélectionner comment le contenu de la sort-key doit être évalué : comme texte ou comme nombre. La colonne Trier recense l'ordre du tri : ascendant ou descendant. Sélectionnez `Texte` et `Ascendant`. Cliquez sur **OK** pour terminer.



Dans le Mode Design, la balise `member` affiche une icône indiquant qu'elle contient un filtre tri `member [↑]`. La sortie HTML de la liste « team-member », triée par le nom de famille, est affichée ci-dessous. Notez que les deux « Edwards » ne sont pas triés par ordre alphabétique (Nadia est recensée avant John, ce qui correspond à l'ordre dans le document XML). Une deuxième sort-key est requise pour trier par nom de famille.

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com

Dans le Mode Design, cliquez avec la touche de droite sur la balise `member` et sélectionnez la commande **Trier sortie** depuis le menu contextuel. Le dialogue Définir Tri de sortie apparaît avec l'instruction de tri `last` recensée. Pour ajouter une autre instruction de tri, ajoutez une nouvelle ronde et saisissez l'élément `first` comme sa sort-key (*capture d'écran ci-dessous*). Cliquez sur **OK** pour terminer.



Dans la sortie HTML, la liste est désormais triée par ordre alphabétique d'après le nom de famille et puis le prénom.

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
John	Edwards	j.edwards@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com

7.5 Paramètres et variables

Les paramètres et variables peuvent être déclarés et référencés dans SPS. La différence entre les deux est que tandis qu'une valeur de variable est définie quand elle est déclarée, un paramètre peut avoir une valeur qui lui est transmise (au moment de l'exécution via la ligne de commande) qui écrase la valeur optionnelle par défaut assignée lorsque le paramètre a été déclaré.

Dans cette section, nous décrivons la fonction disponible pour les paramètres et variables :

- Les [paramètres déclarés par l'utilisateur](#) ²⁸⁵ expliquent comment les paramètres définis par l'utilisateur peuvent être utilisés dans SPS.
- Les [paramètres pour les fragments de design](#) ²⁸⁷ décrivent comment les paramètres peuvent être utilisés avec les fragments de design.
- Les [paramètres pour les sources SPS](#) ²⁸⁹ sont un type spécial de paramètre. Ils sont automatiquement définis par StyleVision pour les sources de schéma (en particulier, les fichiers de travail XML des schémas). Puisque le nom et la valeur d'un tel paramètre sont connus de l'utilisateur, le paramètre peut être référencé au sein de SPS et une valeur transmise au moment de l'exécution depuis la ligne de commande.
- Les [Variables](#) ²⁹⁰ vous permettent de : (i) déclarer une variable avec une certaine étendue et définir leur valeur, et (ii) référencer la valeur des variables déclarées et créer un modèle sur un nœud ou des nœuds sélectionnés par la variable.

7.5.1 Paramètres déclarés par l'utilisateur

Dans un SPS, les paramètres déclarés par l'utilisateur sont déclarés globalement avec un nom et une valeur de string par défaut. Une fois déclarés, il peuvent être utilisés dans les expressions XPath partout dans SPS. La valeur par défaut du paramètre peut être écrasée pour des transformations XSLT individuelles en octroyant à la feuille de style XSLT une nouvelle valeur globale via [StyleVision Server](#).

Utiliser les paramètres

Les paramètres déclarés par l'utilisateur sont utiles dans les situations suivantes :

- Si vous souhaitez utiliser une valeur dans de multiples emplacements ou comme entrée pour différents calculs. Dans ce cas, vous pouvez enregistrer la valeur requise comme valeur de paramètre et utiliser le paramètre dans les emplacements requis et les calculs.
- Si vous souhaitez transmettre une valeur à la feuille de style au moment du traitement. Dans SPS (et la feuille de style), vous utilisez un paramètre avec une valeur par défaut. Au moment du traitement, vous transmettez la valeur désirée au paramètre via [StyleVision Server](#).

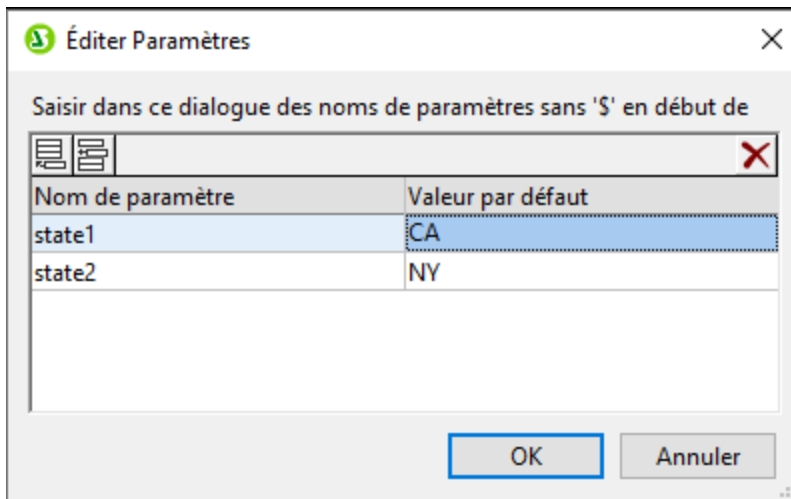
Mécanisme d'utilisation

Travailler avec les paramètres déclarés par l'utilisateur dans SPS consiste en deux étapes :

1. [Déclarer les paramètres requis](#) ²⁸⁶.
2. [Référencer les paramètres déclarés](#) ²⁸⁶.

Déclarer les paramètres

Tous les paramètres définis par l'utilisateur sont déclarés et édités dans le dialogue Éditer les paramètres (*capture d'écran ci-dessous*). Le dialogue Éditer les paramètres est accédé via : la commande [Éditer | Paramètres de feuille de style](#) ⁴⁸².



Déclarer un paramètre implique lui donner un nom et une valeur de string—sa valeur par défaut. Si aucune valeur n'est spécifiée, les valeurs par défaut sont un string vide.

Pour déclarer un paramètre, procédez comme suit :

1. Dans le dialogue Éditer les paramètres, ajoutez ou insérez un nouveau paramètre en cliquant sur les boutons Ajouter ou Insérer. Une nouvelle ligne apparaît.
2. Saisissez le nom du paramètre. Les noms de paramètre doivent commencer par une lettre, et peuvent obtenir les caractères A à Z, a à z, 0 à 9, et du trait souligné.
3. Saisir une valeur par défaut pour ce paramètre. La valeur que vous saisissez est acceptée comme string de texte.

Vous pouvez insérer tout nombre de paramètres et modifiez les paramètres existants à tout moment tout en éditant SPS.

Note :

- Le dialogue Éditer les paramètres contient tous les paramètres définis par l'utilisateur dans une SPS.
- Les paramètres peuvent également être déclarés dans la [barre latérale de l'Aperçu du design](#) ³⁵.

Référencer les paramètres déclarés

Les paramètres peuvent être référencés dans les expressions XPath en préfixant un caractère \$ avant le nom du paramètre. Par exemple, vous pourriez référencer un paramètre dans l'expression XPath du calcul automatique (par ex., `concat('www.', $company, '.com')`).

Note : tandis qu'il s'agit d'une erreur lorsqu'un paramètre non déclaré est référencé, il ne s'agit pas d'une erreur quand on déclare un paramètre et ne le référence pas.

7.5.2 Paramètres pour les fragments de design

Les paramètres pour les fragments de design vous permettent de définir un paramètre sur un fragment de design que vous avez créé et de donner une valeur par défaut à ce paramètre. Dans chaque emplacement où le fragment de design est utilisé dans le design, vous pouvez saisir une valeur de paramètre différente, vous permettant donc de modifier la sortie des fragments de design individuels.

Par exemple, un fragment de design appelé `EmailAddresses` peut être créé avec un paramètre appelé `Domain` qui a la valeur par défaut `altova.com`. Maintenant, disons que ce paramètre est utilisé dans le calcul automatique du fragment de design pour générer les adresses e-mail des employés de l'entreprise. Pour les adresse UE, nous pourrions utiliser le fragment de design `EmailAddresses` et éditer la valeur du paramètre `Domain` en `altova.eu`. De la même manière, dans le modèle des employés japonais, nous pourrions éditer la valeur du paramètre `Domain` en `altova.jp`. Pour les employés US de l'entreprise, nous pourrions laisser la valeur du paramètre de `Domain` inchangé, générant donc une valeur par défaut de `altova.com`.

Utiliser les paramètres pour les fragments de design consiste en deux parties :

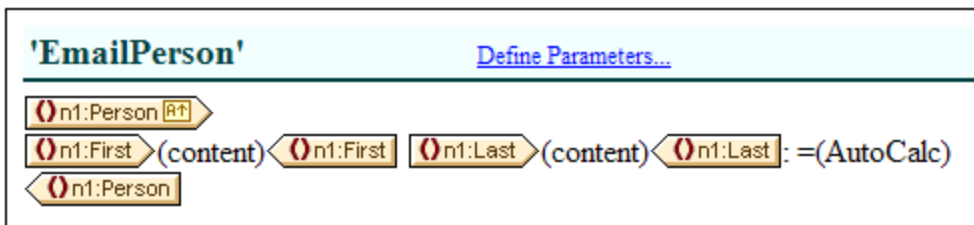
1. [Définir le paramètre](#)²⁸⁷ avec une valeur par défaut sur le fragment de design où il a été créé.
2. [Éditer la valeur du paramètre](#)²⁸⁸ où le fragment de design est utilisé.

Ces parties sont expliquées ci-dessous en détail.

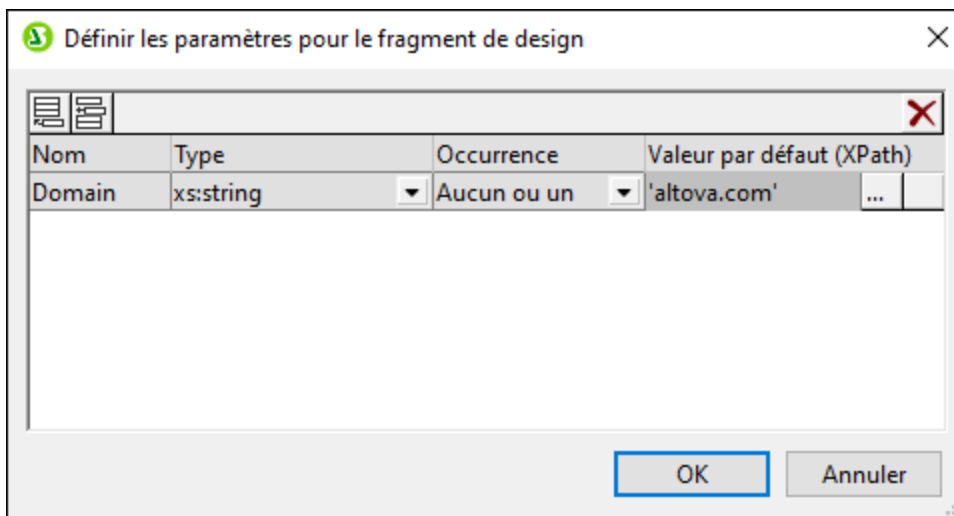
Note : les paramètres pour les fragments de design sont pris en charge dans Authentic View uniquement dans les Enterprise Editions des produits d'Altova.

Définir le paramètre

Chaque fragment de design peut être assigné à tout nombre de paramètres. À cette fin, cliquez sur le lien Définir les paramètres dans la barre de titre du fragment de design (*voir la capture d'écran ci-dessous*).



Ceci fait apparaître le dialogue Définir Paramètres pour les fragments de design (*capture d'écran ci-dessous*). Cliquez sur l'icône **Ajouter** ou **Insérer** en haut à gauche pour ajouter une ligne d'entrée du paramètre. Saisissez ou sélectionnez le nom, le type de données, le nombre d'occurrences, et la valeur par défaut du paramètre. L'attribut *Occurrence* du paramètre spécifie le nombre d'items renvoyés en évaluant l'expression XPath spécifiée comme valeur par défaut du paramètre. L'attribut *Occurrence* est optionnel, par défaut, et est aucun ou un. Vous pouvez ajouter autant de paramètres que vous souhaitez.

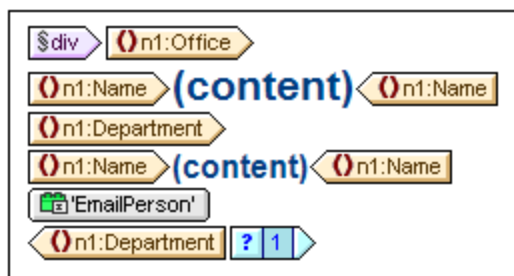


Il existe deux types de l'icône **Supprimer**. L'icône Supprimer à droite de chaque entrée de paramètre supprime la valeur par défaut de ce paramètre. L'icône **Supprimer** en haut à droite du volet du volet supprime le paramètre mis en surbrillance actuellement.

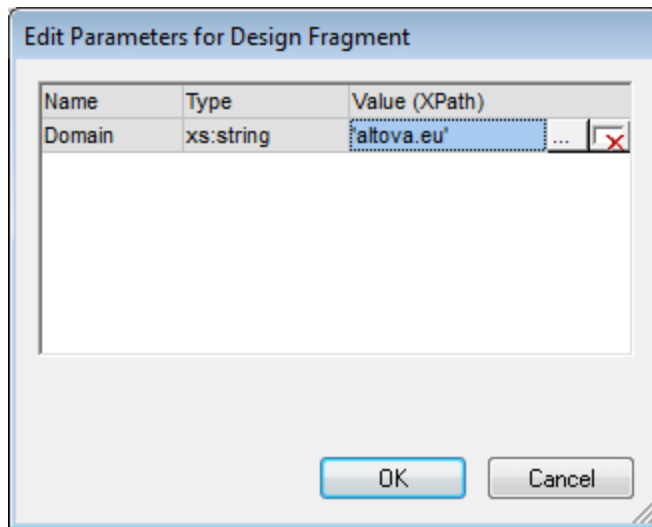
Note : si le fichier SPS utilise XSLT 1.0, alors l'expression XPath que vous saisissez doit retourner un ensemble de nœud. Autrement une erreur est reportée.

Utiliser le paramètre

Une fois que le fragment de design a été créé, il peut être inséré dans de multiples emplacements dans le design (en le glissant depuis l'arborescence de design ou l'arborescence de schéma). La capture d'écran ci-dessous affiche le fragment de design `EmailPerson`, inséré après l'élément `n1:Name`.



Si un paramètre a été défini pour ce fragment de design, alors sa valeur peut être éditée pour cette instance d'utilisation particulière du fragment de design. Faites-ceci en cliquant avec la touche droite de la souris dans le fragment de design et sélectionnant la commande **Éditer Paramètres**. Ceci fait apparaître le dialogue Éditer Paramètres pour les fragments de design (capture d'écran ci-dessous).



Vous pouvez éditer la valeur du paramètre dans ce dialogue. Cliquez sur **OK** pour terminer. La nouvelle valeur de paramètre sera utilisée dans l'instance d'utilisation du fragment de design. Si la valeur de paramètre n'est pas éditée, la valeur de paramètre original (ou par défaut) sera utilisée.

Note : si XSLT 1.0 est utilisé, alors l'expression XPath doit renvoyer un node-set. Autrement une erreur est reportée.

7.5.3 Paramètres pour les sources SPS

Un SPS peut avoir de multiples sources de schéma, où un schéma pourrait être un DTD ou schéma XML sur lequel un document XML est basé, ou un schéma XML qui est généré depuis une BD et sur laquelle une BD est basée.

Dans chaque SPS, il existe un schéma principal, et en option, un ou plusieurs schémas supplémentaires. Lorsque vous ajoutez une nouvelle source de schéma, StyleVision déclare automatiquement un paramètre pour ce schéma et assigne une valeur au paramètre qui est l'URI du fichier de travail XML que vous assignez à ce schéma. Dans le cas de BD, StyleVision génère un fichier XML temporaire depuis la BD, et définit le paramètre pour cibler le nœud du document de ce fichier XML temporaire.

Référencer les paramètres pour les sources

Chaque paramètre SPS pour une source de schéma adresse le nœud du document d'un fichier XML correspondant à ce schéma. Dans StyleVision, le fichier XML de chaque schéma est le fichier de travail XML ou le fichier XML généré depuis une BD. Les paramètres SPS pour les sources peuvent pour cette raison être utilisés de deux manières :

1. Dans les expressions XPath à l'intérieur de SPS, pour situer des nœuds dans de nombreux documents. Le paramètre est utilisé pour identifier le document, et des étapes de localisation ultérieures dans l'expression XPath situent le nœud requis à l'intérieur de ce document. Par exemple, l'expression : `count($XML2//Department/Employee)` renvoie le nombre d'éléments de `Employee` dans tous les éléments `Department` dans le document XML qui est le fichier de travail XML assigné à la source de schéma désignée `$XML2`.

2. Sur la ligne de commande, l'URI d'un autre fichier XML peut être passé comme valeur d'un paramètre SPS pour les sources. Évidemment, le nouveau fichier XML devrait être basé sur le schéma représenté par ce paramètre. Par exemple, si `FileA.xml` et `FileB.xml` sont tous les deux valides selon le même schéma, et que `FileA.xml` est le fichier de travail XML assigné au schéma `$XML3` utilisé dans un SPS, alors quand la transformation XSLT pour ce SPS est invoqué depuis la ligne de commande, `FileB.xml` peut être substitué pour `FileA.xml` en utilisant le paramètre `$XML3="FileB.xml"`. Vous devriez aussi noter que, sur la ligne de commande, les valeurs devraient être saisies pour tous les paramètres SPS pour les sources à l'exception du paramètre pour le schéma principal. Le fichier XML correspondant au schéma principal sera le point d'entrée pour la feuille de style XSLT et sera pour cette raison le fichier XML sur lequel la transformation est exécutée.

7.5.4 Variables

Utiliser des variables consiste en deux parties : (i) [déclarer la variable](#) ²⁹⁰, et (ii) [utiliser la variable](#) ²⁹².

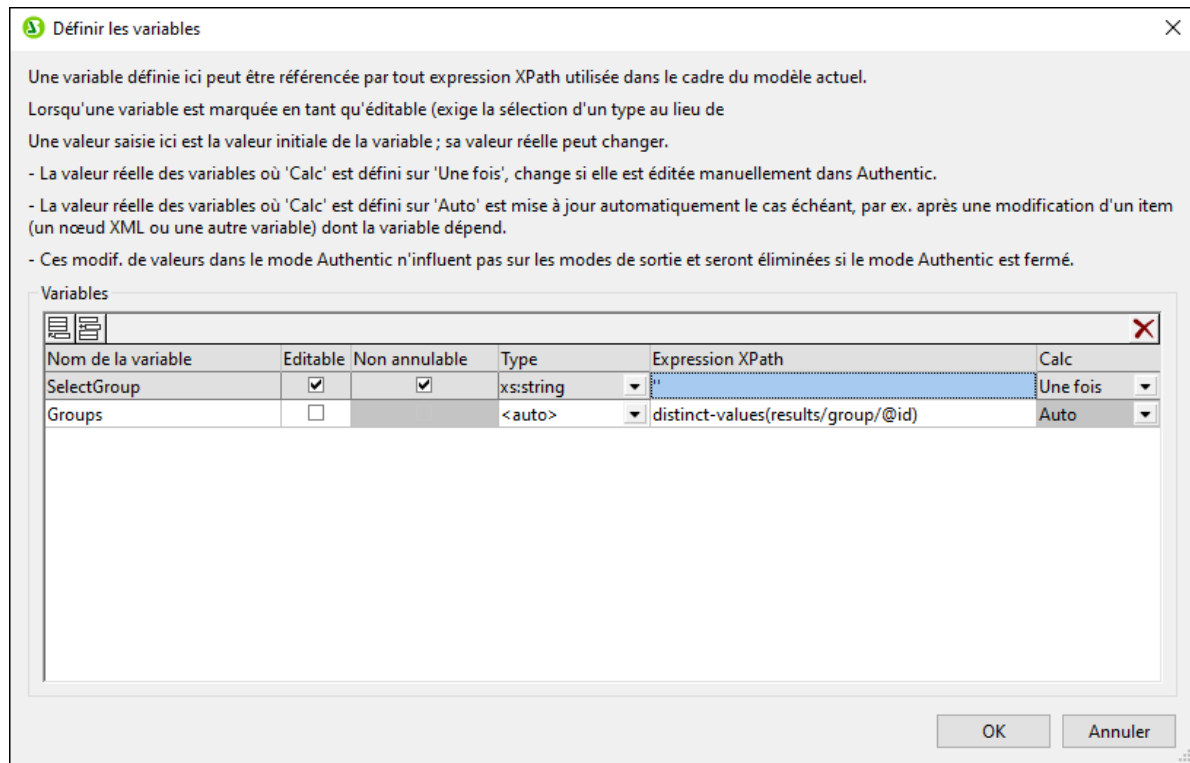
Note : les variables sont prises en charge dans Authentic View uniquement dans les produits des Enterprise Editions d'Altova.

Déclarer une variable

Une variable peut être déclarée sur un modèle inclus dans le design. Il s'agit d'un nom, un type de données et d'une valeur. De plus, vous pouvez spécifier si elle doit être éditable dans les éditions Enterprise de Authentic View. La variable aura ensuite une portée dans ce modèle et peut être utilisé à l'intérieur. Pour déclarer une variable pour qu'elle ait une portée pour le document entier, déclarez la variable sur le modèle racine. Un avantage majeur de déclarer une variable uniquement sur le modèle où nécessaire est que les expressions XPath pour situer le nœud descendant seront plus simples.

Déclarer une variable comme suit :

1. Cliquez avec la touche de droite sur le modèle de nœud sur lequel la variable doit être créée et sélectionnez la commande **Définir les variables**.
2. Dans le dialogue Définir les variables qui apparaît (*capture d'écran ci-dessous*), cliquez sur l'icône **Ajouter la Variable** en haut à gauche sur le volet des Variables, puis saisissez un nom de variable. La valeur de la variable est donnée via une expression XPath. Si vous voulez saisir un string comme valeur de la variable (comme dans la première variable dans la capture d'écran ci-dessous), alors mettez le string entre guillemets. Dans la capture d'écran ci-dessous, la valeur de la variable `SelectGroup` est un string vide. Autrement, le texte sera lu comme un nom de nœud ou un fonction-call.



3. Définir une variable comme *Éditable* (en vérifiant la case à cocher *Éditable*) permet à la variable d'être éditée dans Authentic View (qui est disponible comme aperçu uniquement dans les éditions Enterprise et Professional de StyleVision). Dans ce cas, vous devez aussi définir la valeur de datatype dans le type correct, tel que `xs:string`. Lorsqu'une variable est éditable, la valeur d'origine définie par le designer SPS peut être éditée quand l'utilisateur Authentic View apporte des changements au document dans Authentic View. De tels changements peuvent être une édition explicite de la variable (telle que lorsque la valeur de la variable est créée comme éditable (`contenus`) ou une case de texte éditable et que celle-ci est éditée par l'utilisateur Authentic View), ou lorsqu'un nœud ou une valeur utilisée dans l'expression XPath de la variable est modifiée par l'utilisateur Authentic View.
4. Si la variable est définie comme *Éditable*, alors deux options pertinentes supplémentaires à Authentic View sont activées : *Undoable* et *Calc*. Vérifier l'option *Undoable* génère une étape Undo pour chaque changement effectué à la variable. L'utilisateur Authentic View peut pour cette raison avoir à cliquer sur le cycle Undo pour extraire une valeur précédente de la variable. La valeur *Calc* peut être soit *Once* ou *Auto*. Si cette option est définie comme *Once*, la valeur de la variable est calculée une fois lorsque le modèle contenant la variable est évalué. La valeur peut uniquement être modifiée lorsque l'utilisateur édite explicitement la variable (par exemple, si la variable est créée comme éditable (`contenus`) ou comme case de texte éditable). D'un autre côté, si cette option est définie comme *Auto*, la variable sera recalculée également à chaque fois qu'un nœud ou une valeur utilisé dans l'expression XPath de la variable est modifié.
5. Vous pouvez ajouter autant de variables que vous voulez, mais le nom de la variable ne doit pas être le nom d'une variable déjà déclarée in-scope. Pour supprimer une variable, cliquez sur l'icône **Supprimer** située en haut à droite du volet.
6. Cliquez sur **OK** lorsque vous avez terminé. La balise du modèle aura maintenant une icône \$ pour indiquer qu'une ou plusieurs variables y ont été déclarées.

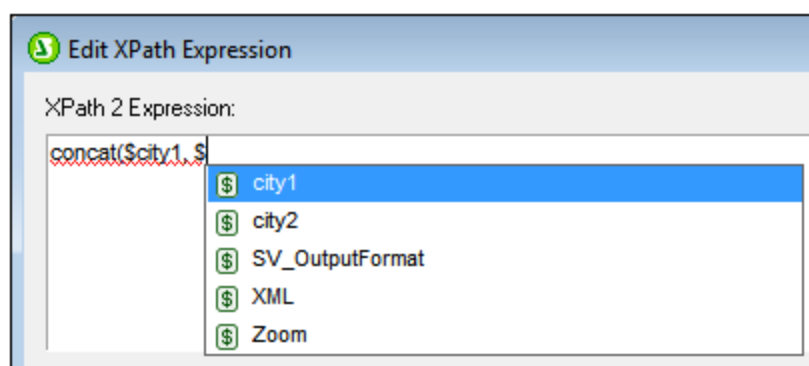
De cette manière, les variables peuvent être créées pour chaque modèle de nœud présent dans le design. Chacune de ces variables aura un nom et une valeur et sera dans leur portée à l'intérieur du modèle sur lequel elle est déclarée. Pour éditer une variable ultérieurement, cliquez avec la touche de droite sur le modèle de

nœud sur lequel la variable a été créée et sélectionnez la commande **Définir les variables** pour accéder au dialogue Définir les variables.

Utiliser une variable

Pour qu'une variable soit utilisée à tout emplacement, elle doit être à la portée dans cet emplacement. Ceci signifie qu'une variable ne peut être utilisée qu'au sein du modèle sur lequel elle est définie. Les variables peuvent aussi être éditées dans Authentic View pour que les utilisateurs peuvent contrôler l'affichage. La valeur éditée est ignorée lorsque la SPS est fermée.

Une variable peut être utilisée dans toute expression XPath, et est référencée dans l'expression XPath en préfixant son nom avec un symbole \$. Par exemple, l'expression XPath \$VarName/Name sélectionne l'élément enfant `Name` du nœud sélectionné par la variable appelée `VarName`.



Lorsque vous saisissez une expression XPath dans le [dialogue Éditer expression XPath](#)⁴²⁶, les variables in-scope apparaissent dans un pop-up (voir la capture d'écran ci-dessus). Sélectionnez une variable dans la pop-up et en appuyant sur **Enter**, ceci insère la référence de la variable dans l'expression.

7.6 Table de contenus, Référencement, Signets

La Table de contenus (TOC) et d'autres mécanismes de référencement fonctionnent en créant des ancres aux points requis dans le document du design, puis revenant à ces références depuis la TOC, les références de texte, les séquences de numérotation automatique et les hyperliens.

Nous nous pencherons brièvement sur le mécanisme d'ancrage (ou bookmarking) en premier lieu, puis sur la mécanique TOC global. Nous faisons ceci car comprendre le mécanisme de bookmarking permettra une meilleure compréhension de la mécanique TOC global.

Le mécanisme de bookmarking

Deux types de mécanismes de bookmarking sont utilisés : simples et complexes. Le mécanisme de bookmarking complexe est celui utilisé pour créer les TOC.

- Un bookmark (signet) simple est créé à un point dans le document de design. Le signet obtient un nom unique qui est utilisé comme cible des liens qui y pointe. Le mécanisme de bookmarking simple et le mécanisme utilisé pour la fonction [Signets et Hyperliens](#) ³²¹. (Notez que les hyperliens peuvent pointer en plus vers les URL à l'extérieur du document.)
- Pour un référencement plus complexe, tel que les TOC et pour numéroté automatiquement les sections de document, créer un signet implique deux parties.
 1. Le document de design est structuré dans une hiérarchie de niveaux requis pour la TOC. Ces niveaux sont connus comme niveaux de TOC. La structuration est obtenue en assignant des niveaux de TOC aux différents points dans la structure du document. Les niveaux de TOC peuvent être imbriqués à l'intérieur d'autres niveaux de TOC en vue de donner une structure de TOC hiérarchique au document. (Par exemple, un niveau de TOC peut être assigné à un chapitre du livre, un autre niveau de TOC peut être attribué à l'intérieur de ce niveau aux sections du chapitre.)
 2. Les signet TOC sont créés à l'intérieur des niveaux TOC. Ces signets TOC identifient les sections du document à différents niveaux qui entrent dans la TOC. De plus, chaque signet TOC doit être défini pour fournir le texte qui apparaîtra dans le composant de référencement.

Une fois que les niveaux TOC et les textes de référence des signets TOC ont été définis, le modèle de TOC contenant les composants de référencement peuvent être conçus.

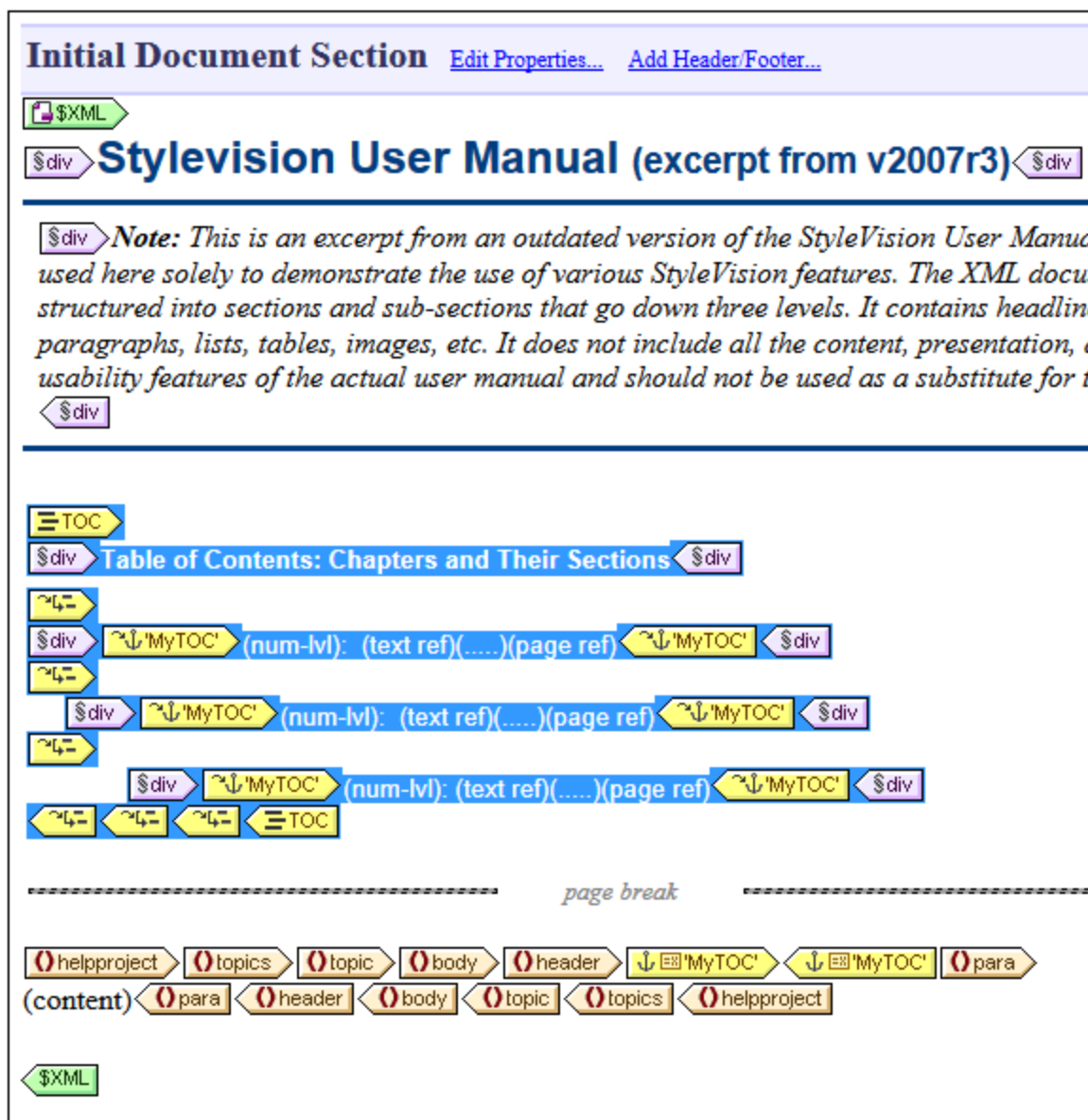
Le mécanisme TOC global est décrit ci-dessous, sous le [mécanisme de TOC](#) ²⁹³. Les différentes fonctions de référencement sont expliquées en détail dans le reste de la section.

Le mécanisme de la TOC

Si vous avez sélectionné [XSLT 2.0 ou XSLT 3.0](#) ⁹⁹ (et non XSLT 1.0) comme version XSLT de votre SPS, vous pouvez créer une table de contenus (TOC)—essentiellement un modèle pour la TOC—à tout emplacement dans le design.

- Il est recommandé que les items du design qui doivent être inclus dans et liés à depuis la TOC sont d'abord [marqués dans le design](#) ²⁹⁶. Ces items peuvent être du contenu statique ou du contenu dynamique. Dans la partie inférieure de la capture d'écran ci-dessous, les balises des signets de la TOC en jaune  'MyTOC' 'MyTOC' à l'intérieur de la balise `header` indiquent que l'item `header` a été marqué (pour inclusion dans le modèle TOC).

- Un [modèle est créé pour la TOC](#) ³⁰³ (en surbrillance dans la capture d'écran ci-dessous). Le modèle TOC contient le design de la TOC ; il peut être situé partout dans le design. Dans cet exemple affiché dans la capture d'écran ci-dessous, le modèle TOC est situé près du haut du document.



Note : Une de ces deux parties peut être créée d'abord, ou les deux parties peuvent être créées parallèlement. Nous recommandons, toutefois, que les signets TOC soient créés avant le modèle TOC.

La TOC est affichée dans HTML. Notez également que : (i) les TOC peuvent être créées avec une structure plate ou hiérarchique (avec une numération correspondante), et (ii) de multiples TOC peuvent être créées à l'intérieur d'un design. En conséquence, un designer de feuille de style peut créer un document avec, disons une TOC (hiérarchique) au niveau du livre et d'autres (également hiérarchiques) au niveau du chapitre, et recense des figures et tables (plates).

Procédure pour créer les TOC

Ci-dessous, vous trouverez une procédure étape-par-étape pour créer une TOC. Les items sont d'abord marqués pour inclusion. Le modèle TOC est construit juste après. (En alternative, vous pouvez d'abord créer un modèle TOC, puis marquer les items pour inclusion. Ou vous pouvez créer le modèle TOC et sélectionnez les items pour inclusion en parallèle.)

1. Assurez-vous que [XSLT 2.0](#)⁹⁹ est la version XSLT sélectionnée.
2. [Structurer le document dans les niveaux TOC](#)²⁹⁸. Si la TOC a de multiples niveaux, structurez le design du document dans une hiérarchie de niveaux de TOC imbriqués. Si la TOC doit avoir une structure plate (à savoir, un niveau uniquement), alors créez au moins un niveau de TOC (dans le design du document) qui inclura les signets TOC.
3. [Créer un ou plusieurs signets TOC](#)³⁰⁰ à l'intérieur de chaque élément dans le design du document. Les signets TOC identifient les composants à l'intérieur de chaque niveau TOC qui doivent apparaître dans la TOC.
4. [Créer un modèle TOC contenant les références du niveau TOC \(levelrefs\)](#)³⁰³. Le modèle TOC doit avoir le nombre requis de références de niveau TOC (levelrefs). Dans le cas d'une TOC à plusieurs niveaux, les levelrefs dans le modèle TOC devraient être imbriqués (voir la capture d'écran ci-dessus).
5. [Créer des références TOC \(TOCrefs\) dans le modèle TOC](#)³⁰⁶. Dans un modèle TOC, configurez une TOCref pour chaque levelref. Chaque TOCref référencera, de par le nom, les signets TOC à l'intérieur du niveau TOC correspondant dans le document. En alternative, la TOCref peut référencer les signets TOC à d'autres niveaux.
6. [Formater les items TOC](#)³⁰⁷. Chaque item de texte dans la sortie TOC est généré par la TOCref dans le modèle TOC. Les définitions TOCref peuvent préciser le numéro de l'item (y compris hiérarchique), le texte de l'item TOC, un leader, et, pour les média paginés, un numéro de page. Chaque TOCref et ses parties individuelles peuvent être formatés séparément comme requis. (Notez que la numérotation automatique peut aussi être définie à l'intérieur du signet TOC dans le corps principal du document. Voir la section [Numérotation automatique](#)³¹⁶ pour plus de détails.)

Mise à jour des numéros de page TOC dans les documents DOCX et RTF

Quand un utilisateur édite un document de sortie DOCX ou RTF dans MS Word de telle manière que la numérotation de page change, il se peut que la TOC ne soit pas mise à jour avec les nouvelles références de page. Il s'agit d'un problème MS Word. Pour mettre à jour les références de page dans la TOC, appuyez sur **Ctrl+A** pour tout sélectionner, puis appuyez sur **F9**. Pour plus d'information, voir [ici](#).

Terminologie

Les noms des composants liés à la TOC principale utilisés dans l'interface sont dans la table ci-dessous. Les composants ont été placés dans deux colonnes différentes en fonction de leur emplacement : dans le **corps du document** ou dans le **modèle TOC** (qui est le modèle qui spécifie la conception de la table des matières proprement dite et qui se trouve généralement au début du document).

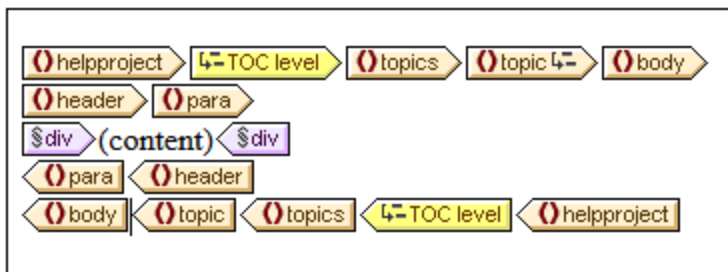
- Les **composants de la TOC dans le corps du document** tracent les items qui seront utilisés dans le modèle TOC.
- Les **composants TOC dans le modèle TOC** référencent les items marqués dans le corps du document. Les composants dans le modèle TOC ont le terme *'reference'* dans leurs noms.

Corps du document	Modèle TOC
Niveau TOC : Les niveaux TOC structurent le document dans une hiérarchie imbriquée.	Références des niveaux (levelrefs) : Correspond à la structure du niveau TOC défini dans le corps du document. Permet aux TOCrefs à un niveau donné de cibler les signets TOC au niveau correspondant.
Signet TOC : A un nom, avec lequel il identifie un nœud dans le document comme item TOC.	Références TOC (TOCrefs) : Référence un signet TOC par son nom.

7.6.1 Items de bookmarking pour l'inclusion TOC

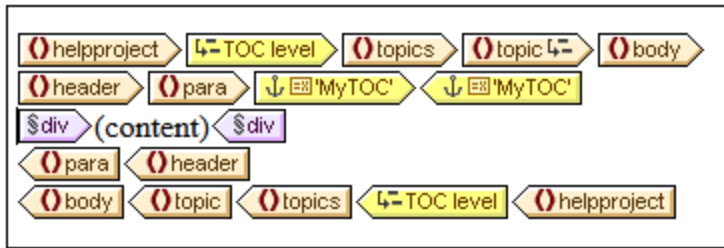
Bookmarker un item dans le design pour inclusion dans une TOC consiste en deux étapes, qui peuvent être faites selon l'ordre souhaité :

1. [Structurer le document de design dans une hiérarchie de niveaux TOC imbriqués](#) ²⁹⁸. Un niveau TOC peut être créé dans le design soit sur un modèle ou autour d'un composant de design. Dans la capture d'écran ci-dessous, un niveau de table des matières a été créé sur le modèle de `sujet`

Lorsqu'un niveau est créé sur un modèle, cela est indiqué par l'icône de niveau à l'intérieur de la balise de début du modèle, par exemple, . Lorsqu'un niveau est créé autour d'un composant, cela est indiqué par des balises de niveau de table des matières  . Dans la capture d'écran ci-dessus, le composant de modèle `topics` est enfermé par un niveau. La différence entre les deux manières de marquer les niveaux est expliquée dans la section [Structurer le Design dans les Niveaux](#) ²⁹⁸. Quand le [modèle TOC est créé](#) ³⁰³, il doit être structuré dans une hiérarchie de niveaux, avec les niveaux du modèle TOC correspondant aux niveaux que vous avez créé dans le design. Même pour les TOC avec une structure plate (un niveau), le design doit avoir le niveau correspondant.

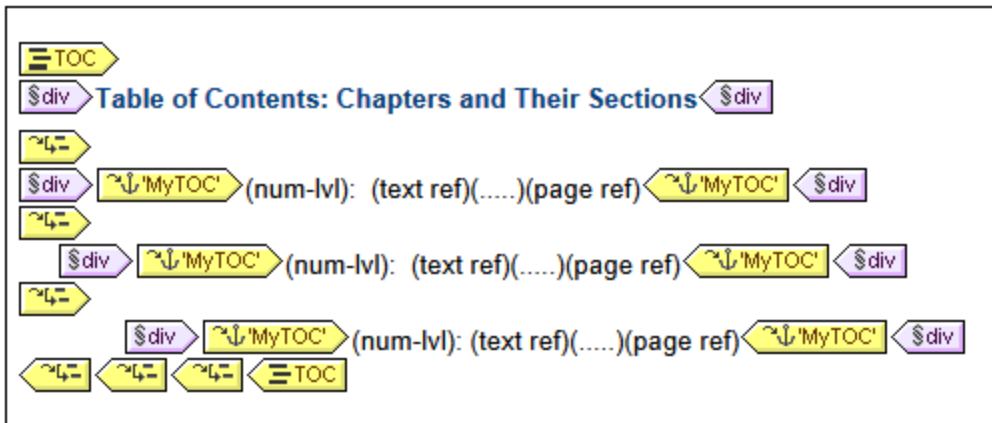
2. [Créer in signet TOC](#) ³⁰⁰ dans le design avec un nom et du texte « TOC-item ». Le signet TOC peut soit être joint ou ne pas joint dans le composant de design ; dans le dernier cas, il est vide. Dans la capture d'écran ci-dessous, le signet TOC n'enferme pas de composant de design.



Le signet TOC sert d'ancrage dans le document. Dans la capture d'écran ci-dessus, le signet TOC (et l'ancrage) est situé au début des instances d'élément `para`. Le signet TOC a deux attributs : (i) le nom qui sera utilisé pour référencer le signet TOC quand vous créez un item TOC dans le modèle TOC, et (ii) un string de texte qui sera utilisé comme texte de l'item TOC correspondant. Comment ces deux attributs sont assignés est décrit dans la section, [Créer des signets TOC](#) ³⁰⁰.

Comment des items marqués sont référencés dans le modèle TOC

Le [modèle TOC](#)³⁰³ est structuré dans des niveaux imbriqués (appelés références de niveau (levelrefs) pour les différencier des niveaux créés dans le corps principal du modèle de design). À l'intérieur de chaque levelref , une référence TOC (TOCref)  est insérée (*voir capture d'écran ci-dessous*). La TOCref à l'intérieur d'une levelref référence les signets TOC utilisant le nom du signet TOC. Chaque signet TOC avec ce nom et niveau correspondant dans le document XML sera créé comme item TOC à ce niveau dans la TOC. Par exemple, la TOCref a indiqué que la balise  référence tous les signets TOC appelés `chapters` au niveau correspondant dans le document XML (lorsque la portée de TOCref a été définie comme `current`). L'attribut de texte des signets TOC instanciés respectifs sera la sortie comme texte de l'item TOC.



Dans la capture d'écran ci-dessus d'un modèle TOC, il existe trois levelrefs imbriquées, à l'intérieur desquels se trouve une TOCref qui contient un modèle de l'item TOC de chaque niveau. Par exemple, dans la première levelref, il existe une TOCref qui référence les signets TOC qui ont un nom de `MyTOC` . En guise de résultat, tous les signets TOC du premier niveau (tel que structuré dans le design) et appelés `MyTOC` seront accédés pour la sortie à ce niveau dans la TOC. La TOCref à l'intérieur de la deuxième levelref référence également des signets TOC ayant un nom de `MyTOC`. En guise de résultat, tous les signets TOC du second niveau du document et appelés `MyTOC` seront utilisés pour les items du second niveau dans la TOC. La troisième levelref fonctionne de la même manière : Les signets TOC appelés `MyTOC` qui apparaissent à l'intérieur du troisième niveau sont référencés pour les items du troisième niveau dans la TOC.

Dans les sous-sections de cette section, nous allons décrire : (i) comment [le design est structuré en niveaux](#)²⁹⁸, et (ii) comment les [signets sont créés](#)³⁰⁰. Comment le [modèle TOC est créé](#)³⁰³ est décrit dans la section, [Créer le modèle TOC](#)³⁰³.

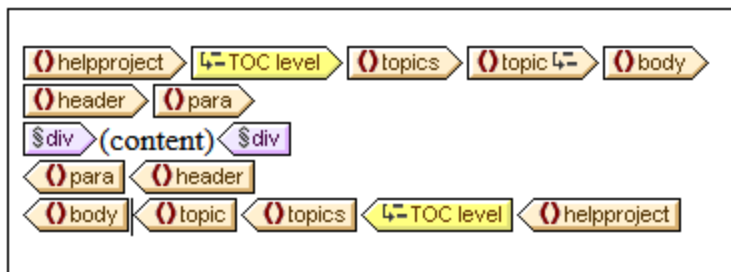
7.6.1.1 Structurer le Design dans les niveaux TOC

La structure hiérarchique que vous souhaitez concevoir pour la TOC est spécifiée comme ensemble des **niveaux imbriqués**. En tant que telle, il s'agit d'une structure hiérarchique qui, bien qu'associée à la structure de document XML, est séparée de celle-ci. Cette structure est spécifiée dans le design du document SPS. Le modèle TOC que vous construisez utilisera une structure correspondant à cette structure hiérarchique. Dans le cas d'une TOC avec une structure plate (un niveau uniquement), le document de design doit avoir au moins un niveau. Si plus d'un niveau existe dans le document, une TOC plate peut ensuite être créée pour tout autre niveau ou pour de multiples niveaux (agrégés ensemble comme un niveau).

Dans le design, les niveaux peuvent être créés dans le modèle principal, dans les modèles globaux, ou dans une combinaison du modèle principal et de modèles globaux. La chose importante à noter est que, peu importe où créé, ces niveaux doivent définir, ensemble, en combinaison, une structure hiérarchique.

Créer des niveaux

Chaque niveau dans le design est créé séparément. Un niveau peut être créé sur un modèle ou autour d'un composant de design. Dans la capture d'écran ci-dessous, un niveau a été créé sur le modèle `topic` (indiqué par ) et un autre autour de l'élément `topics` (indiqué par  ). La différence essentielle entre ces deux méthodes de création de niveaux réside dans le fait que l'option incluse dans un niveau   permet de créer des niveaux autour d'autres composants que les modèles.

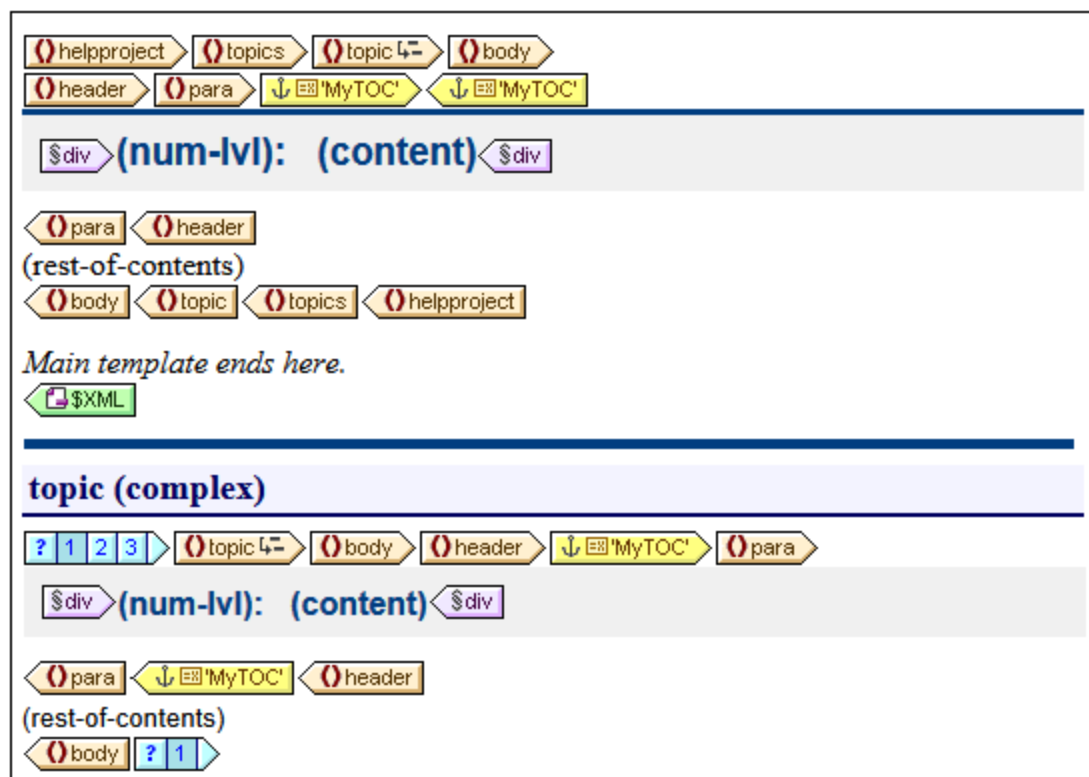


Pour créer un niveau, faites la chose suivante :

1. Sélectionner le composant (modèle ou autre).
2. Cliquez avec la touche de droite, et depuis le menu contextuel, sélectionnez **Modèle sert de niveau** (activé quand un modèle est sélectionné) ou **Enclose With | TOC Level**. Les deux options sont également disponibles dans le menu **Insérer | Insérer Table des matières** : Le **niveau TOC** ou **Modèle sert de niveau**.

Niveaux dans les modèles globaux

Les niveaux peuvent aussi être définis dans les modèles globaux. Dans ces cas, il faut s'assurer que les niveaux créés dans divers modèles globaux, ainsi que ceux dans le modèle principal, définissent **ensemble** une structure hiérarchique lorsque le SPS est exécuté. La capture d'écran ci-dessous montre deux niveaux, un dans le modèle principal (sur le modèle `topic`) et un dans le modèle global pour `topic` (sur le `topic` modèle).



Dans le modèle de contenu représenté par la capture d'écran ci-dessus, `topic` est un élément récursif, à savoir, un élément `topic` lui-même contenir un élément descendant `topic`. Dans le modèle principal (dont la fin est indiquée par la balise `<$XML>`), un niveau a été défini au premier niveau du sujet `topic`. L'instruction `rest-of-contents` dans le modèle principal précise que des modèles seront appliqués pour tous les éléments enfant de `topic/body` à l'exception de `header`. Ceci signifie que le modèle global pour les enfants `topic` de `topic/body` sera traité.

Dans le modèle global pour le sujet, un niveau a été défini dans le modèle du sujet (indiqué par `topic`). Ce deuxième niveau de hiérarchie TOC, qui apparaît au deuxième niveau des éléments `topic`, est imbriqué à l'intérieur du premier niveau de la hiérarchie TOC. Puisque le modèle global a aussi une instruction `rest-of-contents`, le modèle global pour `topic` sera appliqué à tous les éléments `topic` récursifs, créant ainsi des niveaux imbriqués supplémentaires dans la hiérarchie TOC : troisième niveau, quatrième niveau, etc.

En tant que designer, vous devriez être conscient du nombre de niveaux créés dans le design, car lorsque le modèle TOC est créé, vous allez devoir préciser explicitement comment les items TOC pour chaque niveau seront sélectionnés et formatés.

Niveaux dans les TOC plats

Dans une hiérarchie TOC plate, les items TOC seront la sortie à un seul niveau : par exemple, une liste simple des images dans un document.

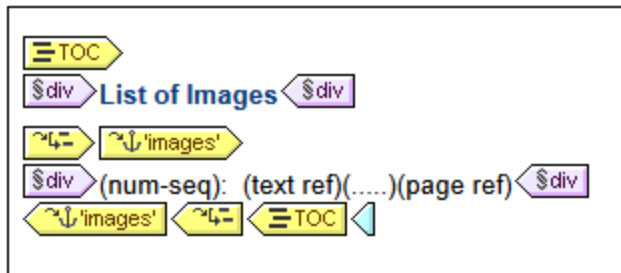
Une hiérarchie plate peut être obtenue de plusieurs manières.

- Le document de design peut être structuré avec juste un seul niveau TOC. Le modèle TOC aura ensuite une seule levelref avec une référence TOC (TOCref) à l'intérieur de celle-ci.

- Si le document de design a plus d'un niveau TOC, alors le modèle TOC pourrait avoir un nombre de levelrefs égal à la position séquentielle du niveau TOC étant référencé. La levelref correspondant au niveau TOC cible contiendra une seule TOCref dans le modèle TOC.
- Si le document de design a plus d'un niveau TOC, alors la seule TOCref dans le modèle TOC doit avoir une portée qui couvre tous les niveaux de document cibles, qui, en effet, seront aplatis en un seul niveau.

Disons que nous voulons recueillir toutes les images dans un document à une seule TOC à hiérarchie plate. Le design du document doit donc contenir au moins un niveau, et ce niveau doit contenir tous les signets TOC requis. Dans le modèle TOC, les images à énumérer doivent être référencées de la manière habituelle : (i) en créant un nombre correspondant de levelrefs ; et (ii) en créant une TOCref à l'intérieur de la levelref correspondante vers le niveau TOC cible. La TOCref aura le nom des signets TOC dans le niveau TOC cible.

Dans le modèle TOC affiché ci-dessous, il existe une levelref contenant une TOCref qui référence les signets TOC appelés `images`. Le champ d'application de la TOCref a été défini comme *Niveau actuel et en dessous*. En conséquence, tous les signets TOC appelés `images` au premier niveau et en dessous (à savoir, dans tout le document) seront référencés.



Si le design contient plus d'un niveau, et qu'une TOC plate est requise, disons, pour les items du deuxième niveau, alors le modèle TOC pourrait avoir deux levelrefs avec une TOCref uniquement à l'intérieur du deuxième niveau (pas de TOCref à l'intérieur du premier niveau). En alternative, la propriété du champ d'application des TOCrefs peut être utilisée pour préciser quel/s niveau/x dans le document du design être recherchés pour des signets d'un nom donné.

7.6.1.2 Créer des Signets TOC

Les signets TOC sont créés à l'intérieur d'un [niveau TOC](#) ²⁹⁸ dans le design de document. Ils peuvent être créés dans le modèle principal et/ou dans les modèles globaux. Un signet TOC a deux buts :

- Il marque un composant (statique ou dynamique) dans le design avec un nom statique que vous lui assignez. Il peut soit être joint ou ne pas joint dans le composant de design ; dans le dernier cas, il est vide. Dans la sortie, le signet TOC est instancié comme ancre identifié par un nom. Cet ancre nommé peut être référencé par les items dans la TOC (modèle).
- Un signet TOC définit également un string de texte qui sera utilisé comme texte dans l'item TOC. Ce string de texte peut être le contenu des éléments enfant du nœud où le marqueur est situé, ou il peut être la sortie d'une expression XPath.

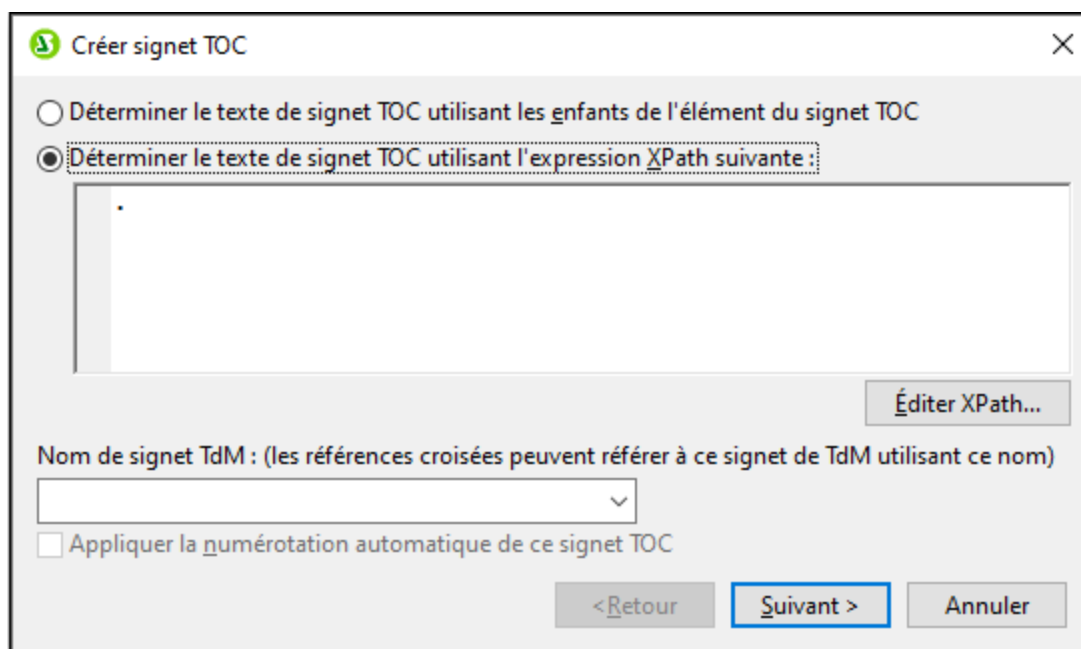
Vous pouvez créer un signet TOC de deux manières :

- En utilisant [Créer l'Assistant du signet TOC](#)³⁰¹, qui vous permet de préciser le nom du signet TOC, son entrée de texte, si la numérotation automatique doit être utilisée ou non, et le niveau à l'intérieur duquel il apparaît.
- En [insérant un signet TOC vide](#)³⁰², les propriétés seront définies ultérieurement.

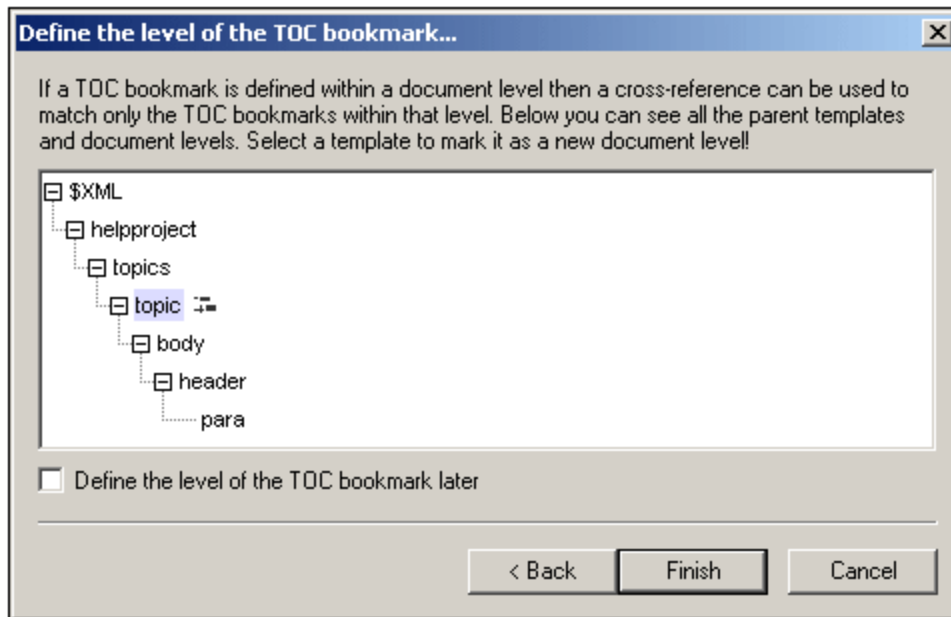
Créer un signet TOC avec l'Assistant de signets TOC

Pour créer un signet TOC utilisant l'Assistant des signets TOC, suivez les étapes suivantes :

1. Placez le curseur à un endroit du design où vous souhaitez insérer le signet TOC. En alternative, sélectionnez le composant de design autour duquel vous souhaitez insérer le signet TOC.
2. Depuis le menu contextuel (obtenu en utilisant la touche de droite) ou depuis le menu **Insérer**, sélectionnez **Insérer Table des matières | Signet TOC (Assistant)**. Si vous joignez un nœud avec un signet TOC, utilisez la commande **Enclose with | TOC Bookmark (Wizard)**. Ceci ouvre l'Assistant Créer Marqueur (*capture d'écran ci-dessous*).



3. Dans le premier écran de l'Assistant (*capture d'écran ci-dessus*) vous : (i) définissez le texte pour un item TOC ; (ii) définissez un nom de signet TOC ; et (iii) spécifiez si ce signet TOC devrait être numéroté dans la sortie. Pour la saisie de texte que vous pouvez sélectionner si le texte des éléments enfant doit être utilisé, ou une expression XPath. Pour le nom du signet TOC, vous pouvez saisir directement ou sélectionner depuis une liste déroulante contenant des noms de noms de signets TOC déjà spécifiés. Une fois que vous avez terminé, cliquez sur **Suivant**.
4. Dans le deuxième écran de l'assistant (*capture d'écran ci-dessous*), vous pouvez créer un niveau TOC sur un modèle si vous le souhaitez. Les modèles ancêtre de l'emplacement d'insertion sont affichés dans l'arborescence. Si un modèle a déjà été créé comme niveau TOC, ceci est indiqué par un symbole. Dans la capture d'écran ci-dessous, le symbole à côté du modèle `chapitre` indique qu'il a déjà été créé comme niveau. Si vous souhaitez créer un niveau supplémentaire à un des modèles ancêtre, sélectionnez ce modèle. En alternative, vous pouvez choisir de définir le niveau plus tard en cochant la case à cocher *Définir niveau plus tard*. Lorsque vous avez fait votre sélection, cliquez sur **Terminer**. (Notez que, si un modèle TOC existe déjà dans un modèle, sélectionner un tel modèle et cliquer sur **Terminer** ne créera pas de nouveau niveau TOC sur ce modèle.)



En cliquant sur **Terminer**, un signet TOC sera créé au point d'insertion et, s'il a été précisé dans le deuxième écran de l'assistant, un niveau TOC sera créé sur un modèle. Le signet TOC qui a été créé dans le niveau TOC se trouvera dans le niveau TOC qui le contient immédiatement. Par exemple, si le niveau TOC est le troisième niveau TOC dans la hiérarchie de niveau TOC, alors le signet TOC inséré se trouvera dans le troisième niveau TOC.

Créer un signet TOC

Pour créer un signet TOC sans attributs (nom de signet TOC, texte de l'item TOC, etc.), suivez les étapes suivantes :

1. Placez le curseur au point du document de design où vous souhaitez insérer le signet TOC, ou sélectionnez le composant de design autour duquel vous souhaitez insérer le signet TOC.
2. Depuis le menu contextuel (obtenu par clic droit) ou depuis le menu **Insérer**, sélectionnez **Insérer Table des matières | Signet TOC**. Un signet TOC est inséré. Ce signet TOC n'a ni de nom ni d'entrée de texte. Ceux-ci peuvent être définis par la suite utilisant les [Commandes d'édition](#) ³⁰³ (voir ci-dessous).

Insérer une numérotation hiérarchique ou séquentielle pour un composant

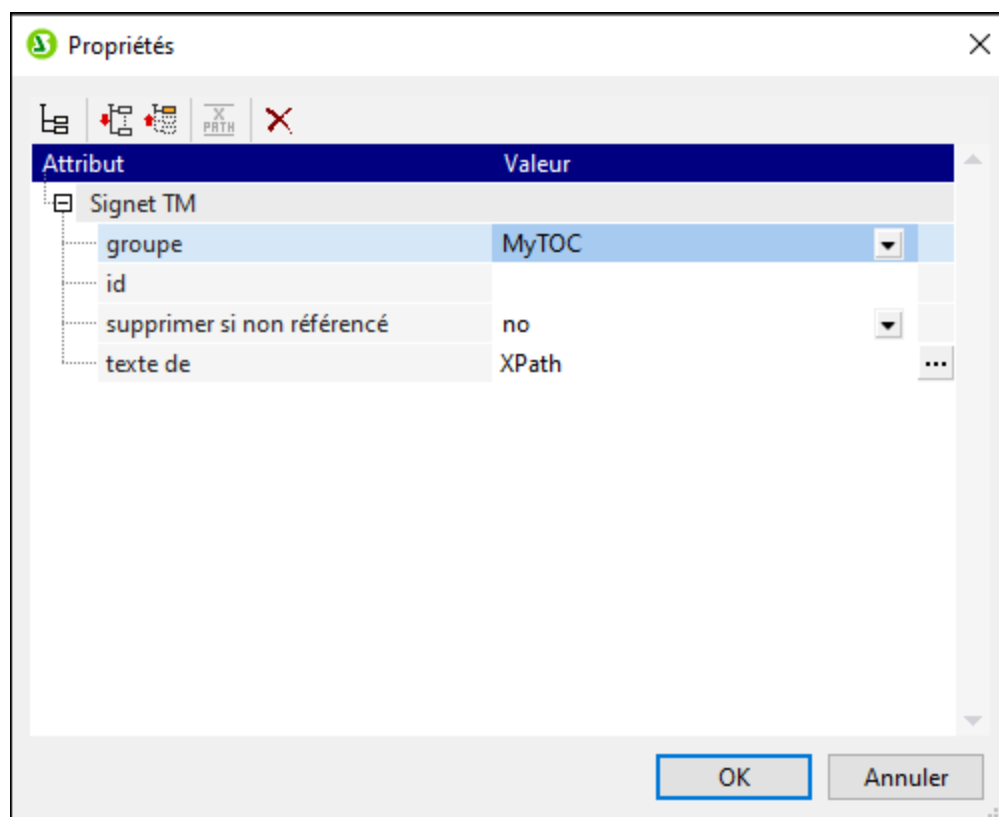
Une numérotation hiérarchique ou séquentielle à l'intérieur du corps de la marge du document de sortie (et non à l'intérieur de la TOC) peut être insérée à l'intérieur (mais pas à l'extérieur) des balises de signets TOC. Clic droit de la souris dans l'emplacement où vous souhaitez insérer la numérotation, puis sélectionnez **Insérer Table des matières | Numérotation hiérarchique / Numérotation séquentielle**. Par exemple, un signet TOC auto-numérotation qui est placé autour du modèle de l'en-tête du chapitre générera la numérotation pour tous les en-têtes de chapitre générés par le le modèle de l'en-tête du chapitre.

Notez que la numérotation est basée sur la structure des niveaux TOC. Donc, par exemple, si un élément d'en-tête du chapitre se trouve dans le premier niveau TOC, alors le quatrième du chapitre sera numéroté 4 car il s'agit de la quatrième instance de l'en-tête du chapitre à l'intérieur du premier niveau TOC. Si les sections d'un chapitre apparaissent au deuxième niveau TOC, alors la troisième section du quatrième chapitre sera

numérotée 4.3. Ceci est dû au fait que, à l'intérieur du premier (chapitre) TOC, il s'agit de la quatrième instance d'un chapitre, et à l'intérieur du deuxième (section) niveau TOC (du quatrième chapitre), il s'agit de la troisième instance d'une section.

Éditer le nom et l'entrée de texte d'un signet TOC

Le nom et l'entrée de texte du signet TOC peuvent être édités dans la fenêtre des Propriétés (*capture d'écran ci-dessous*). Pour éditer ces propriétés, sélectionnez le signet TOC, et éditez soit directement le programme dans la [fenêtre des Propriétés](#) ⁴⁸ ou cliquez avec la touche de droite sur le signet TOC et sélectionnez la propriété que vous souhaitez éditer.



Le signet TOC a les propriétés suivantes : (i) le nom du groupe de signets TOC (*Group*); (ii) une ID unique ; (iii) une option pour supprimer le signet s'il n'est pas référencé ; et (iv) une option (*Text From*) pour préciser l'entrée de texte, qui viendrait du contenu du signet ou d'une expression XPath.

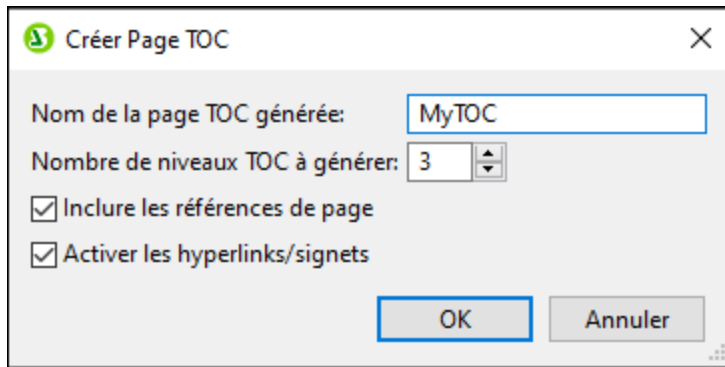
7.6.2 Créer le modèle TOC

Le modèle TOC est le modèle qui produit la table des matières dans la sortie. Il peut être créé partout à l'intérieur du design SPS, et de multiples modèles TOC peuvent être créés dans un design SPS unique.

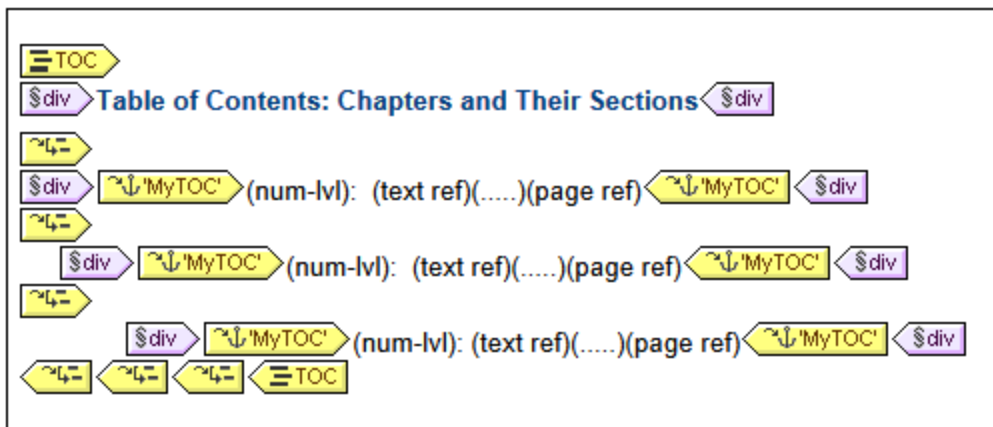
Les étapes à créer un modèle TOC sont les suivantes :

1. Placez le curseur à l'emplacement où le modèle TOC doit être inséré.

2. Cliquez dans la commande de menu **Insérer | Insérer Table des matières | Table des matières**. Ceci ouvre le dialogue Créer page TOC (*capture d'écran ci-dessous*). (En alternative, cette commande peut être accédée via le menu contextuel, qui apparaît quand vous cliquez avec la touche de droite.)



3. Saisir l'information requise dans le dialogue : (i) le nom de la page TOC générée est le nom (TOCref) qui sera utilisé pour référencer les [signets TOC](#) ³⁰⁰ dans le document du design. Si vous sélectionnez de multiples niveaux pour la TOC (références de niveau, pour être plus exact ; prochaine option), le même nom TOCref sera utilisé dans toutes les références de niveau (bien que des noms TOCref individuels peuvent être [édités par la suite](#) ³⁰⁶). (ii) Le nombre de [références de niveau TOC \(levelrefs\)](#) ²⁹⁸ spécifie combien de références de niveau la TOC doit avoir. (iii) Pour les médias imprimés, l'option vers les références de page de sortie (par ex., les numéros de page) est disponible. (iv) Les entrées de texte dans la TOC peuvent être utilisées comme liens dans les signets TOC.
4. Cliquez sur **OK** pour terminer. Le modèle TOC est créé avec le nombre spécifié de levelrefs (*capture d'écran ci-dessous ; le formatage du modèle TOC a été modifié de celui créé initialement*).



À l'intérieur de chaque levelref, il existe une TOCref ayant un nom qui identifie les signets TOC qui doivent être les items TOC pour la levelref. À l'intérieur de chaque TOCref, il existe un modèle par défaut pour l'item TOC, que vous pouvez [éditer à tout moment](#) ³⁰⁷.

Éditer le modèle TOC

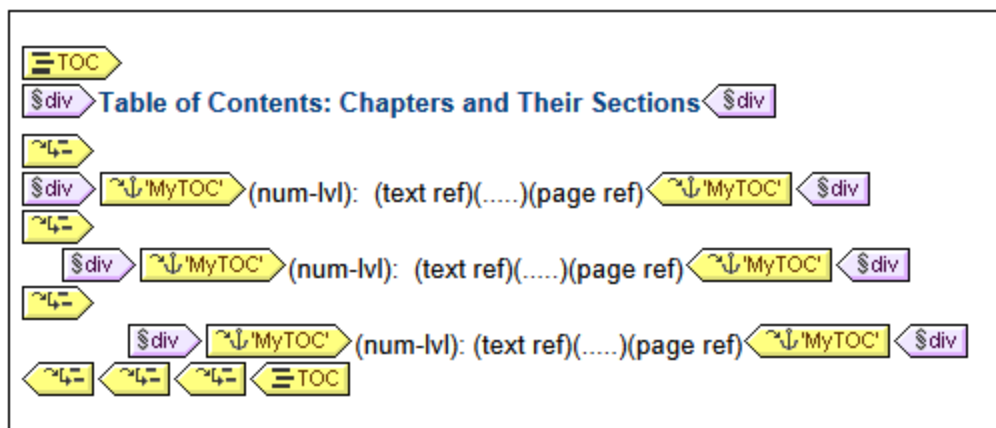
Les options d'édition suivantes sont disponibles :

- Le modèle TOC peut être glissé vers un autre emplacement dans SPS. Notez, toutefois, qu'un changement de nœud contextuel pourrait affecter des expressions XPath à l'intérieur d'un modèle TOC.

- Les [Levelrefs](#) ³⁰⁵ peuvent être ajoutés à ou supprimés de la structure du modèle TOC.
- Les [propriétés des références TOC individuelles](#) ³⁰⁶ (TOCrefs) peuvent être éditées. Le nom et champ d'application d'une TOCref peuvent être changés et vous pouvez choisir si l'item TOC correspondant à la TOCref est créé comme hyperlien ou non.
- Les [TOCrefs](#) ³⁰⁶ peuvent être ajoutés à ou supprimés de toute levelref dans le modèle TOC.
- L'[item TOC](#) ³⁰⁷ à l'intérieur d'une TOCref peut être formaté avec des propriétés CSS utilisant les [mécanismes StyleVision](#) ³⁴³ standard.
- Les fonctions SPS standard (telles les images, les Calculs automatiques, et les composants formatant les blocs) peuvent être insérées partout dans le module TOC.

7.6.2.1 Les Levelrefs dans le modèle TOC

Le [modèle TOC](#) ³⁰³ est structuré dans les **références de niveau (ou levelrefs)**; voir la capture d'écran ci-dessous. Ces niveaux sont initialement créés quand le modèle TOC est créé, et le nombre de levelrefs est spécifié dans le [dialogue Page TOC](#) ³⁰³.



Notez que les levelrefs sont imbriquées. Aux fins du design TOC, il existe une correspondance one-to-one entre les levelrefs dans le modèle TOC et les niveaux du design SPS. Donc, la première levelref du modèle TOC correspond au premier niveau dans le design SPS, la deuxième levelref dans le modèle TOC au deuxième niveau dans le design SPS, etc. Les TOCrefs à l'intérieur d'une levelref donnée dans le modèle TOC identifient les [signets TOC](#) ³⁰⁰ à l'intérieur d'un [champ d'application spécifié](#) ³⁰⁶ dans le design SPS. Par exemple, une TOCref peut spécifier que la TOCref cible les signets TOC dans le niveau du document correspondant, ou cible les signets TOC dans tous les niveaux de document, ou ceux du niveau de document actuel et des niveaux de document inférieurs.

Insérer et supprimer des levelrefs

Les levelrefs peuvent être insérées dans ou supprimées du modèle TOC une fois que le modèle TOC a été créé.

Pour insérer une levelref autour du contenu, sélectionnez le contenu dans le modèle TOC autour duquel la levelref doit être créée, puis, depuis le menu contextuel ou via la barre de menu, sélectionnez la commande **Enclose With | TOC Level Reference**. Vous pouvez également insérer une levelref vide au point d'insertion du curseur avec la commande de menu **Insérer | Insérer Table des matières | Référence de niveau TOC** (aussi disponible dans le menu contextuel).

Pour supprimer une levelref depuis le modèle TOC, sélectionnez la levelref à supprimer et appuyez soit sur la clé **Supprimer** ou sélectionnez **Remove** depuis le menu contextuel. Notez que seule la levelref sera supprimée—et non son contenu.

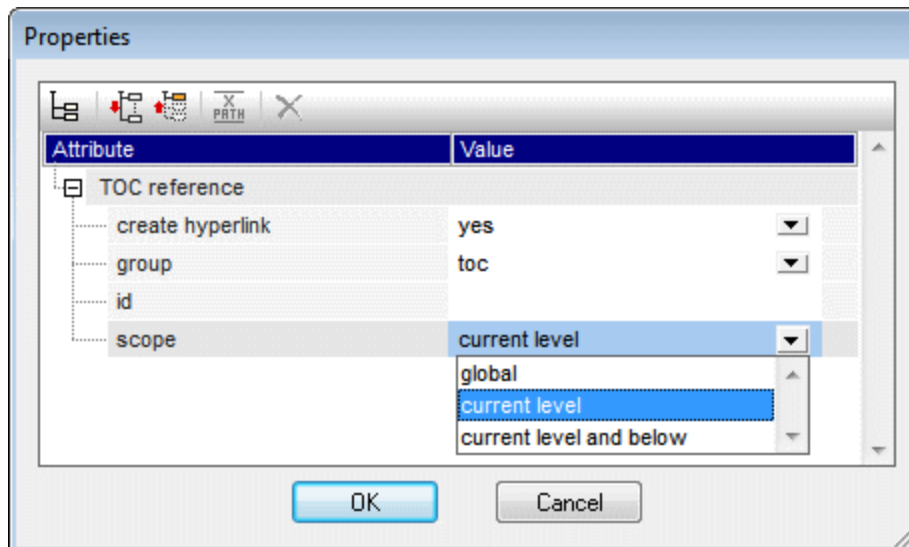
7.6.2.2 Références TOC : Nom, champ d'application, hyperlien

Les références TOC (TOCrefs) apparaissent à l'intérieur des références de niveau (levelrefs) et ont quatre propriétés (voir la capture d'écran ci-dessous) :

- Une propriété **hyperlien** qui peut être activée entre **oui** et **non** pour spécifier quels items TOC correspondants sont créés comme hyperliens ou non.
- Une propriété **group**, qui est le nom de la TOCref et identifie les signets TOC du même nom qui apparaît à l'intérieur du champ d'application spécifié (voir ci-dessous). Les signets TOC identifiés de cette manière fournissent les items à être inclus à la levelref de la TOC.
- Une **id** pour identifier la TOCref.
- Un **scope**, qui spécifie à quels niveaux correspondants dans le design SPS la TOCref s'applique. Trois options sont disponibles : (i) global, (ii) niveau actuel, (iii) niveau actuel et niveaux descendants (voir la capture d'écran ci-dessous).

Pour insérer une TOCref, placez le curseur à l'intérieur d'une levelref et, depuis le menu **Insérer** ou le menu contextuel, sélectionnez **Insérer Table des matières | Référence TOC**.

Pour éditer la propriété TOCref, cliquez avec la touche de droite sur la balise TOCref dans le modèle TOC et sélectionnez la propriété que vous souhaitez éditer (*Créer hyperlien*, *Éditer ID*, *Éditer Groupe*, ou *Éditer champ d'application*). Ceci ouvre la fenêtre des propriétés avec la propriété spécifiée sélectionnée pour l'édition (capture d'écran ci-dessous).

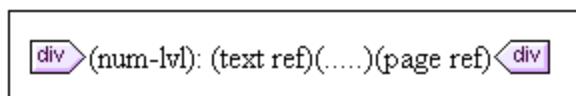


En alternative, avec la balise TOCref sélectionnée, allez directement à la propriété requise dans la fenêtre des Propriétés (groupe de propriétés *référence TOC*).

7.6.2.3 Formater les items TOC

L'item TOC peut contenir jusqu'à quatre composants standard, plus du contenu en option spécifié par l'utilisateur. Les quatre composants standard sont (*voir aussi capture d'écran ci-dessous*) :

- le point de suite de l'item TOC, indiquée dans le modèle TOC par (text ref)
- le leader entre l'entrée de texte et le numéro de page (pour la sortie de média paginée), indiqué par (.....)
- la référence de page de l'item TOC (pour la sortie de média paginée), indiquée par (page ref)
- la numérotation hiérarchique et séquentielle, indiquée par (num-lvl) et (num-seq), respectivement



Lorsqu'un modèle TOC est initialement créé, l'entrée de texte est automatiquement insérée à l'intérieur des TOCrefs. Si l'option Inclure Référence de Page a été sélectionnée, alors le point de suite et les composants de référence de page sont également inclus. Par la suite, des composants peuvent être insérés et supprimés depuis l'item TOC. Pour insérer un composant, placez le curseur au point d'insertion désiré à l'intérieur de l'item TOC, et dans le menu contextuel, sélectionnez **Insérer Table des matières | Référence TOC | Texte d'entrée / point de suite / Référence de page** ou **Insérer Table des matières | Numérotation hiérarchique / Numérotation séquentielle**, le cas échéant. (La numérotation hiérarchique devrait être insérée quand le design est structuré dans des niveaux imbriqués, la numérotation séquentielle quand il n'y pas de hiérarchie, à savoir, uniquement un niveau plat de TOC. Voir la note ci-dessous sur les TOC plats). Pour supprimer un composant, sélectionnez-le et appuyez sur la touche **Supprimer**.

De plus, vous pouvez insérer un contenu statique (par ex., du texte) et du contenu dynamique (par ex., les calculs automatiques) à l'intérieur de l'item TOC.

Niveaux dans les TOC plats

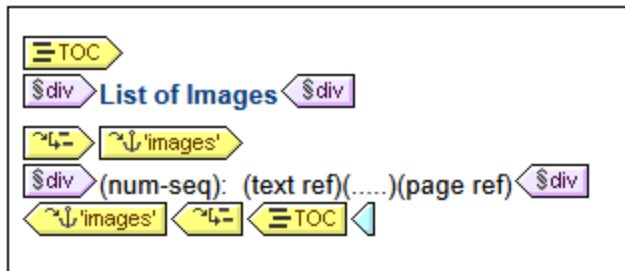
Dans une hiérarchie TOC plate, les items TOC seront la sortie à un seul niveau : par exemple, une liste simple des images dans un document.

Une hiérarchie plate peut être obtenue de plusieurs manières.

- Le document de design peut être structuré avec juste un seul niveau TOC. Le modèle TOC aura ensuite une seule levelref avec une référence TOC (TOCref) à l'intérieur de celle-ci.
- Si le document de design a plus d'un niveau TOC, alors le modèle TOC pourrait avoir un nombre de levelrefs égal à la position séquentielle du niveau TOC étant référencé. La levelref correspondant au niveau TOC cible contiendra une seule TOCref dans le modèle TOC.
- Si le document de design a plus d'un niveau TOC, alors la seule TOCref dans le modèle TOC doit avoir une portée qui couvre tous les niveaux de document cibles, qui, en effet, seront aplatis en un seul niveau.

Disons que nous voulons recueillir toutes les images dans un document à une seule TOC à hiérarchie plate. Le design du document doit donc contenir au moins un niveau, et ce niveau doit contenir tous les signets TOC requis. Dans le modèle TOC, les images à énumérer doivent être référencées de la manière habituelle : (i) en créant un nombre correspondant de levelrefs ; et (ii) en créant une TOCref à l'intérieur de la levelref correspondante vers le niveau TOC cible. La TOCref aura le nom des signets TOC dans le niveau TOC cible.

Dans le modèle TOC affiché ci-dessous, il existe une levelref contenant une TOCref qui référence les signets TOC appelés `images`. Le champ d'application de la TOCref a été défini comme *Niveau actuel et en dessous*. En conséquence, tous les signets TOC appelés `images` au premier niveau et en dessous (à savoir, dans tout le document) seront référencés.



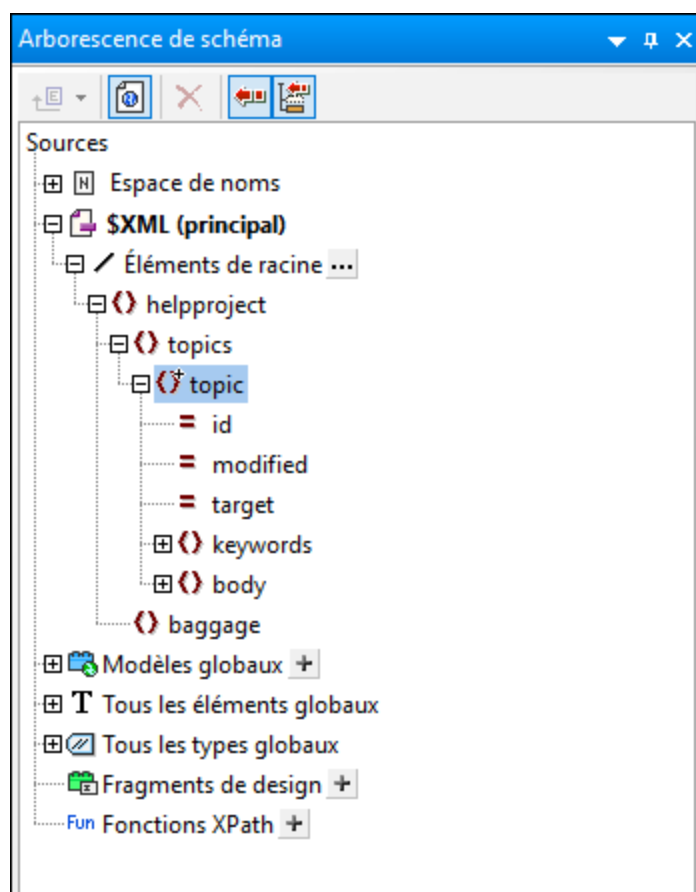
Si le design contient plus d'un niveau, et qu'une TOC plate est requise, disons, pour les items du deuxième niveau, alors le modèle TOC pourrait avoir deux levelrefs avec une TOCref uniquement à l'intérieur du deuxième niveau (pas de TOCref à l'intérieur du premier niveau). En alternative, la propriété du champ d'application des TOCrefs peut être utilisée pour préciser quel/s niveau/x dans le document du design être recherchés pour des signets d'un nom donné.

Formater les items TOC

L'item TOC peut être formaté avec les [styles CSS](#) ³⁴³ via la [barre latérale des Styles](#) ³⁵⁰. Les composants de l'item TOC individuels peuvent être formatés séparément en sélectionnant le composant et en lui attribuant des [propriétés de style](#) ³⁵⁰ dans la barre latérale des Styles.

7.6.3 Exemple : TOC simple

Un fichier exemple SPS pour démontrer l'utilisation de base des TOC, appelé `ChaptersSimple.sps`, se trouve dans le [dossier \(Mes\) Documents](#) ²⁴, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\TOC`. Cet SPS est basé sur un schéma qui définit le modèle de contenu d'un grand document basé sur le chapitre. La structure de schéma est affichée dans la capture d'écran ci-dessous et peut être consultée dans la fenêtre de l'arborescence de schéma de StyleVision quand vous ouvrez `ChaptersSimple.sps`. (Un exemple TOC plus complexe basé sur le même schéma est décrit dans la prochaine section de cette documentation, [Exemple : TOC hiérarchique et séquentiel](#) ³¹².)



L'élément de document est `helpproject`, qui contient un élément enfant `topics`. L'élément `topics` peut contenir un nombre illimité d'éléments `topic`, donc chacun peut contenir à son tour des éléments descendants `topic`. Le premier niveau des éléments `topic` peut être considéré être les chapitres du document, tandis que les éléments descendants `topic` sont des sections, sous-sections, etc.

Cet SPS crée une TOC, située en haut du document, qui recense les noms de chaque chapitre (les rubriques du premier niveau). Créer la TOC implique trois étapes :

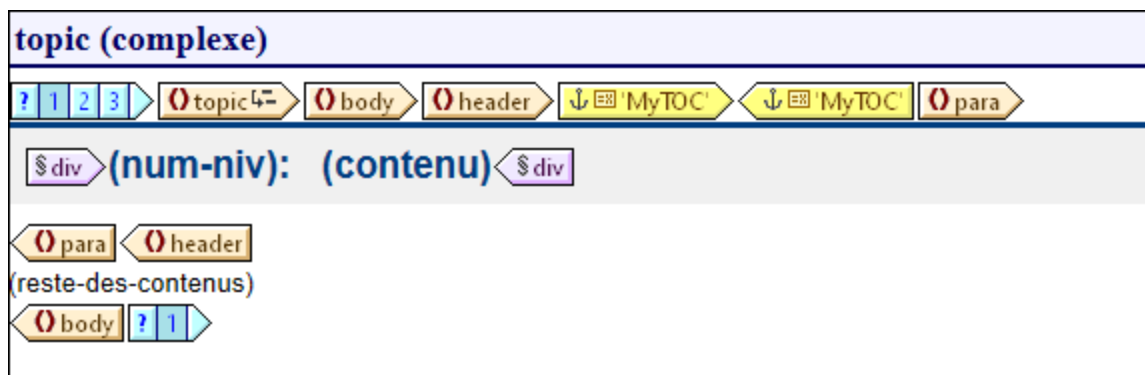
1. [Structurer le Design dans les niveaux TOC](#) ²⁹⁸ : Un ou plusieurs niveaux sont insérés dans le document de design pour structurer le document (sortie) de manière hiérarchique. Cette structure hiérarchique sera celle que la TOC reflétera. Dans notre exemple actuel, pour garder les choses simples, uniquement un **niveau TOC** a été créé—sur le modèle `Topic`. Car il n'existe qu'un niveau dans le design, le **modèle TOC**—quand il est créé par la suite, —ne peut avoir qu'un niveau dans sa structure (par ex., une référence à un niveau).
2. [Créer des Signets TOC](#) ³⁰⁰ : Un **signet TOC** est créé à l'intérieur du niveau TOC qui a été créé dans l'étape 1 (dans le document de design). Ceci permet aux références TOC dans le modèle TOC (qui sera créé dans la prochaine étape) de pointer de retour vers ce signet TOC. Le signet TOC spécifie aussi le texte qui apparaîtra dans l'item TOC qui y pointe.
3. [Créer le modèle TOC](#) ³⁰³ : Ceci est le modèle qui crée la TOC dans le document. Il est structuré en **références des niveaux (levelrefs)**, qui doivent correspondre à la structure des niveaux TOC dans le document de design. Par exemple, s'il existe trois levelrefs imbriqués dans le modèle TOC, alors le document de design doit avoir au moins trois niveaux imbriqués. Dans cet exemple, nous avons un seul levelref qui correspond à un seul niveau TOC dans le document de design. C'est à l'intérieur de la

levelref que la **référence TOC (TOCref)** est placée. C'est la TOCref qui génère les items TOC pour ce niveau dans la TOC.

La structure et les niveaux SPS

Regardez la structure du design dans la SPS. Notez qu'un modèle principal (avec les balises vertes `$XML`) contient une TOC. Le reste du modèle principal spécifie, à travers l'instruction `rest-of-contents`, que les modèles globaux et par défaut doivent être appliqués. Le reste du design SPS —à l'extérieur du modèle principal et après celui-ci—consiste en modèles globaux.

Les définitions TOC (niveaux TOC et signets TOC dans le design) se trouvent dans le modèle global pour `topic` (*capture d'écran ci-dessous*). Dans ce modèle global, une condition a été insérée pour séparer les éléments `topic` selon de combien d'éléments ancêtre `topic` chacun dispose, donc fournissant un traitement séparé (à l'intérieur de différentes branches conditionnelles) pour les chapitres, sections et sous-sections.



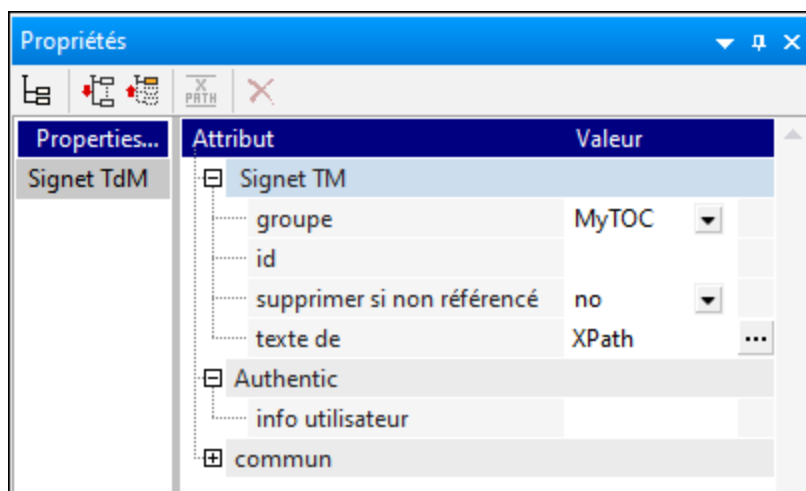
La capture d'écran ci-dessus affiche les contenus de la première branche conditionnelle, pour les éléments `topic` du premier niveau, de type chapitre. Notez qu'un niveau TOC a été créé sur le modèle « start-tag » de cet élément `topic`. Dans les deux autres branches conditionnelles, aucun niveau TOC n'a été créé sur le modèle `topic`. En conséquence, le document s'est vu assigné uniquement un niveau TOC, et ceci est au niveau de l'élément du premier niveau (type de chapitre) `topic`.

Créer le signet TOC

Un signet TOC (*balises jaunes dans la capture d'écran ci-dessous*) a été créé à l'intérieur de l'élément descendant en-tête de `topic` (mais à l'extérieur de l'élément `para`). Ce signet TOC sert d'ancrage pour chaque élément de niveau supérieur, chapitre-type `topic`.



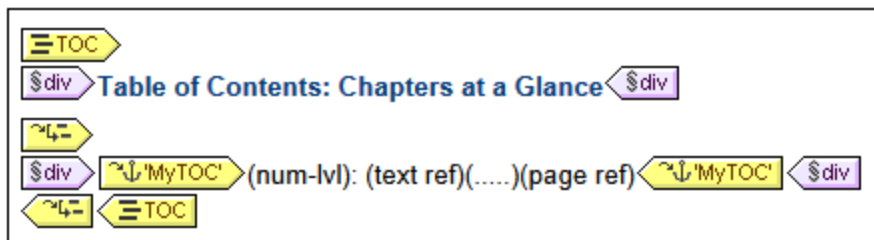
Le propriétés du signet TOC peuvent être éditées dans la barre latérale des Propriétés (*capture d'écran ci-dessous*).

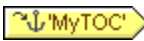


La propriété *Group* définit le groupe de signet TOC (et est le nom du signet TOC). Dans notre exemple, nous avons spécifié la valeur `MyTOC` pour cette propriété. Le groupe de signet sera référencé dans la TOC lorsqu'elle est créée, et il permet à différents groupes TOC d'être spécifiés à l'intérieur du même niveau. La propriété *ID* permet des ID uniques qui doivent être spécifiées pour les instances de signets créées. La propriété *Supprimer si pas référencé* est une option pour supprimer le signet s'il n'est pas référencé. La propriété *Text From* spécifie l'entrée de texte qui sera utilisée comme texte de l'item TOC dans la TOC. Le texte pourrait venir du contenu de signet (le contenu entre les balises de début et de fin du signet dans le design) ou depuis l'expression XPath. Dans notre exemple, une expression XPath est utilisée qui renvoie le texte de l'en-tête, respectivement, de chaque élément `topic` first-level.

Modèle TOC

À l'intérieur du modèle TOC (*capture d'écran ci-dessous*), une seule Référence de niveau (levelref)  a été intégrée. Cette levelref correspond au niveau TOC attribuée à l'élément de premier niveau, de type chapitre `topic` dans le design (*voir 'Structure et Niveaux SPS' ci-dessus*).



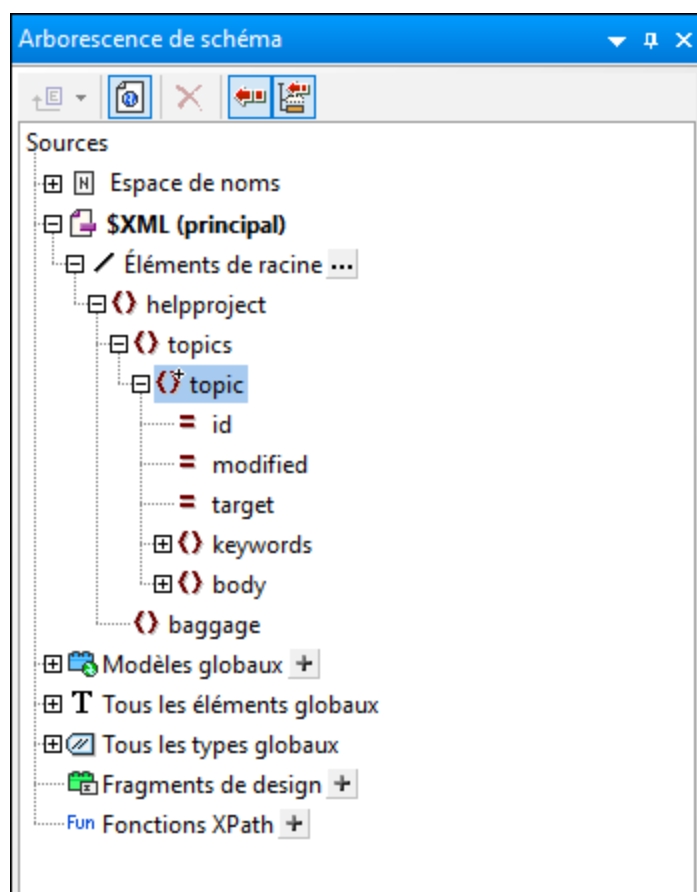
À l'intérieur de cette levelref, une référence TOC (TOCref)  a été insérée. Cette TOCref a été définie pour sélectionner les signets (i) qui sont dans le groupe Signet nommé `MyTOC` (*voir 'Créer le signet TOC' ci-dessus*), et (ii) qui sont dans le champ d'application du niveau actuel uniquement. Ces paramètres peuvent être réalisés dans la barre latérale des Propriétés quand la TOCref est sélectionnée, ou en cliquant avec la touche de droite sur la TOCref dans le design et en sélectionnant la commande d'édition pertinente depuis le menu contextuel..

L'aspect de l'élément TOC est spécifié à l'intérieur des balises TOCref de la TOC. Le format de numérotation, le texte, le point de suite et la référence de la page peuvent être insérés en cliquant avec la touche de droite à l'intérieur des balises TOCref et en sélectionnant le composant à insérer depuis le menu contextuel. Chacun de

ces composants peut être édité en le sélectionnant dans le design et en modifiant ses propriétés dans la barre latérale des Propriétés.

7.6.4 Exemple : TOC hiérarchiques et séquentielles

Un exemple de fichier SPS pour démontrer l'utilisation de des TOC, appelé `ChaptersSimple.sps`, se trouve dans le [dossier \(Mes\) Documents](#)²⁴, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\TOC`. Cet SPS est basé sur un schéma qui définit le modèle de contenu d'un grand document basé sur le chapitre. La structure de schéma est affichée dans la capture d'écran ci-dessous et peut être consultée dans la fenêtre de l'arborescence de schéma de StyleVision quand vous ouvrez `Chapters.sps`.



L'élément de document est `helpproject`, qui contient un élément enfant `topics`. L'élément `topics` peut contenir un nombre illimité d'éléments `topic`, donc chacun peut contenir à son tour des éléments descendants `topic`. Le premier niveau des éléments `topic` peut être considéré être les chapitres du document, tandis que les éléments descendants `topic` sont des sections, sous-sections, etc.

Le SPS contient trois TOC, situées en haut du document, dans l'ordre suivant :

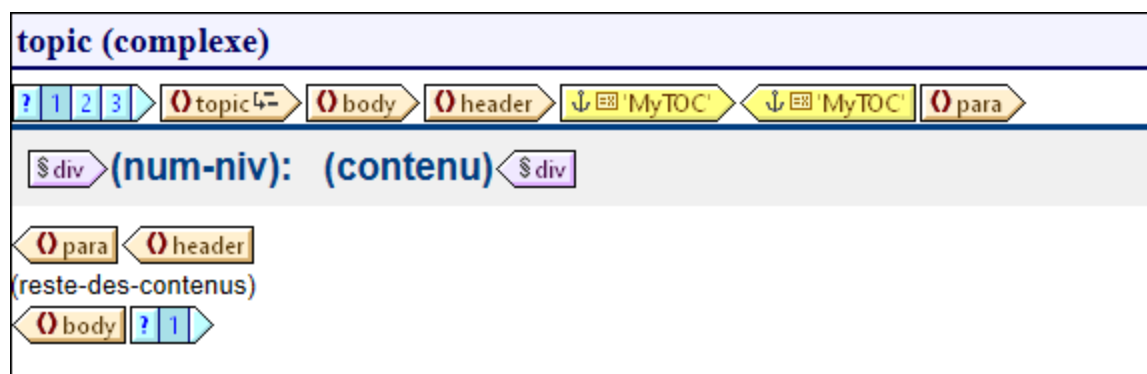
1. Les [chapitres en un coup d'œil](#)³¹³, qui recensent les noms de chaque chapitre (les premières rubriques de premier niveau).

2. Les [chapitres et leurs sections](#)³¹⁴, qui recensent chaque chapitre avec ses sections descendantes (rubriques de premier niveau, plus chaque hiérarchie de rubrique des sous-rubriques du premier niveau le plus inférieur, qui, dans le document XML accompagnant, `chapters.xml`, est la rubrique du troisième niveau)
3. La [liste des images](#)³¹⁵, qui est une liste plate de toutes les images dans le document (à l'exception de la première), recensée par le nom du fichier.

structure SPS

Avant de considérer les TOC en détail, jetez un coup d'œil à la structure du design. Notez qu'un modèle principal (avec les balises vertes `$XML`) contient une TOC. Le reste du modèle principal spécifie, à travers l'instruction `rest-of-contents`, que les modèles globaux et par défaut doivent être appliqués.

Les définitions TOC se trouvent dans les modèles globaux pour `topic` et `image`. Dans le modèle global pour `topic` (*capture d'écran ci-dessous*), un niveau TOC a été créé dans l'élément `topic`, et un signet TOC a été créé à l'intérieur de l'élément enfant `header` (mais en dehors de l'élément `para`).



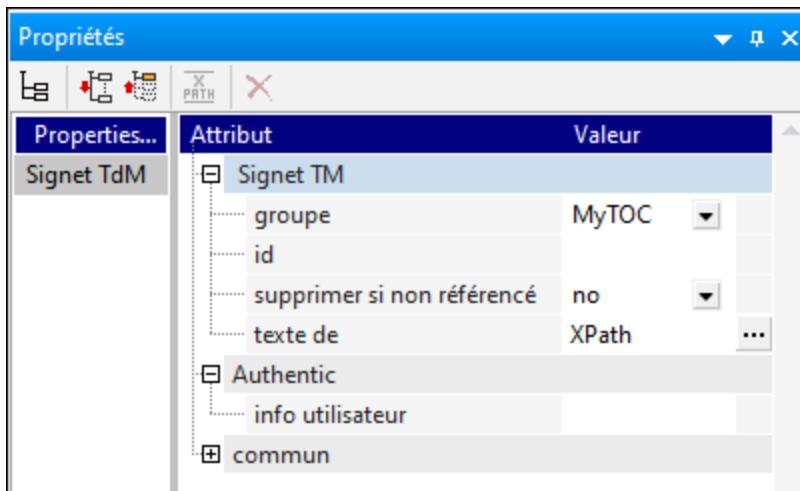
Depuis que l'élément `topic` est récursif, le niveau TOC et le signet TOC seront aussi récursifs. Ceci signifie que, lors de la première récursion, un nouveau niveau hiérarchique subordonné et un nouveau signet TOC sont créés. Ce processus continue pour chaque rubrique descendante, créant ainsi une hiérarchie de niveaux TOC descendants, dont chacun est attribué un signet TOC correspondant. Puisque le formatage de l'en-tête (le titre de la rubrique) pour chaque niveau TOC doit être différent, nous avons enveloppé chaque niveau à l'intérieur d'une branche séparée de la condition avec trois branches. Chaque branche teste le niveau TOC dans lequel une rubrique apparaît : premier, deuxième ou troisième niveau.

Notez que la numérotation hiérarchique (`num-lvl`) a été insérée à l'intérieur d'un niveau. Ceci est fait en cliquant avec la touche de droite de la souris sur l'emplacement requis et en sélectionnant **Insérer Table des matières | Numérotation hiérarchique**. L'effet est d'insérer le numéro hiérarchique correct avant chaque rubrique dans le **flux du texte du document**, par exemple, 3.1 ou 4.2.3.

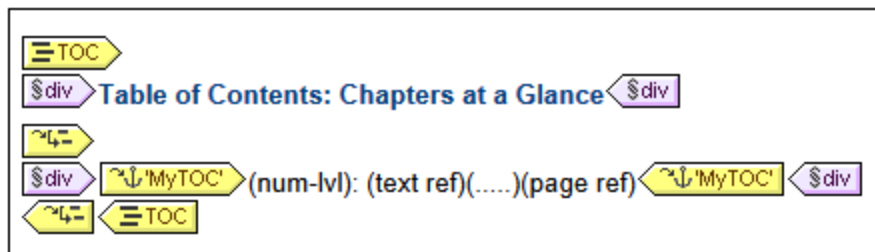
Descriptions TOC

Ci-dessous, vous trouverez une brève description de chaque TOC et les points à noter.

Les chapitres en un coup d'œil : Sélectionnez le signet TOC dans le modèle global pour `topic`. Dans la barre latérale des Propriétés (*capture d'écran ci-dessous*), notez que le texte d'entrée a été défini pour être créé utilisant une expression XPath. Lorsque vous cliquez sur le bouton Éditer dans le champ de valeur de la propriété `Text from`, vous verrez que l'expression XPath a été définie comme `para`. Ceci signifie que les contenus de l'enfant `para` de `header` (puisque le signet TOC a été inséré à l'intérieur de l'élément `header`) seront utilisés comme texte de l'item TOC.

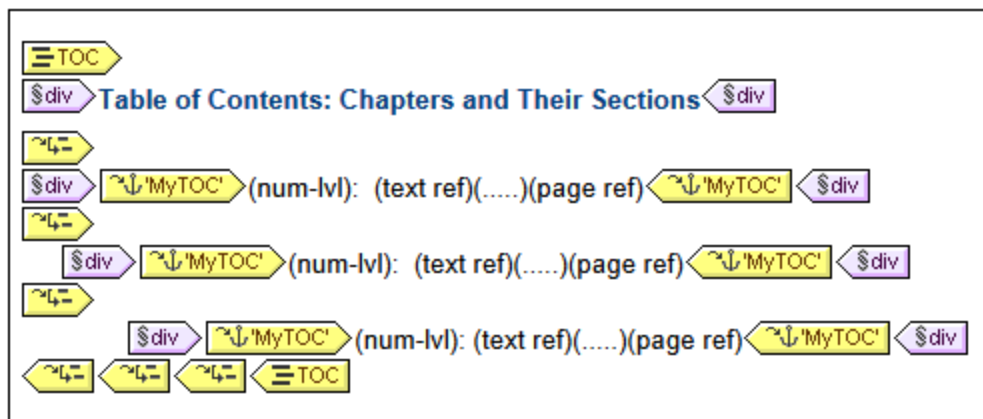


Le modèle TOC lui-même (*capture d'écran ci-dessous*) contient une référence de niveau (levelref) , et la TOCref à l'intérieur de cette levelref  a été définie pour sélectionner les signets TOC nommés MyTOC à l'intérieur du champ d'application du niveau actuel uniquement — qui est le premier niveau. Le résultat étant que les items TOC seront créés uniquement pour les rubriques de premier niveau.



Veuillez noter que la numérotation hiérarchique a été définie en tant que numérotation hiérarchique.

Les chapitres et leurs sections : Dans cette TOC (*capture d'écran ci-dessous*), notez que les trois levelrefs imbriquées ont été définies, chacune contenant une TOCref pour laquelle le champ d'application est le niveau actuel.



Puisque chaque item TOC est contenu dans un bloc `div`, formater les propriétés (y compris l'indentation) peut être défini pour le bloc.

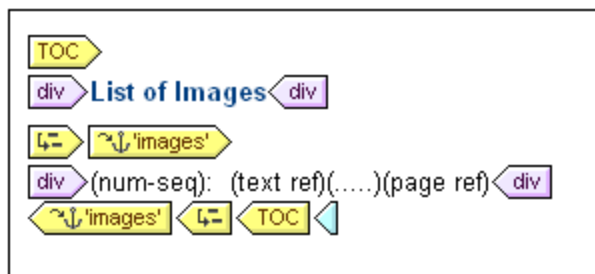
Liste des images : La liste des images est une liste plate. D'abord, considérez à l'intérieur de quels niveaux les images apparaîtront dans le document instancié. L'élément `image` est un enfant de l'élément `para`. Puisque les niveaux ont été créés dans les éléments `topic`, les éléments `image` apparaîtront à l'intérieur du premier, deuxième et/ou troisième niveaux du document. Il est donc nécessaire de créer tout niveau pour l'élément `image`.

Dans le modèle global pour l'`image`, la condition (voir la capture d'écran ci-dessous) active un traitement séparé pour (i) la première image (qui est présentée dans cet exemple), et (ii) les autres images (qui, à des fins d'économie, ne sont pas présentées dans cet exemple).



Notez que le signet TOC est placé à l'intérieur de la deuxième branche de la condition ; ceci signifie que les images sélectionnées dans la première branche ne sont pas bookmarquées. Aussi, la numérotation séquentielle (`num-seq`) des images, insérée avec **Insérer Table des matières | Numérotation séquentielle**, démarrera avec la deuxième image (car la première image est sélectionnée dans la première branche de la condition). Une autre fonction à noter est que la numérotation peut être formatée, comme cela a été fait dans ce cas. Pour voir le formatage, clic droit sur (`num-seq`), puis sélectionnez **Éditer Format**. Dans la boîte de dialogue qui s'ouvre, vous verrez que le formatage a été défini comme 01, indiquant que 0 sera inséré devant les numéros à un chiffre.

Dans le modèle TOC pour les images (capture d'écran ci-dessous), notez qu'il y a une seule TOCref identifiant les signets nommés `images`, et que cette TOCref se trouve à l'intérieur de la levelref. Le champ d'application de la TOCref (éditable dans la fenêtre des Propriétés quand la TOCref est sélectionnée) a été défini comme : niveau actuel et inférieur. Le niveau actuel, déterminé par la levelref, est le premier niveau. Les niveaux ci-dessous seront le deuxième, le troisième, etc. De cette manière, toutes les images du premier niveau vers les bas sont sélectionnées comme items dans la TOC.



Puisque la numérotation sélectionnée est séquentielle, les images sont numérotées de manière séquentielle dans une liste plate.

7.6.5 Numérotation automatique dans le corps du document

Les instances répétitives d'un nœud peuvent être numérotées automatiquement dans le corps principal d'un document utilisant la fonction de numérotation automatique. Par exemple, dans un élément `Book` qui contient de multiples éléments `Chapter`, chaque élément `Chapter` peut être numéroté automatiquement utilisant la fonction de numérotation automatique. Ceci est une manière facile d'insérer la numérotation basée sur la structure du document XML.

Note : la fonction de numérotation automatique se réfère à la numérotation à l'intérieur du corps principal du document. Elle **ne se réfère pas** à la numérotation à l'intérieur des tables des matières (TOC), où la numérotation est considérée être une propriété de l'item TOC.

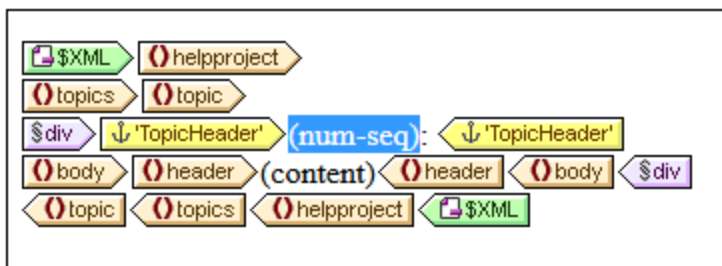
La numérotation automatique peut être soit séquentielle (plate) ou hiérarchique. La numérotation séquentielle fournit une numérotation normale à un seul niveau. La numérotation hiérarchique est basée sur la hiérarchie du niveau TOC créée dans le document et crée la numérotation selon la position de l'élément dans la hiérarchie du niveau TOC.

Une large gamme de formatage est disponible pour les numéros. Dans le cas de numéros hiérarchiques, les jetons de numéro individuels peuvent être formatés séparément. Par exemple, un numéro à trois jetons pourrait avoir le format suivant : A.1.1., là où chacun des trois jetons a un format de numéro différent. Le formatage de numéro est attribué différemment selon qu'il s'agit d'une numérotation séquentielle ou hiérarchique, et a donc des descriptions séparées, chacune dans leurs sections respectives ci-dessous.

Numérotation séquentielle (num-seq)

La numérotation séquentielle (ou plate) peut être insérée à l'intérieur d'un [signet TOC](#) ³⁰⁰ dans le design de document (voir la capture d'écran ci-dessous). Créer une numérotation séquentielle comme suit :

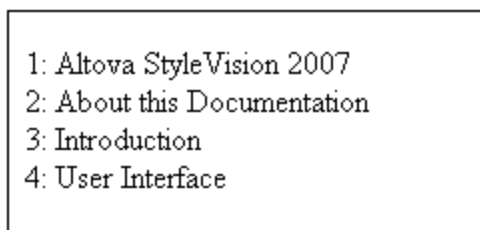
1. Placez le curseur à l'intérieur d'un nœud qui doit être numéroté et créez le signet TOC (clic droit, et sélectionnez **Insérer Table des matières | Signet TOC**). Le signet TOC sera créé. Dans la capture d'écran ci-dessous, nous voulons numéroté l'élément `topic` ; donc le signet TOC a été créé à l'intérieur de l'élément `topic`. L'emplacement exact à l'intérieur de l'élément `topic` dépend où vous voulez avoir la numérotation dans la mise en page. Dans la capture d'écran ci-dessous, la numérotation est placée immédiatement à gauche de l'en-tête du chapitre (titre.)
2. Placez le curseur à l'intérieur des balises du signet TOC, clic droit, et sélectionnez **Insérer Table des matières | Numérotation séquentielle**. Ceci insère l'espace réservé de la numérotation automatique pour la numérotation séquentielle, `(num-seq)` (mis en surbrillance à l'intérieur du signet TOC « TopicHeader » dans la capture d'écran ci-dessous).



3. Si le signet TOC est en train d'être référencé de l'intérieur du modèle TOC, alors vous pouvez saisir les propriétés de signet TOC, tel que requis. Toutefois, si le signet TOC ne sera utilisé que pour la

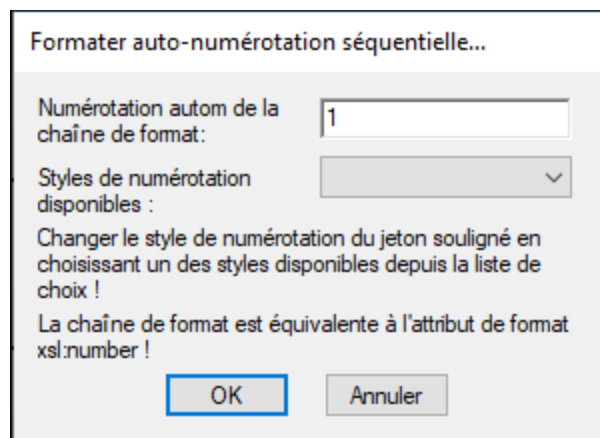
numérotation séquentielle, ce n'est pas la peine de lui donner un nom. Si vous souhaitez lui donner un nom, clic droit et sélectionnez la commande **Edit Group**.

Dans l'exemple affiché dans la capture d'écran ci-dessus, la numérotation séquentielle a été définie sur le nœud `topic`. Le résultat est que chaque élément `topic` obtient un numéro séquentiel, tel qu'affiché dans la capture d'écran ci-dessous. Notez que la numérotation est essentiellement la position de chaque élément `topic` à l'intérieur de la séquence de tous les éléments frère `topic` au niveau de la hiérarchie du document XML.



Note : si la numérotation séquentielle doit être poursuivie sur un autre ensemble de nœuds, utilisez le signet TOC avec le même nom sur les deux nodesets.

Pour formater la numérotation séquentielle, clic droit sur l'espace réservé `num-seq` et sélectionnez la commande **Éditer format**. Ceci fait apparaître le dialogue Formater auto-numérotation séquentielle (capture d'écran ci-dessous).



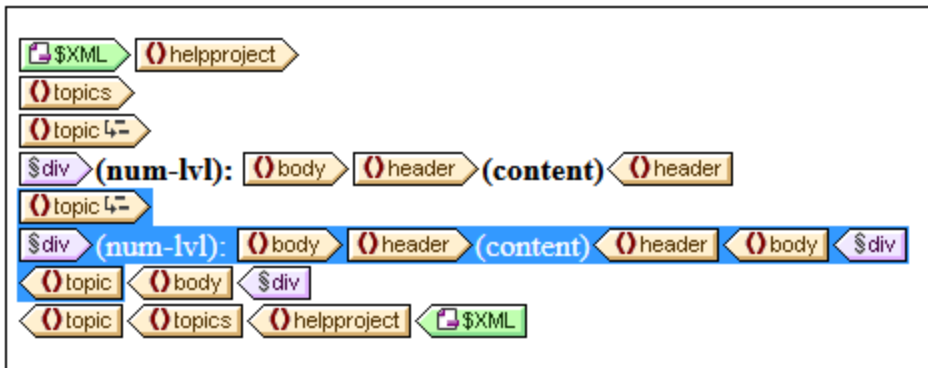
Sélectionnez le format depuis la case déroulante de la liste déroulante *Styles de numérotation disponible* (voir la capture d'écran ci-dessus) et cliquez sur **OK** pour appliquer le format sélectionné.

Numérotation hiérarchique (num-lvl)

La numérotation hiérarchique peut être insérée à l'intérieur d'un [niveau TOC dans le design](#)²⁹⁸. Pour créer une numérotation hiérarchique dans un document, vous devez d'abord structurer le document dans les niveaux TOC. Suivez l'étape telle que décrite dans la section [Structurer le Design dans les Niveaux](#)²⁹⁸. Veuillez noter les points suivants :

- Les niveaux doivent être créés soit sur le nœud à numérotter ou à l'intérieur de celui-ci.
- Les niveaux doivent être imbriqués selon la hiérarchie de la numérotation requise (voir capture d'écran ci-dessous).

- L'espace réservé de la numérotation hiérarchique doit être inséré à l'intérieur du niveau correspondant dans le design (voir la capture d'écran ci-dessous).



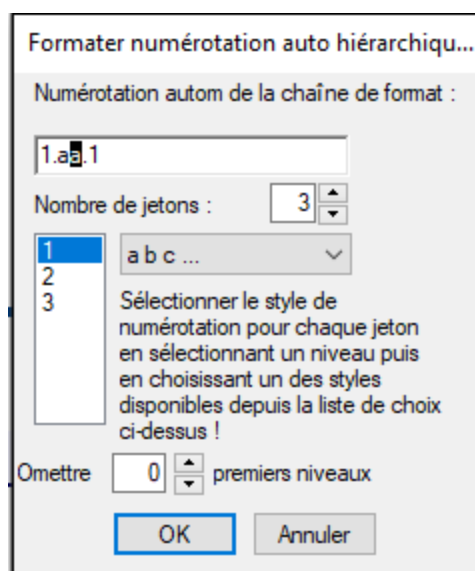
Dans la capture d'écran ci-dessus, il existe deux niveaux. L'élément `topic` est récursif, et un niveau a été créé sur deux éléments `topic` (en cliquant avec la touche de droite sur la balise du nœud et en sélectionnant **Modèle sert de Niveau**). Un élément `topic` (en surbrillance dans la capture d'écran ci-dessus) est imbriqué à l'intérieur de l'autre. Le résultat étant que les niveaux sont également imbriqués. À l'intérieur de chaque niveau, l'espace réservé de la numérotation hiérarchique (`num-lvl`) a été inséré (clic droit à l'intérieur du niveau et sélectionnez **Insérer Table des matières | Numérotation hiérarchique**).

Le résultat du design affiché dans la capture d'écran ci-dessus aura l'aspect suivant.

1: Altova StyleVision 2007
2: About this Documentation
3: Introduction
3.1: What Is an SPS?
3.2: Product Features
3.3: Setting up StyleVision
4: User Interface
4.1: Main Window
4.2: Design Entry Helpers

Le premier niveau est affiché en gras, le deuxième de manière normale.

Pour formater la numérotation hiérarchique, clic droit sur l'espace réservé `num-lvl` et sélectionnez la commande **Éditer format**. Ceci ouvre le dialogue Formater auto-numérotation hiérarchique (capture d'écran ci-dessous).



Sélectionnez d'abord le numéro des jetons dans la liste de choix Jeton. Ce nombre devrait être le même que le numéro des niveaux TOC dans le document. Chaque jeton peut ensuite être formaté séparément. Dans la case inférieure des deux cases d'affichage, sélectionnez le jeton à formater. (Dans la capture d'écran ci-dessus, le deuxième jeton a été sélectionné.) Ensuite, dans la liste de choix déroulante Formater, sélectionnez le style de formatage que vous voulez. Dans la capture d'écran ci-dessus, le formatage en lettres minuscules a été sélectionné pour le deuxième jeton, et ceci est reflété dans la case de l'affichage du dialogue. En plus, les niveaux peuvent être omis en saisissant le numéro requis des niveaux à omettre dans la case Omettre niveaux.

Notez que le formatage est défini sur la numérotation hiérarchique un niveau à la fois. Donc l'espace réservé de la numérotation hiérarchique `num-lvl` à chaque niveau doit être formaté séparément.

Cliquez sur **OK** lorsque vous avez terminé.

7.6.6 Référencement croisé

Une référence croisée est une référence vers une autre partie du document. Dans un SP, une référence croisée est créée en deux parties : Premièrement, en définissant la cible de la référence croisée. Deuxièmement, en définissant le lien vers la cible. Définir une cible consiste à créer un signet TOC à l'intérieur d'un niveau TOC. Le lien vers cette cible est une Référence de texte à l'intérieur de la référence TOC (TOCref). La Référence de texte génère le texte de sortie et sert de lien. Pour cette raison, créer une référence croisée consiste en les trois étapes suivantes :

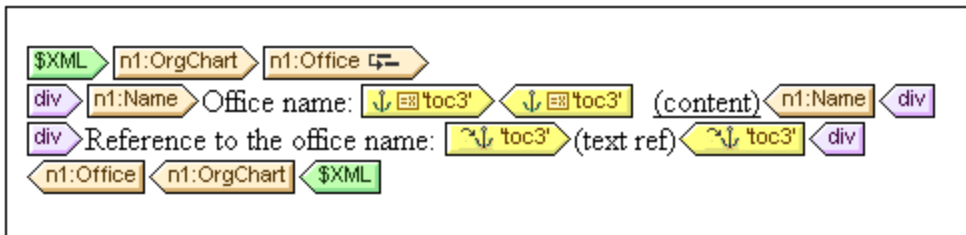
Étape 1 : Niveaux

Le document est structuré en niveaux TOC, tel que décrit dans la section [Structurer le Design en Niveaux](#) ²⁹⁸. Les niveaux TOC seront utilisés au cours du référencement pour spécifier le champ d'application du référencement. Seuls ces signets TOC ayant le nom spécifié et relevant du champ d'application spécifié seront ciblés. Dans la capture d'écran ci-dessous, un niveau a été créé sur l'élément `n1:Office`.

Étape 2 : Créer des Signets TOC

À l'intérieur d'un niveau, un signet TOC est créé en plaçant le curseur à l'emplacement requis, clic droit, et en sélectionnant **Insérer Table des matières | Signet TOC**. Le signet TOC obtient un nom et une expression XPath qui génèrent le texte de sortie. L'expression XPath identifiera généralement un nœud dans le document, dont les contenus sont précisés dans le texte requis.

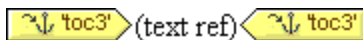
Dans la capture d'écran ci-dessous, le signet TOC à l'intérieur de l'élément `n1:Name`  a un nom de `toc3` et une expression XPath qui situe le nœud actuel. Cela signifie que le texte de sortie sera le contenu du nœud `n1:Name`.



Quand le document XML est traité, l'ancre est créée pour chaque élément `n1:Name`. Cette ancre aura une référence de texte (le texte de la référence croisée) qui est la valeur de l'élément `n1:Name`.

Étape 3 : Créer des Références TOC

Une référence TOC (TOCref) est insérée (menu contextuel, **Insérer Table des matières | Référence TOC**) pour créer un lien avec les ancres générées par un signet TOC.



Dans la capture d'écran ci-dessus, la TOCref appelée `toc3` (*capture d'écran ci-dessus*) est à l'intérieur du même niveau TOC que le signet TOC qu'elle référence (le niveau `Office`). Vous devez aussi spécifier le champ d'application de la TOCref. Le champ d'application spécifie quels niveaux TOC doivent être recherchés pour les signets TOC du même nom que la TOCref. Dans l'exemple affiché ci-dessous, le champ d'application est le niveau actuel. Ceci signifie que les signets TOC à l'intérieur d'un niveau actuel qui sont dénommés `toc3` sont ciblés par cette référence.

La capture d'écran ci-dessus affiche un modèle `n1:Office`. Lorsqu'un nœud `n1:Office` est traité, une ancre est créée avec le texte de sortie qui est le contenu du nœud `n1:Name`. Ceci est en raison du fait que le signet TOC spécifie une expression XPath (via la propriété *Text from* du signet TOC) que les contenus de ce nœud seront le texte de sortie. La TOCref dans la prochaine ligne identifie l'ancre dénommée `toc3`, et le composant de référence de texte génère le texte de sortie du lien vers l'ancre (*texte en mauve dans la capture d'écran ci-dessus*). La sortie aura l'aspect suivant :

Office name: Nanonull, Inc.
 Reference to the office name: Nanonull, Inc.
 Office name: Nanonull Europe, AG
 Reference to the office name: Nanonull Europe, AG

Dans l'exemple ci-dessus, le champ d'application été défini pour le niveau actuel. Vous avez deux autres possibilités pour le champ d'application : (i) un champ d'application global, (ii) le champ d'application pour le niveau actuel et ci-dessous. Avec ces options, il est possible de générer également les signets TOC dans d'autres niveaux du design.

7.6.7 Signets et hyperliens

Dans le document SPS, les signets peuvent être insérés partout à l'intérieur du design. Ces signets sont transformés en ancres dans la sortie, qui peuvent être liées depuis des hyperliens. Les hyperliens ne peuvent non seulement être liés vers des signets, mais aussi vers des ressources externes comme les pages Web. StyleVision propose une flexibilité considérable en matière de création d'URI cibles pour les hyperliens.

Dans cette section, nous décrivons :

- Comment les [signets](#)³²¹ peuvent être insérés dans le SPS.
- Comment les [hyperliens](#)³²³ peuvent être insérés dans le SPS et comment ils se mettent en lien vers les pages cibles.

Note : les liens aux documents externes sont pris en charge dans la spec FO, mais ne sont éventuellement pas pris en charge par le processeur FO que vous utilisez. Vous devriez vérifier avec le support pour cette fonction que vous prévoyez utiliser.

7.6.7.1 Insérer des signets

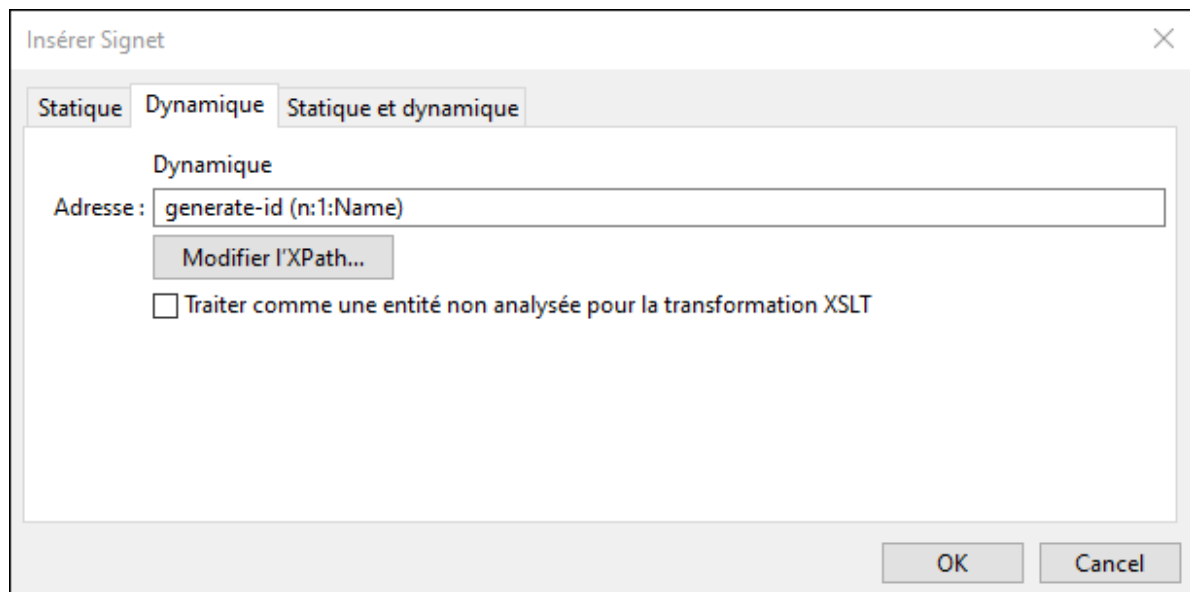
Un signet (ou ancre) peut être inséré partout dans le SPS, à un point d'insertion du curseur ou autour d'un composant SPS.

Les signets sont créés dans le SPS via le dialogue Insérer Signet (*capture d'écran ci-dessous*) Dans ce dialogue, vous définissez un nom du signet. Le nom peut être un nom statique, ou il peut être un nom dynamique qui est (i) dérivé du contenu de document XML, ou (ii) généré arbitrairement avec une expression XPath.

Créer un signet

Pour insérer un signet, faites comme suit :

1. Placez le curseur à l'emplacement où vous souhaitez créer le signet.
2. Sélectionnez la commande de menu [Insérer | Insérer Signet](#)⁴⁹⁸, ou clic droit et sélectionnez **Insérer | Signet**.
3. Dans le dialogue Insérer Signet (*capture d'écran ci-dessous*), sélectionnez un onglet selon si le nom du signet doit être statique (onglet statique), obtenu de manière dynamique depuis le document XML ou généré de manière arbitraire depuis une expression XPath (dynamique), ou composé de deux parties, statique et dynamique (statique et dynamique). Dans la capture d'écran ci-dessous, un signet dynamique est créé, qui a le nom qui est une ID unique pour chaque enfant `Name` du nœud contextuel.



4. Cliquez sur **OK**. Le signet est défini.

Une fois qu'un signet a été créé, il peut être [mis en lien par le biais d'un hyperlien](#) ³²³.

Note : les signets sont créés à l'emplacement spécifié dans le design. Si cet emplacement se trouve à l'intérieur d'un élément répétitif, un signet est créé à l'intérieur de chaque instance de l'élément répétitif. Si un nom statique est donné, chaque signet aura le même nom. Pour cette raison, il est mieux, dans de tels cas (éléments répétitifs), de donner un nom dynamique, qui peut être, par exemple, le nom d'un élément enfant du nœud contextuel (l'élément à l'intérieur duquel le signet est créé). Si le nœud sélectionné pour le nom dynamique pourrait avoir le même contenu dans les multiples instances, alors le caractère unique du nom du signet peut être assuré en utilisant la fonction `generate-id()` pour générer le nom (*voir la capture d'écran ci-dessus*). Pour référencer un tel signet, la même ID peut être générée comme valeur `href` d'un [hyperlien](#) ³²⁶. Dans ce cas, assurez-vous d'utiliser l'identificateur de fragment # devant la fonction `generate-id()`. L'expression XPath serait : `concat('#', generate-id(nodeXXX))`.

Modifier un signet

Une fois que le signet a été créé, son nom peut être modifié via le dialogue Éditer Signets. Ce dialogue est accédé comme suit :

1. Sélectionnez le signet dans le design.
2. Dans la barre latérale des Propriétés, cliquez sur le bouton **Éditer** de la propriété `Nom du signet` (*capture d'écran ci-dessous*) dans le groupe des propriétés *Signet*. Ceci ouvre le dialogue Éditer Signet, qui est identique au dialogue Insérer Signet décrit ci-dessus (*voir la capture d'écran ci-dessus*).



3. Dans le dialogue Éditer Signet, éditez le nom d'un signet soit dans l'onglet statique, dynamique ou statique et dynamique.

Supprimer un signet

Pour supprimer un signet, sélectionnez-le dans le design et appuyez sur la touche **Supprimer**.

7.6.7.2 Définir des hyperliens

Des hyperliens peuvent être créés autour de composants SPS tels un texte ou des images. Les cibles d'hyperliens peuvent être : (i) des signets dans le design SPS, ou des (ii) ressources externes, telles des pages web ou des messages e-mail. Dans cette section, nous discutons d'abord du contenu de l'hyperlien (texte, image, etc.), puis de la cible de l'hyperlien.

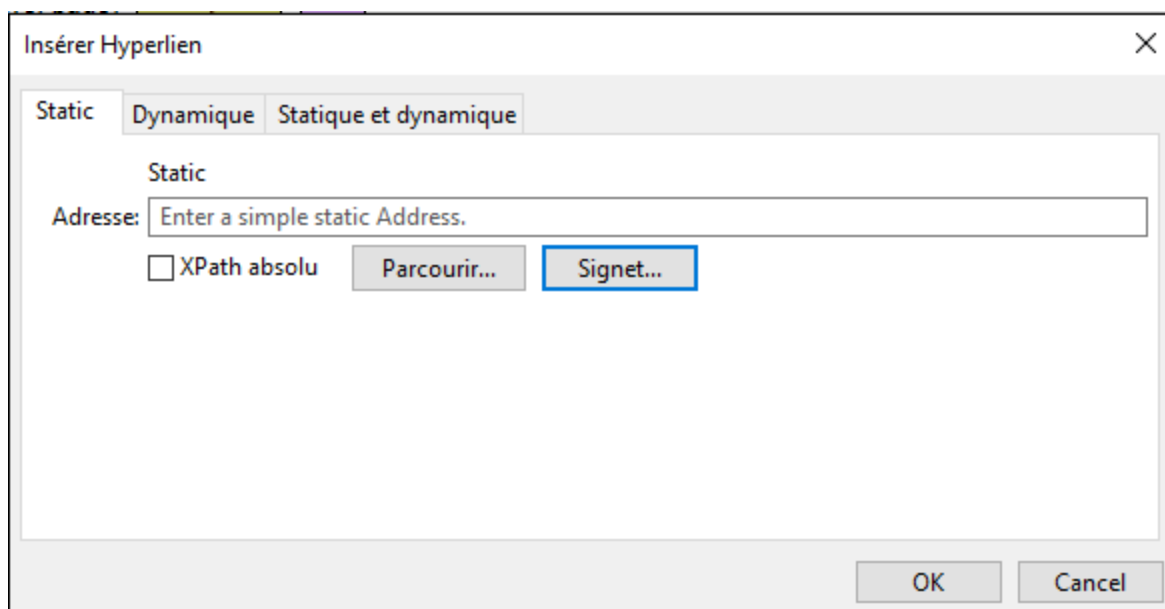
Créer des hyperliens

Un hyperlien peut être créé des manières suivantes :

- Autour d'un texte (statique ou dynamique), de nœuds, d'images, de modèles conditionnels, de calculs automatiques, et de blocs de contenu ou de nœuds ; il ne peut pas être créé autour d'un appareil d'entrée de données tel qu'un champ d'entrée ou liste déroulante—mais il peut être créé autour d'un nœud ou d'un modèle conditionnel dans lequel l'appareil d'entrée de données se trouve. Ceci est le contenu du lien, qui, lorsque cliqué, saute à la cible du lien. Pour créer un hyperlien autour d'un composant dans le SPS, sélectionnez ce composant et utilisez la commande de menu **Enclose With | Hyperlink**.
- Un nouvel hyperlien peut être inséré via la commande de menu **Insérer | Hyperlien**. Le contenu du lien devra être ajouté par la suite à l'intérieur des balises de l'hyperlien nouvellement créé.

Définir la cible de l'hyperlien

La cible de l'hyperlien est créée dans le dialogue Insérer hyperlien (*capture d'écran ci-dessous*), qui est accédée via [Enclose With | Hyperlink](#)⁵¹² ou [Insérer | Hyperlien](#)⁵⁰⁰.



La cible d'un lien peut être soit :

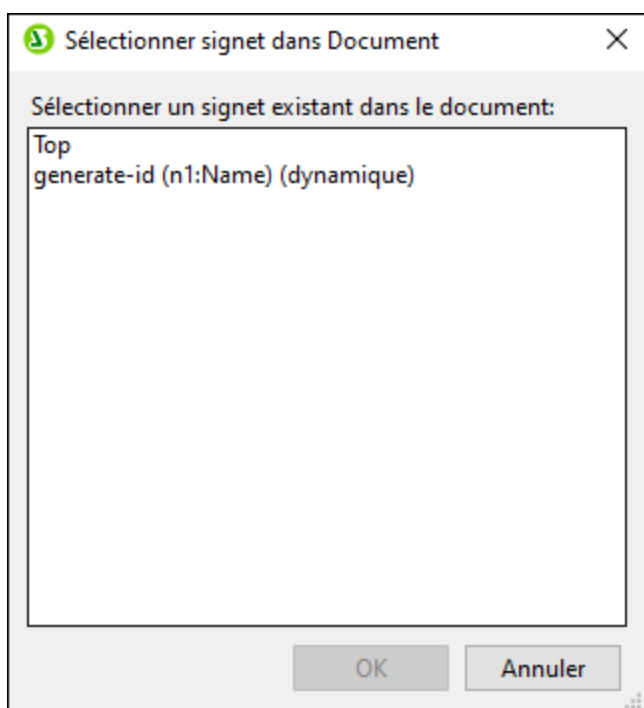
- Un [signet](#) ³²⁴ dans le même design SPS (dans ce cas, l'URI cible doit être un identifiant de fragment),
- [Généré de manière dynamique](#) ³²⁶ pour correspondre à des ancres de signets (ces URI sont aussi des identifiants de fragment),
- Une [ressource externe](#) ³²⁶ ; l'URI peut être statique (directement saisie), dynamique (prise d'un nœud dans le document XML), une combinaison de parties statiques et dynamiques, ou la valeur d'une entité non parsée.

Comment ces cibles sont définies est expliquée ci-dessous. Une fois que l'URI a été définie dans le dialogue Insérer/Éditer hyperlien, cliquez sur **OK** pour terminer.

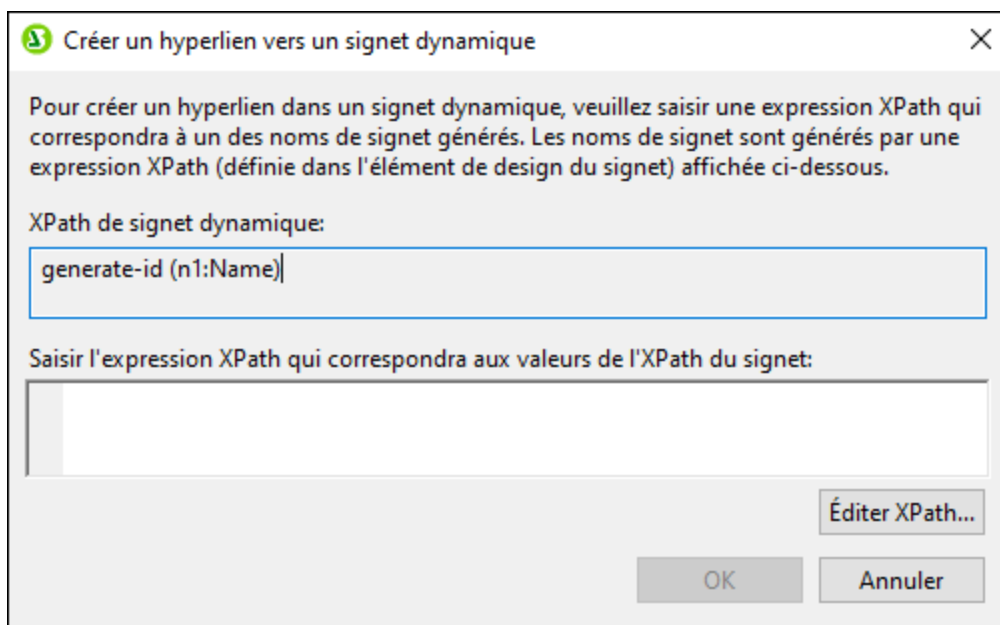
Mettre en lien vers les signets

Pour établir un lien vers un signet, faites comme suit :

1. Dans l'onglet Statique du dialogue Insérer hyperlien, cliquez sur la touche **Signet**. Le dialogue Sélectionner Signet dans un Document s'ouvre (*capture d'écran ci-dessous*). La capture d'écran ci-dessous affiche deux signets : un statique, un dynamique.



2. Pour sélectionner un signet statique comme URI cible, double-cliquez sur le signet statique et cliquez sur **OK**. Si vous double-cliquez sur un signet dynamique, vous serez invité à saisir une expression XPath pour correspondre au signet dynamique sélectionné (*voir la capture d'écran ci-dessous*).



Le [signet dynamique](#)³²¹ est en fait une expression XPath qui génère le nom du signet ; il n'est pas le nom du signet. Le dialogue Créer Hyperlien dans Signet dynamique affiche l'expression XPath du signet dynamique et vous permet de créer une expression XPath qui générera un nom pour correspondre à celui du signet ciblé. Cliquez sur **OK** lorsque vous avez terminé.

Mettre en lien vers des signets ID générés de manière dynamique

Les signets peuvent avoir des [ancres ID générés de manière dynamique](#)³²¹. Si quelqu'un souhaite mettre en lien de tels signets, le problème est le suivant : Puisque les noms des ancres générés de manière dynamique sont générés lors de l'exécution, et donc inconnus au moment du design, comment doit-on définir la valeur href d'un [hyperlien](#)⁵⁰⁰ qui cible une telle ancre ? La réponse est d'utiliser la fonction `generate-id()` une fois de plus, cette fois-ci à l'intérieur de la valeur href de l'[hyperlien](#)⁵⁰⁰. La clé pour comprendre pourquoi ceci fonctionne se trouve dans la propriété de la fonction `generate-id()`. Dans une seule transformation, à chaque fois que la fonction `generate-id()` est évaluée pour un nœud spécifique, elle génère toujours la même ID. Pour cette raison, les ID générées dans le signet et l'hyperlien seront les mêmes.

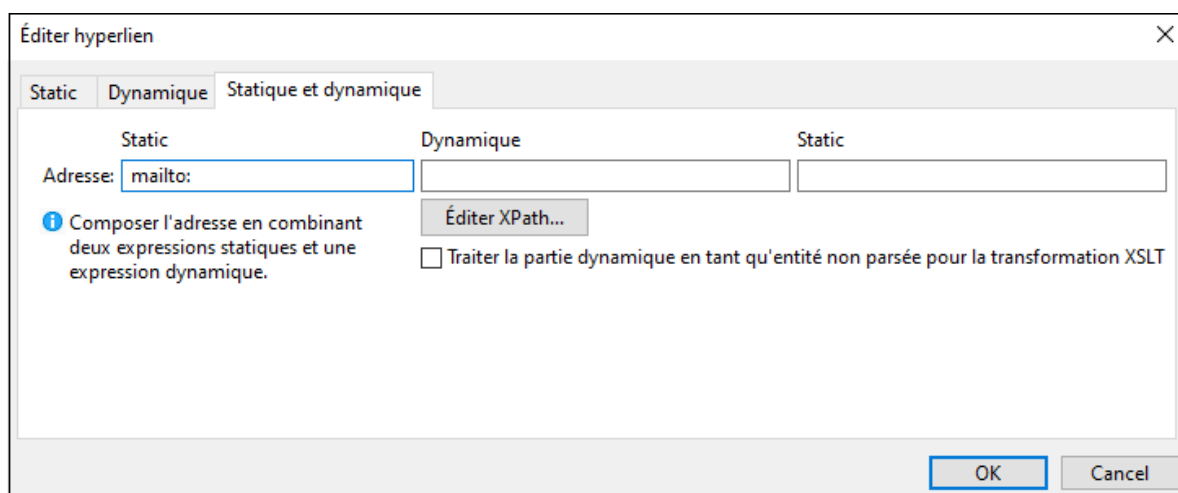
Il ne faut pas oublier deux points :

- Puisque la fonction `generate-id()` doit être évaluée comme expression XPath, utilisez l'onglet Dynamic du dialogue Insérer hyperlien (*voir la capture d'écran ci-dessous*) pour définir la cible de l'hyperlien.
- La valeur évaluée de l'attribut href doit commencer par # (l'identifiant de fragment). Par conséquent, l'expression XPath sera : `concat('#', generate-id(nodeXXX))`. En alternative, dans l'onglet statique et dynamique, saisissez # dans la partie statique de l'adresse et `generate-id(nodeXXX)` dans la partie dynamique.

Mettre en lien vers des ressources externes

Les URI qui localisent des ressources externes peuvent être créées de manière suivante :

- En entrant l'URI directement dans l'onglet statique du dialogue Insérer hyperlien. Par exemple, un lien vers la page d'accueil d'Altova (<http://www.altova.com>) peut être saisi directement dans le champ d'entrée 'Address' de l'onglet statique.
- En sélectionnant un nœud dans la source du document XML dans l'onglet dynamique du dialogue Insérer hyperlien. Le nœud dans la source XML peut fournir un string de texte qui est soit : (i) l'URI à cibler, ou (ii) le nom d'une [entité non parsée](#)³⁶³ qui a l'URI requis comme sa valeur. Par exemple, l'adresse du site Altova peut être contenue comme string de texte dans un nœud.
- En créant l'URI qui a les deux parties statique et dynamique dans l'onglet Static et Dynamic du dialogue Insérer hyperlien. Ceci peut être utile pour ajouter des préfixes statiques (par ex., un protocole) ou des suffixes (par ex., un nom de domaine). Par exemple, des adresses e-mail pourraient être créées en utilisant une partie statique de `mailto:` et une partie dynamique qui prend le contenu du string du nœud `//Contact/@email` (*la capture d'écran ci-dessous crée un lien sur les contenus de l'espace réservé du nœud //Contact/@email, c'est pourquoi le sélecteur abrégé `self::node()` a été utilisé.*). La touche **Éditer XPath** ouvre le [dilaogue Éditer expression XPath](#)⁴³⁸ pour vous aider à créer une partie dynamique de l'hyperlien.



Pour savoir comment utiliser les entités non parsées, consultez la section [URI de l'entité non parsée](#) ³⁶³.

Éditer les propriétés d'hyperliens

Pour éditer un hyperlien, clic droit soit sur la balise de début ou de fin de l'hyperlien (A), et sélectionnez **Éditer URL** depuis le menu contextuel. Ceci ouvre le dialogue Éditer hyperlien (*capture d'écran ci-dessous*). Le dialogue Éditer hyperlien peut aussi être accédé via la propriété `URL` du groupe des propriétés *Hyperlien* dans la fenêtre des Propriétés.

Retirer et supprimer les hyperliens

Pour supprimer un hyperlien, sélectionnez l'hyperlien (en cliquant soit sur la balise de début ou de fin de l'hyperlien (A)), et appuyez sur la touche **Supprimer**. L'hyperlien et son contenu sont supprimés.

8 Présentation et Sortie

Dans le design SPS, un seul ensemble de fonctions de style est défini pour les composants. Ces styles sont convertis dans le markup de style dans les sorties respectives *Authentic View*, *HTML*, *RTF*, *PDF*, *Word 2007+* et *Texte dans l'édition Enterprise* ; *Authentic View*, *HTML*, *RTF*, et *Texte dans l'édition Professional* ; *HTML dans l'édition Basic*).

Note : les formats de sortie basés sur page imprimée, tels RTF, PDF, Word 2007+ et Texte (contrairement à HTML), ne sont pas pris en charge dans l'édition Basic de StyleVision. En conséquence, toutes les fonctions liées à une telle sortie (tel qu'ajouter les en-têtes et bas de page) sont désactivées dans l'édition Basic. Pour pouvoir utiliser ces fonctions, vous devez obtenir une licence pour l'édition Professional (inclut la sortie RTF et Texte) ou l'édition Enterprise (inclus RTF, PDF, Word 2007+ et Texte). Veuillez voir la page de comparaison de l'édition StyleVision sur le [site Web d'Altova](#) pour plus d'information.

Styliser des composants SPS

Tous les styles de composants de SPS sont réalisés utilisant les principes et la syntaxe CSS2. Les styles peuvent être définis dans les feuilles de style externes, globalement pour SPS, ainsi que localement sur un composant. L'ordre en cascade de CSS2 s'applique à SPS, et fournit une flexibilité considérable dans les styles de design. Comment travailler avec les styles CSS est décrit dans la sous-section [Travailler avec les Styles CSS](#) ³⁴³ de cette sous-section.

Les valeurs des propriétés de style peuvent être saisies directement dans les Styles ou barres latérales des Propriétés, ou être définies via les [expressions XPath](#) ³⁵⁴. Les avantages d'utiliser les expressions XPath sont que : (i) la valeur de propriété peut être prise du fichier XML, et (ii) que la valeur de propriété peut être assignée de manière conditionnelle selon un test contenu dans l'expression XPath.

En plus, dans le design SPS, certains éléments HTML sont disponibles comme markup pour les composants SPS. Ces [formats prédéfinis](#) ³²⁹ sont passés dans la sortie HTML. Le formatage inhérent dans un tel markup est, pour cette raison, aussi utilisé pour fournir le styling aux composants SPS. Lorsque les styles CSS sont appliqués aux formats prédéfinis, les styles CSS auront la priorité sur le style inhérent du format prédéfini. Ces formats prédéfinis sont décrits dans la sous-section [Formats prédéfinis](#) ³²⁹ de cette sous-section de cette section.

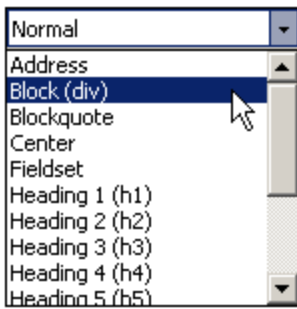
8.1 Formats prédéfinis

StyleVision fournit un nombre de formats prédéfinis, dont chacun correspond à un élément HTML (*capture d'écran ci-dessous*). Quand vous appliquez un Format prédéfini à un composant dans le Design, ce composant est marqué comme composant ayant une sémantique HTML correspondante. Ceci a deux effets :

- Un formatage inhérent au format prédéfini sélectionné est appliqué.
- Le composant est contenu dans le type de composant, *paragraph*, qui [le rend disponible pour le style local](#) ⁽³⁵⁰⁾ par le type de composant.

Assigner des formats prédéfinis

Les formats prédéfinis peuvent être assignés en cliquant sur **Insérer | Paragraphe spécial**, puis le format requis, ou en sélectionnant le format requis depuis la liste déroulante dans la barre d'outils (affiché ci-dessous).



Styles inhérents

Les formats prédéfinis utilisés dans StyleVision ont soit un ou deux des composants de style suivants :

- Un composant text-styling
- Un composant d'espacement.

Par exemple, le format `para (p)` prédéfini a uniquement un composant d'espacement ; il positionne l'espace vertical avant et après le composant sélectionné, et n'applique aucun style de texte. D'un autre côté, le format `Heading 1 (h1)` prédéfini a un composant text-styling et un composant d'espacement.

Les points de style suivants sur les formats prédéfinis devraient être notés :

- Le composant d'espacement d'un format prédéfini s'applique pour tout type de composant SPS, mais le style de texte uniquement s'il peut être appliqué. Par exemple, si vous sélectionnez une image et appliquez un format prédéfini de `Heading 1 (h1)` à celle-ci, alors le composant d'espacement prendra effet, mais pas le composant text-styling.
- Le composant text-styling des formats prédéfinis ne s'applique pas aux appareils d'entrée des données.
- Uniquement un seul format prédéfini s'applique au composant à tout moment donné.
- Le format prédéfini `Preformatted (pre)` applique le formatage équivalent à celui appliqué à l'onglet `pre` de HTML : les sauts de ligne et l'espacement dans le texte sont maintenus et une police à espacement fixe (tel un 'Courier') est utilisée pour l'affichage. Dans le cas de lignes 'run-on' sans sauts de ligne, tel un paragraphe de texte, le format prédéfini `Preformatted (pre)` affichera les lignes de texte sans enveloppement. Si vous souhaitez wrapper un texte, utilisez le format prédéfini `Preformatted, wrapping (pre-wrap)`.

Définir des styles supplémentaires pour un format prédéfini.

Les styles supplémentaires aux styles inhérents peuvent être définis pour un format prédéfini en le sélectionnant et appliquant un [style local via la barre latérale Styles](#)³⁵⁰.

8.2 Échappement de sortie

Un caractère de string de texte est dénommé échappé lorsqu'il est écrit comme référence de caractère ou référence d'entité. Les deux types de références (caractère et entité) sont délimités par une esperluette au début et un point-virgule à la fin. Par exemple :

- La référence de caractère hexadécimal (ou Unicode) du caractère A est `A`
- La référence de caractère décimal du caractère A est `&# 65;`
- La référence d'entité HTML (et XML) du caractère & est `&`
- La référence de caractère hexadécimal (ou Unicode) du caractère & est `&`
- La référence de caractère décimal du caractère & est `&# 38;`
- La référence d'entité HTML (et XML) du caractère < est `<`

Échappement de sortie

L'échappement de sortie se réfère à la manière dont les caractères qui sont **échappés dans l'entrée** sont représentés dans la sortie. Un caractère est dit 'output-escaped' lorsqu'il est représenté dans la sortie comme référence de caractère ou d'entité. Notez qu'un caractère peut uniquement être output-escaped lorsqu'il est échappé dans l'entrée (*voir table ci-dessous pour des exemples*). Dans un SPS, l'échappement de sortie peut être activé ou désactivé pour :

- Fragments de texte statique,
- Espace réservé `contents`, et
- Calculs automatiques

Ceci est fait avec l'attribut `disable-output-escaping` du groupe des propriétés de *Text*. La valeur par défaut de cette propriété est `no`, ce qui signifie que output-escaping ne sera pas désactivé. Ainsi, les caractères qui sont échappés dans l'entrée seront échappés dans la sortie par défaut (*voir table ci-dessous pour les exemples*).

Pour désactiver l'échappement de sortie, suivez les étapes suivantes :

1. Sélectionnez le (i) texte statique, ou (ii) le fragment de texte statique, (iii) l'espace réservé `contents`, ou (iv) le calcul automatique pour lequel vous souhaitez désactiver l'échappement de sortie.
2. Dans la barre latérale des Propriétés, sélectionnez le groupe des propriétés *Text* pour l'item *Text* et définissez l'attribut `disable-output-escaping` comme `yes` pour les différentes sorties individuellement ou pour toutes les sorties. Les valeurs disponibles sont :
 - For *HTML* (pour définir `disable-output-escaping` comme `yes` pour la sortie HTML).
 - For *Authentic* (pour définir `disable-output-escaping` comme `yes` pour la sortie Authentic).
Notez que désactiver l'échappement de sortie pour Authentic View est activé **uniquement dans les Enterprise editions d'Authentic View** (à savoir, dans les Enterprise editions de StyleVision, Authentic Desktop, Authentic Browser et XMLSpy).
 - For *all* (pour définir `disable-output-escaping` comme `yes` pour toutes les sorties à l'exception de Text).

Lorsque l'échappement de sortie est désactivé pour un format de sortie particulier (par exemple, sortie HTML), le texte sélectionné ne sera pas échappé dans ce format de sortie, mais sera échappé dans les autres formats de sortie.

Ci-dessous, vous trouverez quelques exemples de texte avec l'échappement de sortie désactivé et/ou activé.

Un texte statique	désactiver-l'échappement de sortie	Texte de message de sortie
&	non	&
&	oui	&
&	non	&
&	oui	&
<	non	<
<	oui	<
&# 65;	non	&# 65;
&# 65;	oui	A
&lt;	non	&lt;
&lt;	oui	<
&amp;lt;	oui	<
&& lt;	oui	&<

Note : désactiver-Sortie_Échappement est pris en charge dans Authentic View uniquement dans les produits des Enterprise Editions d'Altova.

Utiliser l'output-escaping désactivé parmi les formats de sortie

Si output-escaping est désactivé, le string de texte peut avoir une signification dans une sortie, mais aucune signification dans toutes les autres sorties. Par exemple, prenez en compte le texte d'entrée suivant, qui a échappé des caractères (en surbrillance) :

```
&lt;b>Ce texte est en gras.</b>
```

Si output-escaping est désactivé, la sortie de ce texte sera comme suit :

```
<b>This text is bold.</b>
```

Si output-escaping est désactivé pour la sortie HTML et que cette sortie est consultée dans un navigateur (contrairement à un éditeur de texte), le markup sera significatif pour le navigateur HTML et le texte sera affiché en gras comme suit :

This text is bold.

Toutefois, si consulté dans un autre format de sortie, tel un PDF, le markup qui était significatif dans HTML ne sera pas nécessairement important dans cet autre format de sortie. Dans ce cas particulier cité ci-haut, le texte non échappé (échappement de sortie désactivé) sera sorti en format PDF comme suit :

```
<b>This text is bold.</b>
```

Comme l'exemple ci-dessus le démontre, le texte de sortie obtenu en désactivant le texte obtenu par 'disabling output-escaping' pourrait être interprété comme code dans le format de sortie, mais pas dans un autre. Ceci ne devrait en aucun cas être retenu lorsque vous utilisez la propriété 'Disable-Output-Escaping'.

8.3 Formatage de valeur (formater les types de données numériques)

Le formatage de valeur permet l’affichage des contenus des nœuds de type de données du Schéma XML numériques (voir [liste ci-dessous](#) ³³⁴) dans un format autre que la représentation lexicale de ce type de données. (Par exemple, la représentation lexicale d’un nœud de type de données `xs:date` est `YYYY-MM-DD`, avec un composant de fuseau horaire optionnel, tel que `+02:00.`) Le formatage de valeur est affiché dans HTML. Le formatage de valeur peut aussi être utilisé pour formater le résultat d’un calcul automatique si le résultat du calcul automatique est dans le format lexical d’un des types de données numériques (voir [liste ci-dessous](#) ³³⁴) pour lequel le formatage de valeur est disponible.

Dans les sous-sections de cette section, nous allons décrire :

- comment le [mécanisme de formatage de la valeur fonctionne](#) ³³⁴, et
- la [syntaxe](#) ³³⁷ pour définir le formatage de valeur.

Note : le formatage de valeur ne change pas le format dans lequel les données sont stockées dans le document XML. Dans le document XML valide, les données sont toujours stockées dans le format lexical approprié du type de données du nœud approprié. Le formatage de valeur est appliqué dans l’affichage dans la sortie.

Les types de données numériques pour lesquels le formatage de valeur est disponible

Le formatage de valeur est disponible pour les types de données suivants :

- `xs:decimal`; `xs:integer`; les 12 types intégrés dérivés de `xs:integer`
- `xs:double` et `xs:float` lorsque les valeurs sont entre et incluent 0.000001 et 1,000,000. Les valeurs en dehors de cette portée sont affichées en notation scientifique (par exemple : `1.0E7`), et ne peuvent pas avoir de formatage de valeur qui s’y applique.
- `xs:date`; `xs:dateTime`; `xs:duration`
- `xs:gYear`; `xs:gYearMonth`; `xs:gMonth`; `xs:gMonthDay`; `xs:gDay`

Note : pas tous les formats ne sont disponibles dans l’édition Basic puisque Authentic View n’est pas appuyée dans l’édition Basic.

8.3.1 Mécanisme de formatage de valeur

Le formatage de valeur peut être appliqué à :

- Un [nœud de type de données numérique](#) ³³⁴, tel que `xs:decimal` ou `xs:date` qui est présent dans SPS comme contenu ou champ d’entrée.
- Un calcul automatique qui évalue une valeur qui a un format lexical d’un [type de données numérique](#) ³³⁴.

Définir le formatage valeur

Pour définir le formatage de valeur pour un nœud ou un calcul automatique dans SPS, faites comme suit :

1. Sélectionnez l'espace réservé `contents` ou champ d'entrée du nœud, ou le calcul automatique.
2. Dans la barre latérale des Propriétés, sélectionnez un item, puis le groupe *Content* (ou le groupe *AutoCalc*) des propriétés. Puis, cliquez sur le bouton Éditer [...] de la propriété *Formatage valeur*. En alternative, clic droit et sélectionnez **Éditer formatage valeur** depuis le menu contextuel. Le dialogue *Formatage valeur* apparaît (*capture d'écran ci-dessous*). C'est différent selon si le composant sélectionné était un nœud ou un calcul automatique. Si le composant sélectionné était un nœud, alors le dialogue comme celui du dessous apparaît. Le nœud représenté dans la capture d'écran ci-dessous est de type de données `xs:date`.

Formatage valeur

Format pour le champ content

☐ Non formaté

☒ Formater en tant que valeur de schéma XML

Options pour la valeur de schéma XML

Le type de données dans l'instance XML: date

Formater dans les docs de sortie en: DD Month YYYY

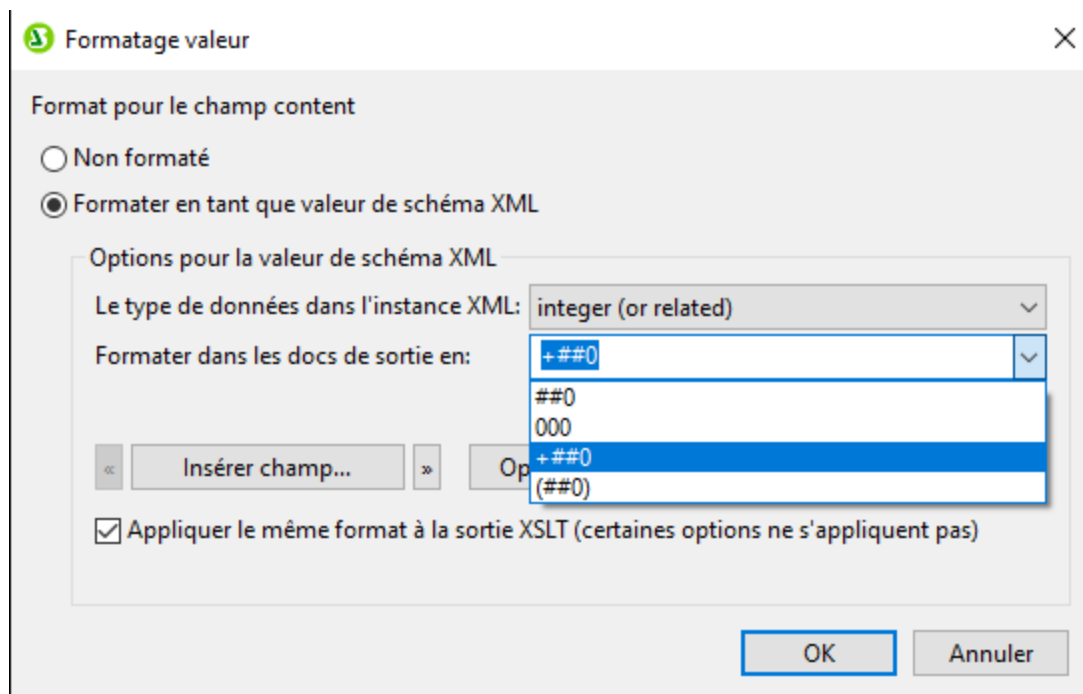
« Insérer champ... » Options de champ... Aller à Erreur

☒ Appliquer le même format à la sortie XSLT (certaines options ne s'appliquent pas)

OK Annuler

Notez que la capture d'écran ci-dessus contient une ligne : Les *formats de type 'date'* et celui du format standard du type de données `xs:date` sont indiqués à côté de la case à cocher *Non formaté*. Pour un nœud d'autres types de données, cette information serait en conséquence différente.

Si le composant sélectionné était un calcul automatique, le dialogue suivant apparaîtrait.



- À cette étape, vous spécifiez si l'affichage de la valeur du composant doit être non formaté ou formaté. Si vous souhaitez que la sortie soit non formatée, sélectionnez le bouton radio *Non formaté*. Autrement, sélectionnez le bouton radio *Formater en tant que valeur schéma XML*. (Si la valeur n'est pas formaté, la sortie a le formatage standard pour le type de données du nœud sélectionné ou le type de nœud du résultat du calcul automatique. Si vous spécifiez *Formater en tant que valeur de schéma XML* pour un calcul automatique, vous devez sélectionner en plus (depuis une liste déroulante) le type de données du résultat du calcul automatique attendu.
- Saisissez la définition du formatage valeur. Cette définition peut être saisie de trois manières : (i) en sélectionnant depuis la liste déroulante des options disponibles pour ce datatype (voir le *champ d'entrée 'Formater dans les docs de sortie' dans la capture d'écran ci-dessus*) ; (ii) en saisissant la définition directement dans le champ d'entrée ; et (iii) en utilisant les boutons **Insérer champ** et **Options de champ** pour créer la définition correctement. Voir [Syntaxe du formatage valeur](#) ³³⁷ pour une description complète de diverses options de formatage.

Erreurs dans la syntaxe

S'il existe une erreur dans la syntaxe, la chose suivante se passe :

- La définition est affichée en rouge.
- Un message d'erreur, également en rouge, est affiché sous le champ d'entrée.
- Le bouton **OK** dans le dialogue Formatage valeur est désactivé.
- Le bouton **Aller à Erreur** dans le dialogue Formatage valeur est activé. En cliquant dessus, le curseur se place au point dans la définition de format où se trouve l'erreur de la syntaxe.

Décalage des données et formats de type de données

Si les données saisies dans un nœud XML ne correspondent pas au format lexical de ce datatype du nœud, ou si le résultat d'un calcul automatique ne correspond au format lexical du type de données attendu, alors le formatage sera indéfini et ne s'affichera pas correctement dans la sortie.

Appliquer le Formatage valeur à la sortie

Le Formatage valeur que vous définissez s'applique à Authentic View, qui est pris en charge dans les éditions Enterprise et Professional.

Certaines définitions du Formatage valeur—pas toutes!—peuvent en plus être appliquées aux sorties HTML. À cette fin, vérifiez la case à cocher Appliquer le même format à la sortie XSLT. Si cette option n'est pas cochée, ou si elle n'est pas disponible alors seul Authentic View affichera le Formatage valeur, tandis que la sortie affichera la valeur dans le format standard pour le datatype du composant (le format lexical).

8.3.2 Syntaxe du Formatage valeur

La syntaxe du Formatage valeur est :

```
([prefix character/s]field[suffix character/s][{field-option1,field-option2,...}])+
```

where **prefix character/s** et **suffix character/s** sont des spécificateurs optionnels utilisés pour contrôler des alignements et l'affichage de symboles positifs / négatifs ;
field peut être un formatage ou texte spécifique au datatype ; et
{field-option(s)} est un qualificateur optionnel, qui permet des options de formatage supplémentaires.

Explication de la syntaxe de définition

La définition du Formatage valeur est construite comme suit :

- La définition est composée d'un ou de plusieurs champs. Par exemple, la définition `DD Month YYYY` a trois champs.
- Les champs peuvent être exécutés ensemble, ou ils peuvent être séparés par les caractères suivants : espace, trait d'union, virgule, point, période, ou par string de texte entre guillemets simples ou doubles. Par exemple, dans la définition : `DD-Month' in the year 'YYYY`, les champs `DD` et `Month` sont séparés par un trait d'union, et les champs `Month` et `YYYY` sont séparés par un string de texte enveloppé par des guillemets simples.
- Un champ peut avoir un/des caractère/s de préfixe et/ou suffixe optionnel/s. Par exemple :
`<+####,##0.00`.
- Un champ peut avoir un ou plusieurs options de champ optionnel. L'/les option/s de champ doit/doivent être contenue/s dans un ensemble d'accolades, et doivent suivre le champ sans espace intervenant. De multiples options de champ pour un champ simple sont séparées par une ", " (comma). Par exemple, dans la définition : `DD Month{uc,ro} YYYY`, `uc` et `ro` enveloppés par des accolades sont des options de champ pour le champ `Month`.

Exemples

Exemple du Formatage valeur pour un type de données `xs:decimal` :

```
"$" (##0.00)
```

Exemples de la sortie :

\$ 25.00
\$ 25.42
\$ 267.56

Exemple du Formatage valeur pour un type de données `xs:date` :

`DD Month{uc,ro} YYYY`

where `uc` et `ro` sont des options de champ pour mettre le champ `Month` fen majuscule et en lecture seule, respectivement

Un exemple de la sortie serait :

24 SEPTEMBER 2003

Types de champs

Un type de champ représente un composant des données et de la manière dont ce composant doit être formaté. Le formatage inhérent dans le type de champ peut être modifié davantage par des modificateurs de préfixe et suffixe ainsi que par les options de champ. Les tables suivantes recensent les types de champ disponibles. Notez que, dans le menu de la liste déroulante du dialogue Formatage valeur, il existe des définitions du Formatage valeur 'type-specific' et 'field-only'. Vous pouvez sélectionner un de ceux et les modifier tel que requis en ajoutant des modificateurs de préfixe, de suffixe et/ou d'options de champs.

Types de champs	Explication
#	Espace si pas de chiffre à la position
0	Zéro si pas de chiffre à la position
.	Marque décimale
,	Séparateur de groupe de chiffres
Y	Année
y	année (base = 1930) ; voir note ci-dessous
MM	Mois, doit avoir une longueur de 2
DD	Jour, doit avoir une longueur de 2
W	Numéro de la semaine
d	Numéro du jour de la semaine (1 à 7)
i	Jour dans l'année (1 à 366)
hh	Heure (0 à 23), doit avoir une longueur de 2

HH	Heure (1 à 12), doit avoir une longueur de 2
mm	Minute, doit avoir une longueur de 2
ss	Seconde, doit avoir une longueur de 2
AM	AM ou PM
am	am ou pm
AD	AD ou BC
ajouter	ajouter ou bc
CE	CE ou BCE
ce	ce ou bce

Types de champs	Explication
Jour de la semaine	Jour de la semaine (Dimanche, Lundi...)
JOUR DE LA SEMAINE	Jour de la semaine (DIMANCHE; LUNDI...)
weekday	Jour de la semaine (dimanche, lundi...)
Wkd (Jds)	Jour de la semaine (Dim, Lun...)
WKD (JDS)	Jour de la semaine (DIM, LUN...)
wkd (jds)	Jour de la semaine (dim, lun...)
Mois	Mois (Janvier, Février...)
MOIS	Mois (JANVIER, FÉVRIER...)
mois	Mois (janvier, février...)
Moi	Mois (Jan, Fév...)
MOI	Mois (JAN, FEV...)
moi	Mois (jan, fév...)

Notes sur longueur de champ et longueur d'entrée

Les points suivants liés à la longueur des composants devraient être notés :

Longueur des champs de date : Lorsque des champs comme MM, DD, HH, hh, mm, et ss sont utilisés, il doivent avoir la longueur de 2 dans la définition. Lorsque les champs y ou Y sont utilisés, le nombre de caractères y ou Y dans la définition détermine la longueur de la sortie. Par exemple, si vous spécifiez YYY, alors la sortie pour une valeur de 2006 serait 006 ; pour une définition de YYYYYY, ce serait 002006. Voir aussi 'Base Year' ci-dessous.

Étendre la longueur du champ : Le symbole * (astérisque) est utilisé pour étendre la longueur d'un champ numérique non sémantique (entiers, décimales, etc). Dans le cas des décimales, il peut être utilisé soit d'un côté ou des deux du point décimal. Par exemple, le Formatage valeur *0.00* assure qu'un nombre aura les zéros tel que spécifié dans le formatage si ces emplacements de chiffres sont vides, de même que tout nombre de chiffres des deux côtés du point décimal.

Note : si un champ ne rend pas de texte, ceci peut être dû à votre paramètre de région dans Windows. Par exemple, Windows renvoie un string vide pour le champ AM/PM si le paramètre de langage régional est allemand.

Modificateurs de préfixe et de suffixe

Les modificateurs de préfixe et de suffixe sont utilisés pour modifier l'alignement textuel et les représentations positives/négatives des champs. La table suivante recense les modificateurs de préfixe et suffixe disponibles.

Préfixe	Suffixe	Explication
<		Aligné à gauche ; défaut pour texte. Pour les numéros, qui sont alignés à droite par défaut, ceci est significatif s'il existe un nombre fixe d'espaces dominants.
>		Aligné à droite ; défaut pour numéros.
?		Symbole Moins adjacent au numéro si négatif ; rien autrement. Il s'agit du défaut pour les numéros.
<?		Symbole Moins aligné à gauche si négatif ; rien autrement. Numéro aligné à gauche, suit le signe moins.
<?>		Symbole Moins aligné à gauche si négatif ; rien autrement. Numéro aligné à droite.
-	-	Symbole Moins adjacent au numéro si négatif ; espace autrement. Emplacement avant le numéro (préfixe), après le numéro (suffixe).
<-	>-	Symbole Moins si négatif ; espace autrement. Numéro et signe adjacent. Aligné à gauche (préfixe); aligné à droite (suffixe).
<->		Symbole Moins aligné à gauche si négatif ; espace autrement. Numéro aligné à droite.
+	+	Signe Plus ou Moins toujours, adjacent au numéro ; avant le numéro (préfixe) après le numéro (suffixe).
<+	>+	Signe Plus ou moins toujours, adjacent au numéro ; aligné à gauche (préfixe) aligné à droite (suffixe).
<+>		Signe plus ou moins toujours, aligné à gauche ; numéro aligné à droite.
(	)	Parenthèses si négatif ; espace autrement. Adjacent au numéro.
<(		Parenthèses si négatif ; espace autrement. Adjacent au numéro. Aligné à gauche.
<(>		Parenthèses si négatif ; espace autrement. Parenthèses gauche alignés à gauche ; numéro et parenthèses à droite adjacents et alignés à droite.
[	]	Parenthèses si négatif ; espace autrement. Adjacent au numéro.
*	*	Nombre de chiffres extensible à gauche (préfixe) ou à droite (suffixe)
—	—	Espace
^	^	Caractère de remplissage (défini dans les options)
	th	Numéros ordinaux en EN (st, nd, rd, ou th)
	TH	Numéros ordinaux en EN (ST ND, RD, ou TH)

Options de champs

Les options de champ permettent des modifications avancées à réaliser dans les champs. Les options suivantes sont disponibles :

Option	Explication
uc	Faire majuscule
lc	Faire minuscule
gauche	Aligné à gauche
droite	Aligné à droite
ro	Lecture (XML) uniquement ; pas d'édition permise
éditer	Le champ est éditable (actif par défaut)
dec=<char>	Spécifie un caractère pour un point décimal (défaut est point)
sep=<char>	Spécifie un caractère pour un séparateur de chiffre (défaut est virgule)
fill=<char>	Spécifie le caractère de remplissage
base=<year>	Année de base pour les champs d'année (<i>voir la note ci-dessous</i>)
pos	N'affiche que les numéros positifs ; entrée des numéros négatifs autorisée

Les options de champs devraient être utilisées pour générer le formatage de numéro pour les langues européennes, qui échangent les virgules et les périodes du système de la langue anglaise : par exemple, 123.456,75.

Le Formatage valeur à utiliser pour obtenir le formatage ci-dessus serait : ###,###.##{dec=, , sep=. }

Notez que le champ maintient le formatage anglais, tandis que ce sont les options de champs `dec` et `sep` qui spécifient le symbole décimal et le séparateur de chiffres. Si le symbole décimal et le séparateur de chiffres ne sont pas spécifiés, ces caractères passent par défaut au symbole décimal et séparateur de chiffre des paramètres régionaux de Windows OS (**Panel de commande | Tous items | Région | Format**).

8.4 Travailler avec les Styles CSS

Le document de design SPS est stylé avec les règles CSS. Les règles de style peuvent être spécifiées :

- Dans les [feuilles de style CSS externes](#)³⁴⁴. Les feuilles de style CSS externes peuvent être ajoutées via la barre latérale de l'[Aperçu de design](#)³⁵ et via la barre latérale [Répertoire de Style Repository](#)⁴⁴.
- Dans les [styles globaux](#)³⁴⁸ pour le SPS, qui peuvent être considérés pour être définis à l'intérieur de SPS et à son démarrage. (Dans la sortie HTML, ces styles globaux sont définis à l'intérieur de l'élément enfant `style` de l'élément `head`.) Les styles globaux sont définis dans la barre latérale du [Répertoire de style](#)⁴⁴.
- [Localement](#)³⁵⁰, dans des composants individuels du document. Dans la sortie HTML, de telles règles dans l'attribut `style` des éléments individuels HTML. Les règles de style locales sont définies dans la barre latérale des [Styles](#)⁴⁶.

Chacune des méthodes ci-dessus pour créer des styles est décrite en détail dans les sous-sections de cette section (*liens ci-dessus*).

Terminologie

Une feuille de style CSS consiste en une ou plus de règles de style. Une règle a l'air de ceci :

```
H1 { color: blue }
```

ou

```
H1 { color: blue;
    margin-top: 16px; }
```

Chaque règle a un sélecteur (dans les exemples ci-dessus, `H1`) et une déclaration (`color: blue`). La déclaration est une liste de propriétés (par exemple, `color`) avec des valeurs (`blue`). Nous nous référons à chaque paire de property-value comme définition de style. Dans StyleVision, les styles CSS peuvent être définis dans la barre latérale [Styles](#)⁴⁶ (styles locaux) et [Répertoire de style](#)⁴⁴ (styles globaux).

Ordre de cascade

L'ordre de cascade de CSS s'applique. Ceci signifie que la préséance des règles est évaluée sur la base de :

1. ***l'origine***. Les feuilles de style externes ont une préséance plus faible que les styles globaux, et les styles globaux ont une préséance plus faible que les styles locaux. Les feuilles de style externes sont considérées être importées, et l'ordre d'importation est important, avec le dernier de deux feuilles de style importées ayant préséance.
2. ***Spécification***. Si deux règles s'appliquent au même élément, la règle avec plus de sélecteurs spécifiques a préséance.
3. ***Ordre***. Si deux règles ont la même origine et spécificité, la règle qui apparaît plus tard dans la feuille de style a préséance. Les feuilles de style sont considérées venir avant la règle définie de la feuille de style importée.

Les Styles CSS dans les SPS modulaires

Quand un module SPS est ajouté à un autre SPS, alors les styles CSS dans le SPS référant ont la priorité sur ceux dans le module ajouté. Lorsque de multiples modules sont ajoutés, alors les styles CSS dans ces

modules situés relativement bas dans la liste de module ont la priorité. For more information sur les SPS modulaires, voir la section [SPS modulaires](#) ²²⁰.

Prise en charge de CSS dans Internet Explorer

Les versions d'Internet Explorer (IE) antérieures à IE 6.0 interprètent certaines règles CSS différemment que IE 6.0 et plus récentes. En tant que designer, il est important de savoir quelle version d'IE vous conserverez. La version IE 6.0 et plus récente propose une prise en charge pour les interprétations plus anciennes et plus récentes, permettant ainsi d'utiliser même l'interprétation plus ancienne dans les versions plus récentes (IE 6.0 et plus récente). Quelle interprétation est utilisée par IE 6.0 et ultérieure est déterminée par une touche dans le code du document HTML. Dans un SPS, vous pouvez [spécifier](#) ⁴⁷² si les documents de sortie HTML devraient être stylés selon l'[interprétation Internet Explorer plus ancienne ou plus récente](#) ¹⁰⁰. Vous devriez définir les styles CSS selon l'interprétation sélectionnée. Pour plus de détails, voir les [Propriétés](#) ⁴⁷³ : [Prise en charge CSS](#) ⁴⁷³.

Note : pour plus d'information sur la spécification CSS, allez à <http://www.w3.org/TR/REC-CSS2/>.

8.4.1 Feuilles de style externes

Cette section décrit comment les feuilles de style CSS externes peuvent être gérées de l'intérieur de la GUI StyleVision. Elle consiste en les parties suivantes :

- [Ajouter une feuille de style CSS externe au SPS](#) ³⁴⁴
- [Consulter les contenus de la feuille de style CSS externe et modifier l'applicabilité média](#) ³⁴⁵
- [Modifier la préséance](#) ³⁴⁶
- [Basculer entre la feuille de style CSS entière et la feuille de style CSS unique](#) ³⁴⁷

Les feuilles de style CSS externes peuvent être gérées de deux barres latérales : la [barre latérale du Référentiel de Style](#) ⁴⁴ et la [barre latérale de l'Aperçu de Design](#) ³⁵. Si un aspect des feuilles de style externes est consultable dans les deux barres latérales (par exemple, la préséance relative de multiples feuilles de style), alors les changements réalisés dans une barre latérale seront automatiquement reflétés dans l'ordre.

Ajouter une feuille de style CSS externe au SPS

Pour assigner une feuille de style CSS externe au SPS, procédez comme suit :

1. Dans l'Aperçu design, sélectionnez un item externe dans la fenêtre du Référentiel de Style (*capture d'écran ci-dessous*).



2. Cliquez sur la touche **Ajouter** en haut à gauche de la barre d'outils du Référentiel de Style (*voir la capture d'écran ci-dessus*).

3. Dans le dialogue Ouvrir qui apparaît, recherchez et sélectionnez le fichier CSS requis, puis cliquez sur **Ouvrir**. Le fichier CSS est ajouté à l'item externe comme faisant partie de sa structure d'arborescence (voir *liste d'arborescence et capture d'écran ci-dessous*).
4. Pour ajouter une feuille de style CSS externe supplémentaire, répétez les étapes 1 à 3. La nouvelle feuille de style CSS sera ajoutée à l'arborescence externe, en dessous de toutes les feuilles de toutes les feuilles de style CSS externes précédemment ajoutées.

Note : vous pouvez aussi ajouter une feuille de style CSS externe via la barre latérale [Aperçu de Design](#) ³⁵.

Consulter et modifier l'arborescence de feuilles de style CSS

L'arborescence des feuilles de style CSS externes est structurée comme suit (voir aussi la capture d'écran ci-dessous) :

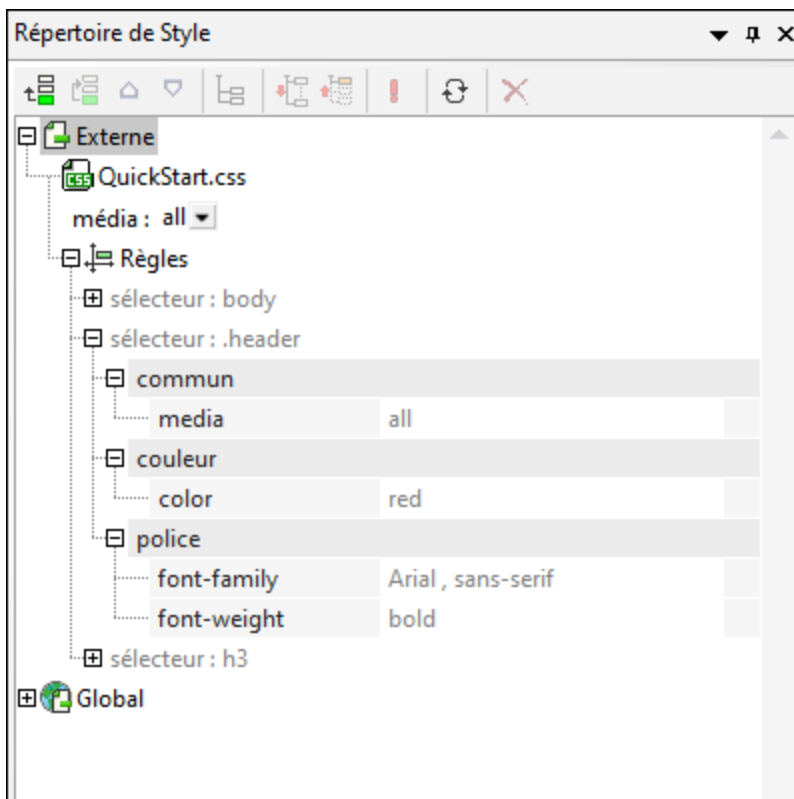
```

- CSS-1.css (emplacement du fichier apparaît sur mouseover)
- Media (peut être défini dans la fenêtre de le Référentiel de style)
- Règles (non-éritable ; doit être édité dans le fichier CSS)
  - Selector-1
    - Property-1
    - ...
    - Property-N
  - ...
  - Selector-N
+ ...
+ CSS-N.css
```

Le média vers lequel cette feuille de style particulière est applicable dans la fenêtre Référentiel de Style.

Procédez en cliquant sur la flèche vers le bas à droite de l'item et sélectionnez le média requis depuis la liste déroulante. Les règles définies dans la feuille de style CSS externe sont affichées, mais ne peuvent pas être éditées. La feuille de style, les Règles, et les items des sélecteurs dans l'arborescence sont élargis et réduits en cliquant sur les symboles + et - à gauche de chaque item (voir la capture d'écran ci-dessous).

Pour supprimer une feuille de style externe, sélectionnez la feuille de style et cliquez sur la touche **Réinitialiser** dans la barre d'outils du Référentiel de Style.



Modifier la préséance des feuilles de style CSS externes

Les feuilles de style CSS assignées à la fenêtre du Référentiel de Style seront importées dans le fichier de sortie HTML utilisant l'instruction `@import`. Dans le fichier HTML, ceci aurait l'aspect suivant :

```
<html>
  <head>
    <style>
      <!--
        @import url("ExternalCSS-1.css");
        @import url("ExternalCSS-2.css") screen;
        @import url("ExternalCSS-3.css") print;
      -->
    </style>
  </head>
  <body/>
</html>
```

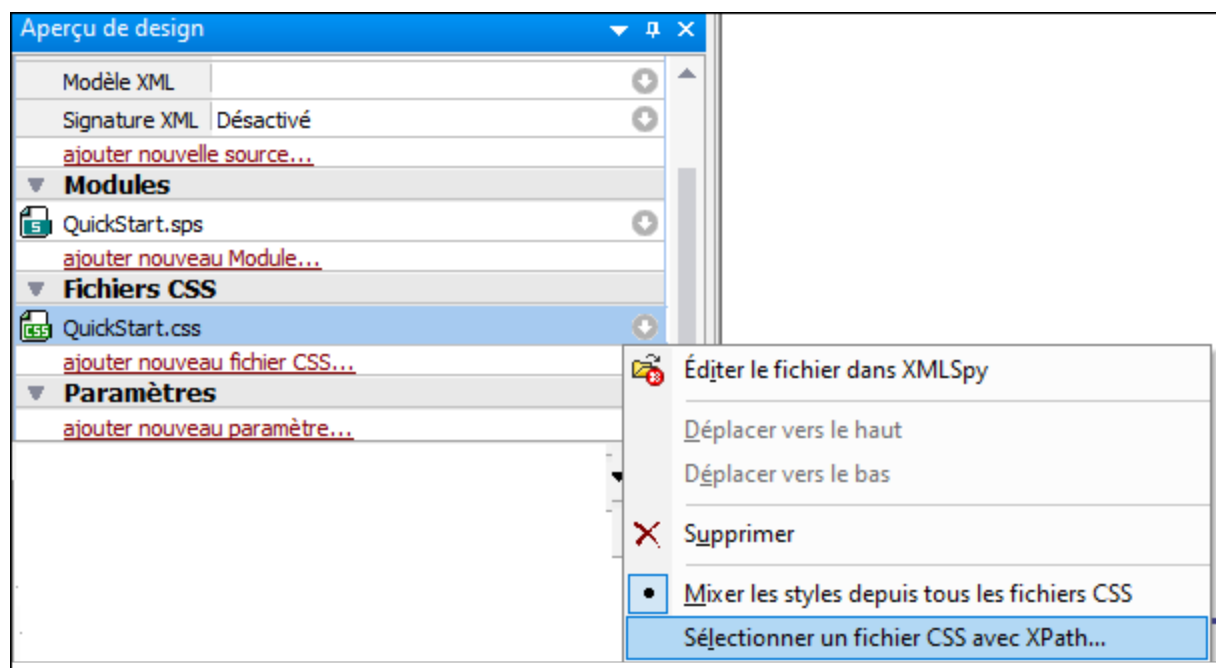
L'ordre dans lequel les fichiers sont recensés dans le fichier HTML correspond à l'ordre dans lequel ils sont recensés dans l'arborescence externe du Référentiel de Style et dans l'arborescence des fichiers CSS de la barre latérale de l'Aperçu de Design. Pour changer l'ordre des feuilles de style CSS dans le Référentiel de style, sélectionnez la feuille de style pour laquelle la préséance doit être changée. Puis, utilisez les touches

Déplacer vers le haut  ou **Déplacer vers le bas**  dans la barre d'outils du Référentiel de style pour repositionner cette feuille de style relative aux autres feuilles de style dans l'arborescence. Dans la barre latérale de l'Aperçu de Design, cliquez sur la touche Éditer de la feuille de style CSS et sélectionnez la commande **Déplacer vers le haut** ou **Déplacer vers le bas** tel que requis.

Important : Notez que si la feuille de style la plus basse qui a la **préséance d'importation la plus importante**, et que la préséance d'importation se réduit avec chaque feuille de style plus élevée dans l'ordre de la liste. L'ordre de la préséance d'importation dans la liste affichée ci-dessus est : (i) `ExternalCSS-3.css`; (ii) `ExternalCSS-2.css`; (iii) `ExternalCSS-1.css`. Lorsque les deux règles CSS, chacune dans une feuille de style différente, utilisez le même sélecteur, la règle dans la feuille de style avec la préséance d'importation la plus élevée s'applique.

Basculer entre tous les fichiers CSS et un seul fichier CSS

Vous pouvez choisir entre : (i) laisser les règles dans tous les fichiers CSS être appliquées avec les règles de cascade déterminant la préséance, ou (ii) laisser les règles dans un seul fichier CSS sélectionné être appliquées. Vous pouvez sélectionner l'option que vous voulez avoir dans la barre latérale de l'Aperçu de Design (voir la capture d'écran ci-dessous). Cliquez sur le bouton **Éditer** de tous fichiers CSS recensés et sélectionnez soit la commande **Mix Styles** ou la commande **Select One**. Cette option est aussi disponible dans le Référentiel de Style (sur toute feuille de style externe).



Si vous cliquez sur la commande **Sélectionner Un fichier CSS avec XPath**, un dialogue apparaît dans lequel vous pouvez saisir l'expression XPath (*capture d'écran ci-dessous*). L'expression XPath doit évaluer le nom d'un des fichiers CSS dans le SPS, exactement comme ces noms sont recensés dans le volet supérieur du dialogue. Si vous saisissez le nom du fichier comme string, notez que, tout comme tous les strings dans les expressions XPath, le string doit être saisi à l'intérieur de guillemets simples.

- **Lorsque les styles sont mixés depuis tous les fichiers CSS :** Dans les sorties Authentic et HTML, toutes les règles depuis tous les fichiers CSS sont appliquées et prises en charge dans tous les composants de design. Les conflits sont résolus sur la base de la priorité du fichier CSS. Dans la, seules les règles de sélecteur non-classe sont appliquées, avec des conflits étant résolus sur la base de la priorité.

8.4.2 Styles Globaux

Les styles globaux sont définis pour le design SPS entier dans le Référentiel de Style et recensés dans le Référentiel de style sous l'en-tête global. Ils sont passés à au document de sortie HTML comme règles CSS. Dans le document HTML, ces règles CSS sont écrites à l'intérieur de l'élément `/html/head/style`.

Dans le Référentiel de Style, un style global est une seule règle CSS consistant en un sélecteur et des propriétés CSS pour ce sélecteur. Créer un style global, pour cette raison, consiste en deux parties :

- Ajouter un nouveau style et déclarer le sélecteur CSS pour celui-ci
- Définir les propriétés CSS pour le sélecteur

Sélecteurs pris en charge

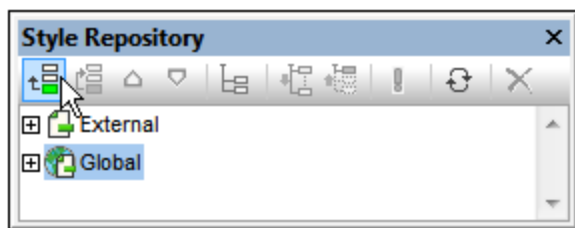
Les [sélecteurs](#) suivants sont pris en charge :

- *Sélecteur universel* : écrit comme `*`
- *Sélecteurs de type* : noms d'élément, tel que `h1`
- *Sélecteurs d'attribut* : par exemple, `[class=maindoc]`
- *Sélecteurs de classe* : par exemple, `.maindoc`
- *Sélecteurs ID* : par exemple, `#header`

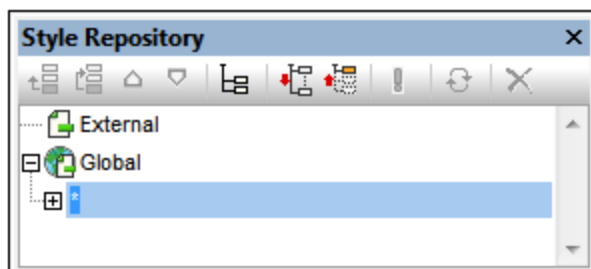
Ajouter un style global

Pour ajouter un style global au design SPS, suivez les étapes suivantes :

1. Dans l'Aperçu de design, sélectionnez un item Global dans la fenêtre du Référentiel de Style (*capture d'écran ci-dessous*).



2. Cliquez sur la touche **Ajouter** en haut à gauche de la barre d'outils du Référentiel de Style (*voir la capture d'écran ci-dessus*). Un style global est inséré dans l'arborescence global avec un sélecteur `*` (qui sélectionne tous les éléments HTML) ; le sélecteur universel est le sélecteur par défaut pour chaque style global nouvellement inséré.
3. Pour changer le sélecteur du sélecteur universel par défaut, double-cliquez sur le sélecteur et éditez-le.



4. Désormais, définissez les valeurs de la propriété CSS pour le sélecteur. Suivez à cette fin les explications dans la section, [Définir les valeurs de style](#) ³⁵².
5. Pour ajouter un autre style global, répétez les étapes 1 à 4. Le nouveau style global sera ajouté à l'arborescence globale, en dessous de tous les styles globaux ajoutés précédemment ajoutées.

Note :

- Les styles globaux peuvent aussi être insérés avant un style global sélectionné dans l'arborescence global en cliquant sur la touche **Insérer** dans la fenêtre du Référentiel de style. Les touches **Ajouter** et **Insérer** sont aussi disponibles via le menu contextuel qui apparaît quand vous cliquez avec la touche de droite sur un sélecteur global.
- Un style global avec un sélecteur qui est un élément HTML peut être inséré en cliquant avec la touche de droite sur l'élément dans l'arborescence global, puis en sélectionnant **Ajouter Sélecteur | HTML | HTML`ElementName`**.

Éditer et supprimer les styles globaux

Les deux sélecteurs de style ainsi que leurs propriétés peuvent être édités dans la fenêtre du Référentiel de style.

- Pour éditer un sélecteur, double-cliquez sur le nom du sélecteur, puis placez le curseur dans le champ de texte et éditez.
- Pour information sur la définition et l'édition des valeurs de la propriété de style, voir [Définir les valeurs de style](#) ³⁵². (Les propriétés de style peuvent être affichées de trois manières possibles. Ces manières et la façon dont vous pouvez basculer entre elles sont décrites dans [Opinions des définitions](#) ⁴⁷).

Pour supprimer un style global, sélectionnez le style et cliquez sur la touche **Réinitialiser** dans la barre d'outils du Référentiel de Style.

Modifier la préséance des styles globaux

Les styles globaux qui sont assignés dans la fenêtre du Référentiel de style sont placés comme règles CSS dans l'élément `/html/head/style`. Dans le fichier HTML, ils auraient l'aspect suivant :

```
<html>
  <head>
    <style>
      <!--
        h1  { color:blue;
              font-size:16pt;
            }
        h2  { color:blue;
              font-size:14pt;
            }
        .red { color:red;}
        .green { color:green;}
        .green { color:lime;}
      -->
    </style>
  </head>
</body/>
</html>
```

L'ordre dans lequel les styles globaux sont recensés dans Authentic View et le document HTML correspond à l'ordre dans lequel ils sont recensés dans l'arborescence global du Référentiel de style. L'ordre dans Authentic View et le document HTML ont une signification. Si deux sélecteurs sélectionnent le même nœud, alors le sélecteur qui apparaît plus bas dans la liste des styles globaux a préséance. Par exemple, dans le document HTML ayant une liste partielle donnée ci-dessus, s'il existait un élément `<h1 class="green">`, alors trois styles globaux correspondent à cet élément : celui avec le sélecteur `h1` et les deux sélecteurs de classe `.green`. La propriété de couleur du sélecteur `.green` avec la couleur `lime` s'appliquera car elle apparaît après le sélecteur `.green` avec la couleur `green` et a une préséance élevée pour cette raison. (Les sélecteurs de classe ont toujours une préséance plus élevée que les nœuds de sélecteurs, donc les deux sélecteurs `.green` auront une préséance plus élevée que le sélecteur `h1` indépendamment de leurs positions respectives pertinentes au sélecteur `h1`.) La taille de la police du style `h1`, s'appliquera, toutefois, à l'élément `<h1>` car il n'existe aucun sélecteur avec une préséance plus élevée qui correspond à l'élément `<h1>` et a une propriété `font-size`.

Pour changer la préséance du style global, sélectionnez ce style et utilisez les touches **Déplacer vers le haut** et **Déplacer vers le bas** dans la barre d'outils du Référentiel de Style pour repositionner ce style global par rapport aux autres styles dans l'arborescence. Par exemple, si le style global `.green` était déplacé vers une position avant le style `.red`, alors la propriété de couleur du style `.red` aurait préséance sur celle du style `.green`.

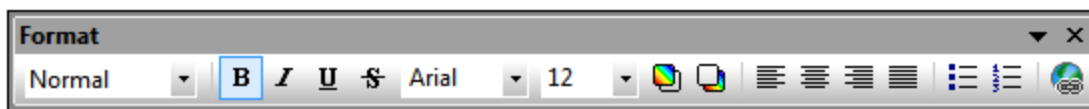
Notez, toutefois, que les sélecteurs de classe ont préséance sur des sélecteurs de type. Donc, si l'ordre du sélecteur était modifié à `.red .green h1 h2`, alors `h1` et `h2` serait toujours vert.

8.4.3 Styles locaux

Lorsque les styles sont définis localement, les règles de style sont définies directement dans le composant. Ces règles locales ont préséance sur les règles de style global et les règles de style dans les feuilles de style CSS externes qui sélectionnent ce composant. Les styles définis localement sont des styles CSS et sont définis soit via la [barre d'outils Format](#)⁴⁴⁵ ou dans la barre latérale [Styles](#)⁴⁴⁶. (Ceci est opposé aux styles globaux, qui sont définis dans la barre latérale du [Répertoire de Style](#)⁴⁴⁴.)

Styles locaux via la barre d'outils Format

Vous pouvez sélectionner le contenu dans le design et appliquer les styles locaux via la barre d'outils Format (*capture d'écran ci-dessous*).

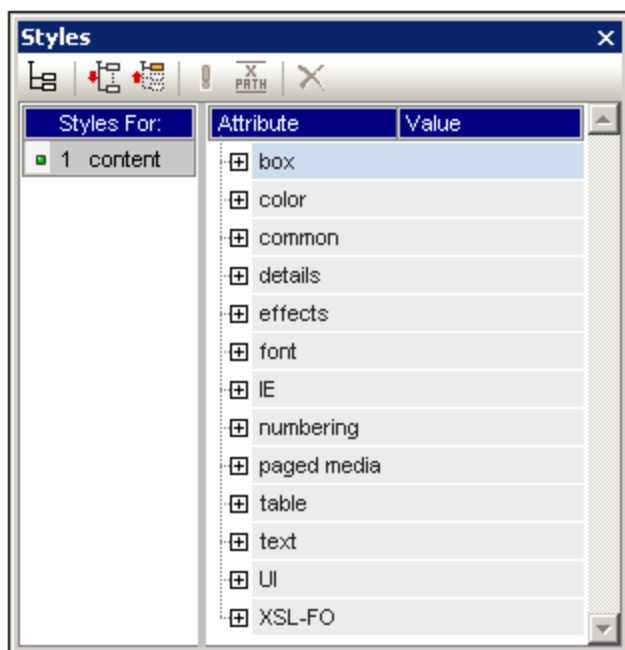


Vous pouvez appliquer du formatage HTML prédéfini (tel que `div`, `h1`, `pre`, etc.), style de texte, couleur d'arrière-plan, argument de texte, listes, et hyperliens. Voir la section [barre d'outils Format](#)⁴⁴⁵ pour plus de détails.

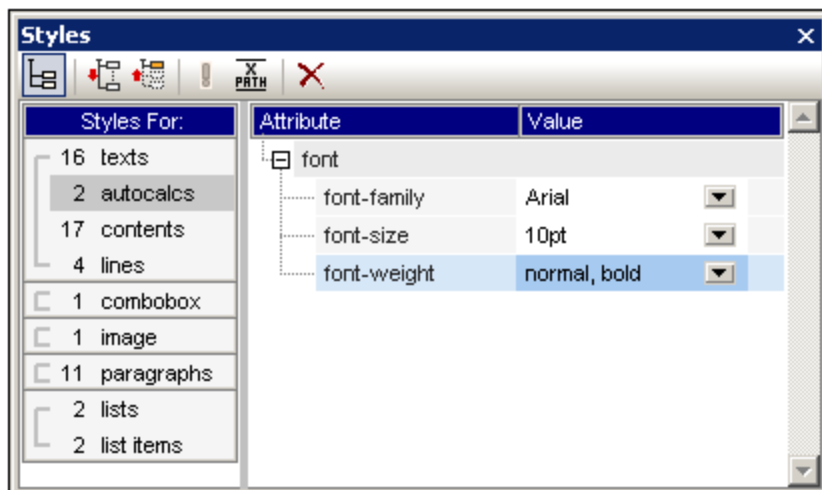
Styles locaux via la barre latérale Styles

Définir un style localement via la barre latérale Styles consiste en trois parties :

1. Le composant à styler est sélectionné dans le Mode Design. Tout composant dans le design à l'exception des balises de nœuds peut être stylé. Le composant sélectionné dans le Mode Design apparaît ensuite dans la colonne *Styles-For* de la barre latérale des Styles (voir capture d'écran ci-dessous). Dans la capture d'écran ci-dessous, un composant `content` était sélectionné dans le Mode Design et apparaît donc dans la colonne *Styles-For*.



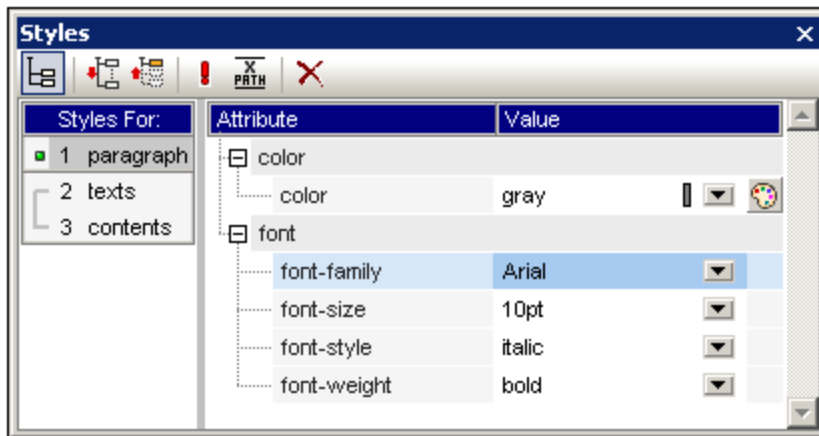
Très souvent, le composant sélectionné dans le Mode Design peut contenir d'autres composants. Dans ce cas, tous les composants dans la sélection sont affichés, organisés par le component-type, dans la colonne *Styles-For* de la barre latérale des Styles. La capture d'écran ci-dessous affiche les différents component-types contenus dans la sélection du Mode Design. À gauche de chaque component-type se trouve le nombre d'instances de ce component-type dans la sélection. Par exemple, dans la capture d'écran ci-dessous, il existe 16 composants texte et deux composants de calcul automatique (entre autres) dans la sélection du Mode Design. Vous pouvez aussi sélectionner une gamme de composants en maintenant la touche **Shift** appuyée tout en sélectionnant le deuxième composant à la fin de l'étendue.



- Après avoir fait la sélection dans le Mode Design, vous sélectionnez, dans la colonne *Styles-For*, le **component-type** que vous souhaitez styler. S'il existe plus d'une instance du component-type que vous sélectionnez, alors les styles que vous définissez seront appliqués à toutes ces instances. Donc, par exemple, si vous sélectionnez l'item *16-texts* de la capture d'écran ci-dessus, alors les styles que vous définissez (voir *Étape 3 ci-dessous*) seront appliqués à tous les 16 composants `text`. Si vous souhaitez styler, disons, quatre de ces composants de texte différemment, vous devez sélectionner et styler chacun des quatre composants séparément. Si deux composants du même component-type ont été stylés différemment et que les deux sont sélectionnés dans le Mode Design, alors les styles des deux instances sont affichés dans le volet *Définitions de style*. Dans la capture d'écran ci-dessus, par exemple, tandis qu'un calcul automatique a un font-weight normal, l'autre a un font-weight gras. Lorsque l'item *2-Autocalcs* est sélectionné dans le volet *Styles-For*, les deux font-weights sont affichés.
- Après la sélection, dans la colonne *Styles-For*, le component-type à style, les styles sont définis dans le [volet des Définitions de Style](#)⁴⁶. Vous trouverez la description à cette fin dans la section [Définir les valeurs de style](#)³⁵².

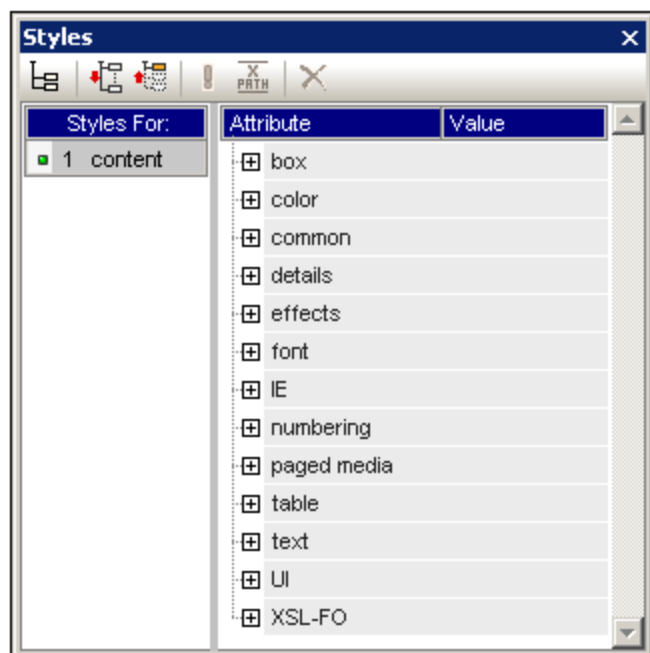
8.4.4 Définir des valeurs de Style

Pour les composants type sélectionnés dans la colonne *Styles-For*, les propriétés de style sont définies dans le [volet des Définitions de style de la barre latérale des Styles](#)⁴⁶ (capture d'écran ci-dessous). Vous pouvez sélectionner plus d'un component-type dans la colonne *Styles-For* si vous voulez—en sélectionnant les component-types supplémentaires avec la touche **Ctrl** appuyée, ou en sélectionnant une gamme de component-types dans la colonne *Styles-For* avec la touche **Shift** appuyée. Lorsque de multiples component-types sont sélectionnés, toute valeur de style que vous définissez dans le volet *Définitions de Style* est appliquée à toutes les instances des component-types sélectionnés.



Groupe de Style de propriété

Les propriétés de style disponibles dans la colonne *Style-Definitions* sont organisées en groupes tels qu'affichés dans la capture d'écran ci-dessous.

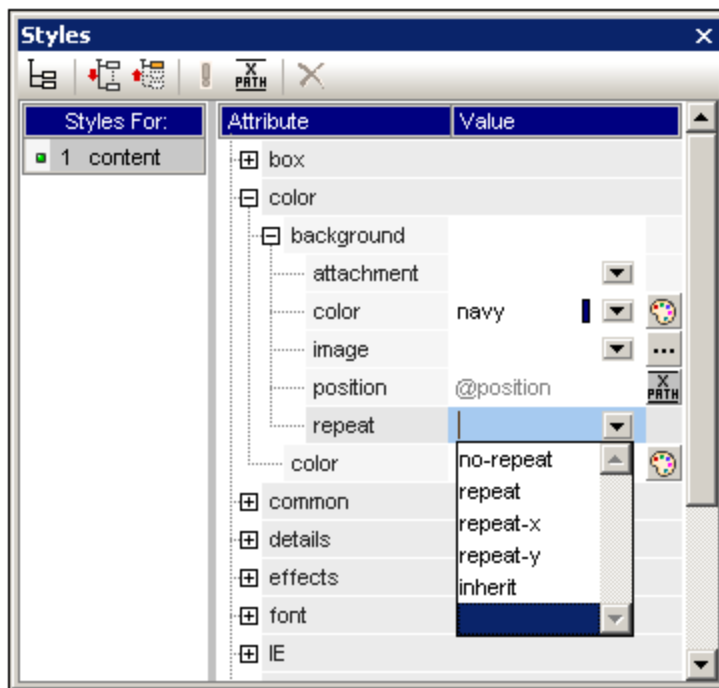


L'affichage des propriétés peut être modifié utilisant les [boutons de la barre d'outils](#) ⁴⁶ [List Non-Empty](#) ⁴⁶, [Expand All](#), et [Collapse All](#) ⁴⁶ [***](#) ⁴⁶. Chaque groupe des propriétés de style peut être élargi pour accéder aux propriétés de style ou sous-groupes de propriétés de style (voir la capture d'écran ci-dessous).

Saisir des valeurs de style

Les valeurs de la propriété de style (valeurs de style pour court) peuvent être saisies des manières suivantes, toutes sont affichées dans la capture d'écran ci-dessous :

- Saisir directement dans la colonne Valeur. À cette fin, sélectionnez une propriété, double-cliquez sur sa colonne Valeur, saisissez la valeur utilisant le clavier et appuyez sur **Saisir** ou cliquez n'importe où dans la GUI.
- En sélectionnant une valeur à partir de la liste déroulante dans la liste de choix pour cette propriété. Cliquez sur la flèche du bas de la zone de liste déroulante des options style-valeur. Dans la capture d'écran ci-dessous, les options pour la propriété (background-) repeat sont affichées. Sélectionner la valeur requise depuis la liste déroulante.
- En utilisant l'icône à droite de la colonne Valeur pour cette propriété de style. Deux types d'icônes sont disponibles, et celles-ci sont disponibles uniquement pour les propriétés pour lesquelles elles sont pertinentes : (i) une palette de couleur pour sélectionner les couleurs (dans la capture d'écran ci-dessous, voir la propriété (background-) color), et (ii) un dialogue pour parcourir les fichiers (dans la capture d'écran ci-dessous, voir la propriété (background-) image).



- Les valeurs pour les styles peuvent aussi être attribuées via une [expression XPath](#) ³⁵⁴.

Modifier ou supprimer une valeur de style

Si une valeur de style est saisie de manière incorrecte ou est invalide, elle est affichée en rouge. Pour modifier la valeur, utilisez toute méthode applicable décrite dans la section précédente, [Saisir des valeurs de propriété](#) ³⁵³.

Pour supprimer une valeur de style (ou, en d'autres termes, pour réinitialiser une valeur de style), cliquez sur la touche Reset dans la barre d'outils de la barre latérale des Styles. En alternative, vous pouvez double-cliquer dans la colonne Valeur de la propriété et supprimer la valeur utilisant la touche **Supprimer** et/ou **Retour Arrière**, puis en appuyant sur **Enter**.

8.4.5 Propriétés de style via XPath

Les styles peuvent être assignés pour concevoir les composants via des expressions XPath. Ceci permet aux valeurs des propriétés de style d'être prélevées des données XML ou de l'expression XPath elle-même. Aussi, en utilisant la fonction `doc()` de XPath 2.0/3.0, les nœuds dans tout document XML accessible peuvent être traités. Ce ne sont non seulement des définitions de style qui peuvent être tirées de données XML extraites ; cette fonction permet également de faire des choix de style qui sont conditionnels en matière de structure ou contenu des données XML. Par exemple, utiliser la déclaration `if...else` de XPath 2.0/3.0, deux couleurs d'arrière-plan différentes peuvent être sélectionnées dépendant de la position d'un élément dans la séquence. Donc, lorsque ces éléments sont présentés comme lignes dans une table, les lignes impaires numérotées peuvent être présentées avec une couleur d'arrière-plan tandis que les lignes paires numérotées sont présentées avec une autre (*voir l'exemple ci-dessous*). Aussi, dépendant du contenu d'un nœud, la présentation peut varier.

Les propriétés de style pour lesquelles les expressions XPath sont activées

Les expressions XPath peuvent être saisies pour les propriétés de style suivantes :

- Toutes les propriétés disponibles dans la barre latérale des Styles
- Les groupes de propriétés *Common*, *Event*, et *HTML* dans la barre latérale des Propriétés

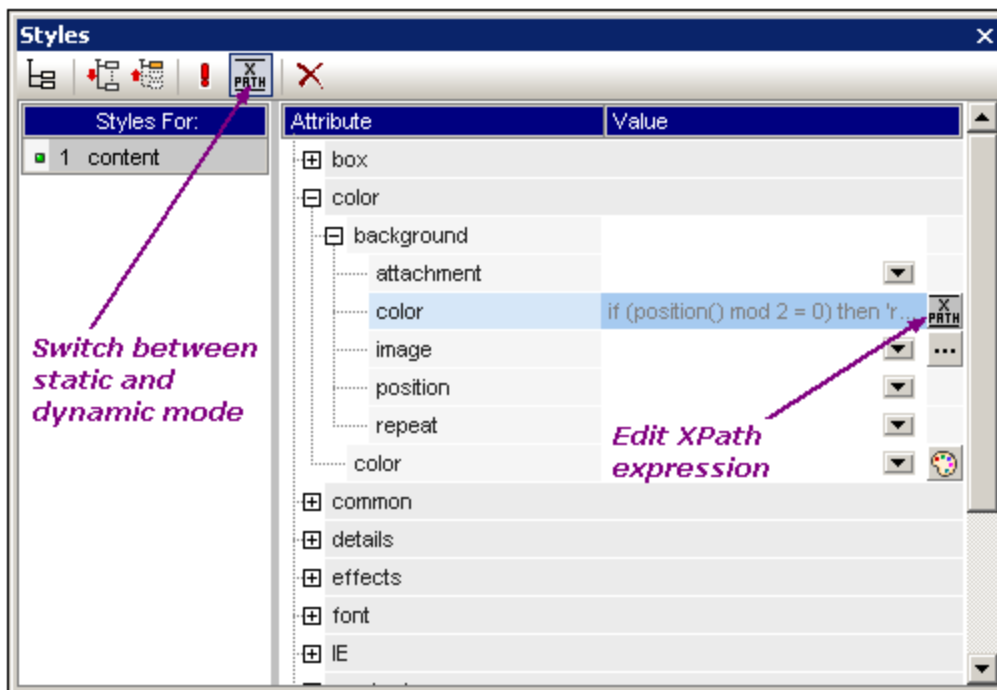
Mode statique et mode dynamique (XPath) pour les valeurs de propriétés

Pour ces propriétés où les [expressions XPath sont activées](#) ³⁵⁴, deux modes sont disponibles :

- Le mode statique, où la valeur de la propriété est saisie directement dans la colonne Valeur de la barre latérale. Par exemple, pour la couleur d'arrière-plan d'un composant de design, la valeur `red` peut être saisie directement dans la barre latérale.
- Dynamique, ou mode XPath, où une expression XPath est saisie. L'expression est évaluée au moment de l'exécution, et le résultat est saisi comme valeur de la propriété. Par exemple, pour la couleur d'arrière-plan d'un composant de design, l'expression XPath suivante peut être saisie : `/root/colors/color1`. Au moment de l'exécution, le contenu du nœud `/root/colors/color1` sera extrait et saisi comme la valeur de la propriété de la couleur d'arrière-plan.

Basculer entre les modes statique et dynamique (XPath)

Pour chaque propriété pour laquelle les expressions XPath sont activées, le mode statique est sélectionné par défaut. Pour basculer une propriété vers le mode dynamique (XPath), sélectionnez cette propriété et cliquez sur l'icône XPath dans la barre d'outils de la barre latérale (*capture d'écran ci-dessous*).



Si une valeur statique était présente pour cette propriété, elle est désormais délogée et le mode est basculé vers dynamique. Le [dialogue Éditer l'expression XPath](#) ⁴²⁶ s'ouvrira. C'est dans ce dialogue que vous saisissez l'expression XPath pour cette propriété. Cliquez sur **OK** lorsque vous avez terminé.

Après avoir saisi une expression XPath pour la propriété, une touche d'expression **Éditer XPath** apparaît dans la colonne Valeur pour cette propriété (*capture d'écran ci-dessus*). Cliquer sur cette touche afin d'éditer l'expression XPath par la suite. Si vous souhaitez basculer de retour vers le mode statique, cliquez sur l'icône XPath dans la barre d'outils. Ceci effacera l'expression XPath et basculera la propriété vers le mode statique.

Note : deux points importants. D'abord, seul un mode (statique ou dynamique), et la valeur/expression pour ce mode, est actif à tout moment. Toute valeur/expression qui existait précédemment pour l'autre mode est effacé ; donc basculer vers l'autre mode présentera ce mode avec un champ d'entrée vide. (Afin de retourner vers une valeur/expression précédente, utilisez la [commande Undo](#) ⁴⁷⁶.) Deuxièmement, si vous re-sélectionnez une propriété après avoir édité davantage le SPS, alors cette propriété sera ouverte dans le mode dans lequel il était précédemment.

Créer et éditer la définition XPath

La définition XPath est créée et éditée dans le [dialogue de l'expression Éditer XPath](#) ⁴²⁶. Ce dialogue est accessible de deux manières :

- À chaque fois que vous basculez vers le mode dynamique d'une propriété du mode statique (en cliquant sur l'icône XPath dans la barre d'outils de la barre latérale), le [dialogue Éditer expression XPath](#) ⁴²⁶ apparaît. Vous pouvez désormais créer les expressions XPath. (Notez que cliquer sur l'icône de la barre d'outils lorsqu'elle est déjà en mode dynamique bascule le mode vers le mode statique ; il n'ouvre pas le dialogue Éditer l'expression XPath.)
- Le [dialogue Éditer expression XPath](#) ⁴²⁶ s'ouvre également lorsque vous cliquez sur le bouton **Éditer expression XPath** dans le champ Valeur d'une propriété qui a déjà une expression XPath définie pour celle-ci. Le dialogue contiendra l'expression XPath déjà définie pour la propriété que vous pouvez éditer maintenant.

Une fois que vous avez saisi ou édité l'expression XPath dans votre champ de saisie, cliquez sur **OK** pour terminer.

Valeurs retournées par les expressions XPath

Les avantages les plus importants d'utiliser les expressions XPath pour définir une valeur de propriété sont : (i) la valeur de propriété peut être prise depuis un fichier XML (à la place d'être directement saisie) ; et/ou (ii) une expression XPath peut tester certaines conditions associées au contenu ou à la structure du document XML étant traité, et sélectionner une valeur en fonction en conséquence. Les expressions XPath renvoient les valeurs dans les deux catégories suivantes :

- **Contenu du nœud XML**
L'expression XPath peut aborder les nœuds dans : (i) le document XML étant traité par SPS, ou (ii) tout document XML accessible. Par exemple, l'expression `Format/@color` accéderait l'attribut `color` de l'enfant `Format` du nœud contextuel. La valeur de l'attribut `color` sera définie comme valeur de la propriété pour laquelle l'expression XPath a été définie. Un nœud dans certains documents XML peut être accédé utilisant la fonction `doc()` de XPath 2.0. Par exemple, l'expression `doc('Styles.xml')//colors/color-3` extrairait la valeur de l'élément `color-3` dans le document XML `Styles.xml` et définissez cette valeur comme valeur de la propriété pour laquelle l'expression XPath a été définie.
- **Expression XPath**
La valeur de la propriété peut venir de l'expression XPath elle-même, et non du document XML. Par exemple, la couleur d'arrière-plan d'un élément qui est sorti comme ligne peut être décidée pour alterner selon si la position de la ligne a des nombres impairs ou pairs. Ceci peut être réalisé à l'aide

de l'expression XPath 2.0/3.0 suivante : `if (position() mod 2 = 0) then 'red' else 'green'`. Notez que la valeur de renvoi de cette expression est soit le string `red` ou le string `green`, et elle sera définie comme valeur de la propriété pour laquelle l'expression XPath a été définie. Dans l'exemple cité, les valeurs de propriété étaient saisies comme chaînes littérales. Ils peuvent également provenir d'un document XML, par exemple : `if (position() mod 2 = 0) then doc('Styles.xml')//colors/color-1 else doc('Styles.xml')//colors/color-2`. Inversement, l'expression XPath pourrait être un string simple, par exemple : `'green'` : Toutefois, il s'agit de la même chose que si vous saisissez la valeur statique `green` pour la propriété.

8.4.6 Styles Composite

Un style composite est un groupe de propriétés de style du texte CSS qui ont été associées à un attribut d'un nœud de document d'instance XML. De plus, tout groupe de propriétés de style du texte CSS stocké dans la feuille de style est également considéré être un Style composite. Les Styles composite peuvent ensuite être spécifiés dans les composants de design suivants :

- [Calculs automatiques](#) ²⁶¹
- [L'espace réservé \(contenus\)](#) ¹¹⁰
- [Éléments de design du paragraphe \(bloc\)](#) ¹¹²
- [Cellules de table](#) ¹²⁵

Avantages des Styles composite

Les Styles composite offrent les avantages suivants :

- Les propriétés de Styling sont dans les données XML et peuvent donc être éditées par l'utilisateur.
- Les propriétés de style des composants de design recensés ci-dessus peuvent être une combinaison de propriétés stockées dans les données et propriétés XML assignées dans SPS.
- Dans la phase de design SPS, le designer SPS peut rapidement basculer entre les multiples styles Composite associés à un élément.

Saisir le Style Composite dans l'attribut XML

Un Style Composite (composé de multiples propriétés de style) est saisi comme valeur-attribut d'un élément dans le document XML source. Par exemple, l'attribut `desc-style` dans le document source XML recensé ci-dessous contient un Style Composite par défaut :

```
<Desc desc-style="font-family:Verdana; font-size:12pt; color:blue">
```

Vous pouvez également définir plus d'un Style Composite d'un élément. Dans ce cas, chaque Style Composite doit être saisi dans un attribut séparé :

```
<Desc styleBlue="font-family:Verdana; font-size:12pt; color:blue"
      styleRed="font-family:Verdana; font-size:12pt; color:red">
```

Lorsque des Styles Composite sont disponibles sur un élément, vous pouvez basculer parmi des Styles Composite lorsque vous définissez une valeur pour la propriété *Style Composite* d'un composant de design (voir ci-dessous).

Note : les attributs qui seront utilisés pour accéder aux Styles Composite Styles doivent être définis dans le schéma de source pour que le document XML soit valide.

Propriétés CSS text-styling prises en charge

Les styles CSS suivantes peuvent être utilisés dans les Styles Composite :

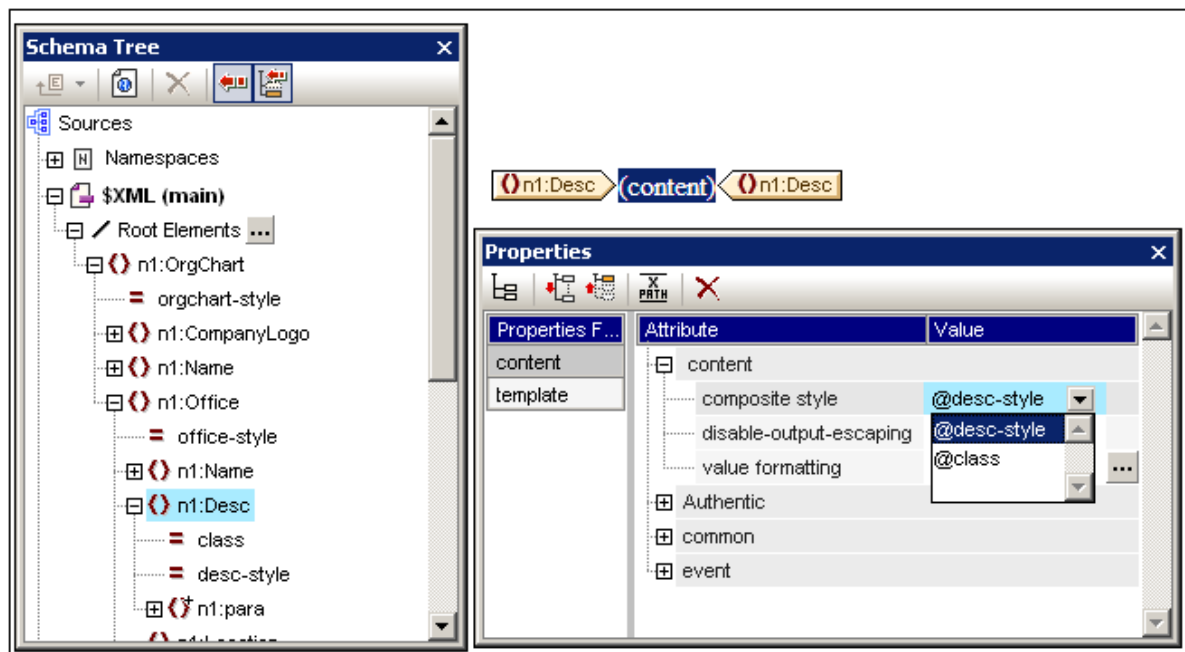
font-family	font-size	font-weight	font-style
color	background-color	text-align	text-decoration

Définir un attribut comme valeur de Style Composite

Si vous définissez le Style Composite d'un composant de design pour qu'il soit un attribut, alors l'utilisateur Authentic View peut éditer ce Style Composite.

Pour définir un attribut comme Style Composite, suivez les étapes suivantes :

1. Dans le Mode Design, sélectionnez le composant de design auquel vous voulez assigner un attribut comme Style Composite. Dans la capture d'écran ci-dessous, un espace réservé (contenus) de l'élément `Desc` a été sélectionné.



2. Dans la liste déroulante de la propriété `Style Composite` du composant `Content` (voir la barre latérale des Propriétés en bas à droite de la capture d'écran ci-dessus), les attributs de l'élément contextuel sont affichés. Sélectionnez l'attribut que vous souhaitez définir comme Style Composite du composant de design.

Définir une expression XPath comme valeur de Style Composite

Vous pouvez saisir une expression XPath en tant que la valeur d'une propriété de *Style Composite*. Dans ce cas, toutefois, puisque le Style Composite est stocké dans le SPS (pas dans le document source XML), Authentic View ne sera pas à même d'éditer le Style Composite.

Pour définir l'expression XPath comme valeur de la propriété *Style Composite*, cliquez sur l'icône **XPath** dans la barre d'outils de la barre latérale des Propriétés, puis saisissez l'expression XPath dans le dialogue XPath qui apparaît. L'expression XPath sera évaluée comme modèle de valeur d'attribut ; la valeur renvoyée sera la valeur d'un attribut HTML `style` (et son équivalent est un format de sortie non-HTML).

Par exemple, tenez compte de l'expression XPath suivante créée dans l'espace réservé (contenus) de l'élément `n1:Person`.

```
if (number(n1:Shares) gt 1000) then 'color:red' else 'color:green'
```

Ce que cette expression fera : Si l'élément `n1:Person` a un élément enfant `n1:Shares` avec une valeur supérieure à 1000, alors le contenu de l'élément `n1:Person` est la sortie en rouge ; autrement, tous les éléments `n1:Person` sont la sortie en vert. La valeur renvoyée par l'expression XPath est passée vers le document de sortie comme valeur d'un attribut HTML `style` (ou de son équivalent dans des formats de sortie non-HTML).

Dans la feuille de style XSLT générée depuis le SPS, cette expression XPath sera évaluée comme un modèle de valeur d'attribut, comme ceci :

```
<span style="{if (number(n1:Shares) gt 1000) then &apos;color:red&apos; else  
&apos;color:green&apos;}">
```

Dans la sortie HTML, une des lignes suivantes serait générée dépendant de la manière dont la condition est évaluée :

```
<span style="color:red">
```

ou

```
<span style="color:green">
```

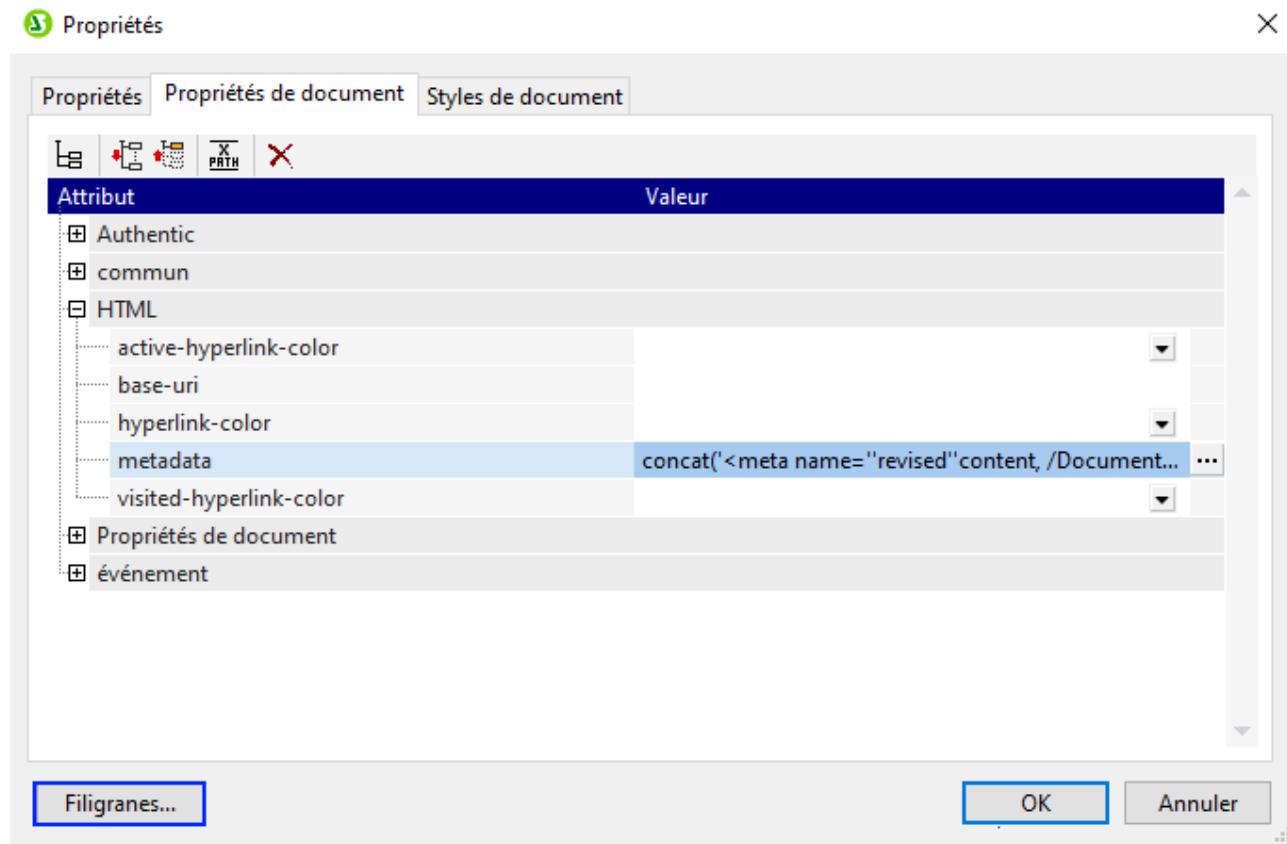
Note : les modèles de valeur de l'attribut sont des constructions XSLT qui permettent à la valeur d'un attribut d'être lues comme expression XPath. Elles sont délimitées par des accolades et permettent que la valeur de l'attribut soit assignée de manière dynamique.

8.5 Propriétés Document HTML

Les propriétés de la sortie du document HTML peuvent être spécifiées soit dans l'onglet *Propriétés de document* dans le dialogue des Propriétés de la Section initiale du document ou dans l'Aperçu des Propriétés quand le [modèle principal dans la fenêtre d'arborescence de design](#) ⁴⁰ est sélectionné.

Par le biais des propriétés de la Section initiale du document

Cliquez sur l'hyperlien *Éditer Propriétés* dans la barre de titre Section initiale du document, puis sélectionnez l'onglet *Propriétés de document*.



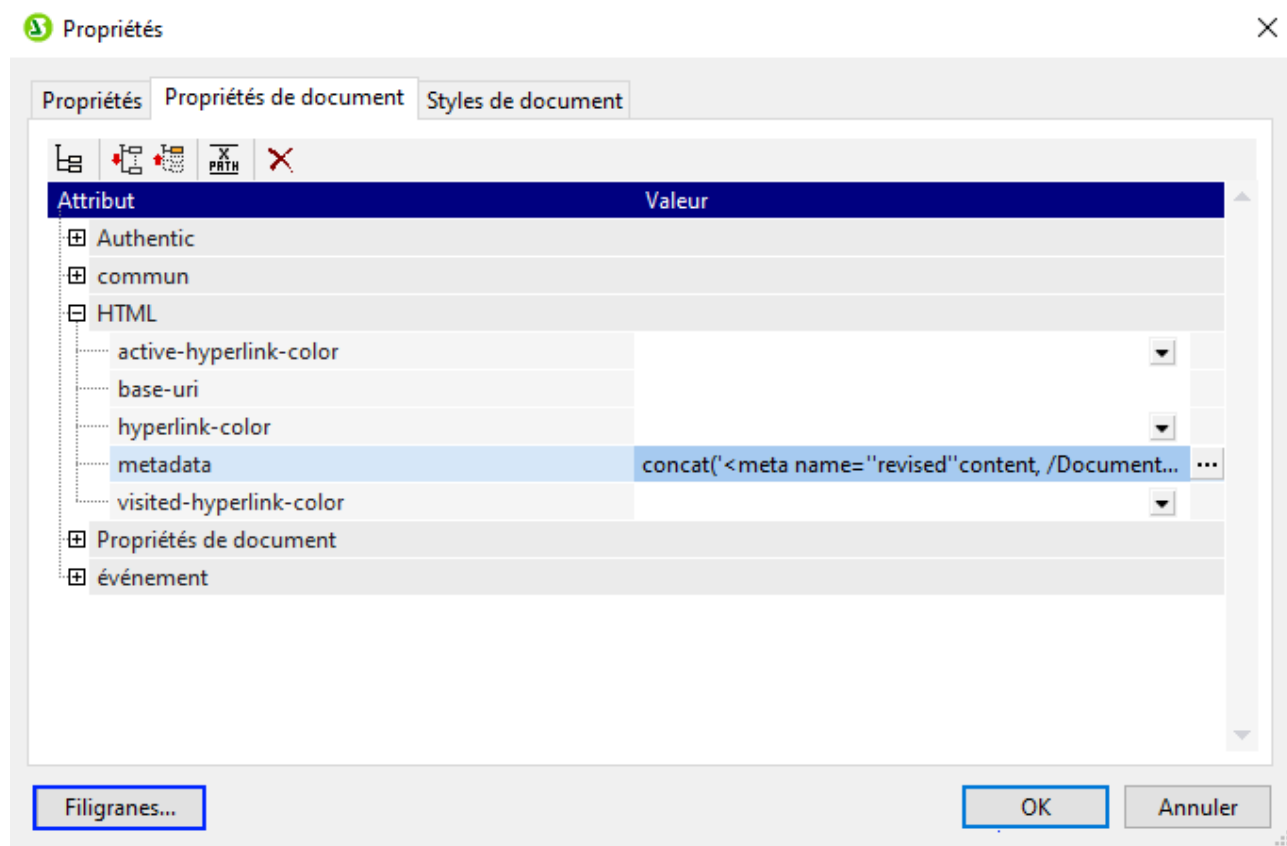
Ici, vous pouvez définir diverses variétés du document de sortie HTML qui sera généré. Notez les points suivants :

- Il s'agit de propriétés HTML qui seront appliquées au niveau du document, par exemple, les propriétés `class` et `id` de la section des propriétés *Commun*.
- Les sections *Propriétés de document* et *HTML* contiennent des propriétés génériques qui sont associées au document HTML dans son ensemble, tel des couleurs de lien, et l'information qui va dans les balises `meta` du document HTML.
- La propriété `metadata` de la section *HTML* vous permet de saisir tout texte que vous choisissez pour aller dans l'élément `HEAD` du document HTML (voir les captures d'écran ci-dessus et ci-dessous, qui sont de l'Enterprise Edition de StyleVision et contiennent les fonctions spécifiques de cette édition). Le texte que vous saisissez pourraient, par exemple, être un élément `script` ou `meta`, ou plusieurs de ces éléments. Vous pouvez saisir les éléments HTML directement comme texte (sans guillemets) ou,

comme dans les captures d'écran, comme expression XPath. Dans l'exemple de capture d'écran, l'expression XPath définit une méta-balise pour la date de révision d'un document.

Par le biais des propriétés du modèle principal

Dans la barre latérale [Arborescence de design](#) ⁴⁰, sélectionnez le modèle principal. Dans la barre latérale des Propriétés (*capture d'écran ci-dessous*), vous pouvez désormais définir les propriétés du document de sortie HTML.



Les propriétés dans ce dialogue sont exactement les mêmes que celles dans la balise *Propriétés de document* décrit ci-dessus.

9 Fonction additionnelle

En plus des procédures de l'[édition du contenu](#)¹⁰⁹, la [structure](#)¹⁸⁵, [avancé](#)²⁶⁰, et [présentation](#)³²⁸ décrites dans cette documentation, StyleVision fournit des fonctions supplémentaires diverses. Celles-ci sont recensées ci-dessous et décrites en détail dans les sous-sections de cette section.

- [Travailler avec des dates](#)³⁸⁵. Les données peuvent être manipulées et formatées, tel que requis.
- [URI d'entité non parsées](#)³⁶³. Les URI peuvent être stockées dans des entités non parsées dans le DTD sur lequel le document XML est basé. La fonction de l'URI d'entité non parsée permet aux images et hyperliens d'utiliser ces URI comme URI cible.
- [Utiliser des Scripts](#)³⁸⁹. StyleVision contient un Éditeur JavaScript dans lequel les fonctions JavaScript peuvent être définies. Ces fonctions sont ensuite disponibles pour être utilisées comme gestionnaires d'événement partout dans SPS, et prendront effet dans le document de sortie HTML.
- [Importation HTML](#)³⁹⁴. Un fichier HTML peut être importé dans StyleVision et les fichiers XML, XSD et SPS peuvent être créés de celui-ci.
- [Nouveautés de XSLT](#)³⁶⁵. Une SPS peut être créée de XSLT-pour-HTML ou de XSLT-pour-FO. La structure et le style du modèle dans le XSLT seront créés dans le SPS. Vous pouvez ensuite modifier les composants SPS et ajouter du contenu et des formats au SPS.

Voir aussi

- [Barre latérale des propriétés](#)⁴⁸

9.1 URI d'entité non parsés

Si vous utilisez une DTD et avez déclaré une entité non parsée dans celui-ci, vous pouvez utiliser l'URI associé à cette entité pour l'image et les cibles d'hyperlien dans SPS. Ceci est utile si vous voulez utiliser le même URI de multiples fois dans SPS. Cette fonction utilise la fonction XSLT `unparsed-entity-uri` pour passer l'URI de l'entité non parsée du DTD à la sortie

Utiliser cette fonction requiert que le DTD, le document XML et les documents SPS soient édités de manière appropriée, comme suit :

1. Dans le DTD, les [entités non parsées doivent être déclarées](#) ³⁶³, avec (i) l'URI, et (ii) la notation (qui indique le type de ressource de l'entité StyleVision).
2. Dans le document XML, l'entité non parsée doit être [référéncée](#) ³⁶³. Ceci est réalisé en donnant les noms des entités non parsées requises.
3. Dans le SPS, les entités non parsées peuvent être utilisées comme cible [images](#) ¹⁵² et [hyperliens](#) ³²³ en [accédant correctement aux valeurs de nœud dynamiques pertinentes comme entités non parsées](#) ³⁶³.

Déclarer et référencer les entités non parsées

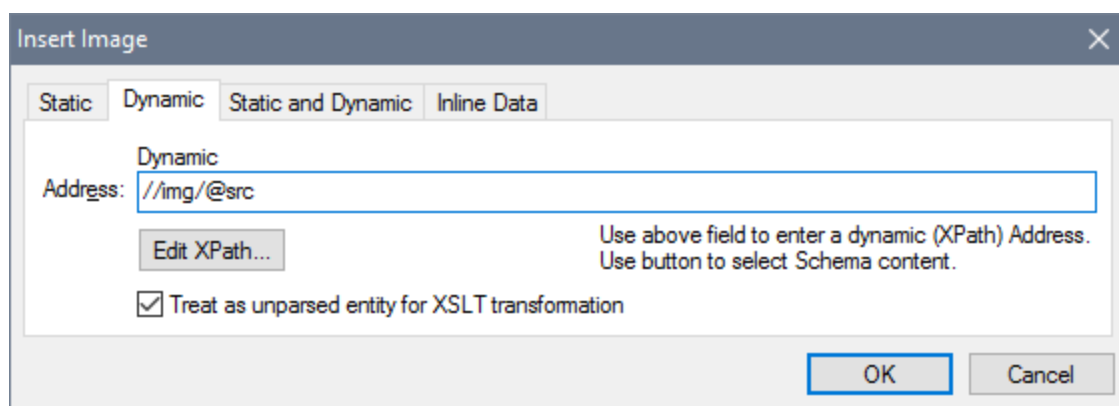
Ci-dessous, vous trouverez une liste allégée d'un document XML. Il s'agit d'un sous-ensemble DTD interne qui déclare deux entités non parsées, un avec une notation GIF (indiquant une image GIF) et l'autre avec la notation LNK (indiquant un hyperlien). Les nœuds `img/@src` et `link/@href` dans la référence de code XML référence les entités non parsées en donnant leurs noms.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE document SYSTEM "UEURIDoc.dtd" [
<!ENTITY Picture SYSTEM "nanonull.gif" NDATA GIF>
<!ENTITY AltovaURI SYSTEM "http://www.altova.com" NDATA LNK>
]>
<document>
  <header>Example of How to Use Unparsed Entity URIs</header>
  <para>...</para>
  
  <link href="AltovaURI">Link to the Altova Website.</link>
</ document>
```

Images SPS et hyperliens qui utilisent les entités non parsées

Images et hyperliens dans SPS qui référence les URI d'entité non parsés sont utilisés comme suit :

1. Insérez l'image ou l'hyperlien via le menu **Insert**.
2. Dans le dialogue Éditer de l'objet, sélectionnez les propriétés de l'onglet Dynamic (*capture d'écran ci-dessous*), et saisissez l'expression XPath qui sélectionne le nœud contenant le nom de l'entité non parsée. Dans l'exemple du document XML donné ci-dessus, ces nœuds seraient, respectivement, les nœuds `//img/@src` et `/link/@href`.



3. Puis vérifiez la case à cocher 'Treat as Unparsed Entity' en bas du dialogue. Ceci fait que le contenu du nœud sélectionné est lu comme entité non parsée. Si une entité non parsée de ce nom est déclarée, l'URI associé avec cette entité non parsée est utilisé pour localiser la ressource (image ou hyperlien).

Lorsque la feuille de style est traitée, l'URI associé avec le nom d'entité est substitué pour le nom de l'entité.

Note : si l'URI est un URI relatif, alors le processeur XSLT l'élargit à un URI absolu appliquant l'URI de base de la DTD. Par exemple, si l'entité non parsée est associée à l'URI relatif "nanonull.gif", alors cet URI sera élargi à `file:///c:/someFolder/nanonull.gif`, où la DTD est dans le dossier `someFolder`.

9.2 Nouveauté du fichier XSLT, XSL-FO ou FO

Un design SPS peut être basé sur les fichiers XSLT existants qui ont été conçus pour la sortie HTML ou les fichiers XSLT avec les commandes XSL-FO pour la sortie dans les fichiers PDF ou FO. Ceci signifie que les fichiers SPS ne doivent pas être conçus de zéro, mais peuvent prendre un fichier XSLT déjà existant comme point de départ.

Les étapes pour créer un SPS de XSLT

Les étapes pour créer un fichier SPS depuis un fichier XSLT, XSLT-for-FO, ou FO sont comme suit.

1. Sélectionner la commande **Fichier | Nouveauté | Nouveauté du fichier XSLT, XSL-FO ou FO**.
2. Dans le dialogue Ouvrir qui apparaît, parcourez pour le fichier que vous souhaitez.
3. Dans le prochain dialogue, vous serez invité à sélectionner un schéma sur lequel la SPS doit être basée. Sélectionnez le schéma que vous désirez.
4. Un SPS basé sur la structure et le formatage dans le fichier XSLT ou FO sera créé et affiché dans le [Mode de design](#) ²⁸.
5. Vous pouvez modifier maintenant la SPS de la manière usuelle. Par exemple, vous pourriez glisser dans les nœuds de l'[arborescence de schéma](#) ³⁷, modifier le style et la présentation ou ajouter un style supplémentaire, et utiliser la fonction de StyleVision comme [calculs automatiques](#) ²⁶¹ et [modèles conditionnels](#) ²⁶⁶.
6. Vous pouvez enregistrer la SPS et utiliser [Working XML File](#) ³⁵ pour consulter des [formats de sortie divers](#) ³⁰. Par la suite, vous pouvez [générer des feuilles de style et fichiers de sortie](#) ⁴⁶⁹ en utilisant la commande [Enregistrer fichiers générés](#) ⁴⁶⁹.

Exemple

Les exemples discutés ci-dessous sont situés dans le [dossier \(Mes\) Documents](#) ²⁴, C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2027\StyleVisionExamples\Tutorial\NewFromXSLT. Ce dossier contient des fichiers : SimpleExample.xslt, SimpleExample.xsd, et SimpleExample.xml.

Le fichier XML est affiché ci-dessous.

📄 Exemple de fichier XML utilisé dans les graphiques : YearlySales.xml

```
<?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>
    <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>
  </Region>
</Data>
```

```

        <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>

```

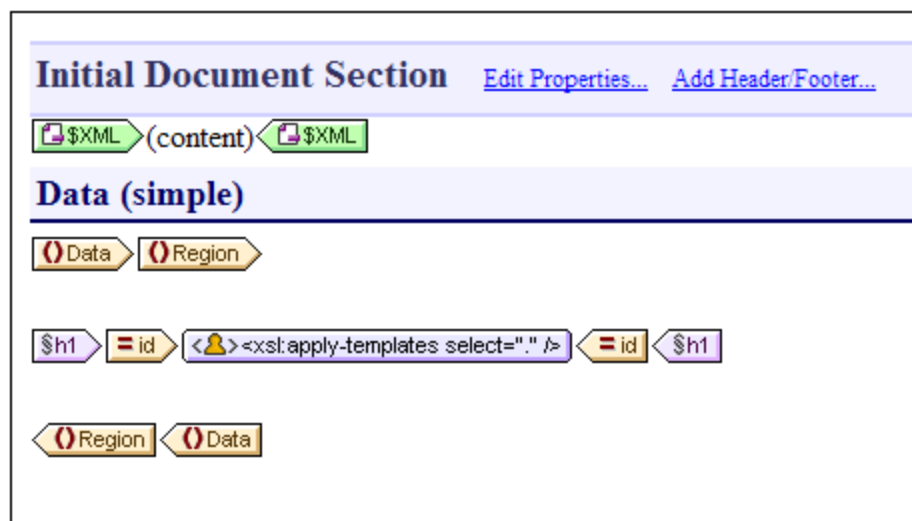
Le fichier XSLT est comme suit :

```

<?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">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Simple Example for New From XSLT</title>
      </head>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="Data">
    <xsl:for-each select="Region">
      <h1 style="color:red">
        <xsl:apply-templates select="@id"/>
      </h1>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

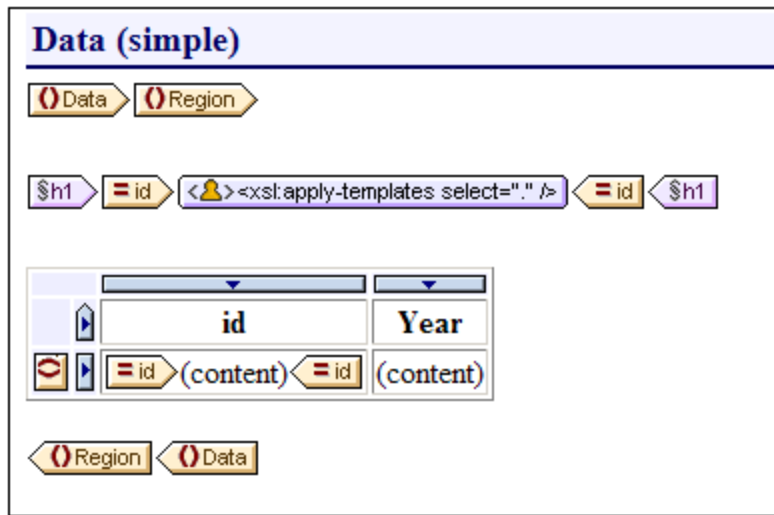
Suivez les étapes 1 à 4 recensées ci-dessus pour obtenir la SPS dans le Mode de design. La SPS aura l'aspect suivant :



Notez que les deux modèles dans XSLT ont été créés dans la SPS. Maintenant, basculez vers l'Aperçu HTML (capture d'écran ci-dessous), et notez que le style de l'élément `h1` (`color:red`) a été passé à la SPS.



Dans le Mode Design, sélectionnez l'élément `h1` et modifiez sa couleur en noir (dans la barre latérale des Styles, dans le groupe des propriétés *Color*). Puis, depuis l'arborescence de schéma, glissez l'élément `Year` et créez-le comme table à l'emplacement affiché dans la capture d'écran ci-dessous. Inversez les contenus des deux colonnes pour que l'ID `Year` soit dans la première colonne.



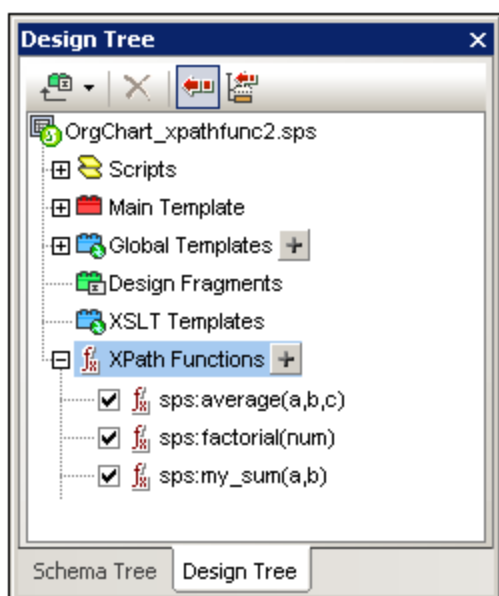
Vous pouvez effectuer des changements supplémentaires dans le contenu, la structure, et les propriétés de présentation du document, puis consultez la sortie et enregistrer les fichiers utilisant la commande [Enregistrer fichiers générés](#) ⁴⁶⁹.

9.3 Fonctions XPath définies par l'utilisateur

Le designer SPS peut définir les fonctions personnalisées **XPath 2.0/3.0**. Une fonction XPath définie par l'utilisateur peut être réutilisée dans tout composant de design qui accepte une expression XPath, par exemple, dans les calculs automatiques, conditions, et des listes déroulantes.

Définir et éditer les fonctions XPath définies par l'utilisateur

Les fonctions XPath définies par l'utilisateur sont créées (et par la suite accéder pour l'édition) dans chaque barre latérale de l'arborescence de schéma ou la barre latérale de l'arborescence de design (*voir la capture d'écran ci-dessous*). Toutes les fonctions XPath définies par l'utilisateur dans la SPS sont recensées sous l'élément des Fonctions XPath dans l'arborescence de schéma et dans les barres latérales de l'arborescence de schéma et peuvent être accédées par une de ces barres latérales.



Pour créer une fonction XPath définie par l'utilisateur, cliquez sur l'icône  de l'élément des Fonctions XPath. Ceci ouvre le dialogue des Fonctions XPath (*capture d'écran ci-dessous*). Si vous souhaitez éditer une fonction qui a déjà été créée, double-cliquez sur son entrée dans la liste des fonctions XPath. Le dialogue des Fonctions XPath (*capture d'écran ci-dessous*) apparaîtra et la définition de la fonction peut être éditée.

Fonctions XPath

Nom de la fonction:

Type de retour (option):

Paramètres

Nom	Type (en option)	Occurrence
-----	------------------	------------

Corps de fonction (Expression XPath)

```
$XML/Trade/Stock[@name=$XML/Trades/Selection/Stock]
```

XPath...

OK Annuler

Une fois que la fonction XPath définie par l'utilisateur est créée, elle est disponible pour utilisation partout dans le design.

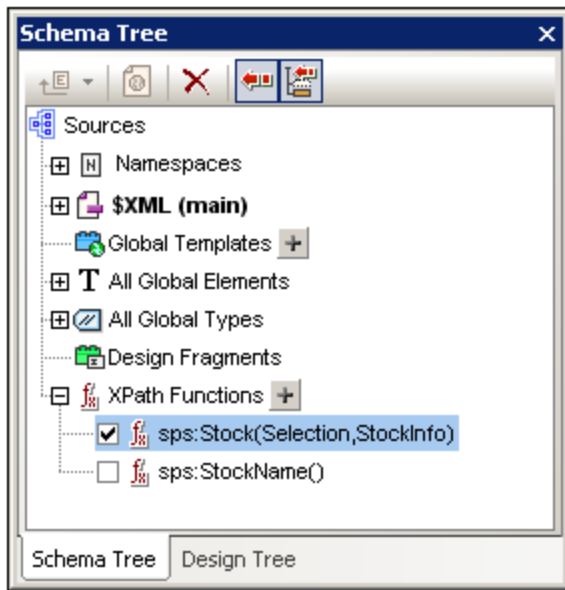
Espace de nom des fonctions XPath définies par l'utilisateur

Les fonctions XPath définies par l'utilisateur sont créées dans l'espace de nom :

<http://www.altova.com/StyleVision/user-xpath-functions>. Cet espace de nom est lié au préfixe `sps:`, donc les fonctions XPath définies par l'utilisateur doivent être appelées utilisant le préfixe d'espace de nom. Par exemple, `sps:MyFunction()`.

Activer et désactiver les fonctions XPath définies par l'utilisateur

Chaque fonction XPath définie par l'utilisateur peut être activée ou désactivée en cochant ou décochant la case à cocher, respectivement, à gauche de la saisie de la fonction dans la liste des fonctions XPath définies par l'utilisateur (voir la capture d'écran ci-dessous).



Cette fonction est utile si les deux fonctions ont le même nom. Une telle situation pourrait, par exemple, arriver quand un module SPS importé contient une fonction ayant le même nom.

Appeler une fonction XPath définie par l'utilisateur

Une fonction XPath définie par l'utilisateur peut être appelée dans une expression XPath à tout emplacement dans le design. Par exemple, la fonction XPath définie par l'utilisateur `sps:MyFunction` définie ci-dessus peut être appelée, par exemple, avec l'expression XPath suivante dans un calcul automatique :

```
sps:MyFunction() /@name.
```

Cette expression XPath serait évaluée comme suit :

1. La fonction `sps:MyFunction()` est évaluée. La fonction est définie comme suit :
`$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]`. Quand la fonction est évaluée, elle renvoie l'élément `/Trades/Stock` qui a un attribut `name` avec la valeur qui correspond à l'élément `/Trades/Selection/Stock`.
2. Le résultat de l'Étape 1 est renvoyée à l'expression XPath dans l'appel de fonction. Maintenant, la valeur de l'attribut `name` de cet élément `/Trades/Stock` est renvoyée comme valeur du calcul automatique.

Supprimer une fonction

Pour supprimer cette fonction, sélectionnez-la dans la liste des Fonctions XPath dans l'arborescence de schéma ou la barre latérale de l'arborescence de design, puis cliquez sur l'icône **Supprimer Item** dans la barre d'outils de la barre latérale. En alternative, cliquez de la touche droite de la souris sur la Fonctions XPath et sélectionnez **Supprimer Item** depuis le menu contextuel.

9.3.1 Définir une fonction XPath

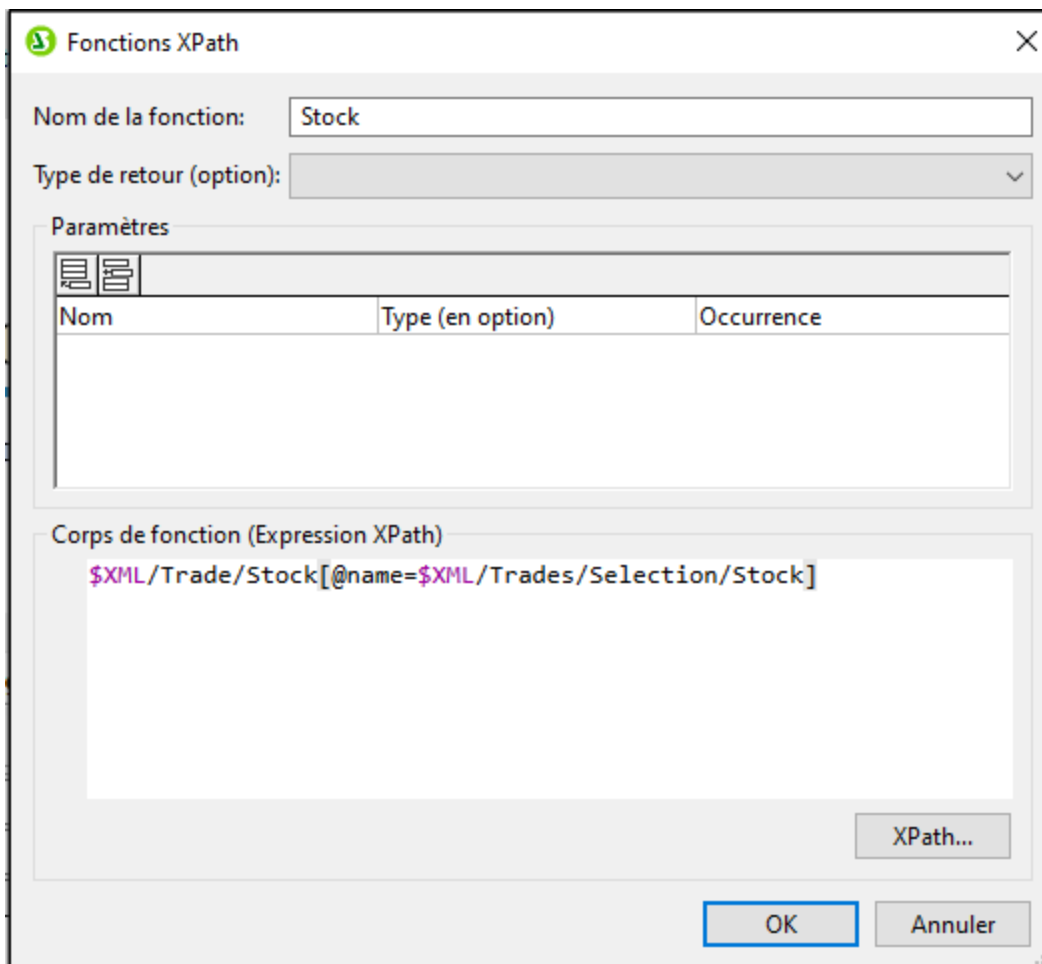
Une fonction XPath définie par l'utilisateur requiert : (i) un nom (un string de texte), et (ii) une définition (une expression XPath).

De plus, vous pouvez spécifier un ou plusieurs **paramètres** pour la fonction. Une fonction XPath définie par l'utilisateur peut aussi avoir un **return type** optionnel (spécifié en sélectionnant un type depuis la liste déroulante de la liste déroulante *Return Type*). Un type de retour est utile si vous souhaitez vérifier que ce type de données de la valeur retournée est conforme au type de données sélectionné. Notez que la valeur de retour n'est pas convertie en type de données sélectionné. S'il y a un décalage de type, une erreur est renvoyée. Si aucun type de retour n'est spécifié, aucune vérification de type de données n'est exécutée.

Une fois que la fonction XPath définie par l'utilisateur est créée, elle est disponible pour utilisation partout dans le design. Dans la feuille de style XSLT, elle est créée comme élément `xsl:function` qui est un enfant de l'élément `xsl:stylesheet`, tel qu'affiché dans la liste ci-dessous.

```
<xsl:stylesheet>
...
  <xsl:function name="sps:Stock">
    <xsl:sequence select="$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]"/>
  </xsl:function>
  <xsl:function name="sps:Average" as="xs:decimal">
    <xsl:param name="a" as="xs:integer"/>
    <xsl:param name="b" as="xs:integer"/>
    <xsl:param name="c" as="xs:integer"/>
    <xsl:sequence select="avg( ($a, $b, $c) )"/>
  </xsl:function>
</xsl:stylesheet>
```

La fonction `sps:Stock` affichée dans la capture d'écran ci-dessous et recensée ci-dessus renvoie l'élément `/Trades/Stock` qui a un attribut `name` avec la valeur qui correspond au contenu de l'élément `/Trades/Selection/Stock`. La fonction `sps:Average` recensée ci-dessus renvoie la moyenne des trois valeurs de paramètre d'entrée. La définition de fonction utilise la fonction `avg()` de XPath 2.0/3.0. Le type de données de renvoi est spécifié pour être le type `xs:decimal`, qui est le type de données renvoyé par la fonction `avg()` quand vous évaluez les valeurs d'entrée du type de données `xs:integer`. Si le type de renvoi est spécifié, alors le type de données de la valeur de renvoi est vérifié s'il est conforme au type spécifié. S'il ne l'est pas, alors une erreur est renvoyée.



Définir la fonction

Pour définir une fonction, cliquez sur l'icône  de l'élément des fonctions XPath dans l'arborescence de schéma ou l'arborescence de design. Ceci ouvre le dialogue des Fonctions XPath (*capture d'écran ci-dessous*). Si vous souhaitez éditer une fonction qui a déjà été créée, double-cliquez sur son entrée dans la liste des fonctions XPath. Puis, saisissez un nom pour la fonction et une définition dans le volet Corps de fonction. Les définitions de paramètre peuvent être saisies si requises (voir les deux prochaines sections, [Paramètres et Séquences](#) ³⁷⁷ et [Paramètres et Nœuds](#) ³⁸², pour les détails). Un type retour pour la fonction peut aussi être spécifié (*voir ci-dessus*).

Le point le plus important duquel il faut se souvenir quand vous écrivez l'expression XPath qui définit la fonction est qu'il n'existe **pas de nœud contextuel** pour l'expression XPath. Si l'expression XPath doit localiser un nœud, alors le nœud contextuel pour l'expression peut être fourni des manières suivantes :

1. L'expression XPath commence par la racine du document. La racine du document est spécifiée dans la première étape d'emplacement de l'expression XPath comme \$XML. Par exemple, l'expression XPath \$XML/Trades/Stock[1] situe le premier élément enfant Stock de l'élément /Trades. La variable \$XML (qui situe la racine du document du schéma principal) est définie globalement par StyleVision dans tous les designs SPS.
2. Le nœud contextuel peut être passé comme paramètre. Voir la section [Paramètres et Nœuds](#) ³⁸² ci-dessous pour une explication.

Dans les cas suivants, les erreurs sont renvoyées :

- Si un paramètre est défini mais n'est pas utilisé dans le corps de la définition.
- Si le type de données de la valeur renvoyée par la fonction ne correspond pas au type de renvoi défini pour la fonction.
- Si une fonction dans la SPS contient une erreur, une erreur XSLT est générée pour tout le design, même si la fonction contenant l'erreur n'est pas appelée. Notez, toutefois, qu'une fonction peut être désactivée en décochant sa case à cocher dans la liste des fonctions XPath définies par l'utilisateur. Lorsqu'elles sont désactivées dans le design, la fonction n'est pas incluse dans le document XSLT généré depuis les design. De cette manière, l'expression XPath contenant une erreur peut être exclue depuis XSLT et aucune erreur XSLT ne sera générée.

9.3.2 Réutiliser les fonctions pour localiser les nœuds

Dans la section précédente, nous avons vu comment une fonction XPath peut être créée pour localiser un nœud. La fonction `sps:Stock` qui est définie telle qu'affichée dans la capture d'écran ci-dessous retourne l'élément `/Trades/Stock` qui a un attribut `name` avec la valeur qui correspond au contenu de l'élément `/Trades/Selection/Stock`.

Fonctions XPath

Nom de la fonction:

Type de retour (option):

Paramètres

Nom	Type (en option)	Occurrence
-----	------------------	------------

Corps de fonction (Expression XPath)

```
$XML/Trade/Stock[@name=$XML/Trades/Selection/Stock]
```

XPath...

OK Annuler

Nous pourrions modulariser les étapes de l'emplacement de l'expression XPath

`$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]` dans des fonctions XPath séparées. Par exemple, comme suit :

- La fonction `sps:Stocks()`, avec la définition : `$XML/Trades/Stock`
- La fonction `sps:Stocks()`, avec la définition : `$XML/Trades/Selection/Stock`

Toute l'expression XPath peut ensuite être écrite dans une autre expression XPath comme suit :

```
sps:Stocks() [ @name=sps:SelectedStock() ]
```

Lorsque les fonctions XPath sont créées de cette manière pour localiser un nœud ou ensemble de nœud, ces fonctions peuvent être réutilisées dans d'autres expressions XPath dans tout le design SPS, simplifiant donc considérablement l'écriture d'expressions XPath complexes.

9.3.3 Paramètres dans les fonctions XPath

Une fonction XPath définie par l'utilisateur peut se voir attribuer un nombre quelconque de paramètres. Les paramètres de la fonction sont définis dans le volet *Paramètres* du dialogue Fonctions XPath (*voir capture d'écran ci-dessous*). Ces paramètres peuvent ensuite être utilisés dans la définition de la fonction XPath définie par l'utilisateur (dans le volet *Corps de fonction*).

Mécanisme de la fonction XPath définie par l'utilisateur

Les étapes ci-dessous expliquent le fonctionnement d'une fonction XPath.

1. Dans un appel de fonction (par exemple, dans un calcul automatique), le nombre d'arguments dans l'appel de fonction doit correspondre au nombre de paramètres définis pour la fonction définie par l'utilisateur (tels que définis dans le volet *Paramètres* de la fonction définie par l'utilisateur ; *voir capture d'écran ci-dessous*). De plus, le nombre d'éléments soumis par chaque argument (dans l'appel de fonction) doit correspondre à la définition *Occurrence* du paramètre correspondant. Si une restriction de type de données a été spécifiée pour un paramètre (dans la colonne *Type* du volet *Paramètres*), la ou les valeurs soumises par l'argument doivent correspondre à ce type de données.
2. Les arguments transmis aux paramètres de la fonction sont ensuite utilisés dans la fonction XPath (telle que définie dans le volet *Corps de fonction* ; *voir capture d'écran ci-dessous*). Le résultat obtenu en évaluant l'expression XPath est ensuite vérifié par rapport à la définition facultative du *Type de retour* (*voir capture d'écran ci-dessous*). Si le type de données est conforme aux attentes, le résultat est utilisé dans l'expression XPath à partir de laquelle la fonction a été appelée.

Ordre des paramètres

L'ordre des paramètres de la fonction définie par l'utilisateur est important car, lorsque la fonction est appelée, les arguments soumis dans l'appel de fonction seront attribués aux paramètres selon l'ordre dans lequel ils sont définis dans le volet *Paramètres* (*voir capture d'écran ci-dessous*).

Fonctions XPath

Nom de la fonction:

Type de retour (option):

Paramètres

Nom	Type (en option)	Occurrence
ContextStock	▼	Exactement un ▼
Selection	▼	Exactement un ▼
StockInfo	▼	Exactement un ▼

Corps de fonction (Expression XPath)

`$ContextStock[@name=$Selection]/$StockInfo`

XPath...

OK Annuler

Ainsi, si la fonction XPath définie par l'utilisateur `sps:Stock` est définie comme dans la capture d'écran ci-dessus et si elle est appelée avec l'expression XPath suivante :

```
sps:Stock($XML, Node1, Node2)
```

alors ces trois arguments — `$XML`, `Node1`, `Node2` — seront attribués, dans cet ordre, respectivement aux paramètres `$ContextStock`, `$Selection` et `$StockInfo`.

Notez que chaque argument de l'appel de fonction est séparé du suivant par une virgule. Ainsi, chaque argument, délimité par les virgules dans l'appel de fonction, sera transmis au paramètre correspondant (tel qu'ordonné dans le volet *Paramètres* ; voir capture d'écran ci-dessus).

L'ordre des paramètres dans le volet *Paramètres* peut être contrôlé à l'aide des icônes **Apposer**, **Insérer** et **Supprimer** du volet *Paramètres*.

Type de données des paramètres

Il est également possible de définir les types de données des paramètres de la fonction définie par l'utilisateur. Si un type de données est spécifié, le type de données de l'argument entrant sera vérifié par rapport au type de données du paramètre, et une erreur sera renvoyée si les types ne correspondent pas. Cette fonctionnalité permet de vérifier les données d'entrée (provenant des arguments de l'appel de fonction).

Occurrence

Chaque paramètre de la fonction XPath définie par l'utilisateur peut être considéré comme une séquence. La propriété *Occurrence* d'un paramètre spécifie le nombre d'éléments qui doivent être soumis pour ce paramètre par l'argument correspondant de l'appel de fonction.

Dans les définitions de fonction et les appels de fonction, les virgules sont utilisées pour séparer un paramètre ou un argument d'un autre, ainsi que pour séparer les éléments d'une séquence. Il est donc important de noter le contexte dans lequel une virgule est utilisée : pour séparer des paramètres/arguments ou pour séparer des éléments de séquence.

- Dans les paramètres/arguments, si nécessaire, des parenthèses sont utilisées pour délimiter les séquences — dans la définition de la fonction (paramètres) ou dans l'appel de fonction (arguments).
- Dans les séquences, les parenthèses sont ignorées.

Dans ce contexte, il convient de noter les exemples et points suivants :

- **Parenthèses dans les paramètres/arguments** : plusieurs fonctions XPath prennent une seule séquence comme argument, par exemple les fonctions `avg()` et `count()`. Si cette séquence est énumérée à l'aide de séparateurs de virgule ou d'opérateurs de plage, elle doit être placée entre parenthèses afin d'indiquer sans ambiguïté qu'il s'agit d'une séquence unique — et non de plusieurs séquences séparées par des virgules. Par exemple, dans la fonction `avg(count($a), $b, $c)`, la fonction XPath 2.0 `avg()` prend la séquence unique `(count($a), $b, $c)` comme argument. Étant donné que les éléments de la séquence sont énumérés, formant une séquence de trois éléments, la séquence doit être placée entre parenthèses et soumise comme argument unique à la fonction `avg()` : `avg((count($a), $b, $c))`. Sans la paire de parenthèses intérieures, la définition de la fonction `avg()` aurait trois paramètres, ce qui serait une erreur (car la fonction `avg()` attend un seul argument composé d'une seule séquence).
- **Pas de parenthèses dans les paramètres/arguments** : de même, la fonction `count()` prend également une seule séquence comme argument à un paramètre. Cependant, étant donné que dans notre exemple `count($a)`, la séquence unique n'est pas une liste énumérée séparée par des virgules, mais est récupérée par la variable/le paramètre `$a`, l'argument n'a pas besoin d'être entouré d'une paire de parenthèses internes : l'expression `count($a)` est donc correcte.
- **Parenthèses et virgules dans les appels de fonction** : dans un appel de fonction, les parenthèses doivent être utilisées correctement afin que chaque argument corresponde à un paramètre (tel que défini dans le volet *Paramètres* du dialogue Fonctions XPath). Par exemple, si une fonction XPath définie par l'utilisateur nommée `MyAverage()` est définie avec l'expression XPath 2.0 : `avg(count($a), $b, $c)`, alors l'appel de fonction suivant serait valide : `MyAverage((1, 2, 3), 4, 5)`. Les valeurs correspondant aux trois paramètres `$a`, `$b` et `$c` seraient respectivement la séquence `(1, 2, 3)`, la séquence singleton `4` et la séquence singleton `5`. Les séquences singleton peuvent, de manière facultative, être placées entre parenthèses. La valeur renvoyée par `MyAverage()` dans ce cas serait `4`.

9.3.3.1 Paramètres et Séquences

Il est important de noter la relation entre les paramètres et les séquences, et comment les paramètres et les séquences sont utilisées dans l'expression XPath. Nous utilisons les définitions suivantes pour rendre ces relations plus claires :

- Une **séquence** consiste en éléments qui sont soit des valeurs atomiques ou des nœuds. Une virgule peut être utilisée pour construire une séquence, en la plaçant entre les éléments d'une séquence et permettant ainsi à la séquence d'être construite.
- Une fonction XPath peut être définie pour prendre les **paramètres**. Par exemple, dans l'expression XPath 2.0 `count($a)`, la partie à l'intérieur des parenthèses de la fonction est le paramètre de la fonction et doit être une séquence d'éléments.
- Un **argument** consiste en un ou plusieurs items dans l'appel de la fonction. Par exemple, la fonction `count(//Person)` a un argument : `//Person`. Cet argument est valide car il renvoie une séquence de nœuds, qui correspond à la signature de la fonction `count()`. (La signature d'une fonction spécifie le nombre de paramètres et le type de données prévu pour chaque paramètre. Elle spécifie aussi ce que la fonction retournera et le type de données de l'objet retourné)
- La fonction `substring('StyleVisionExamples', 6, 6)`—qui retourne le string `Vision`—a trois arguments. Ceci est valide conformément à la signature de la fonction `substring()`, et est spécifié par celle-ci. Lorsqu'un appel de fonction a de multiples arguments, ceux-ci sont séparés par des virgules.

Parenthèses comme délimiteurs de séquence

Un point clé à noter lorsque des expressions XPath sont créées est ceci : *Les parenthèses sont utilisées pour délimiter les séquences qui utilisent le séparateur ou l'opérateur d'intervalle pour énumérer les séquences. En guise de résultat, chaque séquence délimitée par les parenthèses est lue comme un paramètre (dans les définitions de la fonction) ou comme un argument (dans les appels de fonction).*

Les parenthèses ne sont pas nécessaires autour d'une expression chemin (ou localisateur), (exemple d'une expression chemin : `//Person/@salary`), car une expression chemin peut être lue de manière non ambiguë comme un paramètre ou un argument. Il s'agit en fait d'un paramètre/argument à une séquence.

Vous trouverez ici quelques exemples pour illustrer les points réalisés ci-dessus :

- `avg(10, 20, 30)` La `avg` fonction de XPath 2.0 prend une séquence d'items comme seul argument. Puisque cette séquence est une énumération séparée par une virgule, la paire intérieure de parenthèses est nécessaire pour délimiter la seule séquence obligatoire. Sans les parenthèses intérieures, l'argument aurait trois arguments et serait invalide pour cette raison. (La paire extérieure de parenthèses est la parenthèse de la fonction.)
- `avg(//Person/@salary)` Cette expression chemin sélectionne les nœuds attribut `salary` de tous les éléments `Person` et retourne leurs valeurs attribut comme séquence à évaluer (c'est-à-dire, pour établir une moyenne). Aucune parenthèse n'est requise car la séquence n'est pas énumérée avant que l'argument ne soit lu. L'argument est la seule expression chemin (ou localisateur). L'expression chemin est évaluée et les valeurs retournées sont soumises à la fonction comme items d'une séquence.
- `count(10 to 34)` Ceci est une énumération via l'opérateur d'intervalle. L'opérateur d'intervalle `'to'` génère une séquence d'items séparés par une virgule (les entiers de 10 à 34) avant que l'argument ne soit lu. En guise de résultat, la fonction `count()` a une séquence séparée par une virgule de 25 items à l'intérieur de son argument. Pour lire ceci comme argument à une seule séquence, il est nécessaire de délimiter les parenthèses comme requis. Sans de telles parenthèses, l'appel de fonction aurait 25 arguments à la place d'une—invalidant ainsi l'appel de fonction, puisque la fonction `count()` doit, selon sa signature, avoir uniquement un argument.
- `Count(10(10 à 34, 37))` Les parenthèses intérieures indiquent que tout à l'intérieur d'elles est le seul argument de l'appel de fonction—une seule séquence consistant en 26 items.
- `count(//Person)` Aucune parenthèse de délimitation de séquence n'est requise autour d'un seul argument. L'argument est une expression chemin qui collecte les nœuds `//Person` dans le document XML et retourne ces nœuds comme les items de la séquence à compter.

Utiliser les paramètres XPath dans les fonctions XPath

Quand les paramètres sont utilisés dans la définition d'une fonction XPath définie par l'utilisateur, assurez-vous (i) qu'un nombre d'arguments dans l'appel de la fonction vers cette fonction XPath définie par l'utilisateur est correcte, et (ii) que les arguments évaluent correctement ce type et l'occurrence prévus.

Fonctions XPath

Nom de la fonction:

Type de retour (option):

Paramètres

Nom	Type (en option)	Occurrence
a	xs:integer	Exactement un
b	xs:integer	Exactement un
c	xs:integer	Exactement un

Corps de fonction (Expression XPath)

```
avg( ($a, $b, $c) )
```

La capture d'écran ci-dessus définit trois paramètres (dans le volet *Parameters*), puis utilise ces paramètres (dans le volet *Corps de fonction*) pour définir une fonction XPath.

Chaque paramètre qui est défini dans les *Parameters* peut être considérée être une seule séquence. Le nombre d'items autorisés à l'intérieur d'une seule séquence est spécifiée avec la propriété *Occurrence*. Dans la définition ci-dessus, chaque paramètre est défini (dans sa propriété *Occurrence*) comme une séquence simple (à savoir, une séquence d'exactly un élément). Chaque argument de l'appel de fonction doit donc en être une séquence d'un seul item. Cette propriété *Type* spécifie le type de données des items de la séquence.

Dans la définition de notre fonction XPath (dans le volet *Function Body*), chaque paramètre fournit un item de la séquence pour définir une moyenne. Puisque les paramètres XPath constituent ensemble une séquence, la séquence doit être entre parenthèses pour assurer que toute la séquence est lue comme le paramètre de la fonction `avg()`. Si, lors de l'exécution, un des arguments dans l'appel de fonction (correspondant aux trois paramètres) n'est pas une séquence singleton (simple), une erreur est retournée.

Ci-dessous, vous trouverez des exemples de l'utilisation du paramètre XPath dans les appels vers la fonction XPath `ThreeAverage()` affichée dans la capture d'écran ci-dessus. Dans le Mode Design, vous pouvez insérer un calcul automatique et lui donner les expressions XPath recensées ci-dessous pour voir les résultats. La fonction a été définie pour prendre une séquence de trois entiers et en définir une moyenne..

- `sps:ThreeAverage(10,20,30)` retourne 20. Il existe trois arguments valides dans l'appel de la fonction, correspondant aux trois paramètres XPath.
- `sps:ThreeAverage( (10), (20), (30) )` retourne 20. Il existe trois arguments d'entrée valides, correspondant aux trois paramètres XPath. Chaque argument d'entrée a été enveloppé de parenthèses (qui sont redondants, puisque chaque séquence est une séquence singleton ; toutefois, cette redondance n'est pas une erreur).
- `sps:ThreeAverage( (10),20,30 )` retourne 20. Il existe trois arguments d'entrée valides, correspondant aux trois paramètres XPath. Le premier argument a été enveloppé de parenthèses (redondant, mais pas une erreur).
- `sps:ThreeAverage( (10,20), (30), (40) )` retourne une erreur car le premier argument n'est pas valide. Il ne s'agit pas d'une séquence singleton, tel que requis par la définition du premier paramètre `$a ('Exactly one')`.
- `sps:ThreeAverage( (10,20,30) )` retourne une erreur car seul un argument d'entrée est soumis, à l'intérieur des parenthèses. Par ailleurs, l'argument est invalide car la séquence n'est pas une séquence singleton.

Si la propriété *Occurrence* d'un paramètre est définie comme *At-least-one* (comme dans la définition affichée dans la capture d'écran ci-dessous), alors ce paramètre est défini comme séquence d'un-ou-plus d'items.

Fonctions XPath

Nom de la fonction:

Type de retour (option):

Paramètres

Nom	Type (en option)	Occurrence
a	xs:integer	Au moins un
b	xs:integer	Exactement un
c	xs:integer	Exactement un

Corps de fonction (Expression XPath)

```
avg( (count($a), $b, $c) )
```

XPath...

OK Annuler

Dans la définition ci-dessus, le premier paramètre a été défini comme séquence d'un ou plusieurs items, les deux prochains paramètres comme séquences singleton. La fonction a été définie pour compter le nombre d'items soumis par le premier paramètre, ajouter le résultat de la somme des deux entiers soumis par les deux autres paramètres, puis diviser le résultat par trois pour obtenir la moyenne. Veuillez noter les points suivants :

- La séquence qui est le paramètre de la fonction `avg()` est enveloppée entre parenthèses. Ceci est fait pour spécifier que la fonction `avg()` prend une seule séquence consistant en trois items comme son paramètre. La seule séquence consiste en trois entiers : le premier soumis par la fonction `count()` ; le deuxième et le troisième sont les deux paramètres `b` et `c`.
- L'argument de la fonction `count()` n'est pas entre parenthèses de la délimitation de séquence car l'argument est une seule séquence explicite.

Ci-dessous, vous trouverez des exemples de l'utilisation du paramètre dans les appels vers la fonction XPath `Average()` affichée dans la capture d'écran ci-dessus.

- `sps:Average((1,2),3,4)` returns 3. Il existe trois arguments d'entrée valides, correspondant aux trois paramètres. Le premier argument est entre parenthèses pour le délimiter. Lorsque la fonction `count()` opère sur celui-ci, la fonction retournera la valeur 2, qui sera le premier item de la séquence soumise à la fonction `avg()`.
- `sps:Average( 4,4,4 )` retourne 3. Il existe trois arguments d'entrée valides. Le premier argument est autorisé d'être une séquence d'un item (voir la propriété *Occurrence* de son paramètre correspondant). Aucune parenthèse n'est requise pour indiquer les arguments séparés.

Points d'intérêt supplémentaires

Les points supplémentaires suivants devraient être notés :

- Si un paramètre est défini comme ayant une occurrence *At-least-one*, alors une fonction telle que `MyAverage()` pourrait être définie avec une expression XPath telle que `avg(($a))`. Cette fonction accepterait un argument qui est une seule séquence d'un-ou-plus d'items. La fonction peut être appelée comme suit : `sps:MyAverage((2,3,4))`, et elle renverra la valeur 3. L'argument d'entrée doit être entre parenthèses pour assurer que l'entrée est lue comme une séquence unique plutôt que de trois séquences singleton (qui serait le cas s'il n'y avait pas de parenthèses enveloppantes).
- Si un paramètre XPath `$a` est défini ayant une occurrence *None-or-one*, alors une fonction telle que `MyAverage()` pourrait être définie avec une expression XPath telle que `avg(($a, $b, $c))`. Cette fonction accepterait trois séquences comme son argument, avec la possibilité de la première séquence étant vide. Si la première séquence est vide, alors une séquence vide doit soumettre explicitement comme premier argument d'entrée. Autrement une erreur est reportée. Si la fonction était appelée comme suit : `sps:MyAverage(30,20,10)`, elle renverrait la valeur 20. La fonction pourrait aussi être appelée avec : `sps:MyAverage((),20,10)`, retournant 15 (notez que la séquence vide compte : comme valeur d'entrée de vide ; pour une valeur retour de 10, le premier item devrait être 0). Toutefois, ce qui suit générerait une erreur : `sps:MyAverage(20,10)`, car aucune séquence vide n'est fournie et, en conséquence, le troisième argument d'entrée est considéré être absent.

Exemples complexes

Outre le fait d'avoir l'avantage d'être capable de réutiliser une expression XPath, les fonctions XPath définies par l'utilisateur peuvent aussi activer la construction de fonctions XPath complexes personnalisées qui ne sont pas disponibles dans l'ensemble de la fonction XPath 2.0. Par exemple, une fonction factorielle pourrait facilement être créée avec une expression XPath qui prend une séquence singleton comme son seul paramètre. Si le paramètre `$num` est le nombre à factoriser, alors l'expression XPath à créer la fonction serait :

```
if ($num < 2) then 1 else $num * sps:Factorial($num - 1)
```

Si la fonction était appelée `Factorial()`, alors le factoriel, disons 6, pourrait être obtenu en appelant la fonction avec : `sps:Factorial(6)`.

9.3.3.2 Paramètres et Nœuds

Lorsque vous utilisez des paramètres dans les fonctions XPath qui localisent les nœuds, il est important de se souvenir que la fonction n'a pas de nœud contextuel, peu importe d'où dans le design elle est appelée. Le nœud contextuel peut être fourni soit dans l'expression XPath utilisée pour définir la fonction (qui est, dans le volet *Corps de fonction*) ou dans l'expression XPath qui est utilisée pour appeler la fonction XPath. Dans le dernier cas, le contexte peut être fourni via les arguments dans l'appel de fonction.

Prenez en compte la fonction XPath définie par l'utilisateur `Stock()`, qui est définie avec les trois paramètres tels qu'affichés dans la capture d'écran ci-dessous.

Fonctions XPath

Nom de la fonction:

Type de retour (option):

Paramètres

Nom	Type (en option)	Occurrence
ContextStock	<input type="text"/>	Exactement un
Selection	<input type="text"/>	Exactement un
StockInfo	<input type="text"/>	Exactement un

Corps de fonction (Expression XPath)

`$ContextStock[@name=$Selection]/$StockInfo`

XPath...

OK Annuler

La définition du corps de fonction est `$ContextStock[@name=$Selection]/$StockInfo`, qui utilise les trois paramètres mais ne contiennent pas d'information de nœud contextuel. L'information du nœud contextuel peut être fournie dans l'expression XPath qui appelle la fonction, par exemple de cette manière :

```
sps:Stock( $XML/Trades/Stock, $XML/Trades/Selection/Stock, @name )
```

L'appel de fonction a trois arguments, dont chaque valeur fournit soit l'information contextuelle ou du 'node-locator'. En alternative, les expressions XPath peuvent être utilisées comme appel de fonction et donner les mêmes résultats :

```
sps:Stock( /Trades/Stock, /Trades/Selection/Stock, @name )
sps:Stock( /Trades/Stock, //Selection/Stock, @name )
```

La variable `$XML`, qui renvoie la racine du document, peut être laissée de côté dans les appels de fonction depuis les composants de design car dans les expressions des composants du design les nœuds contextuels sont connus.

Notez que dans l'appel de fonction recensé ci-dessus, il existe trois arguments d'entrée correspondant respectivement aux trois paramètres définis pour la fonction XPath définie par l'utilisateur:

- `$ContextStock` = `$XML/Trades/Stock` (l'élément `/Trades/Stock`)
- `$Selection` = `$XML/Trades/Selection/Stock` (l'élément `/Trades/Selection/Stock`)

- `$StockInfo = @name`

L'expression XPath dans la définition de fonction est :

```
$ContextStock[@name=$Selection]/$StockInfo
```

Lorsque les arguments d'entrée sont substitués, l'expression XPath dans la définition de fonction devient :

```
$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]/@name
```

Il est important de noter que ce sont les ensembles de nœud qui sont passés à la fonction, pas les strings de texte.

C'est de cette manière que le nœud contextuel et les étapes de l'emplacement sont passés à la fonction via les paramètres. La fonction peut ensuite être évaluée pour localiser et retourner les nœuds requis.

9.4 Travailler avec des dates

Si le document source contient des nœuds, qui prennent des valeurs de date, utilisant les types de données `xs:date` ou `xs:dateTime` dans le schéma XML sous-jacent rend disponible la date et l'heure des fonctions puissantes de manipulation de XPath 2.0/3.0 (voir [exemples ci-dessous](#)³⁸⁵). StyleVision prend en charge les types de données `xs:date` ou `xs:dateTime` en fournissant une gamme de possibilités de [formatage de date](#)³⁸⁵ via la fonction de [formatage d'entrée](#)³³⁴.

Note : les données de la date et de l'heure ne peuvent pas être manipulées avec XPath 1.0. Toutefois, avec XPath 1.0, vous pouvez utiliser « Input Formatting » pour fournir le [formatage de date](#)³⁸⁵.

Les calculs de la date avec XPath 2.0

Les données impliquant les données peuvent être manipulées avec les expressions XPath 2.0 dans les [calculs automatiques](#)²⁶¹. Ci-dessous, vous trouverez quelques exemples de ce qui peut être atteint avec les expressions XPath 2.0.

- Les fonctions de XPath 2.0 `current-date()` et `current-dateTime()` peuvent être utilisées pour obtenir la date et date-heure actuelles, respectivement.
- Les dates peuvent être soustraites. Par exemple : `current-date() - DueDate` retournerait une valeur `xdt:dayTimeDuration` ; par exemple, quelque chose comme `P24D`, qui indique une différence positive de 24 jours.
- Les unités de temps peuvent être extraites depuis des durées qui utilisent les fonctions de XPath 2.0. Par exemple : `days-from-duration(xdt:dayTimeDuration('P24D'))` retournerait l'entier 24.

Ici, se trouve une expression XPath 2.0 dans le calcul automatique. Il calcule un intérêt annuel de 4% sur un montant exigible sur une base par-jour et retourne la somme du montant principal et de l'intérêt accumulé :

```
if      (current-date() gt DueDate)
then  (round-half-to-even(InvoiceAmount +
    (InvoiceAmount*0.04 div 360 *
        days-from-duration((current-date() - DueDate))), 2))
else  InvoiceAmount
```

Un tel calcul serait possible avec XPath 2.0 uniquement si l'élément `DueDate` était défini pour être un type de données tel que `xs:date` et le contenu de l'élément est saisi sous sa forme lexicale correcte, à savoir `YYYY-MM-DD[±HH:MM]`, où le composant du fuseau horaire (préfixé par `±`) est optionnel.

9.4.1 Formater les dates

Une date dans un document XML est enregistrée sous le format spécifique au type de données de son nœud. Par exemple, la valeur d'un nœud `xs:date` aura le format `YYYY-MM-DD[±HH:MM]`, tandis que la valeur d'un nœud `xs:dateTime` aura le format `YYYY-MM-DDTHH:MM:SS[±HH:MM]`. Ces formats sont des représentations lexicales de ces données. Par défaut, c'est la représentation lexicale de ces données qui est affichée dans Authentic View et dans la sortie. Toutefois, dans SPS, la fonction de formatage de la valeur peut être utilisée

pour afficher des données sous des formats alternatifs dans Authentic View et, dans certains cas, en option dans la sortie.

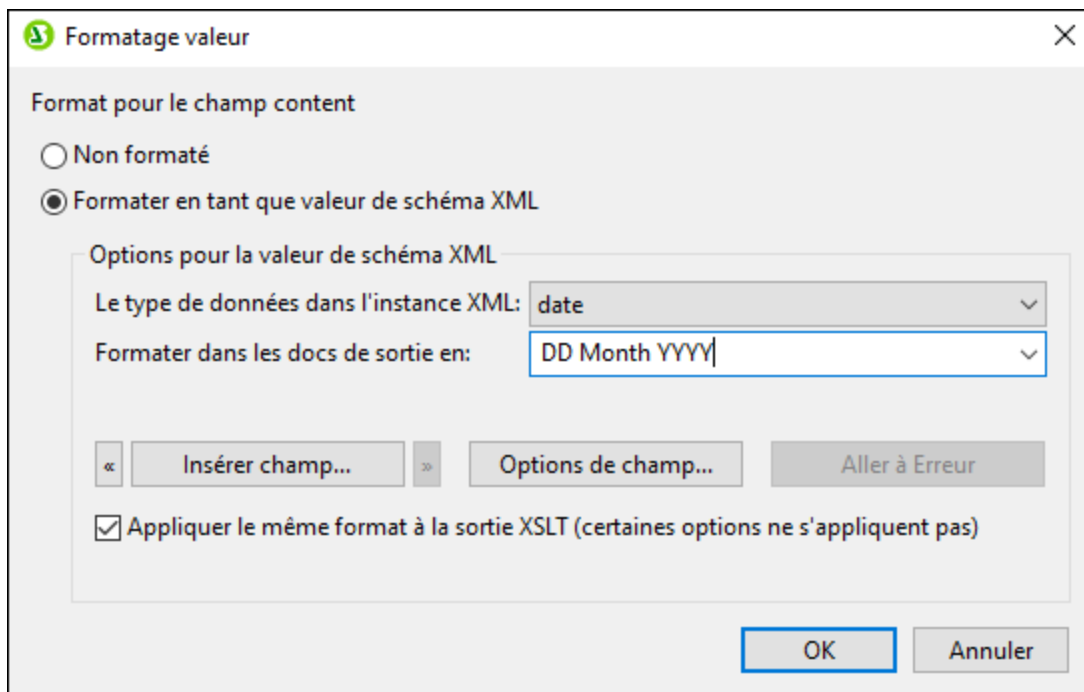
Le formatage de valeur pour les dates peut être utilisé pour définir les formats personnalisés pour les nœuds et les calculs automatiques des types de données suivants :

- xs:date
- xs:dateTime
- xs:duration
- xs:gYear
- xs:gYearMonth
- xs:gMonth
- xs:gMonthDay
- xs:gDay

Utiliser le formatage de la valeur pour formater les nœuds de date

Pour formater les dates en alternative au format lexical du nœud de date, faites comme suit :

1. Sélectionnez l'espace réservé `contents` ou champ d'entrée du nœud. Notez que le formatage de la valeur ne peut être appliqué aux nœuds créés **comme contenu ou champ d'entrée**.
2. Dans la barre latérale des Propriétés, sélectionnez l'item `autocalc`, puis le groupe des propriétés *AutoCalc*. Cliquez ensuite sur le bouton Éditer [...] de la propriété *Formatage de la valeur*. Ceci affiche le dialogue *Formatage de valeur* (capture d'écran ci-dessous).



Par défaut, le bouton radio *Non formaté* (le format lexical standard pour le type de données du nœud) est sélectionné.

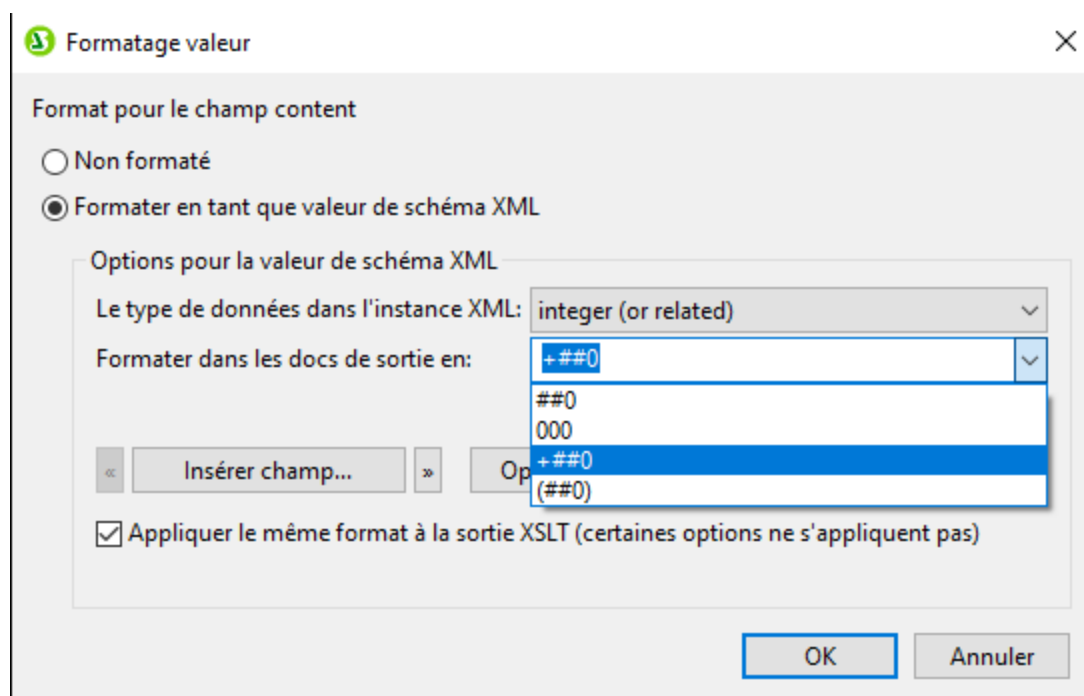
3. Pour définir un format alternatif, sélectionnez le bouton radio *Format*.
4. Vous pouvez sélectionner maintenant un format de date prédéfini depuis la liste déroulante de la zone de liste déroulante (capture d'écran ci-dessous), ou définir votre propre format dans le champ d'entrée

de la zone de liste déroulante. Voir [Syntaxe de formatage de valeur](#)³³⁷ pour les détails sur la syntaxe à utiliser quand vous définissez votre propre format.

Utiliser le formatage de la valeur pour formater les Calculs automatiques

Lorsque les calculs automatiques évaluent une valeur qui est un format de date, le Formatage de valeur peut être utilisé pour afficher le résultat. Procédez comme suit :

1. Sélectionnez le Calcul automatique dans le design.
2. Dans la barre latérale des Propriétés, sélectionnez l'item `content`, puis le groupe des propriétés *AutoCalc*. Cliquez ensuite sur le bouton Éditer [...] de la propriété *Formatage de la valeur*. Ceci fait apparaître le dialogue Formatage valeur (capture d'écran ci-dessous).



Par défaut, le bouton radio *Non formaté* est sélectionné.

3. Pour définir un format alternatif, sélectionnez le bouton radio *Format*.
4. Dans le volet Options pour valeur de schéma XML, dans la zone de liste déroulante *Type de données*, sélectionnez le type de données `date` auquel le calcul automatique sera évalué. Dans le *Format*, vous pouvez sélectionner maintenant un format de date prédéfini depuis la liste déroulante de la zone de liste déroulante, ou définir votre propre format dans le champ d'entrée de la zone de liste déroulante. Voir [Syntaxe de formatage de valeur](#)³³⁷ pour les détails sur la syntaxe à utiliser quand vous définissez votre propre format.

Appliquer le Formatage valeur à la sortie

Le Formatage valeur que vous définissez s'applique à Authentic View. De plus, certaines définitions du Formatage valeur—pas toutes—peuvent aussi être appliquées aux sorties HTML. À cette fin, vérifiez la case à cocher Appliquer le même format à la sortie XSLT. Si cette option n'est pas vérifiée ou si elle n'est pas disponible, alors seul Authentic View affichera le Formatage valeur ; la sortie affichera la valeur dans son format lexical (pour les nœuds) ou, dans le cas des calculs automatiques, dans le format vers lequel le calcul automatique évalue.

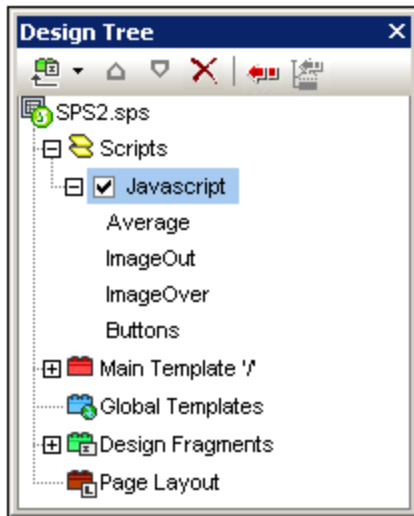
9.5 Utiliser des Scripts

Dans StyleVision, vous pouvez définir les fonctions JavaScript pour chaque SPS dans l'éditeur de JavaScript (disponible comme onglet dans le Mode Design). Les définitions de fonction créées de cette manière sont stockées dans l'en-tête du document HTML et peuvent être appelées de l'intérieur du corps du document HTML. De telles fonctions sont utiles quand :

- Vous souhaitez atteindre un résultat complexe d'instructions de script. Dans ce cas, il est utile d'écrire tous les scripts requis, comme fonctions séparées, dans un emplacement (l'en-tête) et faire référence ensuite aux fonctions dans le document de design.
- Vous souhaitez utiliser un script particulier aux emplacements multiples dans le document de design.

Comment définir les fonctions dans l'Éditeur JavaScript est décrit dans la sous-section [Définir les fonctions JavaScript](#) ³⁹⁰.

Dans la GUI, toutes les fonctions JavaScript qui sont définies pour un SPS donné dans l'Éditeur JavaScript sont recensées dans la fenêtre d'arborescence de design sous l'entrée Scripts (*capture d'écran ci-dessous*). La capture d'écran ci-dessous indique que quatre fonctions JavaScript, *Average*, *ImageOut*, *ImageOver* et *Buttons*, sont actuellement définies dans le SPS actif.



Les fonctions définies dans l'Éditeur JavaScript sont disponibles comme appels de gestionnaire d'événement à l'intérieur de la GUI. Lorsqu'un composant dans le document de design est sélectionné, toute fonction définie peut être attribuée à une propriété de gestionnaire d'événement dans le groupe de la barre latérale des Propriétés. Pour voir comment attribuer une fonction JavaScript à un gestionnaire d'événement est décrit dans la section [Attribuer Fonction aux gestionnaires d'événement](#) ³⁹¹.

Scripts dans les SPS modulaires

Lorsqu'un [module SPS est ajouté à un autre module SPS](#) ²²⁰, les scripts dans le module ajouté sont disponibles à l'intérieur du SPS, et peuvent être utilisés comme gestionnaires d'événement via la barre latérale des Propriétés dans le SPS de référence. Pour plus d'information sur l'utilisation des SPS modulaires, voir la section [SPS modulaires](#) ²²⁰.

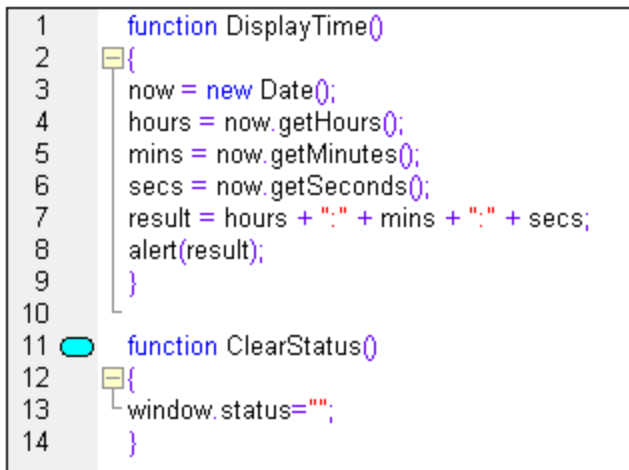
9.5.1 Définir fonctions JavaScript

Pour définir les fonctions JavaScript, suivez les étapes suivantes :

1. Dans le Mode Design, basculez vers l'Éditeur JavaScript Editor en cliquant sur l'onglet Mode Design et sélectionnez JavaScript (*capture d'écran ci-dessous*).



2. Dans l'Éditeur JavaScript, tapez dans les définitions de fonction (*voir capture d'écran ci-dessous*).



La capture d'écran ci-dessus affiche les définitions de deux fonctions JavaScript : `DisplayTime` et `ClearStatus`. Elles ont été décrites pour le SPS actif. Elles seront saisies dans l'en-tête du fichier HTML comme suit :

```
<script language="javascript">

<!-- function DisplayTime()
{
    now = new Date();
    hours = now.getHours();
    mins = now.getMinutes();
    secs = now.getSeconds();
    result = hours + ":" + mins + ":" + secs;
    alert(result)
}

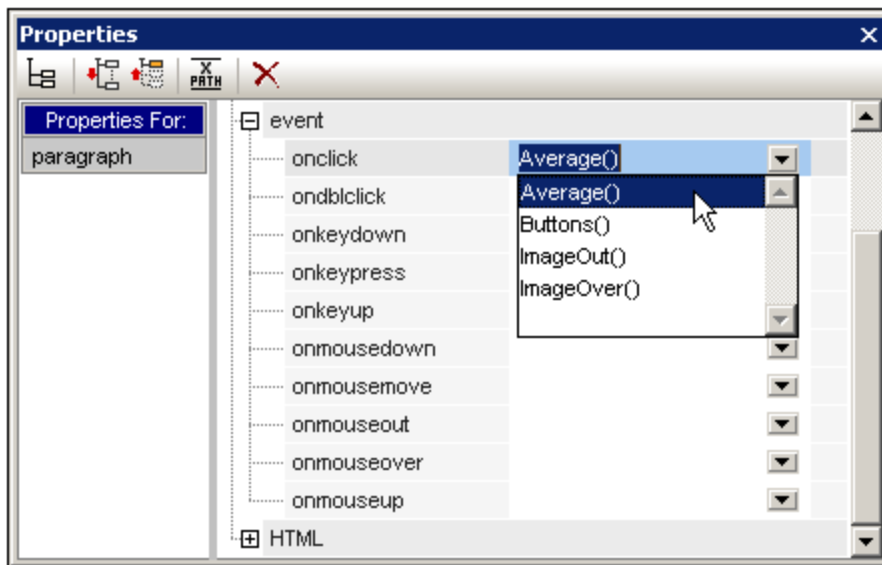
function ClearStatus()
{
    window.status="";
}
-->

</script>
```

Ces fonctions peuvent désormais être appelées de partout dans le document HTML. Dans StyleVision, toutes les fonctions définies sont disponibles comme options qui peuvent être attribuées à une propriété de gestionnaire d'événement dans le groupe de propriété *Event* dans la barre latérale des Propriétés. Voir [Attribuer Fonction aux gestionnaires d'événement](#)³⁹¹ pour les détails.

9.5.2 Attribuer Fonctions comme gestionnaires d'événement

Dans la GUI StyleVision, vous pouvez attribuer des fonctions JavaScript comme gestionnaires d'événement pour des événements qui apparaissent sur rendus HTML de composants SPS. Ces gestionnaires d'événement seront utilisés dans la sortie HTML. Le gestionnaire d'événement pour une événement disponible - tel que `onclick`—est défini en attribuant une fonction globale comme gestionnaire d'événement. Dans la barre latérale des Propriétés, les fonctions globales définies dans l'Éditeur JavaScript sont disponibles comme gestionnaires d'événement dans les zones déroulantes de chaque événement dans le groupe des propriétés *Events* pour chaque composant sélectionné (*capture d'écran ci-dessous*).



Pour attribuer une fonction à gestionnaire d'événement, suivez les étapes suivantes :

1. Sélectionnez le composant dans SPS pour lequel le gestionnaire de l'événement doit être défini. Le composant peut être un nœud ou contenu de tout genre, dynamique ou statique.
2. Dans la barre latérale des Propriétés, sélectionnez le groupe *Event*. Ceci résulte en les événements disponibles étant affichés dans la colonne Attribut (*capture d'écran ci-dessus*).
3. Dans la colonne Valeur de l'événement requis, cliquez la flèche qui pointe vers le bas de la zone de liste déroulante. Ceci fait défiler une liste de toutes les fonctions dans l'Éditeur JavaScript.
4. Depuis la liste déroulante, sélectionnez la fonction requise comme gestionnaire d'événement pour cet événement.

Dans la sortie HTML, lorsque cet événement est déclenché dans la composant pour lequel le gestionnaire d'événement est défini, la fonction JavaScript est exécutée.

9.5.3 Fichiers JavaScript externes

Un SPS peut accéder à des fichiers JavaScript externes de deux manières :

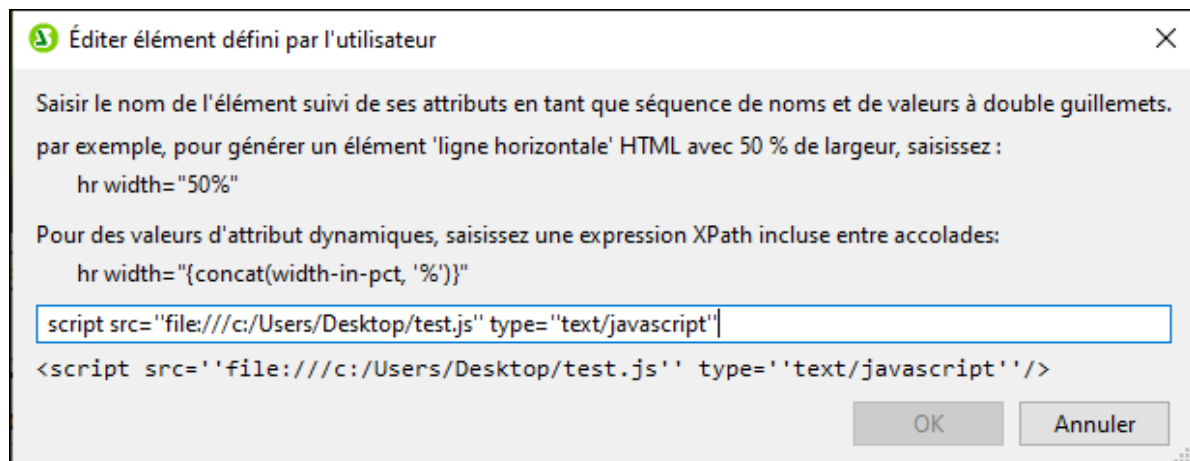
1. [En créant un élément défini par l'utilisateur ou Bloc XML défini par l'utilisateur](#)³⁹². Ces objets design peuvent contenir un `SCRIPT` élément qui a accès au fichier JavaScript externe. Notez que l'emplacement de l'Élément défini par l'utilisateur ou Bloc XML défini par l'utilisateur se trouve à l'intérieur de l'élément `BODY` du design (et pour cette raison, à l'intérieur de l'élément `BODY` de la sortie HTML, et non à l'intérieur de l'élément `HEAD`).
2. [En ajoutant un script dans l'Éditeur Javascript](#)³⁹³ qui a accès au fichier externe. Un script qui est ajouté de cette manière sera situé dans l'élément `HEAD` de la sortie HTML.

Éléments définis par l'utilisateur et Blocs XML définis par l'utilisateur

Les fichiers JavaScript externes peuvent être accédés par les [Éléments définis par l'utilisateur](#)¹²² et les [Blocs XML définis par l'utilisateur](#)¹²³. En utilisant les mécanismes, un élément `SCRIPT` qui accède au fichier JavaScript externe peut être inséré dans tout emplacement à l'intérieur de l'élément `BODY` du document HTML de sortie.

Un [Élément défini par l'utilisateur](#)¹²² pourrait être inséré comme suit :

1. Placez le curseur à l'emplacement dans le design où l'élément `SCRIPT` qui a accès au fichier JavaScript doit être inséré.
2. Depuis le menu **Insert** ou le menu contextuel, sélectionnez la commande pour insérer un [Élément défini par l'utilisateur](#)¹²².



3. Dans le dialogue qui apparaît (*voir la capture d'écran ci-dessus*), saisissez l'élément `SCRIPT` comme affiché ci-dessus, donnant l'URL du fichier JavaScript comme la valeur de l'attribut `src` de l'élément `SCRIPT` : par exemple, `script type="text/javascript" src="file:///c:/Users/mam/Desktop/test.js"`
4. Cliquez sur **OK** pour terminer.

Vous pouvez aussi utiliser un [Bloc XML définir par l'utilisateur](#)¹²³ pour atteindre le même résultat. À cette fin, utilisez la même procédure telle que décrite ci-dessus pour les Éléments définis par l'utilisateur, la seule différence étant (i) qu'un [Bloc XML défini par l'utilisateur](#)¹²³ est inséré à la place de l'[Élément défini par](#)

[l'utilisateur](#)¹²², et (ii) que l'élément `SCRIPT` est inséré comme bloc XML complet, avec des balises de début et de fin.

Éditeur JavaScript

L'[Éditeur JavaScript](#)³⁹⁰ vous permet d'insérer un script externe dans l'élément `HEAD` de la sortie HTML. Faites ceci en saisissant, dans l'Éditeur JavaScript, le fragment de script suivant, à l'extérieur de toute autre définition de fonction que vous créez.

```
var script = document.createElement('script');
script.type = 'text/javascript';
    script.src = 'file:///c:/Users/Desktop/test.js';
var head = document.getElementsByTagName('head')[0];
head.appendChild(script)
```

Le fichier JavaScript externe situé par l'URL dans `script.src` est accédé depuis l'intérieur de l'élément `HEAD` du document de sortie HTML.

9.6 Importation HTML

Dans StyleVision, vous pouvez importer un fichier HTML et créer les documents suivants basés sur celui-ci :

- Un document SPS basé sur le design et la structure du fichier HTML importé.
- Un schéma XML, dans lequel les composants de document HTML sont créés comme éléments ou attributs de schéma. En option, des éléments et attributs supplémentaires qui ne sont pas liés au document HTML peuvent être créés dans le schéma défini par l'utilisateur.
- Un document XML avec : (i) une structure basée sur le schéma XML que vous avez créé et (ii) le contenu du fichier HTML.
- Les feuilles de style XSLT basées sur le design dans le Mode Design.

HTML-à-XML : étape -par-étape

Le mécanisme d'importation HTML, qui permet la création de fichiers basés sur le fichier HTML importé, consiste en les étapes suivantes :

1. [Créer un nouvel SPS via l'Importation HTML](#) ³⁹⁴. Lorsqu'un fichier HTML est importé dans StyleVision, un nouveau document SPS est créé. Le document HTML est affiché dans le Mode Design avec les balises markup HTML. Un schéma XML défini par l'utilisateur avec un élément de document appelé `UserRoot` est créé dans la fenêtre de l'arborescence de schéma. Il s'agit du schéma sur lequel SPS est basé. Le contenu du document et markup HTML affiché dans le Mode Design à ce stade est inclus dans le SPS comme contenu statique.
2. [Créer le schéma et le Design SPS](#) ³⁹⁶. Créer le schéma en (i) glissant les composants depuis le document HTML à l'emplacement requis dans l'arborescence de schéma (dans la fenêtre de l'arborescence de schéma) ; et, en option, (ii) en ajoutant vos propres nœuds à l'arborescence de schéma. Dans la fenêtre Design, le contenu HTML qui a été utilisé pour créer des nœuds dans l'arborescence de schéma sera affiché maintenant avec les balises de nœud du schéma autour du contenu. Le contenu HTML qui n'a pas de nœud de schéma correspondant continuera à être affiché sans balises de nœud de schéma.
3. Dans le document de Design, attribuer le formatage aux nœuds, affiner les règles de traitement, ou ajouter du contenu statique tel que requis. Ces modifications auront un effet uniquement sur SPS et l'XSLT généré. Il n'aura pas d'effet sur le schéma généré ou le fichier XML.
4. Une fois que vous avez complété l'arborescence de schéma et le design du SPS, vous pouvez [générer et enregistrer](#) ⁴⁰⁰ les points suivants :
 - Un schéma XML correspondant à l'arborescence de schéma que vous avez créé ;
 - Un fichier de données XML avec une structure basée sur le schéma et le contenu pour les nœuds de schéma qui sont créés avec l'espace réservé (`content`) dans le design SPS ;
 - un SPS (fichier `.sps`) et/ou une feuille de style XSLT basée sur votre design.

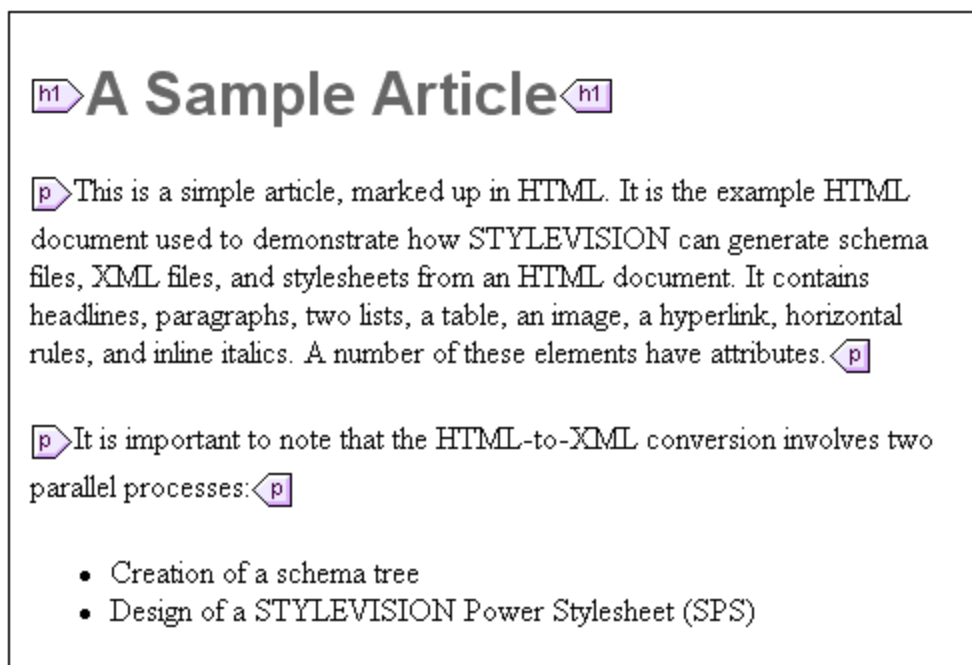
9.6.1 Créer un nouvel SPS via l'Importation HTML

Pour créer un nouveau fichier SPS depuis un document HTML, procédez comme suit :

1. Sélectionner la menu commande **Fichier | Nouveauté | Nouveauté du fichier HTML**.
2. Dans le dialogue Ouvrir qui s'affiche, cherchez le fichier HTML que vous souhaitez et cliquez sur OK. Sélectionnez-le et cliquez sur **Ouvrir**.

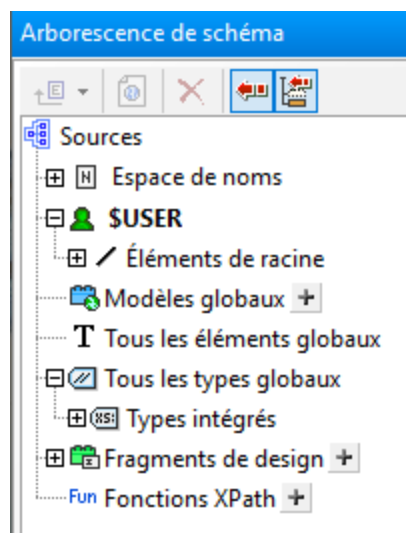
3. Vous serez invité à demander si les chemins relatifs doivent être convertis en chemins absolus. Faites votre choix en cliquant soit sur **Oui** ou **Non**.

Un nouveau document SPS est créé. Le document est affiché dans le Mode Design et est balisé avec des formats HTML prédéfinis disponibles dans StyleVision (*capture d'écran ci-dessous*).



Notez que le document HTML est affiché à l'intérieur du modèle principal. Il n'existe pas de modèle global.

Dans la barre latérale de l'arborescence de schéma, un schéma défini par l'utilisateur est créé (*capture d'écran ci-dessous*) avec un élément racine (élément de document) appelé `UserRoot`.



Notez qu'il n'existe pas d'élément global dans la liste Tous les Éléments globaux.

La structure et le design SPS

Le SPS contient un seul modèle - le modèle principal - qui est appliqué au nœud XML interne temporaire. Ce document XML a la structure du schéma défini par l'utilisateur qui a été créé dans la fenêtre de l'arborescence de schéma. Dans le Mode Design, à ce stade, les composants du document HTML à l'intérieur du modèle principal sont inclus dans le SPS comme composants statiques. La représentation de ces composants HTML dans Authentic View sera contenu non-éditable, non-XML. Les feuilles de style XSLT contiendront ces composants HTML comme éléments de résultat littéral. Le schéma, à ce stade, a uniquement l'élément du document `Root` ; en conséquence, le document XML interne temporaire contient uniquement l'élément de document `Root` sans nœud enfant.

Lorsque vous créez des sélections HTML comme éléments et attributs dans le schéma défini par l'utilisateur, vous pouvez faire ceci de deux manières :

1. En **convertissant** la sélection en un élément ou attribut. Dans le design, les balises de nœud sont insérées avec un espace réservé (`content`) à l'intérieur de la balise. Dans le schéma, un élément ou attribut est créé. Dans le document XML, la sélection est convertie en contenu de texte dans le nœud de schéma qui est créé dans le document XML. Les contenus du nœud créé dans le document XML sera inséré de manière dynamique dans la sortie obtenue via le SPS.
2. En **entourant** la sélection avec un élément ou attribut. Dans ce design, la sélection est entouré par les balises de nœuds ; aucun espace réservé (`content`) n'est inséré. Ceci signifie que la sélection est présente dans le design SPS comme contenu statique. Dans le schéma, un élément ou attribut est créé. Dans le document XML, le nœud est créé, mais vide. Le texte statique qui se trouve à l'intérieur des balises de nœud du schéma dans le design sera la sortie ; aucun contenu dynamique ne sera sorti pour ce nœud, sauf si l'espace réservé (`content`) pour ce nœud est explicitement inséré dans le design.

La signification de l'espace réservé (`content`) est qu'il indique les emplacements dans le design où les données du document XML seront affichées (dans la sortie) et peuvent être éditées (dans Authentic View).

9.6.2 Créer le schéma et le design SPS

Le schéma est créé en glissant les sélections du Mode Design dans le schéma défini par l'utilisateur. Vous faites une sélection à la fois. La sélection est déposée dans un nœud dans l'arborescence de schéma (relatif à celui du nouveau nœud qui sera créé, soit comme enfant ou frère). Vous sélectionnez le type du nœud à créer (élément ou attribut) et si la sélection doit être convertie en un nouveau nœud ou entourée par celui-ci.

La sélection

La sélection dans le Mode Design peut être une des choses suivantes :

- Un nœud dans le document HTML.
- Un string de texte à l'intérieur d'un nœud.
- Des strings de texte adjacents parmi les nœuds.
- Une image.
- Un lien.
- Une table.
- Une liste.
- Un combinaison de tout ce qui est mentionné ci-dessus.

Dans cette section, nous expliquons le processus en général pour chaque sélection. Les cas spéciaux de tables et de listes sont discutés plus en détail dans la section [Créer des Tables et Listes comme Éléments/Attributs](#) ³⁹⁸.

Pour faire une sélection, cliquez sur un composant de document HTML mettez en surbrillance le string de texte requis. Si de multiples composants doivent être sélectionnés, cliquez et glissez au-dessus des composants désirés pour mettre la sélection en surbrillance. Notez que StyleVision élargit la sélection au début et à la fin de la sélection afin de sélectionner les éléments à niveau élevé jusqu'à ce que les premiers et derniers éléments appartiennent au même parent.

L'emplacement de l'arborescence de schéma

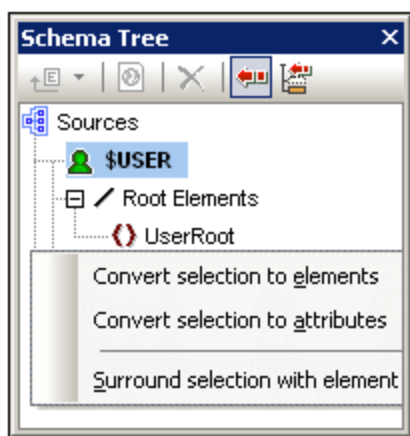
En glissant la sélection au-dessus du nœud de l'arborescence du schéma désiré, le pointeur de la souris sera modifié en l'un des symboles suivants.

- Déposer le nœud quand un symbole Créer un Frère ↓ apparaît, crée la sélection comme nœud frère du nœud sur lequel la sélection est déposée.
- Déposer le nœud quand le symbole Créer symbole Enfant ↘ apparaît, crée la sélection comme nœud enfant du nœud sur lequel la sélection est déposée.

Vous devriez sélectionner le nœud sur lequel la sélection doit être déposée selon si la sélection doit être créée comme frère ou enfant de ce nœud.

Sélectionner comment le nœud est créé

Lorsque vous déposez la sélection (*voir section précédente*), un menu contextuel apparaît (*capture d'écran ci-dessous*) dans lequel vous faites deux choix : (i) si le nœud doit être créé comme élément ou attribut ; (ii) si la sélection doit être convertie en nœud ou si le nœud doit juste envelopper la sélection.



Veuillez noter les points suivants :

- Lorsqu'une sélection est convertie en nœud (élément ou attribut), les balises de nœud, ensemble avec un espace réservé contenu (`content`), remplacent la sélection dans le design. Dans le design et la sortie, le contenu de texte de la sélection est supprimé du contenu statique. Dans la sortie, le texte de la sélection apparaît comme contenu dynamique du nœud dans le document XML.
- Si un nœud HTML est converti en un nœud XML, les balises de nœud XML sont insérées à l'intérieur des balises de nœud HTML.

- Lorsqu'une sélection (y compris les sélections de nœuds HTML) est entourée par un nœud XML, les balises de nœuds XML sont insérées avant et après la sélection. Dans le design et la sortie, le contenu de texte de la sélection est retenu comme texte statique.
- Les balises de nœuds insérées avec le chemin nécessaire (à savoir, avec les balises de nœuds ancêtre qui établissent un chemin relatif au nœud contenant). Le chemin sera absolu ou relatif dépendant du contexte du nœud dans le design.
- Comment créer les nœuds depuis la table et recenser les sélections telles que décrites dans [Créer des Tables et Listes comme Éléments/Attributs](#)³⁹⁸.

Ajouter et supprimer les nœuds dans le schéma

Vous pouvez ajouter les nœuds supplémentaires (qui ne sont pas basés sur une sélection HTML) au schéma défini par l'utilisateur. À cette fin, vous pouvez cliquer avec la touche de droite sur un nœud et sélectionner la commande requise depuis le menu contextuel. De manière alternative, vous pouvez utiliser les icônes de la barre d'outils de la barre latérale de l'arborescence de schéma.

Pour supprimer un nœud, sélectionnez le nœud, puis utilisez soit le menu contextuel, soit l'icône de la barre d'outils. Notez, toutefois, que quand un nœud est supprimé, quelques chemins dans le design pourraient être invalides.

Modifier le design

Vous pouvez modifier la structure du design en glissant les composants autour et en insérant les composants statiques et dynamiques. Les styles peuvent également être modifiés en utilisant les diverses capacités de style de StyleVision.

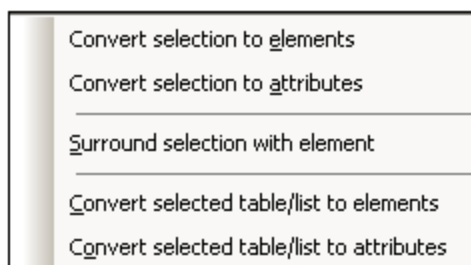
9.6.3 Créer des Tables et Listes comme Éléments/Attributs

Les Tables et listes dans le document HTML peuvent être convertis en nœuds élément ou attribut dans le schéma XML pour qu'ils conservent la structure de la table ou de la liste dans le schéma.

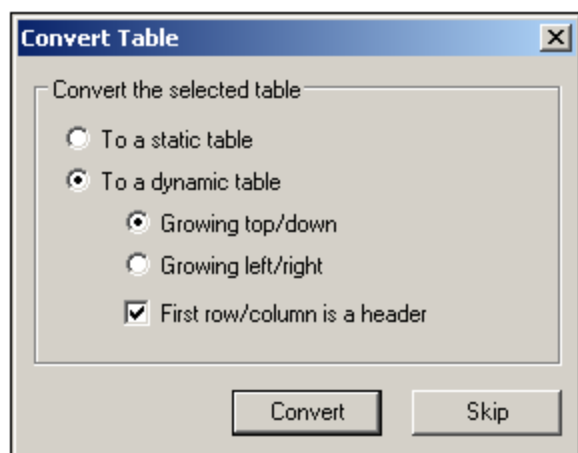
Convertir une table en éléments/attributs

Pour convertir une table aux nœuds de schéma, suivez les étapes suivantes :

1. Sélectionnez la table HTML en y mettant du texte en surbrillance.
2. Glissez le nœud dans l'arborescence de schéma comme frère ou enfant duquel vous voulez le créer.
3. Déposez le nœud quand le symbole Créer comme Frère ↓ ou Créer comme symbole Enfant ↘ apparaît.
4. Dans le menu contextuel qui apparaît maintenant (*capture d'écran ci-dessous*), sélectionnez la commande **Convertir la table/liste en éléments** ou **Convertir la table/liste sélectionnée en attributs** selon si vous souhaitez créer les contenus des cellules de table comme éléments ou attributs, respectivement.



5. Dans le dialogue Convertir Table qui apparaît (*capture d'écran ci-dessous*), sélectionnez si la table créée dans le SPS doit être une table statique ou dynamique.



Si l'option **table statique** est sélectionnée, alors pour chaque table dans la cellule, un nœud de schéma est créé. Dans le design, chaque nœud est inséré avec l'espace de nœud (`content`). Les données dans les cellules de table sont copiées dans le document XML interne temporaire (et dans le document XML généré). L'option **table dynamique** est disponible quand la structure de toutes les lignes dans la table est identique. Lorsqu'elles sont créées dans SPS, les lignes de la table dynamique sont représentées par une seule ligne dans le design (car chaque ligne a la même structure). Les données de la table seront rédigées dans un fichier XML. La table dynamique peut s'agrandir haut/bas (les lignes sont arrangées de manière verticale l'une par rapport à l'autre) ou gauche/droite (les lignes deviennent des colonnes et s'élargissent de gauche à droite). Si vous indiquez que la première ligne/colonne a une en-tête, alors (i) une ligne d'en-tête contenant les en-têtes de colonne comme texte statique est incluse dans le design; et (ii) l'élément de schéma/attribut de nœuds prennent les textes d'en-tête comme leurs noms. Si la première ligne/colonne n'est pas indiquée comme en-tête, alors aucune ligne d'en-tête n'est incluse dans le design.

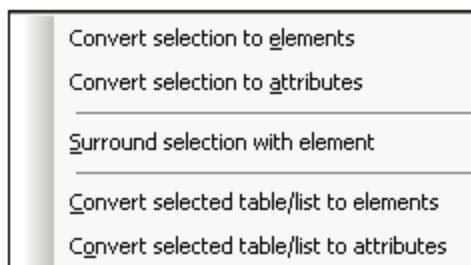
6. Une fois que vous avez sélectionné l'/les option/s requises, cliquez sur **Convertir** pour terminer.

Convertir une liste en éléments/attributs

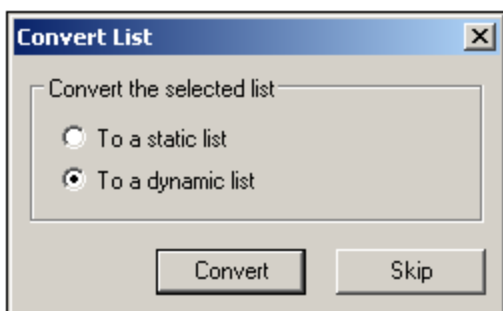
Pour convertir une liste aux nœuds de schéma, suivez les étapes suivantes :

1. Sélectionnez la liste HTML en y mettant du texte en surbrillance.
2. Glissez le nœud dans l'arborescence de schéma comme frère ou enfant duquel vous voulez le créer.
3. Déposez le nœud quand le symbole Créer comme Frère ↓ ou Créer comme symbole Enfant ↘ apparaît.
4. Dans le menu contextuel qui apparaît maintenant (*capture d'écran ci-dessous*), sélectionnez la commande **Convertir la table/liste en éléments** ou **Convertir la table/liste sélectionnée en**

attributs selon si vous souhaitez créer les contenus des cellules de table comme éléments ou attributs, respectivement.



5. Dans le dialogue Convertir Liste qui apparaît (*capture d'écran ci-dessous*), sélectionnez si la liste créée dans SPS devrait être une liste statique ou dynamique.



Si l'option **liste statique** est sélectionnée, alors pour chaque item de liste, un nœud de schéma est créé. Dans le design, chaque nœud inséré avec le texte de l'item de la liste HTML inclus dans le contenu statique de l'item de liste. Si l'option **liste dynamique** est sélectionnée, alors l'item de liste est représenté par un seul nœud d'item de liste dans le design. Dans le design, l'élément de l'item de liste est inséré avec l'espace réservé (*content*).

6. Une fois que vous avez sélectionné l'option requise, cliquez sur **Convertir** pour terminer.

9.6.4 Générer la sortie

Une fois SPS accompli, vous pouvez générer la sortie suivante utilisant la commande **Fichier | Enregistrer Fichiers générés** :

- Schéma défini par l'utilisateur généré, qui est le schéma que vous avez créé dans la barre latérale de l'arborescence de schéma.
- Les données XML générées définies par l'utilisateur, qui est un document XML basé sur le schéma que vous avez créé et contenant les données importées depuis le fichier HTML.
- Les feuilles de style pour la sortie HTML.
- La sortie HTML.

9.7 Interface ASPX pour les applications Web

Si un rapport HTML ou des données BD ou XML pour Internet doit être créé avec un SPS, alors la procédure usuelle pour créer le rapport avec StyleVision il se ferait comme suit :

1. Si les données source sont dans une BD, alors, avec le SPS actif terminé dans StyleVision, générez un fichier XML depuis la BD. (Si les données source sont dans un fichier XML, alors cette étape n'est pas requise.)
2. Également du SPS, générez le fichier XSLT-pour-HTML.
3. Transformez le fichier XML utilisant le fichier XSLT-pour-HTML généré.
4. Placez le fichier HTML résultant sur le serveur.

Pour une application web, le fichier HTML pourrait être obsolète si les données source (BD ou XML) sont modifiées. Mettre à jour le fichier HTML sur le serveur Web avec les nouvelles données nécessiterait : (i) pour les données basées pour la BD, la régénération du fichier XML, (ii) transformer dans le nouveau fichier XML utilisant XSLT-pour-HTML, et (iii) placer le fichier HTML de résultat sur le serveur.

StyleVision fournit une solution pour rapidement mettre à jour les pages web HTML. Il s'agit d'une fonction pour générer automatiquement une application ASPX. Tous les fichiers d'application ASPX requis (le fichier `.aspx`, le fichier XSLT, et les fichiers de code) sont générés par StyleVision. Ces fichiers peuvent être placés sur le serveur ensemble avec le fichier BD source ou le fichier XML et le fichier XSLT-pour-HTML. À chaque fois que le fichier `.aspx` - qui est le fichier de l'interface web - est actualisé, la chose suivante se passe : (i) pour les données basées sur BD, un nouveau fichier XML est généré depuis la BD ; pour les données basées sur XML, cette étape n'est pas requise ; (ii) le fichier XML est transformé utilisant le fichier XSLT-pour-HTML qui est sur le serveur ; et (iii) la sortie de la transformation est affichée sur la page de l'interface web. De cette manière, la page de l'interface web affichera rapidement les dernières données BD ou XML mises à jour.

Générer des fichiers pour une solution ASPX.

Après avoir créé le SPS basé sur BD ou le SPS basé sur XML, suivez les étapes suivantes pour créer une solution ASPX :

1. Avec un SPS actif dans StyleVision, générez les fichiers ASPX en cliquant sur la commande, **Fichier | Design Web | Générer l'Application Web ASPX**. Les fichiers de l'application ASPX seront créés dans l'emplacement du dossier que vous spécifiez. Le dossier dans lequel vous générez une application ASPX contiendra les fichiers suivants, entre autres :
 - `Readme.doc`
 - `SPSFilename.aspx`
 - `SPSFilename.xslt`
 - `SPSFilename.cs`
2. Placez le fichier BD ou fichier XML sur le serveur, dans le même dossier que l'application ASPX. Le fichier `.aspx` est un point d'entrée de l'application. Actualiser ce fichier fera que les données BD ou XML y affichées seront mises à jour.

Note : vous devrez avoir l'[application RaptorXML d'Altova](#) installée pour que la transformation XSLT soit exécutée correctement. Si vous avez des problèmes avec la transformation, allez au fichier `ReadMe.doc` pour le détails concernant la configuration RaptorXML.

Comment cela fonctionne ?

Le dossier dans lequel vous générez une application ASPX contiendra les fichiers suivants, entre autres :

- `Readme.doc`
- `SPSFilename.aspx`
- `SPSFilename.xslt`
- `SPSFilename.cs`

`SPSFilename.aspx` est l'URL du document de sortie. Le `SPSFilename.aspx` exécute un code C# stocké dans le fichier `SPSFilename.cs`. Ce code C# lit le contenu XML (des fichiers ou une base de données, tel que requis) et le transmet à RaptorXML, ensemble avec le fichier `SPSFilename.xslt`. (RaptorXML contient un moteur de transformation XSLT d'Altova. Il peut être téléchargé depuis le site web d'Altova.) RaptorXML réalise une transformation du contenu XML, utilisant le fichier XSLT fourni. Le résultat est un document HTML, que l'application web affiche ensuite dans le navigateur. Lorsque le contenu XML change, par exemple en raison des changements effectués à la base de données, aller au `SPSFilename.aspx` (ou rafraîchissez la page dans le navigateur) récupérera les données les plus récentes depuis une base de données ou un fichier XML file et rendra un document mis à jour.

9.7.1 Exemple : Localhost sur Windows 7

La procédure décrite ci-dessous définit votre hôte local pour servir une application ASPX. Pour plus d'informations, voir le fichier `Readme.docx` dans le dossier d'application ASPX%. Le dossier et le fichier sont générés quand vous sélectionnez la commande **Fichier | Design Web | Générer Application Web ASPX** avec le fichier SPS actif.

Installer RaptorXML

Assurez-vous que la dernière version de RaptorXML est installée. RaptorXML contient un moteur de transformation Altova. Il sera utilisé pour transformer le fichier XML (généré par BD).

Activer les services d'information Internet (le serveur web de Microsoft)

Si les services d'information Internet (IIS) ne sont pas activés, exécutez les étapes ci-dessous pour l'activer. L'étape 5 affiche comment tester si IIS ont été activés.

1. Allez à *Control Panel | Programs and Features | Turn Windows features on or off*.
2. Cochez la case à cocher *Services d'information Internet*. La case à cocher 'en mode trois-états changera en *Partly checked*.
3. De plus, cochez la case à cocher *Internet Information Services | World Wide Web Services | Application Development Features | ASP.NET*.
4. Cliquez sur **OK**. Lorsque le processus est achevé, vous aurez un dossier appelé `C:/inetpub/wwwroot`. Il s'agit du dossier racine du serveur web.
5. En guise de test, allez à `localhost` dans le navigateur. Ceci affichera l'écran d'accueil IIS

Dans StyleVision, générez l'application ASPX.

Générez l'application ASPX comme suit :

1. Il est recommandé que la base de données et le fichier SPP soit dans le même dossier.
2. Une fois que le fichier SPS a été achevé, exécutez la commande *Files | Web Design | Generate ASPX Web Application*.
3. Dans le dialogue qui s'ouvre, créez un dossier en-dessous de `C:/inetpub/wwwroot` et sélectionnez ce dossier, par exemple : `C:/inetpub/wwwroot/Test1`.
4. Sur confirmation de votre sélection du dossier, StyleVision y générera les fichiers suivants : `<FileName>.aspx`, `<FileName>_AltovaDataBaseExtractor.cs`, `Web.config`, et un dossier `App_Code` contenant divers fichiers.

Note : afin d'enregistrer les fichiers dans `C:/inetpub/wwwroot`, vous devrez exécuter StyleVision comme administrateur. Faites ceci en redémarrant StyleVision. Cliquez avec la touche de droite de la souris sur le fichier exécutable StyleVision ou avec un raccourci et sélectionnez **Exécuter comme administrateur**.

Faites de sorte à ce que ASPX est conscient de l'application générée

Effectuez les étapes suivantes pour faire de sorte à ce que ASPX est conscient de l'application que vous générez avec StyleVision:

1. Allez à *Control Panel | Administrative Tools | Internet Information Services (IIS) Manager*.
2. Dans le volet *Connexions*, élargissez l'arborescence pour afficher le dossier (par exemple, `Test1`). L'icône du dossier sera un dossier jaune standard à ce point.
3. Dans le menu contextuel du dossier, exécutez la commande *Convertir en application*, puis cliquez sur **OK** dans le dialogue. L'icône du dossier sera un globe maintenant.
4. Dans le volet *Connexions*, élargissez l'arborescence pour afficher le *Pool d'application* et sélectionnez-le.
5. Dans le menu contextuel pour *DefaultAppPool* (disponible dans le volet principal quand vous sélectionnez l'item *Pools d'application* dans le volet *Connexions*), sélectionnez la commande **Paramètres avancés**.
6. Pour la propriété *Identité*, sélectionnez *Compte personnalisé* et saisissez votre nom utilisateur et mot de passe Windows.
7. Pour la propriété *Activer Applications 32-Bit*, saisissez `True`. (Ceci est le cas pour que les pilotes de base de données ont accès). Cette étape s'applique uniquement dans les versions 64-bit de Windows 7.

Exécuter l'application

Dans le navigateur, allez à `localhost/Test1/<FileName>.aspx` (en partant du fait que `Test1` est le nom du dossier dans lequel l'application ASPX a été enregistrée). Le HTML transformé sera affiché dans le navigateur. Actualiser la page ASPX affichera les dernières données de la base de données ou du fichier XML.

Note : si le navigateur est bloqué à ce point, assurez-vous que RaptorXML est correctement mis sous licence.

9.8 Fichier PXF : Conteneur pour fichiers SPS et associés

Un design SPS qui utilise XSLT 2.0 ou 3.0 peut être enregistré comme fichier 'Portable XML Form' (PXF). Le format de fichier PXF a été développé spécialement par Altova pour packager le design SPS avec des fichiers associés (tels que le fichier de schéma, le fichier source XML, les fichiers d'image utilisés dans le design, et les fichiers XSLT pour la transformation de l'XML source en format de sortie). L'avantage du format de fichier PXF est que tous les fichiers requis pour l'édition de l'Aperçu Authentique et pour la génération de la sortie de l'Aperçu Authentique peuvent être distribués facilement en un seul fichier.

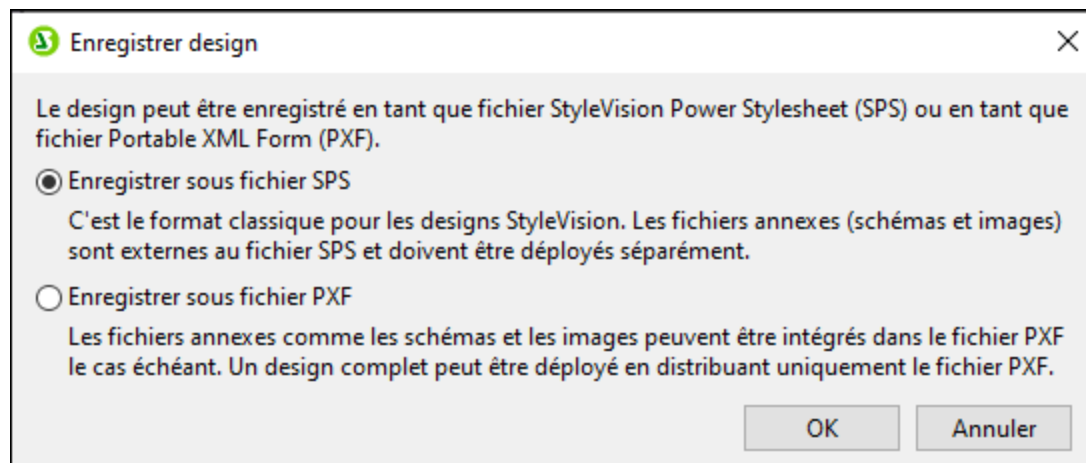
Cette section décrivant l'utilisation des fichiers PXF est organisée en deux parties :

- [Créer un fichier PXF](#) ⁴⁰⁴
- [Éditer un fichier PXF](#) ⁴⁰⁷
- [Déployer un fichier PXF](#) ⁴⁰⁸

Note : les fichiers PXF peuvent être créés uniquement depuis des SPS conçus avec XSLT 2.0 ou 3.0.

9.8.1 Créer un fichier PXF

Pour créer un fichier PXF qui contiendra un design SPS plus les fichiers associés, ouvrez le design SPS dans StyleVision et sélectionnez la commande **Fichier | Enregistrer sous**. Ceci fait apparaître le dialogue Enregistrer Design (*capture d'écran ci-dessous*).

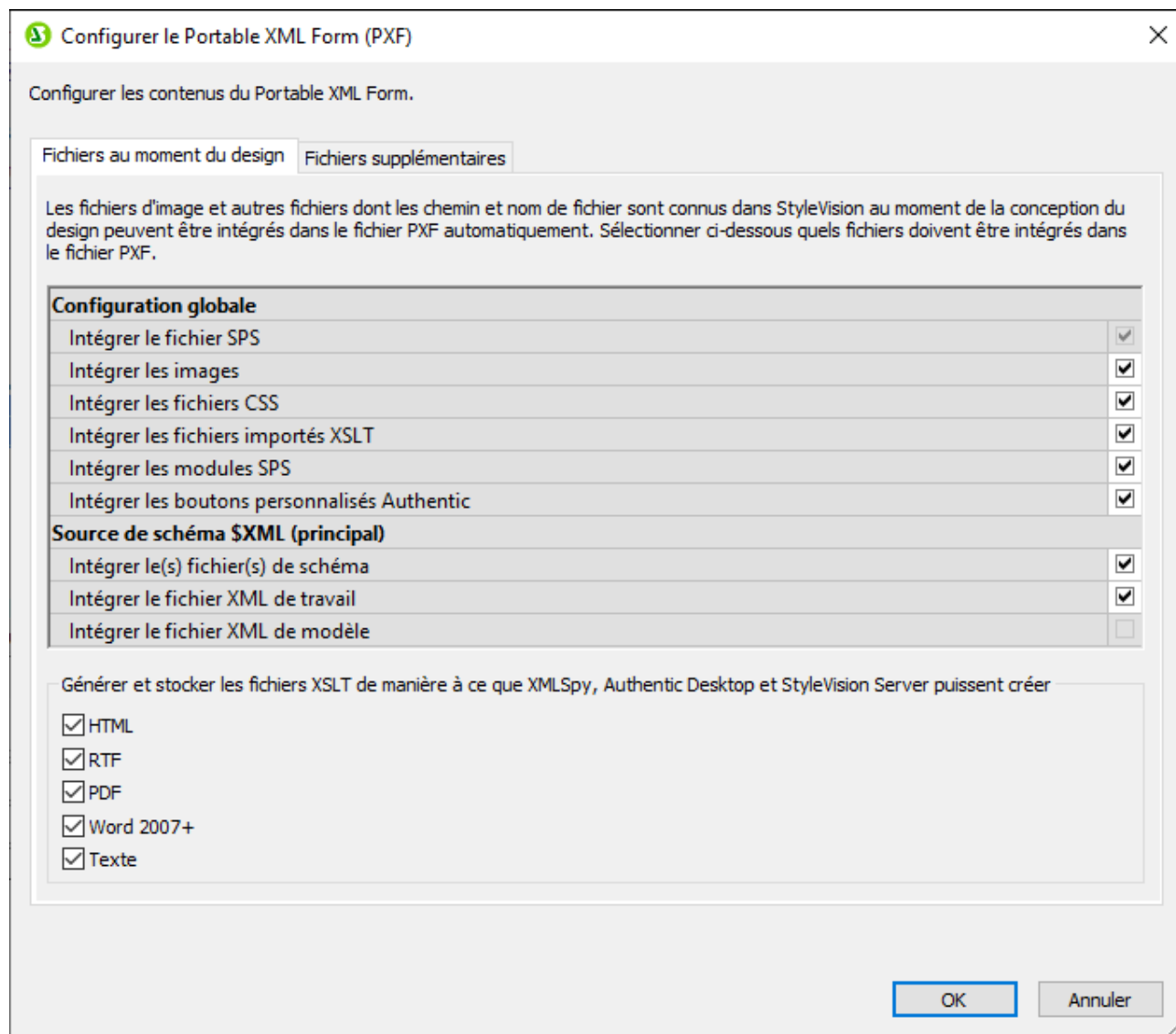


Le **format SPS** est le format Altova standard pour les designs StyleVision. Dans cette section, nous nous penchons sur le format PXF et donc nous ne prendrons pas en compte le format SPS ici. Enregistrer un fichier comme SPS est décrit en détail dans la [section Référence de l'utilisateur](#) ⁴⁶⁸.

Enregistrer sous PXF

La sélection de l'option PXF fait apparaître le dialogue Enregistrer sous familier des systèmes Windows. Enregistrer du travail exactement tel que décrit pour la [commande](#) ⁴⁶³ [Enregistrer Design](#) ⁴⁶³ —avec l'étape supplémentaire de sélectionner les fichiers que vous souhaitez inclure dans le fichier PXF. Une fois que vous avez spécifié le nom du fichier PXF, le dialogue Configurer PXF (*capture d'écran ci-dessous*) apparaîtra, dans

lequel vous sélectionnez/désélectionnez les fichiers que vous souhaitez intégrer.



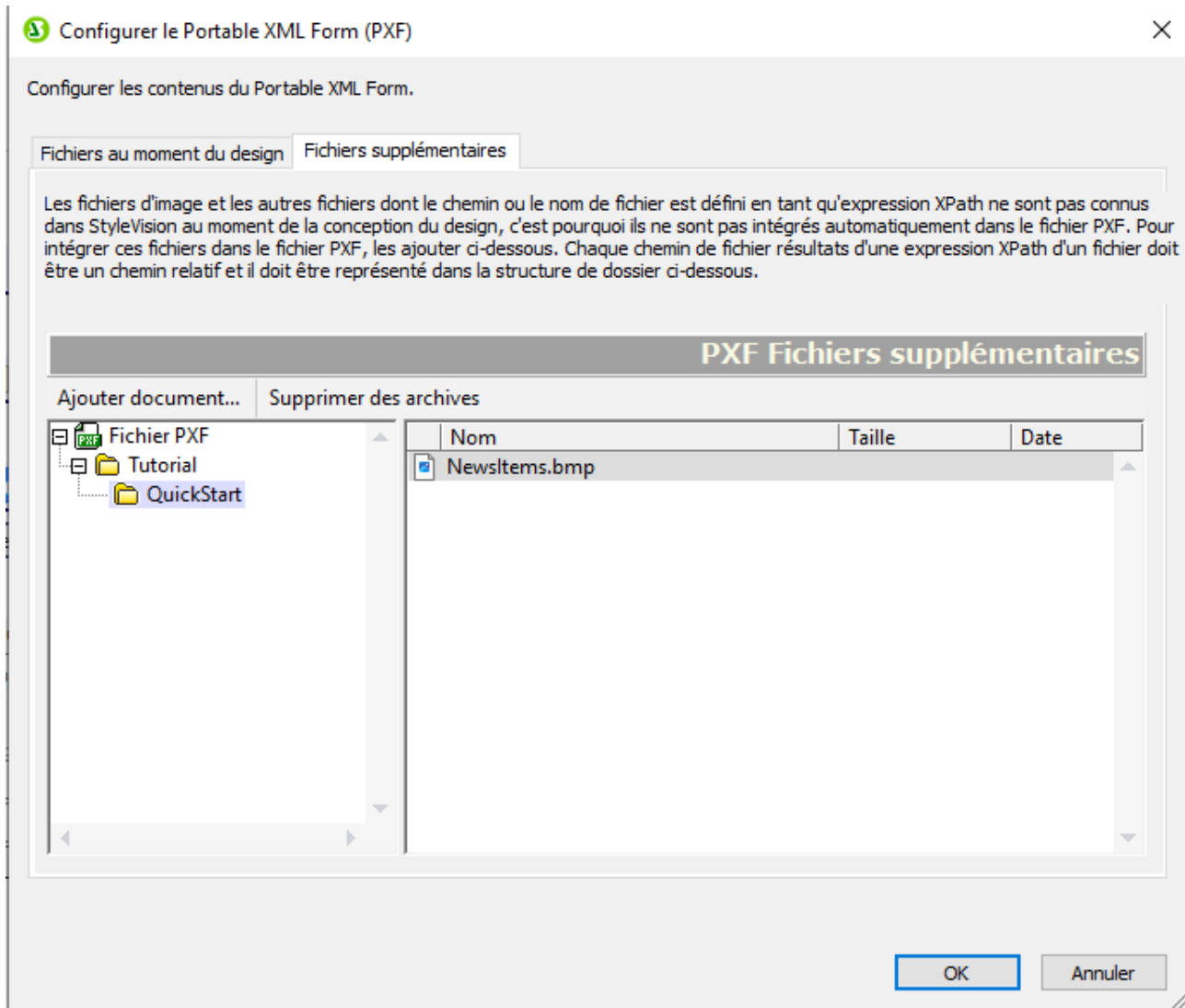
Dans le volet Configuration globale de l'onglet *Fichiers Design-time*, vous pouvez sélectionner/désélectionner les fichiers source liés au design à être intégrés/omis. De plus, vous pouvez choisir d'intégrer les fichiers XSLT générés depuis le design. Dans le volet des fichiers XSLT, sélectionnez les formats de sortie pour lesquels vous souhaitez générer et intégrer les fichiers XSLT. Si un fichier XSLT est inclus dans le fichier PXF file et que le fichier PXF est ouvert dans le Mode Authentic d'un produit Altova, alors le bouton de la barre d'outils pour générer et consulter ce format de sortie est activé dans le Mode Authentic (*capture d'écran ci-dessous*).



Note : Si les fichiers XSLT pour les sorties ne sont pris en charge que dans une édition supérieure de StyleVision (*de la plus élevée à la plus basse : Enterprise, Professional, Basic*) étaient créés dans un fichier PXF et que si ce fichier PXF était ouvert dans une édition inférieure, alors en enregistrant le fichier PXF file, les fichiers XSLT pour les sorties qui ne sont pas pris en charge dans l'édition inférieure ne seront pas enregistrés.

Une invite apparaît, demandant si vous souhaitez continuer à enregistrer le fichier PXF. Vous pouvez ensuite passer à l'enregistrement sans les formats non pris en charge ou abandonner l'enregistrement et garder les formats non pris en charge.

Dans l'onglet *Fichiers supplémentaires* (capture d'écran ci-dessous), vous pouvez préciser des fichiers supplémentaires que vous souhaitez inclure qui ne sont pas des fichiers 'design-time'. Ceux-ci pourraient être, par exemple, des fichiers image référencés dans le design par l'URL générée avec une expression XPath. Dans la capture d'écran ci-dessous, le fichier image `NewsItems.bmp` situé dans le dossier `Tutoriel/Images` est sélectionné pour être inclus dans le fichier PXF.



Pour inclure un fichier supplémentaire dans le fichier PXF, cliquez sur le bouton **Ajouter Document**, puis recherchez le fichier que vous souhaitez. Le dialogue Ouvrir (dans lequel vous recherchez le fichier requis) ouvre le dossier dans lequel la SPS est située. Les fichiers de ce dossier ou tout dossier descendant peuvent être sélectionnés. Une fois qu'un fichier supplémentaire a été ajouté au fichier PXF, celui-ci et la structure du dossier y menant sont affichés. La capture d'écran ci-dessus indique que le fichier supplémentaire `NewsItems.bmp` est un dossier dénommé `Images`, qui est lui-même contenu dans un fichier dans lequel se trouve le fichiers SPS.

Si un fichier est sélectionné depuis un dossier situé à tout niveau au-dessus du dossier contenant le fichier SPS, une erreur est rapportée.

Dans le design SPS, toute référence à un fichier supplémentaire doit être faite avec un chemin relatif et doit utiliser la structure du dossier affichée dans le volet *Fichiers supplémentaires*. Par exemple, `NewsItems.bmp` dans la capture d'écran ci-dessus doit être référencé avec un chemin relatif : `Images/NewsItems.bmp`.

Note : afin d'enregistrer les fichiers PXF, l'option *Intégration de l'image dans RTF et Word 2007+ (Fichier | Propriétés | Images)* doit être sélectionnée.

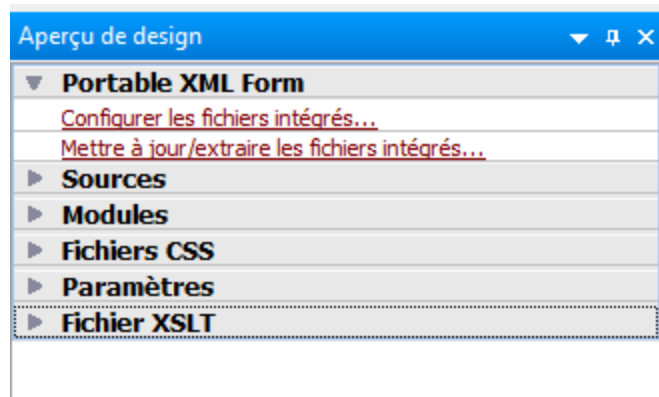
9.8.2 Éditer un fichier PXF

Un fichier PXF peut être ouvert dans StyleVision via la commande **Fichier | Ouvrir** ⁴⁵⁸ et édité. Ces éditions peuvent être de deux types :

- La configuration du fichier PXF peut être éditée
- Le contenu des fichiers de composants individuels tel que SPS et Authentic XML peut être édité dans StyleVision, tandis que d'autres fichiers de composants (comme les fichiers image et CSS) peuvent être édités dans des applications externes. Tous les fichiers de composants doivent toutefois explicitement être mis à jour dans StyleVision.

Point d'entrée pour l'édition PXF

Le point d'entrée pour éditer la configuration PXF et pour mettre à jour le fichier PXF est l'item PXF dans la barre latérale de l'Aperçu Design (*capture d'écran ci-dessous*).

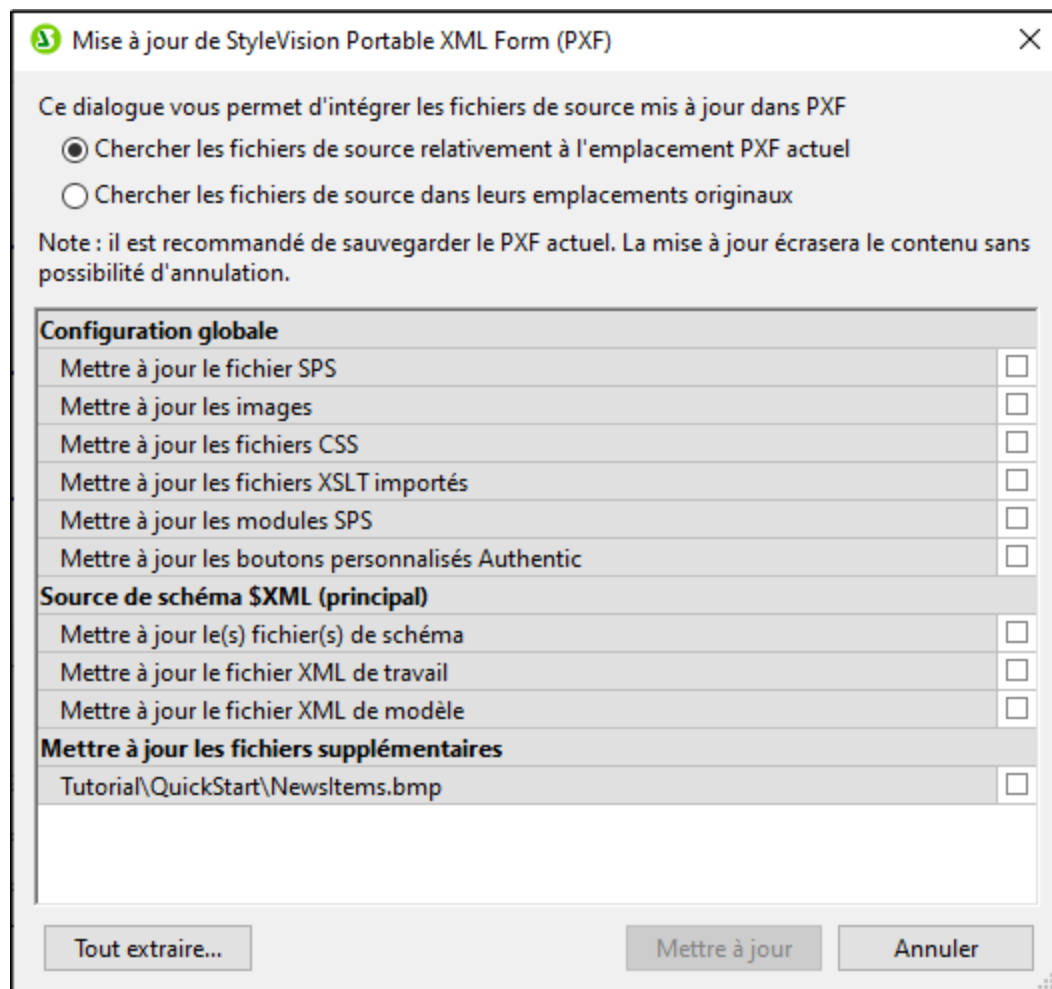


Configurer les fichiers intégrés

Cliquer sur le lien *Configurer les fichiers intégrés* dans l'Aperçu de design (*voir la capture d'écran ci-dessus*) ouvre le dialogue Configurer 'Portable XML Form (PXF)'. Les options de configuration sont exactement les mêmes que décrit dans la section, [Créer un Fichier PXF](#) ⁴⁰⁴.

Mettre à jour les fichiers intégrés

Cliquer sur le lien *Mettre à jour les fichiers intégrés* dans l'Aperçu de design (voir la capture d'écran ci-dessus) ouvre le dialogue Mise à jour de 'Portable XML Form (PXF)' (capture d'écran ci-dessous).



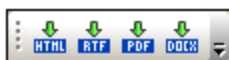
Premièrement, sélectionnez si les fichiers source devraient être extraits par rapport à leurs emplacements PXF actuels ou depuis leurs emplacements d'origine. Puis, vérifiez les fichiers que vous souhaitez mettre à jour et cliquez sur **Mette à jour**. Un nouveau fichier PXF sera créé et écrasera le fichier PXF existant. Pour cette raison, avant la mise à jour, il est hautement recommandé que vous sauvegardiez le fichier PXF d'origine.

9.8.3 Déployer un fichier PXF

Une fois qu'un fichier PXF a été créé, il peut être transporté, téléchargé, copié et enregistré comme tout autre fichier de données. Puisque le fichier PXF peut contenir tous les fichiers requis pour éditer un fichier XML file dans le Mode Authentic et générer des rapports de sortie, c'est le seul fichier dont un utilisateur Authentic a besoin pour se lancer et générer la sortie.

Un fichier PXF peut être ouvert dans [Authentic View des produits Altova](#)¹⁹. Pour vous donner une idée de comment vous pouvez utiliser un fichier PXF, vous trouverez une liste de quelques scénarios d'utilisation dans XMLSpy:

- Le fichier PXF est ouvert via la commande **Fichier | Ouvrir**. Le fichier XML intégré sera affiché dans Authentic View utilisant le SPS intégré, et peut être édité dans Authentic View. La commande **Fichier | Enregistrer** enregistre les changements dans le PXF (le XML intégré est modifié).
- Le fichier PXF file ne contient aucun fichier XML intégré et est ouvert via la commande **Fichier | Ouvrir**. Si aucun fichier XML n'est inclus, alors le fichier modèle XML basé sur le design SPS est ouvert dans le Mode Authentic. La commande **Fichier | Enregistrer** enregistrera ce fichier XML comme fichier intégré dans le fichier PXF.
- Dans le produit Altova XMLSpy, un fichier XML peut être associé à un fichier PXF pour que le SPS intégré du fichier PXF soit utilisé pour l'édition Authentic View. L'association est faite via la commande de menu **Authentic | Attribuer une feuille de style StyleVision**. Lorsque les changements sont enregistrés, ils seront enregistrés dans le fichier XML ; le fichier PXF sera inchangé.
- Si une feuille de style XSLT pour un des formats de sortie a été intégrée dans le fichier PXF, alors l'utilisateur Authentic View sera capable de générer la sortie dans ce format. Ceci est fait avec le bouton de la barre d'outils 'output-generation' approprié (*capture d'écran ci-dessous*). Dans Authentic View, les boutons de la barre d'outils output-generation individuels ne seront activés que si le fichier PXF a été configuré pour contenir la feuille de style pour cette sortie. Par exemple, si le fichier PXF a été configuré pour contenir les feuilles de style XSLT pour HTML et PDF, alors uniquement les boutons de la barre d'outils seront activés tandis que ceux pour la sortie RTF et DocX (Word 2007+) seront désactivés.



Note : si un fichier PXF est situé sur un serveur web et sera utilisé avec Authentic Browser Plug-in, assurez-vous que le serveur ne bloque pas le fichier. Vous pouvez faire ceci en ajoutant (via le panel d'administration IIS, par exemple,) le type MIME pour PXF suivant, (.pxf) extensions de fichier : `application/x-zip-compressed`.

10 Traitement automatisé

La fonctionnalité de StyleVision ensemble avec les différents fichiers XSLT et de sortie générés par StyleVision fournit des possibilités d'automatisation puissantes. Cette section décrit leurs capacités.

Fonction de génération de fichier de StyleVision

Une fois que vous avez créé un design SPS avec StyleVision, vous pouvez générer plusieurs fichiers XSLT et de sortie depuis la GUI, dépendant quelle édition de StyleVision vous utilisez (Enterprise, Professional ou Basic). Les fichiers suivants peuvent être générés avec la commande [Fichier | Enregistrer les fichiers générés](#)⁴⁶⁹ :

- Les fichiers XSLT pour les sorties HTML.
- Les sorties HTML.

Comme vous le voyez de la liste ci-dessus, les fichiers qui peuvent être enregistrés avec StyleVision sont de deux types :

1. Les fichiers XSLT générés par le design SPS et
2. les fichiers de sortie finale (telle que HTML).

Les processus pour générer les fichiers de sortie finale HTML sont tous des processus à une étape dans lesquels le document XML est transformé par une feuille de style XSLT en format de sortie.

StyleVision Server et RaptorXML : générer les fichiers de l'intérieur de la GUI

Outre la génération de feuilles de style XSLT et les formats de sortie requis via la commande GUI de StyleVision ([Fichier | Enregistrer les fichiers générés](#)⁴⁶⁹), vous pouvez générer les fichiers de sortie utilisant deux autres méthodes :

1. Avec StyleVision Server, qui appelle la fonction de génération de fichier de StyleVision sans ouvrir la GUI, vous pouvez produire les différents types de sortie.
2. Avec [RaptorXML](#)⁴¹⁴, une application autonome d'Altova qui contient le validateur XML (+XBRL) d'Altova, et les moteurs XSLT et XQuery. Les Moteurs XSLT dans RaptorXML peuvent être utilisés pour les transformations de XML en format de sortie en traitant les documents XML avec des feuilles de style XSLT. Le fichier XSLT devra être créé en avance pour qu'il puisse être utilisé par RaptorXML. (RaptorXML ne prend pas de SPS comme paramètre d'entrée.) Les avantages d'utiliser les RaptorXML sont : (i) vitesse, en guise de résultat permettant les transformations rapides de grands fichiers ; et (ii) en plus d'une interface de ligne de commande, RaptorXML fournit des interfaces pour COM, Java et .NET, et peut être facilement appelé pour cette raison de l'intérieur de ces environnements. Comment utiliser RaptorXML pour les transformations est expliqué dans la sous-section [RaptorXML](#)⁴¹⁴.
3. De multiples transformations peuvent être exécutées selon des déclencheurs prédéfinis (tels que l'heure quotidienne) utilisant Altova StyleVision Server à l'intérieur du flux de travail de Altova FlowForce Server. Ceci est décrit dans la section [Automatisation avec FlowForce Server](#)⁴¹⁷.

10.1 Interface de ligne de commande

La fonction de StyleVision peut être appelée depuis la ligne de commande de deux manières :

- En appelant le [programme exécutable de StyleVision](#) ⁴¹¹. Ceci fournit un accès à la fonction de génération du fichier XSLT de StyleVision. Les fichiers XSLT sont générés depuis le fichier SPS.
- En utilisant [StyleVision Server](#) ⁴¹² pour générer les fichiers de sortie (HTML, etc). Les fichiers de sortie sont générés depuis le fichier PXF, qui est un pack d'un fichier SPS avec ses fichiers associés (XML, XSD, fichiers d'image, etc). Le fichier PXF est généré depuis StyleVision.

Comment utiliser la ligne de commande

Il existe deux manières dont vous pouvez utiliser la ligne de commande :

- La commande peut être saisie à elle seule sur la ligne de commande et exécutée immédiatement. Par exemple, dans une fenêtre d'invite à la commande, vous pouvez saisir une commande pour StyleVision ou [StyleVision Server](#) ⁴¹², et appuyer sur **Enter** afin d'exécuter la commande.
- Une série de commandes peut être saisie dans le **fichier batch** pour le traitement batch. Par exemple :

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

Lorsque le fichier batch est traité, les commandes sont exécutées et les fichiers sont générés.

La fonction de StyleVision dans les tâches prévues

En utilisant l'outil Scheduled Tasks (Tâches prévues) de Windows, les commandes de StyleVision peuvent être définies pour être exécutées selon le calendrier prédéfini. Soit une seule commande ou un fichier batch peuvent être spécifiés comme tâches à exécuter. Comment créer de telles commandes est décrit dans [Comment automatiser le traitement](#) ⁴¹⁹.

10.1.1 StyleVision

L'utilisation de la syntaxe pour la ligne de commande est :

```
StyleVision [<SPS File>] [<options>]
```

où

StyleVision	appelle StyleVision, qui est situé dans le dossier d'application StyleVision
<SPS File>	spécifie le fichier SPS
<options>	Une ou plus d'options recensées ci-dessous.

Lorsqu'une commande est exécutée, StyleVision est exécuté en silence (à savoir sans que le GUI n'est ouverte), et génère les fichiers de sortie, et se ferme. Si une erreur ou un avertissement est signalé, la GUI est ouverte et le message correspondant est affiché dans une boîte de message.

Note : pour que la SPS puisse chargée correctement dans StyleVision, les fichiers de travail XSD et XML que la SPS utilise doivent être à leurs emplacements spécifiés pour eux dans la SPS.

Options

Les options doivent être saisies quel que soit l'ordre. Notez que les options associées aux sorties FO, RTF, PDF et Word 2007+ sont disponibles dans l'édition Enterprise, ou uniquement les éditions Enterprise et Professional ; ces options sont indiquées avec les mots *Enterprise edition* ou *éditions Enterprise et Professional* dans la liste ci-dessous.

- **Sortie de fichier XSLT**

<code>-OutXSLT=<file></code>	Écrit XSLT-pour-HTML au fichier spécifié
<code>-OutXSLRTF=<file></code>	Écrit XSLT-pour-RTF dans le fichier spécifié (<i>éditions Enterprise et Professional</i>)
<code>-OutXSLText=<file></code>	Écrit XSLT-pour-Text dans le fichier spécifié (<i>éditions Enterprise et Professional</i>)
<code>-OutXSLFO=<file></code>	Écrit XSLT-pour-FO dans le fichier spécifié (<i>édition Enterprise uniquement</i>)
<code>-OutXSLWord2007=<file></code>	Écrit XSLT-pour-Word 2007+ dans le fichier spécifié (<i>édition Enterprise uniquement</i>)

Exemples

```
StyleVision "QuickStart.sps" -outxslt="QuickStartHTML.xslt"
StyleVision "C:\Test\QuickStart.sps" -outxslt="C:\Test\QuickStartHTML.xslt"
```

Points à noter

Notez les points suivants :

- Les chemins peuvent être absolus ou relatifs et devraient utiliser une barre oblique inversée.
- Si le nom de fichier ou le chemin qui y emmène contient un espace, puis le chemin entier devrait être enveloppé de guillemets. Par exemple : "c:\My Files\MyXML.xml" or "c:\MyFiles\My XML.xml".
- Les commandes, chemins et dossiers, ainsi que les noms de fichier sont sensibles à la casse.

10.1.2 StyleVision Server

StyleVision Server peut être utilisé via son interface de ligne de commande (CLI) sur les systèmes Windows et Linux pour convertir des fichiers XML en documents HTML, PDF, RTF et DOCX. La commande CLI `generate` de StyleVision Server prend un fichier XML et un [fichier PXF](#) ⁴⁰⁴ comme ses deux arguments et les formats de sortie désirés comme ses paramètres. Les feuilles de style XSLT pour la transformation sont obtenues depuis le [fichier PXF](#) ⁴⁰⁴ soumises comme entrée.

Un avantage d'utiliser la CLI de StyleVision Server au lieu de la CLI de RaptorXML Server est que StyleVision Server peut prendre les fichiers PXF comme son entrée (RaptorXML prend un fichier XSLT comme son entrée). StyleVision Server est toutefois utilisé de manière la plus efficace lorsqu'il est utilisé comme faisant partie d'un flux de travail d'Altova FlowForce. Un flux de travail FlowForce peut démarrer des tâches de transformation selon les déclencheurs prédéfinis : Les multiples fichiers peuvent être transformés automatiquement à l'intérieur d'un réseau quand la tâche FlowForce est déclenchée. Voir la section [Automatisation avec FlowForce Server](#)⁴¹⁷ pour plus d'information.

Pour plus d'information sur le CLI de StyleVision Server, voir la [documentation StyleVision Server](#).

Fichiers de sortie

StyleVision Server peut générer un ou plus de fichiers suivants depuis le fichier PXF spécifié :

- Le/s fichier/s HTML (.html) utilisant les fichiers XML et XSLT-pour-HTML spécifiés dans le PXF, ou utilisant les fichiers XML alternatifs

10.2 Utiliser RaptorXML

Altova RaptorXML est la troisième génération d'Altova, XML hyper rapide et Processeur XBRL. **Le traitement XBRL est disponible uniquement dans le RaptorXML + XBRL Server.** Il a été optimisé pour s'adapter aux tout derniers standards et environnement de calcul parallèles. Conçu pour fonctionner sur plusieurs plateformes, le moteur profite de l'ubiquité actuelle des ordinateurs multicœurs pour fournir un traitement ultra-rapide des données XML et XBRL.

RaptorXML est disponible dans deux éditions :

- L'édition RaptorXML Server, qui peut être accédée par un réseau et peut transformer de multiples fois en une seule fois.
- L'édition RaptorXML+XBRL Server, qui peut être accédée par un réseau, peut transformer de multiples fichiers en une seule fois, et de plus prendre en charge la validation XBRL.

Pour plus d'informations sur RaptorXML, voir le [site web Altova](#).

Cas d'utilisation typiques

La fonction de RaptorXML qui serait la plus pertinente pour les utilisateurs StyleVision est la fonction de transformation XSLT. Généralement, cette fonction serait utilisée comme suit

1. Une feuille de style XSLT est générée depuis une SPS avec la commande [Fichier | Enregistrer les fichiers générés](#)⁴⁶⁹. Notez que RaptorXML ne peut pas être utilisé pour générer des feuilles de style XSLT depuis un fichier SPS.
2. La feuille de style XSLT générée est utilisée pour transformer des documents XML avec RaptorXML. Avec RaptorXML, vous pouvez générer les sorties HTML.

Bénéfices de RaptorXML

Les avantages d'utiliser RaptorXML sont les suivants :

- RaptorXML offre une validation très rapide et une transformation XSLT, et est très utile pour cette raison en ce qui concerne la gestion de grands fichiers.
- Utilisation facile de la ligne de commande, COM, Java et interfaces .NET.
- [Automatisation et planification](#)⁴¹⁹ avec l'utilisation de fichiers batch et les processus de planification comme le processus des Tâches prévues (Scheduled Tasks) de Windows.

Pour une description sur comment RaptorXML peut être utilisé pour automatiser la production des documents de sortie (comme HTML) depuis les documents de source XML, voir la section [Comment automatiser le processus](#)⁴¹⁹.

Pour plus des informations supplémentaires et plus détaillées sur RaptorXML, y compris l'utilisation de COM de RaptorXML, Java, et les interfaces .NET, voir la documentation [documentation de l'utilisateur RaptorXML](#).

10.2.1 Sortie PDF

La génération de la sortie PDF depuis un document XML requiert deux étapes :

1. Le document XML est transformé par une feuille de style XSLT. Un moteur de transformation XSLT (comme celui de RaptorXML) est utilisé pour cette transformation. Le résultat est un document FO.
2. Le document FO est traité par un processeur FO (tel que FOP d'Apache) pour générer la sortie PDF. StyleVision peut être défini pour passer le résultat FO d'une transformation XSLT dans un processeur FO. Dans StyleVision, le résultat de la génération d'un PDF est affiché dans la fenêtre d'Aperçu PDF ou peut être enregistré comme fichier (via la commande [Fichier | Enregistrer les fichiers générés](#)⁴⁶⁹).

RaptorXML et PDF

Puisque RaptorXML ne fournit pas de paramètres pour diriger la sortie FO d'une transformation XSLT vers un processeur FO, vous aurez un document FO comme résultat de l'étape de transformation XSLT (la première étape du processus de génération du PDF à deux étapes).

Le document FO doit maintenant être transmis à un processeur FO pour un traitement de deuxième étape du FO en PDF. Les instructions pour exécuter cette étape varient selon le processeur qui est utilisé. Par exemple, dans le cas d'un processeur FOP Apache, la commande simple suivante peut être utilisée pour identifier le document FO d'entrée et spécifier le nom et l'emplacement du document PDF de sortie :

```
fop -fo input.fo -pdf output.pdf
```

FOP propose d'autres paramètres, et ceux-ci sont recensés dans la [référence de l'utilisateur FOP](#).

FOP et XSLT

Une option FOP vous permet de spécifier un fichier d'entrée XML d'entrée, un fichier d'entrée XSLT et un fichier de sortie PDF :

```
fop -xml input.xml -xslt input.xslt -pdf output.pdf
```

Dans cette situation, FOP utilise son moteur XSLT intégré pour exécuter la transformation à première étape de XML-à-FO. Elle transmet ensuite le résultat du document FO au FOP pour le traitement à deuxième étape de FO-en-PDF.

Vous devriez être conscients, toutefois, que le moteur FOP intégré peut ne pas prendre en charge toutes les fonctions XSLT qui sont pris en charge par StyleVision et RaptorXML. En conséquence, il pourrait y avoir des erreurs si une feuille de style XSLT générée par StyleVision est spécifiée comme entrée pour une transformation XML utilisant le moteur XSLT intégré de FOP. Dans de tels cas, utilisez le moteur XSLT du serveur RaptorXML(+XBRL) pour transformer en FO, et puis fournissez le fichier FO au FOP pour le transformer en PDF.

Traitement batch vers le PDF

Une manière simple et rapide pour générer un PDF en utilisant RaptorXML pour la transformation XSLT à première étape et FOP pour le traitement FO en deuxième étape serait d'écrire un fichier batch qui combine les deux commandes. Par exemple :

```
raptorxmlserver xslt --input=Test.xml --output=Test.fo Test.xslt
fop -fo input.fo -pdf output.pdf
```

La première commande appelle RaptorXML et produit `test.fo` comme sortie. La deuxième commande transmet `test.fo` au processeur FOP, qui génère le fichier PDF `test.pdf`. Pour plus d'information sur le

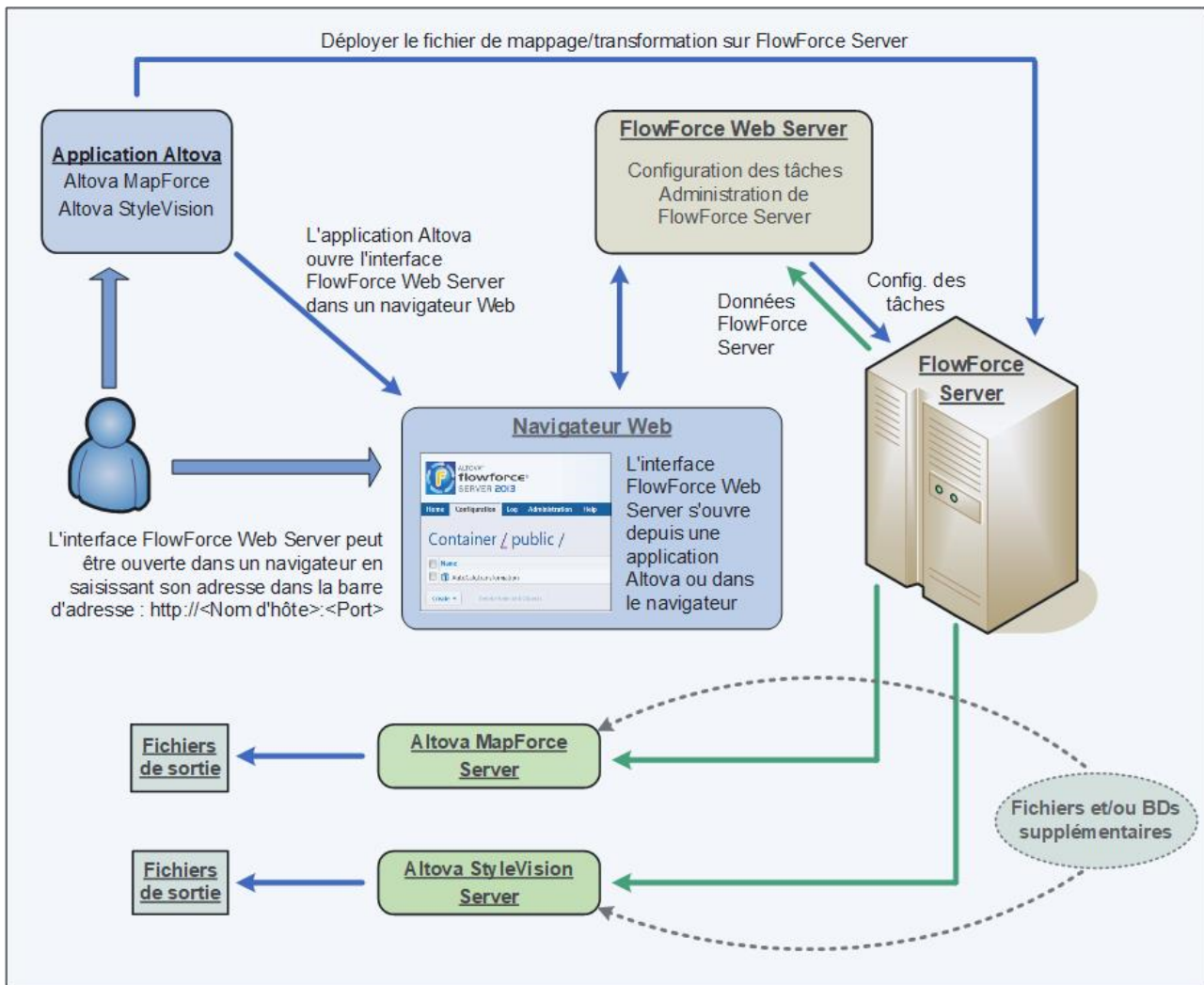
traitement batch et comment les fichiers batch peuvent être utilisés pour automatiser les processus, voir la section suivante : [Comment automatiser le traitement](#)⁴¹⁹.

10.3 Automatisation avec FlowForce Server

Les transformations peuvent être automatisées sur un réseau à l'aide du serveur FlowForce d'Altova, disponible sur les systèmes Windows et Linux. L'exemple fonctionne comme suit :

1. Depuis StyleVision, un [fichier PXF](#)⁴⁰⁴ est [déployé sur FlowForce Server](#)⁴⁷⁰ (avec la commande [Fichier | Déployer vers FlowForce](#)⁴⁷⁰) comme un fichier `.transformation`. Le fichier `.transformation` contient tous les fichiers et informations requis pour exécuter les transformations comme conçues dans le SPS. (Dans le diagramme ci-dessous, le déploiement est représenté par la ligne du connecteur circulant en parallèle au niveau supérieur.)
2. Une fois que le fichier `.transformation` a été déployé vers FlowForce Server, les tâches peuvent être créées dans FlowForce qui utilisent le fichier `.transformation` pour générer les transformations selon les déclencheurs spécifiés dans la définition de la tâche. (Un déclencheur pourrait, par exemple, être un moment spécifique au quotidien.) Les tâches de FlowForce sont créées dans l'interface de FlowForce Web Server (affichée dans le centre du diagramme ci-dessous), qui peuvent être accédées depuis StyleVision ou par le biais de l'adresse HTTP. Pour information sur la création des tâches FlowForce, voir la [documentation FlowForce](#).
3. Au moment de l'exécution, Altova FlowForce Server passe les instructions de transformation et les fichiers pertinents au StyleVision Server, qui effectue ensuite de la transformation (*voir le diagramme ci-dessous*).

Le rôle de StyleVision Server dans le flux de travail FlowForce est affiché dans le diagramme ci-dessous. (Le rôle de MapForce Server dans le flux de travail est également affiché puisque les tâches de FlowForce peuvent être créées qui envoient les mappages MapForce d'Altova au serveur MapForce d'Altova pour exécution.)



Outre le fait d'être invoqué par une tâche FlowForce, StyleVision Server peut aussi être invoqué par le biais de sa ligne de commande. L'utilisation est décrite dans la [documentation de StyleVision Server](#).

10.4 Comment automatiser le traitement

Un fichier batch (un fichier texte enregistré avec l'extension de fichier `.bat`) contient une séquence de commandes qui sera exécutée depuis la ligne de commande. Lorsqu'un fichier batch est exécuté, chaque commande dans un fichier batch sera exécutés en retour, en commençant par le premier et traitant par par le biais de la séquence. Un fichier batch est pour cette raison utile dans les situations suivantes :

- Exécuter une série de commandes automatiquement (*voir ci-dessous*).
- Créer une chaîne de commandes de traitement, là où la commande requiert l'entrée produite par une commande précédente. (Par exemple, un fichier XML produit comme sortie d'une transformation est utilisé comme entrée de la transformation suivante.)
- Planification d'une séquence de tâches à être exécutées à un moment donné.

Fichier batch avec une séquence de commandes

Une séquence de commande à être exécutée est saisie comme suit :

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

Lorsque le fichier est traité, les commandes sont exécutées et les fichiers générés. Le fichier batch file ci-dessus utilise StyleVision pour générer trois fichiers XSLT depuis un fichier SPS.

11 Commandes de menu et Référence

Cette section contient une description complète des barres d'outils symboles de Mode Design et des commandes de menu de StyleVision. Elle est divisée en plusieurs parties :

- Explication des [symboles utilisés en mode Design](#) ⁴²¹.
- Description du [dialogue Éditer expression XPath](#) ⁴²⁶.
- Description de toutes les [barres d'outils avec leurs icônes](#) ⁴⁴³, ainsi qu'une explication sur la personnalisation de l'affichage des barres d'outils.
- Toutes les commandes de menu.

Tandis que la section Référence de l'utilisateur contient une description des commandes individuelles, les mécanismes derrière les diverses fonctions de StyleVision sont expliqués en détail dans les sections correspondantes. Les mécanismes ont été organisés dans les groupes suivants :

- [Contenu du fichier SPS](#) ¹⁰⁹
- [Structure du fichier SPS](#) ¹⁸⁵
- [Fonctionnalités avancées du fichier SPS](#) ²⁶⁰
- [Présentation du fichier SPS](#) ³²⁸
- [Fonctionnalités supplémentaires du fichier SPS](#) ³⁶²

Voir aussi

- [Interface utilisateur](#) ²⁶
- [Tutoriel de démarrage rapide](#) ⁵⁴

11.1 Symboles du Mode Design

Une conception SPS contiendra généralement plusieurs types de composants. Chaque composant est représenté dans la conception par le symbole spécifique. Ces symboles sont recensés ci-dessous et organisés en les groupes suivants :

- [Nœuds dans le document XML](#) ⁴²¹
- Contenu de document XML
- Dispositifs de saisie de données
- Formats prédéfinis
- Objets XPath
- Objets URI

Chacun de ces types de composant peut :

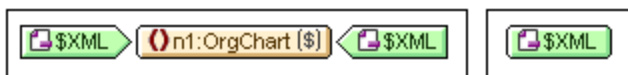
- être déplacé utilisant 'glisser et déposer' ;
- être coupé, copié, collé et supprimé utilisant (i) les commandes dans [Éditer menu](#) ⁴⁷⁶, ou (ii) les raccourcis standard Windows pour ces commandes ;
- avoir du formatage y appliqué ;
- voir un menu contextuel s'ouvrir lorsque vous y cliquez avec la touche de droite de la souris.

Nœuds dans le document XML

Les nœuds d'élément et d'attribut dans le document XML sont représentés dans le document de conception SPS par des balises. Chaque nœud a une balise de début et une balise de fin. Double-cliquer soit sur la balise de début, soit sur la balise de fin réduit ce nœud. Lorsqu'un nœud est réduit, tout son contenu est masqué. Double-cliquer sur un nœud réduit l'élargit et affiche son contenu.

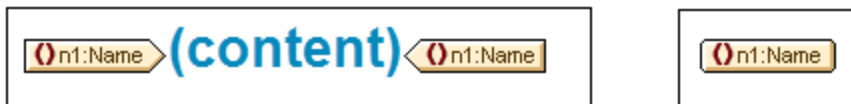
Les types suivants de nœud sont représentés :

- **Nœud du document**



Le **nœud du document** (indiqué par `$XML`) représente le document XML en entier. Il est indiqué par une balise verte `$XML` quand la source de schéma est associée à un document XML, et avec `$DB` lorsque la source de schéma est associée à une BD. Le nœud du document dans la capture d'écran à gauche est élargi et contient l'élément `OrgChart`, qui est réduit. Le nœud du document dans la capture d'écran à droite est réduit ; son contenu est masqué.

- **Nœud d'élément**



Un **nœud d'élément** est inséré ensemble avec tous ses éléments ancêtre si les ancêtres ne sont pas présents au point d'insertion. Dans la capture d'écran ci-dessus, le nœud de l'élément `Name` est affiché étant élargi (*gauche*) et réduit (*droite*).

- **Nœud d'attribut**



Un **nœud d'attribut** est inséré ensemble avec tous ses éléments ancêtre si les ancêtres ne sont pas présents au point d'insertion. Les noms d'attribut contiennent le préfixe `=`. Dans la capture d'écran ci-dessus, le nœud de l'attribut `href` est affiché étant élargi (*gauche*) et réduit (*droite*).

Les nœuds sont inclus dans la conception comme modèles de nœud. Pour obtenir des informations sur les différents types de modèles pouvant être inclus dans la conception, voir la section, [Modèles et fragments de design](#) ²³⁴.

Contenu du document XML

Le contenu du document XML est représenté par deux espaces réservés :

- `(contents)`
- `(rest-of-contents)`

L'espace réservé `contents` représente les contenus d'un seul nœud. Tout le contenu du texte du nœud est la sortie. Si le nœud est un nœud attribut ou un nœud d'élément `text-only`, la valeur du nœud est la sortie. Si le nœud est un nœud d'élément qui contient du contenu mixte ou du contenu `element-only`, le contenu de texte de tous les descendants est la sortie. Dans les termes XSLT, l'espace réservé `contents` est équivalent à l'élément `xsl:apply-templates` avec son attribut `select` défini pour ce nœud.

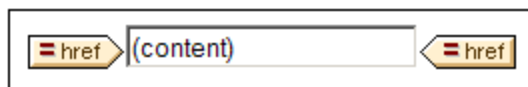
Note : lorsqu'il est appliqué à un nœud d'élément, l'espace réservé `contents` ne sort pas les valeurs d'attributs de cet élément. Pour sortir les nœuds d'attribut, vous devez explicitement inclure l'attribut dans le modèle (principal ou global).

L'espace réservé `rest-of-contents` applique des modèles au reste des éléments enfant du nœud actuel. Le modèle appliqué pour chaque élément enfant dans ce cas sera soit un modèle global (si un modèle est défini pour cet élément), soit le modèle défaut pour les éléments (qui sort tout simplement du texte des éléments `text-only`, et applique les éléments aux éléments enfant). Par exemple, prenez en compte un élément `book`, qui contient des éléments enfant : `title`, `author`, `isbn` et `pubdate`. Si la définition de `book` spécifie que seul l'élément enfant `title` est la sortie, alors aucun des autres éléments enfant (`author`, `isbn` et `pubdate`) ne seront sortis lorsque cette définition est traitée. Si, toutefois, la définition de `book` inclut l'espace réservé `rest-of-contents` après la définition pour l'élément `title`, alors pour chaque autre élément enfant (`author`, `isbn` et `pubdate`), un modèle global (si aucune n'existe pour cet élément), ou le modèle par défaut pour les éléments, sera appliqué.

Dispositifs de saisie de données

Afin d'aider l'utilisateur Authentic View à modifier correctement le document XML et à saisir des données valides, des dispositifs de saisie de données peuvent être intégrés à la conception. Vous pouvez attribuer un des appareils d'entrée des données suivants à un nœud :

- **Champs d'entrée (ligne individuelle ou multi-ligne)**



- **Listes de choix**



- **Cases à cocher**



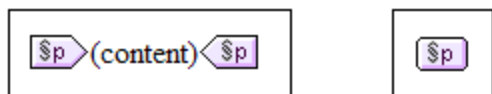
- **Boutons radio**



Ces balises peuvent être réduites et élargies en double-cliquant sur la balise, respectivement. Pour une description détaillée sur l'utilisation de chacun de ces appareils de saisie des données, voir [Appareils d'entrée des données](#) ¹⁵⁸.

Formats prédéfinis

Les formats prédéfinis sont affichés en balises mauves, qui peuvent être élargies/réduites en double-cliquant dessus.

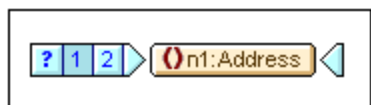


La capture d'écran ci-dessus affiche les balises pour le format prédéfini `p` (`para`), *expanded (à gauche)* et *collapsed (à droite)*. Pour appliquer un format prédéfini, mettez en surbrillance les items autour desquels le format prédéfini doit apparaître (en cliquant sur un composant et/ou en marquant du texte), et [insérez le format prédéfini](#) ³²⁹.

Objets XPath

StyleVision présente deux mécanismes qui utilisent les expressions XPath :

- **Modèles conditionnels**



Les balises **Condition** sont bleues. La balise de début contient des cellules. La cellule la plus à gauche contient un point d'interrogation. D'autres cellules contiennent chacune soit (i) un nombre, en

commençant par le chiffre un, pour chaque condition *when* ; et/ou (ii) un astérisque pour la condition optionnelle *otherwise*. Une branche Condition peut être sélectionnée en cliquant dessus. Le nombre de la branche condition sélectionnée est en surbrillance dans la balise de début, et le modèle pour cette branche est affiché (à l'intérieur des balises de début et de fin de la condition). L'expression XPath pour la branche de condition sélectionnée est également mise en évidence dans l'arborescence de conception. Notez que les balises de conditions ne peuvent pas être élargies/réduites.

- **Calculs automatiques**



Les **Calculs automatiques** sont représentés dans le Mode Design par l'objet `=(AutoCalc)` (voir la capture d'écran ci-dessus). L'expression XPath pour le Calcul automatique sélectionné est mise en évidence dans l'arborescence de conception. Le dialogue pour éditer le Calcul automatique est [accédé par le biais de la barre latérale des Propriétés](#) ²⁶¹.

Objets URI

Trois objets basés sur l'URI peuvent être insérés dans une conception :

- **Images**

Si une image est insérée dans la conception SPS et est accessible par StyleVision, elle devient visible dans le Mode Design. Si elle ne peut pas être accédée, sa place est marquée dans la SPS par un espace réservé de l'image.

- **Signets (Ancres)**



Les balises **Signets** sont jaunes et indiquées par le caractère `A` (capture d'écran ci-dessus). Un signet est créé avec la commande **Insérer | Insérer Signet**, et peut être vide ou contenir du contenu. Le contenu doit toujours être inséré après avoir créé l'ancre. Les balises d'ancre peuvent être élargies (capture d'écran ci-dessus gauche) ou réduites (capture d'écran ci-dessus droite).

- **Liens**



Les balises **Lien** sont jaunes et indiquées par le caractère `A` (capture d'écran ci-dessus). Un lien est créé avec la commande **Insérer | Hyperlien**. L'objet autour duquel le lien est créé peut être inséré dans la conception avant ou après la création du lien. Si un item doit être créé comme lien, il doit être sélectionné et le lien créé autour de lui. Les balises Lien peuvent être élargies (capture d'écran ci-dessus à gauche) ou réduites (capture d'écran ci-dessus à droite).

 **Voir aussi**

- [Barre d'outils](#) 443
- [Barres latérales de design](#) 32
- [Procédures d'édition du contenu](#) 109

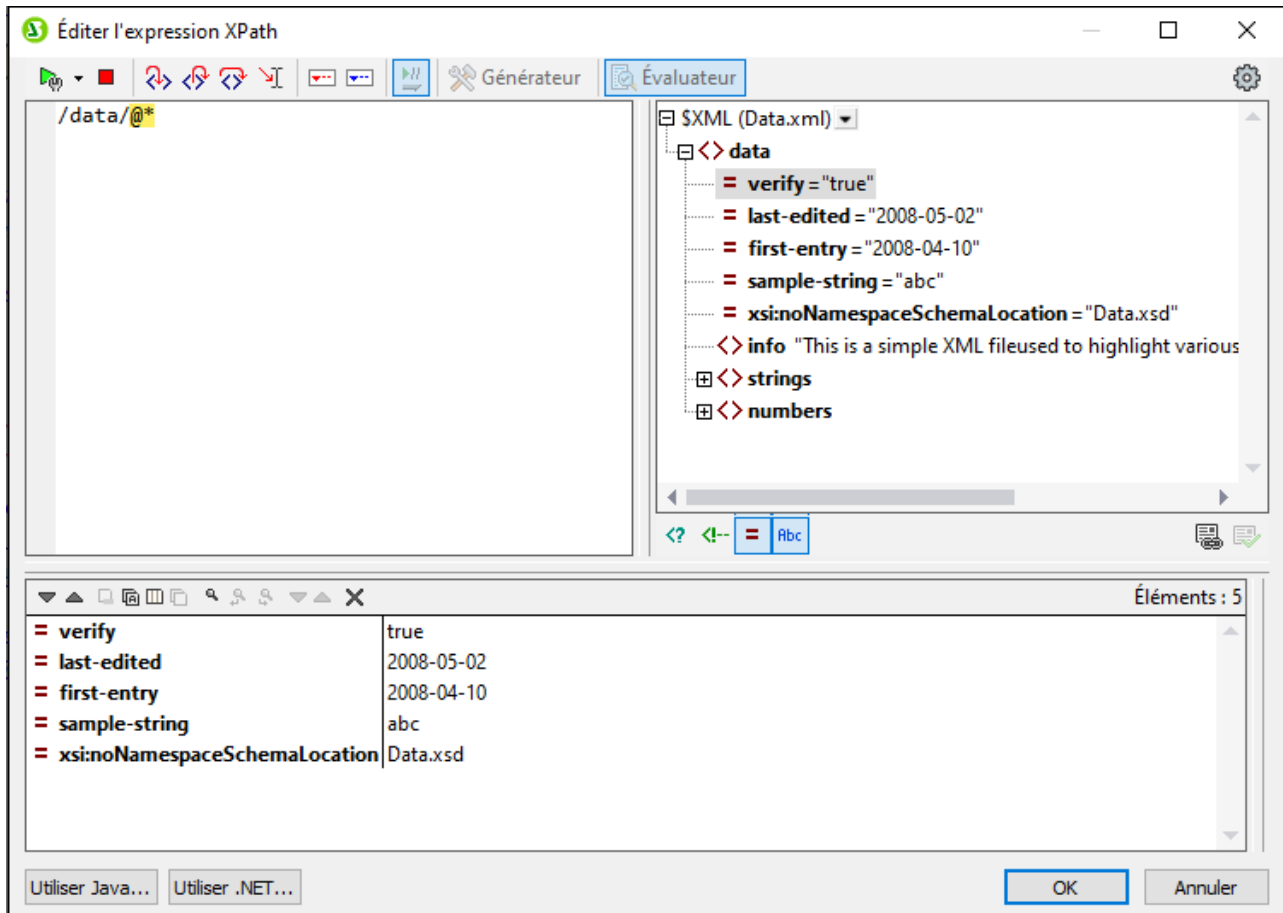
11.2 Éditer le dialogue de l'expression XPath

Le dialogue **Éditer l'expression XPath** (*capture d'écran ci-dessous*) est utilisé pour créer, tester, et éditer les expressions XPath. Il est accessible à tous les endroits dans le Mode Design où l'expression XPath peut être saisie, comme pour la saisie des expressions pour le [traitement conditionnel](#)²⁶⁶ ou les valeurs de [Styles](#)⁴⁶ et des [Propriétés](#)⁴⁸.

Le dialogue prend automatiquement en charge la **version XPath** qui correspond à la [version XSLT](#)⁹⁹ de la SPS (XPath 1.0 pour XSLT 1.0 ; XPath 2.0 pour XSLT 2.0 ; et XPath 3.1 pour XSLT 3.0). Pour changer de version XPath, basculez la [version XSLT de la SPS](#)⁹⁹.

Mise en page du dialogue

Le dialogue Éditer l'expression XPath contient les volets suivants (*voir la capture d'écran ci-dessous*) : (i) un volet d'expression (*en haut à gauche*) ; (ii) un volet Sources (*en haut à droite*) ; (iii) un volet de Résultats (*en bas*). Dans le Mode Générateur, le volet de Résultats est augmenté avec des volets d'assistant à la saisie.

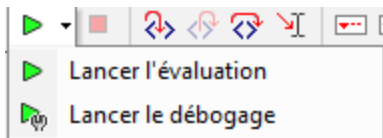


Mode d'évaluation et mode de débogage

Le dialogue Éditer l'expression XPath a deux modes :

- Le [Mode d'évaluation](#) ⁴²⁷, dans lequel une expression XPath est évaluée par rapport au/x fichier/s de travail XML attribué/s. L'expression est saisie dans le **Volet Expression**, et le résultat est affiché dans le **Volet Résultats**. Vous pouvez cliquer sur les nœuds dans le résultat pour aller à ce nœud dans le volet Sources du dialogue.
- Le [Mode Débogage](#) ⁴³¹, dans lequel vous pouvez déboguer une expression XPath en tant qu'elle s'applique au/x fichier/s de travail XML attribué/s. 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/le 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.

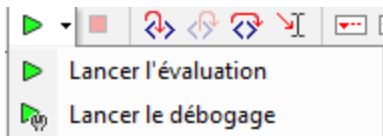
Générateur d'expression XPath/XQuery

Dans les deux modes, le [Générateur d'expression](#) ⁴³⁸ peut être utilisé pour vous aider à construire des expressions syntaxiquement correctes. Allumer ou éteindre le [Générateur d'expression](#) ⁴³⁸ avec la touche

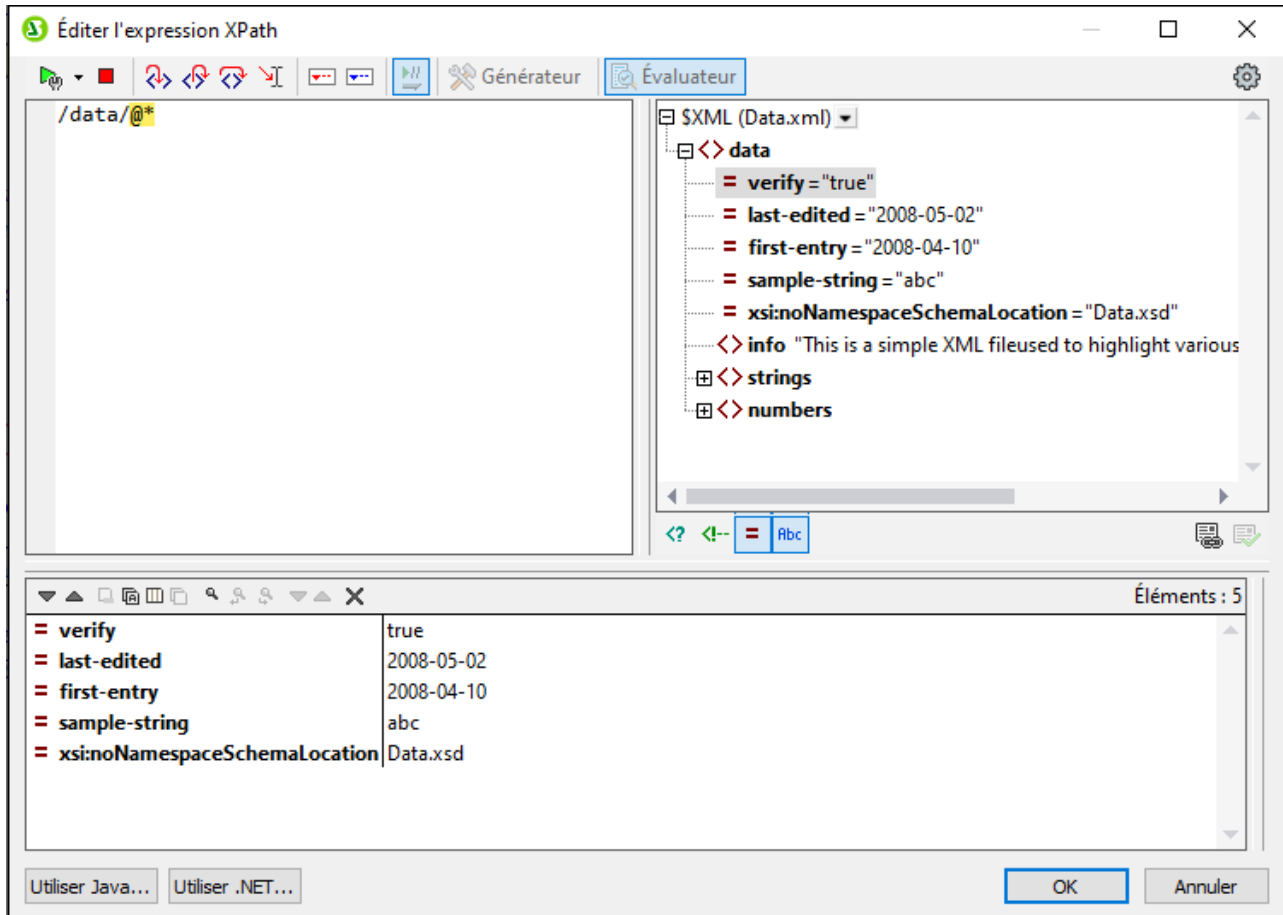
Mode générateur dans la barre d'outils principale .

11.2.1 Évaluateur

Sélectionnez le Mode d'évaluation en sélectionnant **Lancer évaluation** dans le menu déroulant **Lancer Évaluation/Débogage** (*voir capture d'écran ci-dessous*).



Dans le Mode d'évaluation, cliquez sur le bouton **Évaluateur** (*voir la capture d'écran ci-dessous*). L'évaluateur a les volets suivants (*voir capture d'écran ci-dessous*) : (i) un volet d'expression (*en haut à gauche*); (ii) un volet Sources (*en haut à droite*); (iii) un volet de Résultats (*en bas*).



L'expression XPath et son évaluation

L'expression XPath est saisie dans le volet Expression. Les résultats de l'évaluation sont affichés dans le volet *Résultats* (voir capture d'écran ci-dessus).

Notez les points suivants :

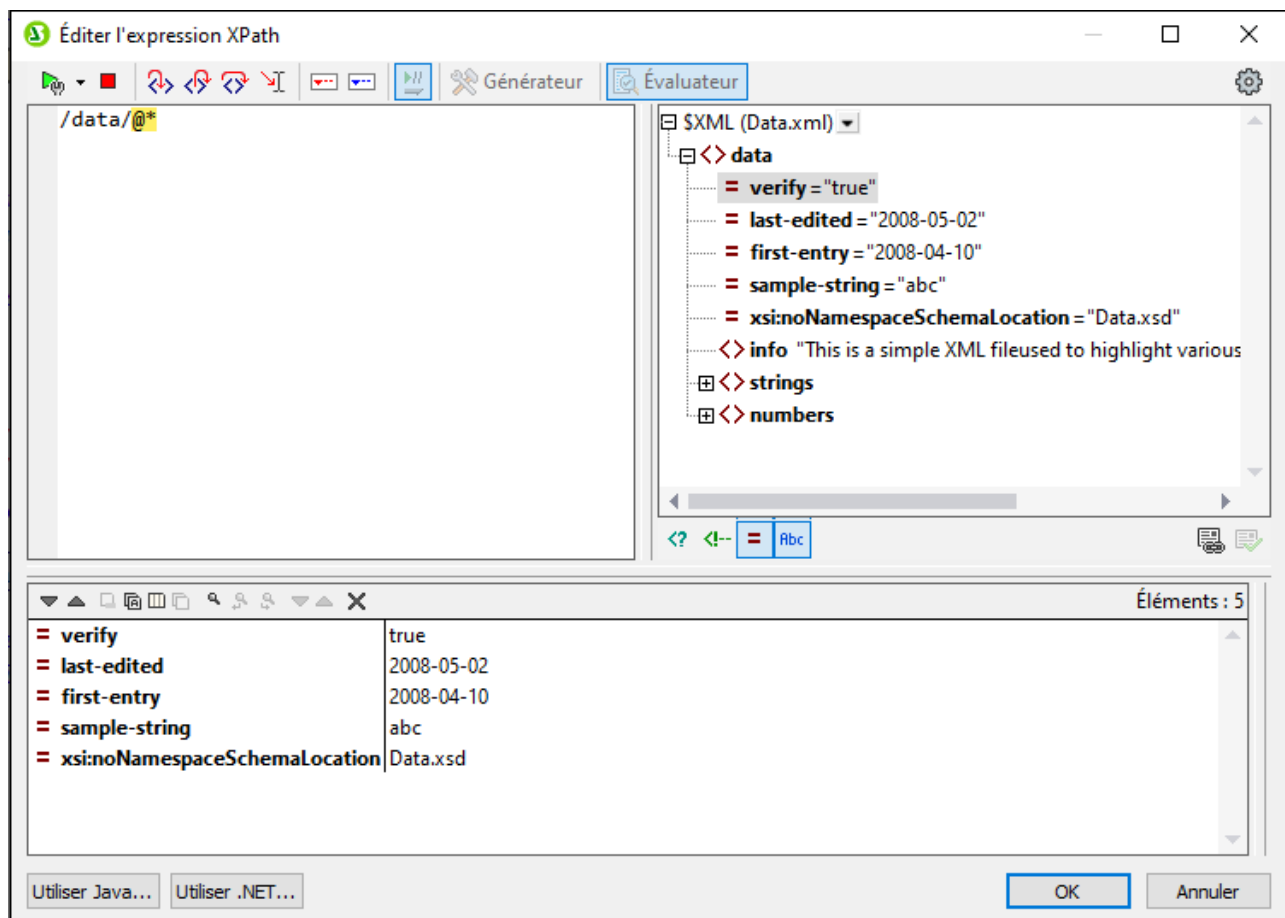
- Pour qu'une évaluation soit évaluée par rapport à un fichier XML, ce fichier doit être attribué comme [fichier de travail XML d'une des sources de la SPS](#) ⁹⁷.
- Les résultats peuvent être affichés même si vous saisissez l'expression (sélectionnez l'icône **Évaluer** en tapant dans la barre d'outils), ou ils peuvent être affichés lorsque vous cliquez sur le bouton de la barre d'outils **Lancer Évaluation/Débugage (F5)** (situé en haut à gauche dans la barre d'outils).
- Pour saisir l'expression du localisateur XPath d'un nœud dans une arborescence source, double-cliquez sur ce nœud dans le volet Sources.
- Dans le volet Sources, vous pouvez activer/désactiver l'affichage des : (i) instructions de traitement, (ii) commentaires, (iii) attributs et (iv) éléments. Faites ceci par le biais des boutons en-dessous du volet Sources.
- Le nœud contextuel est celui du composant de design à l'intérieur duquel l'expression est créée. Pour définir un autre nœud en tant que nœud contextuel de l'expression : (i) sélectionnez le nœud dans le volet Sources, et (ii) cliquez sur **Définir l'évaluation de contexte** (situé ci-dessous dans le volet Sources). Pour enregistrer ce nœud contextuel pour l'expression, cliquez sur le bouton de commande

du toggle **Se souvenir du contexte d'évaluation** (situé en-dessous du volet Sources). Notez, toutefois, que le nœud contextuel actuel pour l'expression sera le nœud contextuel du composant de design actuel, et ceci est le nœud contextuel qui sera utilisé au moment de l'exécution.

- Vous pouvez utiliser les fonctions des langages de programmation Java et .NET dans l'expression XPath. Les boutons **Utiliser Java** et **Utiliser .NET** en bas du dialogue affichent des boîtes d'info avec des explications sur comment utiliser les fonctions d'extension Java et .NET dans les expressions XPath. Pour plus d'information sur le sujet, voir la section [Fonctions d'extension](#) ⁵⁵⁸ de cette **documentation**.
- Pour créer l'expression sur plusieurs lignes (pour une meilleure lecture), utilisez 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. **Veillez 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 procéder comme suit : (i) Placez le curseur au point dans l'expression XPath où vous souhaitez saisir le chemin de localisation ; (ii) dans l'arborescence de Sources, double-cliquez le nœud que vous souhaitez cibler. 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.

Volet Résultats

Le volet Résultats est affiché dans la capture d'écran ci-dessous, en bas. Veuillez noter qu'il dispose de sa propre barre d'outils.



Le volet Résultats propose les fonctions suivantes :

- La liste de résultat 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 le nœud de texte) - à l'opposé des littéraux générés par l'expression - en cliquant sur ce nœud dans le volet Résultats, ceci met en surbrillance le nœud correspondant dans le document XML dans l'arborescence de Sources.
- 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).

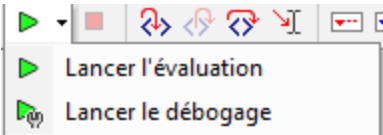
Barre d'outils du volet Résultats

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
<i>Suivant, Précédent</i>	Sélectionne, respectivement, l'item suivant et précédent dans la liste de résultats
<i>Copier la ligne de texte sélectionnée dans le presse-papiers</i>	Copie la colonne de valeur de l'item de résultat sélectionné dans le presse-papiers. Afin de copier toutes les colonnes, activer la commande <i>La copie contient toutes les colonnes</i> (voir ci-dessous)
<i>Copier tous les messages dans le presse-papiers</i>	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
<i>La copie contient toutes les colonnes</i>	Passes entre la copie (i) de toutes les colonnes, ou (ii) uniquement la colonne de valeur. Le séparateur de colonne est un espace unique
<i>Recherche</i>	Ouvre un dialogue <i>Chercher</i> pour chercher un string, y compris des caractères spéciaux, dans la liste de résultat
<i>Chercher précédent</i>	Cherche l'occurrence précédente du terme qui a été saisi en dernier dans le dialogue <i>Chercher</i>
<i>Trouver suivant</i>	Cherche l'occurrence suivante du terme qui a été saisi en dernier dans le dialogue <i>Chercher</i>
<i>Élargir avec des enfants</i>	Élargit l'item sélectionné et tous ses descendants
<i>Comprime avec les enfants</i>	Comprime l'item sélectionné et tous ses descendants
<i>Supprimer</i>	Supprime la liste de résultats

11.2.2 Débogueur

Le débogueur vous permet de déboguer une expression XPath dans le contexte d'un [fichier de travail XML](#) ⁹⁷. Pour accéder au débogueur, sélectionnez **Lancer débogage** dans le menu déroulant **Lancer Évaluation/Débogage** (*capture d'écran ci-dessous*). Le mode Débogueur est activé. Vous pouvez ensuite alterner entre le Générateur (pour obtenir de l'aide avec la génération d'expression) et l'Évaluateur (pour déboguer l'expression). Pour lancer le débogage, cliquer sur **Lancer Évaluation/Débogage (F5)**



Une fois que vous avez saisi une expression, vous pouvez commencer à déboguer en cliquant sur **Lancer Évaluation/Débogage (F5)** (après vous être assuré que vous êtes dans le mode Débogueur).

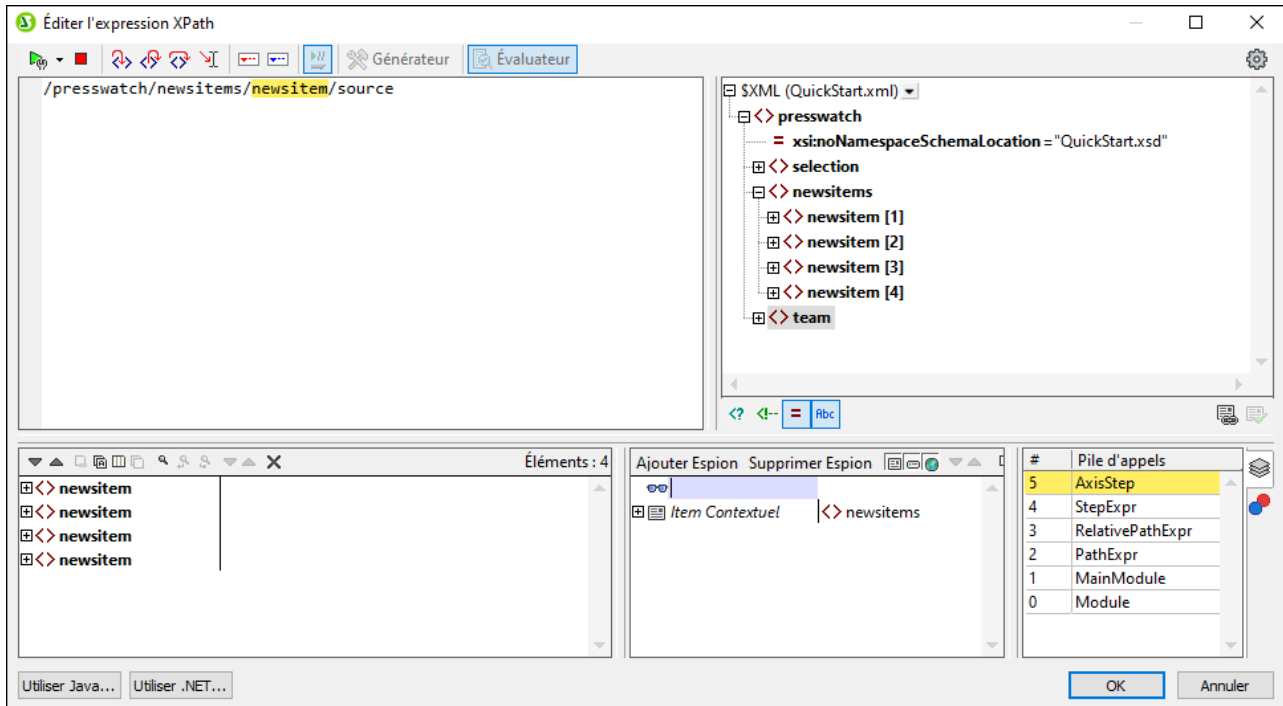
Touches pour le Mode Débogage

	Lancer Évaluation/Débogage (F5)	Lance le débogueur
	Passer au générateur	Passes au mode Générateur d'expression qui vous propose des assistants à la saisie sensibles au contexte pour vous assister dans la construction d'expressions
	Évaluation sur saisie	Active l'évaluation des expressions alors que l'expression est en cours de saisie

Mise en page du Mode Débogage

Dans le Mode Débogage, deux volets supplémentaires sont ajoutés au volet Résultats (*voir capture d'écran ci-dessous*):

- le volet Pile d'appel et Points de débogage, dont chacun a un onglet séparé dans le volet
- le volet Variables et Expressions Watch ; les expressions watch et les variables sont affichées dans le même volet.



Le Mode Débogueur propose les fonctions suivantes :

- Vous permet de pénétrer 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 **Entrer dans (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 dans les endroits où vous souhaitez interrompre l'évaluation et contrôler les résultats à ces points. 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 point de trace. L'évaluation ne s'interrompt pas (sauf aux points d'arrêt), mais les résultats du point de trace seront affichés dans une liste dans le volet Résultats.
- Des expressions Watch peuvent être utilisées pour contrôler l'information (comme des données de document ou des aspects de l'évaluation). Cela est particulièrement utile au niveau des points d'arrêt.
- Les variables qui sont exposées, y compris leurs valeurs, sont affichées dans le volet Variables et Expressions Watch.
- Les appels du processeur d'une étape d'évaluation sont affichés dans l'onglet Pile d'appel du volet Pile d'appel et Points de débogage.
- Si des points d'arrêt et des points de trace ont été définis, alors ils seront affichés dans l'onglet Points de débogage du volet Pile d'appel et Points de débogage.

Pour plus d'information à propos de ces fonctions, voir leur description ci-dessous.

Lancer le débogueur

Les étapes générales pour déboguer une expressions XPath sont les suivantes :

1. Saisir l'expression XPath dans le volet d'expression.
2. Définir tous les points d'arrêt ou les point de trace que vous souhaitez. Un point d'arrêt est un point au niveau duquel l'évaluation est interrompue. Un point d'arrêt est un point dans l'évaluation qui est enregistré ; les points de trace fournissent donc un chemin traçable des résultats d'évaluation.
3. 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 débogueur** plusieurs fois pour passer dans chaque point d'arrêt jusqu'à la fin de l'évaluation.
4. Utiliser la fonction Pas à pas détaillé/sortant/principal pour se rendre étape par étape par le biais de l'évaluation.

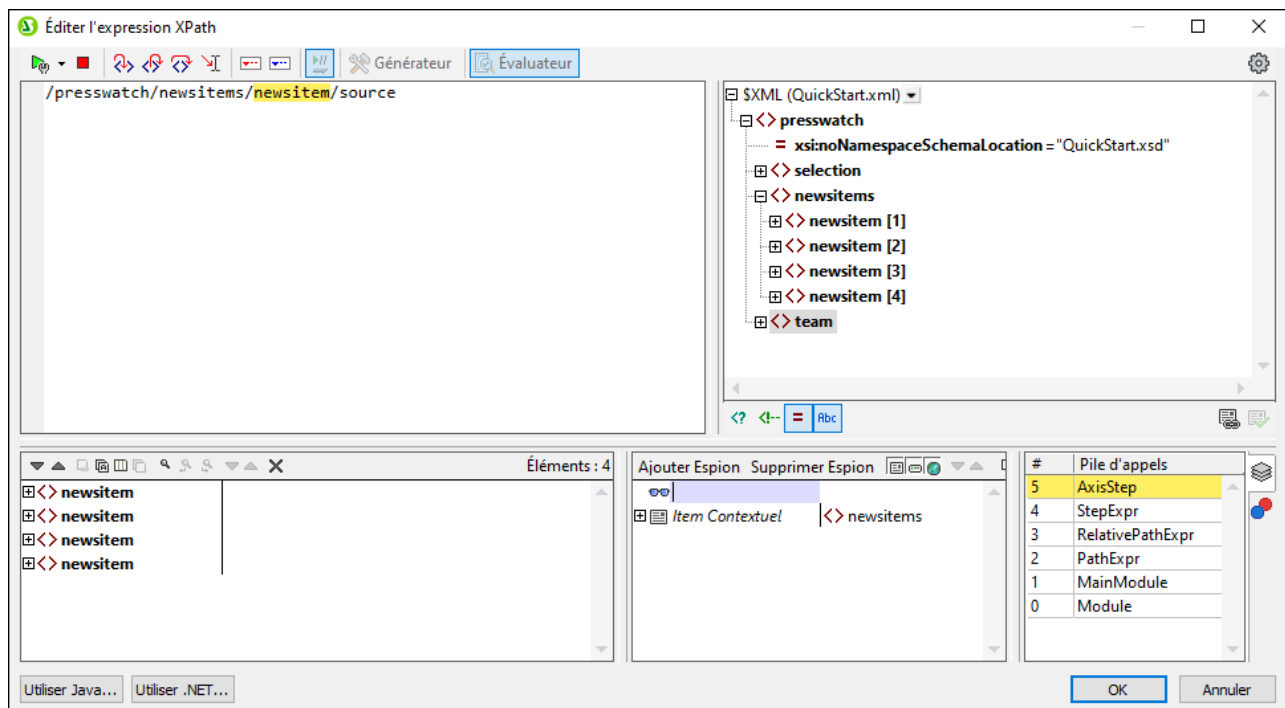
Boutons de débogage

	Lancer débogueur (F5)	Lance le débogueur. L'évaluation se rend directement à la fin, en s'arrêtant uniquement pour les points d'arrêt
	Arrêter le débogueur (Shift+F5)	Termine l'évaluation et arrête le débogueur
	Pas à pas détaillé (F11)	Passe dans l'évaluation, une étape à la fois.
	Pas à pas sortant (Shift+F11)	Sort de l'étape d'évaluation actuelle, et va dans l'étape parent
	Pas à pas principal (Ctrl+F11)	Passe par-dessus les étapes descendantes
	Insère/Supprime le point d'arrêt (F9)	Insère/Supprime un point d'arrêt au niveau de l'étape d'expression dans laquelle vous avez placé le curseur
	Insère/Supprime Point de trace (Shift+F9)	Insère/Supprime un point de trace au niveau de l'étape d'expression dans laquelle vous avez placé le curseur

Entre, sort et passe par-dessus les étapes d'évaluation

La fonction *Pas à pas détaillé* vous permet de passer étape par étape à travers l'évaluation. Chaque clic de cette commande vous fait passer dans l'étape suivante de l'évaluation ; l'étape actuelle est soulignée par le marquage dans l'expression (voir *capture d'écran ci-dessous*). La fonction *Pas à pas sortant* vous amène à une étape dans un niveau supérieur que l'étape actuelle, alors que la fonction *Pas à pas principal* passe par-dessus les étapes de niveau inférieur et vous amène à l'étape suivante dans le même niveau. Vous pouvez essayer la fonction *Pas à pas* en utilisant l'expression affichée dans la capture d'écran ci-dessous et en cliquant sur les trois touches *Pas à pas* pour tester leur fonctionnement.

La capture d'écran ci-dessous montre l'évaluation lorsque le traitement a été interrompu lorsque vous avez atteint l'étape de localisation **newsitem**. À cette étape, le résultat affiche les quatre nœuds **newsitem**.



Points d'arrêt

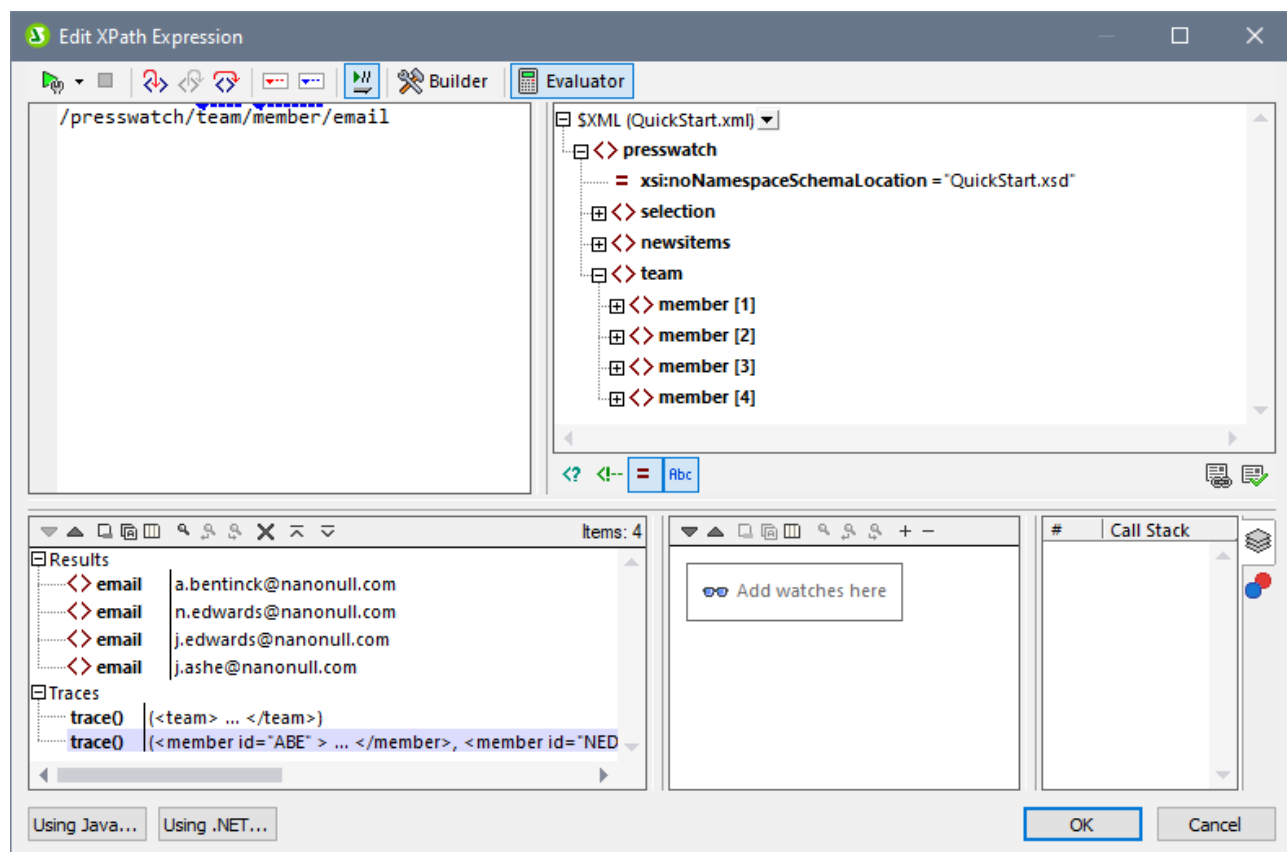
Des point d'arrêt sont des points dans lesquels vous souhaitez que le débogueur s'arrête après l'avoir lancé avec la touche **Lancer débogueur**. Ils sont utiles lorsque vous souhaitez analyser une partie spécifique de l'expression. Lorsque le débogueur s'arrête au point d'arrêt, vous pouvez contrôler 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 où vous souhaitez trouver le 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 pointillés rouge. Pour retirer un point d'arrêt, le sélectionner et cliquer sur **Insérer/Supprimer point d'arrêt (F9)**.

Voir aussi [Points de débogage](#) ⁴³⁷ ci-dessous.

Points de trace

Des 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 particulières de l'expression. Par exemple, dans la capture d'écran ci-dessous, des points de trace ont été définis dans le nœud **team** et le nœud **member**. Les résultats dans ces points de trace sont affichés dans l'arborescence *Traces*.

Pour définir un point de trace, placer le curseur à l'endroit où vous souhaitez trouver le point de trace, et cliquer sur la touche de la barre d'outils **Insérer/Supprimer point de trace (F9)**. Les points de trace seront marqués avec une surligne en pointillés bleu (voir *capture d'écran ci-dessous*). Pour supprimer un point de trace, le sélectionner et cliquer sur **Insérer/Supprimer points de trace (F9)**.

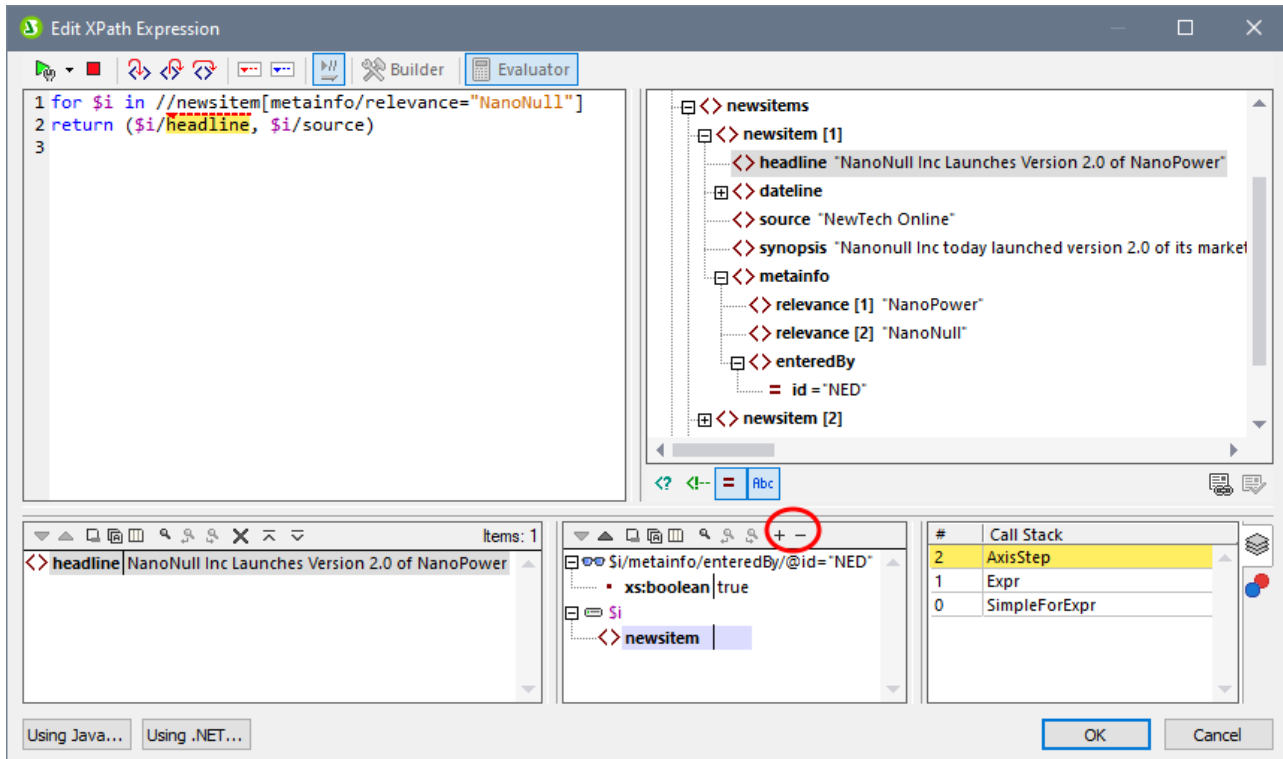


Note : si un point d'arrêt et un point de trace sont tous deux définis sur une partie de l'expression, la ligne supérieure est composée de tirets rouges et bleus alternés.

Voir aussi [Points de débogage](#) ⁴³⁷ 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 a été interrompu au point d'arrêt dans la `headline`. La variable `$i` est dans la portée de cette étape d'évaluation. Donc `$i` est affichée avec sa valeur actuelle, qui dans la capture d'écran ci-dessus est le premier nœud `newsitem`.

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/metainfo/enteredBy/@id="NED"` est utilisée pour tester si l'item news a été saisi par le membre de l'équipe avec l'id de `NED`. Le résultat `true` dans le cas du premier newsitem nous dit que cette condition a été satisfaite.
- Pour trouver des données dans le cadre d'un contexte particulier. Par exemple, dans le cadre du contexte d'un élément `company`, nous pouvons 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, un string adapté peut être généré pour indiquer le nombre total de newsitems.

Pour saisir l'expression watch, cliquez sur **Ajouter entrée Watch** dans la barre d'outils du volet (encadré en rouge dans la capture d'écran ci-dessus), puis double-cliquez la nouvelle entrée watch pour saisir l'expression et cliquez sur **Entrée** lorsque vous avez terminé. Pour supprimer une expression watch, la choisir et cliquer sur **Supprimer Entrée watch sélectionnée** dans la barre d'outils. Si, au cours du débogage, l'expression ne peut

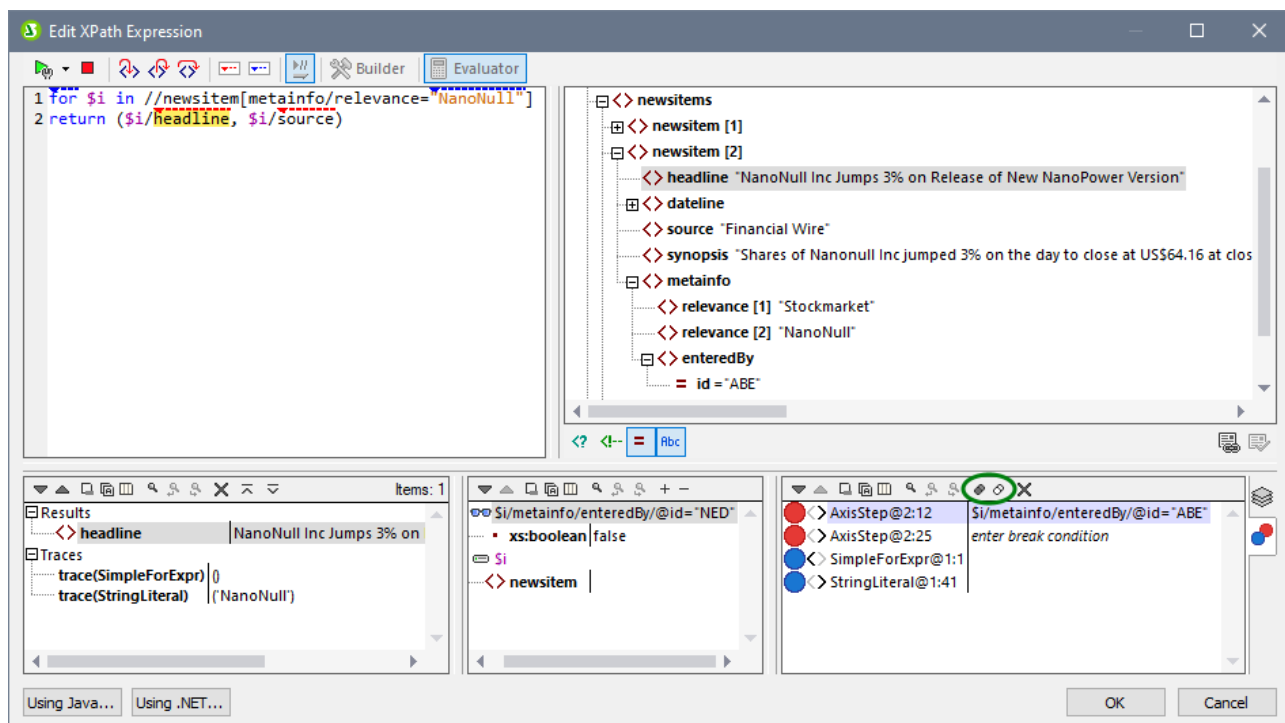
pas être évaluée correctement pour une raison quelconque (par exemple si une de ses variables est hors de portée), l'expression watch passe au rouge.

Pile d'appel

L'onglet *Pile d'appel* du volet Pile d'appel et Points de débogage (volet en bas à droite dans la capture d'écran ci-dessus) affiche les appels du processeur jusqu'à ce point du 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.

Points de débogage

L'onglet Points de débogage du volet Pile d'appel et Points de débogage (volet en bas à 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, `AxisStep@2:12` signifie qu'il y a un point de débogage sur la ligne 2, caractère 12 de l'expression dans le volet d'expression.



Veuillez noter les fonctions suivantes :

- Pour les points d'arrêt, vous pouvez saisir une **break condition** en (i) double-cliquant sur *Saisir break condition* dans le volet Déboguer Points, (ii) saisissant l'expression pour la condition, et (iii) en appuyant sur **Enter**. Ce point d'arrêt sera uniquement activé si la condition évalue à **true**. Par exemple, dans la capture d'écran ci-dessus, la break condition `$i/metainfo/enteredBy/@id="ABE"` activera le point d'arrêt sur le titre de chaque newsitem qui a été saisi par le membre de l'équipe avec l'id **ABE**. La capture d'écran montre que l'évaluation s'est interrompue à ce point d'arrêt. (Notez également que l'expression Watch à ce point d'arrête renvoie **false**.)
- Vous pouvez activer/désactiver tous les points de débogage en cliquant sur leur touche de la barre d'outils respective : **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, il est désactivé pour toutes les évaluations jusqu'à ce qu'il soit réactivé ultérieurement.

- Vous pouvez activer/désactiver les points d'arrêt individuels dans leurs menus contextuels respectifs.

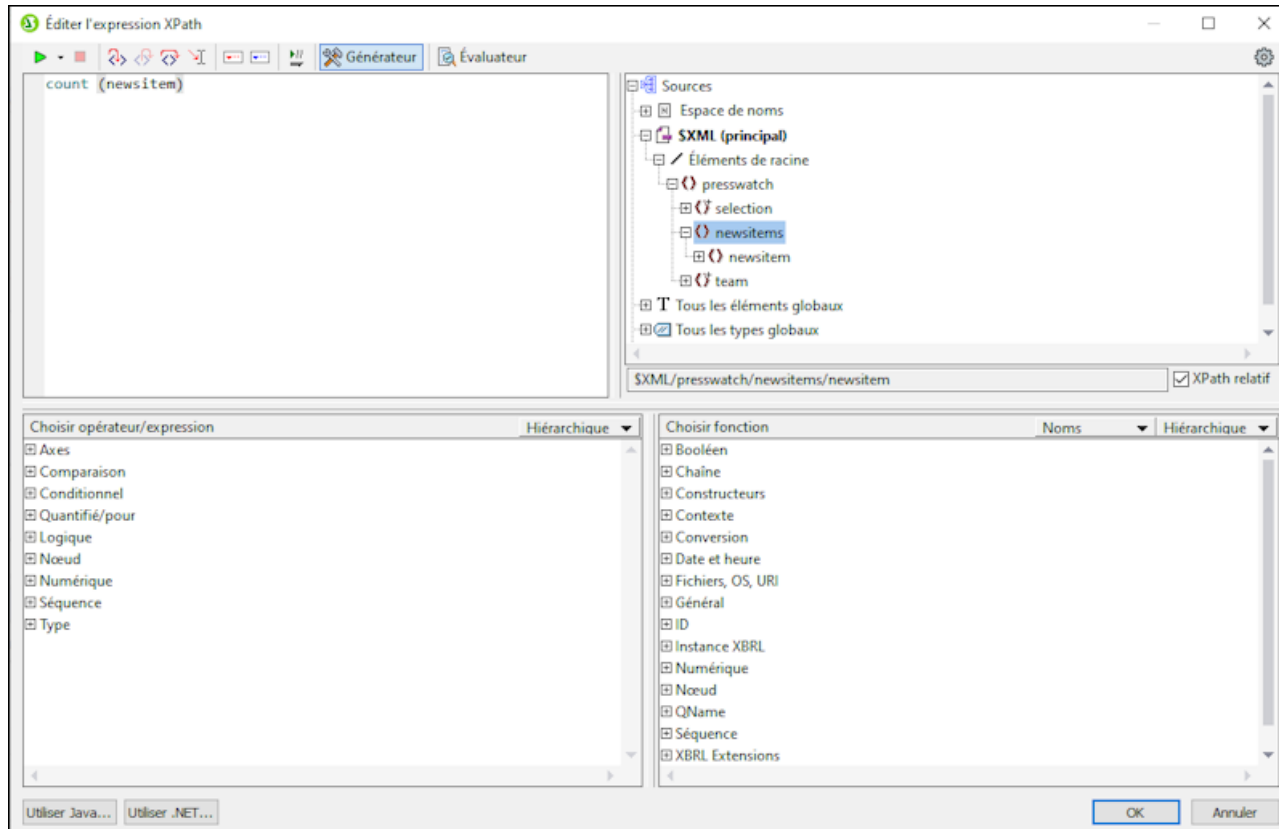
Commandes de barre d'outils dans les volets

Les volets du dialogue Éditer Expression XPath dans le Mode Débogage (*voir capture d'écran ci-dessus*) contiennent des touches qui proposent des fonctions de navigation, de recherche et de copie. Ces boutons, en commençant par la gauche, sont décrits dans la table ci-dessous. Les commandes correspondantes sont aussi disponibles dans le menu contextuel des items recensés.

Icône	Fonction
<i>Suivant, Précédent</i>	Sélectionne, respectivement, l'item suivant et précédent dans la liste de résultats
<i>Copier la ligne de texte sélectionnée dans le presse-papiers</i>	Copie la colonne de valeur de l'item de résultat sélectionné dans le presse-papiers. Afin de copier toutes les colonnes, activer la commande <i>La copie contient toutes les colonnes</i> (<i>voir ci-dessous</i>)
<i>Copier tous les messages dans le presse-papiers</i>	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
<i>La copie contient toutes les colonnes</i>	Passes entre la copie (i) de toutes les colonnes, ou (ii) uniquement la colonne de valeur. Le séparateur de colonne est un espace unique
<i>Recherche</i>	Ouvre un dialogue <i>Chercher</i> pour chercher un string, y compris des caractères spéciaux, dans la liste de résultat
<i>Chercher précédent</i>	Cherche l'occurrence précédente du terme qui a été saisi en dernier dans le dialogue <i>Chercher</i>
<i>Trouver suivant</i>	Cherche l'occurrence suivante du terme qui a été saisi en dernier dans le dialogue <i>Chercher</i>
<i>Élargir avec des enfants</i>	Élargit l'item sélectionné et tous ses descendants
<i>Comprime avec les enfants</i>	Comprime l'item sélectionné et tous ses descendants
<i>Supprimer</i>	Supprime la liste de résultats

11.2.3 Générateur d'expression

Lorsque la touche **Générateur** est cliquée dans le dialogue de l'Expression XPath (*voir la capture d'écran ci-dessous*), les volets de l'assistant à la saisie vous aident à construire une expression XPath pour qu'elle devienne visible. Double-cliquez sur une entrée dans l'assistant à la saisie pour la saisir à l'endroit actuel du curseur dans l'Expression XPath.



Il existe trois volets d'assistants à la saisie :

- Une arborescence de schéma pour saisir les nœuds de l'élément et de l'attribut dans l'expression XPath. Si la case à cocher *XPath relatif* est cochée alors le chemin de l'emplacement s chemin vers le nœud sélectionné est saisi relativement au nœud contextuel (le nœud dans le design dans lequel l'expression XPath est construite. Le nœud contextuel est affiché sous le volet d'arborescence du schéma. Une expression XPath absolue qui commence par la racine du document, et est utile pour le nœud sélectionné si la case à cocher *Chemin relatif* n'est pas cochée.
- Un volet d'assistant à la saisie pour les opérateurs et les expressions. Ceux-ci contiennent : (i) axes (`ancestor::`, `parent::`, etc), (ii) opérateurs (par exemple `eq` et `div`), et (iii) expressions (`for # in # return #`, etc). Les items du volet peuvent groupés soit alphabétiquement soit par catégorie fonctionnelle. Sélectionnez l'option que vous voulez choisir *Hiérarchique* ou *Plat* depuis le menu de la liste déroulante dans la barre de titre du volet.
- Un assistant à la saisie avec des fonctions de la version XPath active est classé soit par ordre alphabétique, soit groupé par catégorie fonctionnelle. Sélectionnez l'option que vous voulez choisir *Hiérachique* ou *Plat* depuis le menu de la liste déroulante dans la barre de titre du volet. L'option *Names/Types* vous permet de choisir si les arguments des fonctions sont affichés en tant que noms ou types de données.

Fonctions du générateur

- Pour consulter une description de texte d'un item dans chaque volet, passez au-dessus de 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).

- Les signatures sont données soit avec les noms ou types de données des arguments et de la sortie de la fonction. Choisir *Noms* ou *Types* depuis le menu déroulant dans la barre de titre du volet.
- 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 des espaces réservés (le symbole #).
- Si le (i) texte est sélectionné dans le champ d'édition d'expression, et (ii) qu'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.

Une fois que vous avez saisi une fonction dans l'expression, planer au-dessus du nom de la fonction affiche la signature de la fonction et un descriptif texte de la fonction. Si différentes signatures existent pour une fonction ayant le même nom, celles-ci sont indiquées par un facteur de surcharge en bas de l'affichage. Si vous placez le curseur à l'intérieur des parenthèses de la fonction et appuyez sur **Ctrl+Shift+Spacebar**, vous pouvez consulter les signatures des diverses surcharges du nom de la fonction.

Créer des expressions XPath

Le dialogue Éditer Expression XPath vous aide à générer des expressions XPath de la manière suivante.

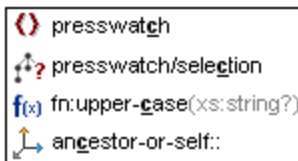
- *Nœud contextuel et arborescence de schéma*
La zone de texte *Sélection* dans le volet *Sources* vous montre immédiatement le nœud contextuel. L'expression sera insérée à l'emplacement à l'intérieur de ce nœud contextuel, et elle sera évaluée avec ce nœud comme son contexte.
- *Insérer un nœud depuis l'arborescence de schéma*
Dans le volet *Sources*, le schéma entier est affiché. Double-cliquez sur un nœud dans l'arborescence de schéma pour l'insérer dans l'expression XPath. Si la zone de texte *Chemin relatif* est cochée, le nœud sélectionné sera inséré avec une expression du chemin de l'emplacement qui est relatif au nœud contextuel.
- *Information relative à l'espace de nom*
L'arborescence de schéma dans le volet *Sources* contient un item d'espace de nom. Élargir cet item affiche tous les espaces de noms déclarés dans la feuille de style. Cette information peut être utile pour vérifier les préfixes d'un espace de nom que vous voulez utiliser dans une expression XPath.
- *Insérer des axes, opérateurs et expressions XPath*
Le volet *Sélectionner opérateur/Expression* recense les axes XPath (`ancestor::`, `parent::`, etc.), opérateurs (par exemple, `eq` et `div`), et les expressions (`for # in # return #`, etc.) pour la version XPath sélectionnée comme version XSLT pour la SPS. L'affichage peut passer entre une liste alphabétique et une liste hiérarchique (qui regroupe les items selon leur fonctionnalité). Pour insérer un axe, un opérateur ou un axe dans l'expression XPath, double-cliquez sur l'item requis.
- *Insérer des fonctions XPath*
Le volet *Sélectionner fonction* recense les fonctions XPath de manière alphabétique ou groupée selon la fonction (cliquez sur l'icône respective en haut du volet pour basculer entre les deux arrangements). Chaque fonction est montrée avec sa signature. Si une fonction a plus d'une signature, cette fonction est affichée autant de fois qu'elle a de signatures. Les arguments dans une signature sont séparés par des virgules, et chaque argument peut avoir un indicateur d'occurrence (? indique une séquence de zéro ou un item du type spécifié ; * indique une séquence de zéro ou plus d'items du type spécifié). Les arguments peuvent être affichés comme noms ou comme types de données ; sélectionnez *Noms* ou *Types* dans la barre de titre du volet. Chaque fonction spécifie également le type retour de cette fonction. Par exemple : `=> date ?` indique que le type de données de retour

attendu est une séquence d'aucun ou un item `date`. Placer la souris au-dessus d'une fonction affiche une description succincte de la fonction. Pour insérer une fonction dans l'expression XPath, double-cliquez la fonction requise.

- Java et les fonctions d'extension .NET peuvent être utilisés dans les expressions XPath, vous permettant d'accéder les fonctions des langages de programmation. Les touches **Java** et **.NET** en bas du dialogue ouvrent des cases d'information avec des explications sur comment utiliser Java et les fonctions d'extension .NET dans les expressions XPath. Pour plus d'information sur le sujet, voir la section [Fonctions d'extension](#) ⁶³⁵ de cette **documentation**.

Édition intelligente directement pendant la saisie de texte

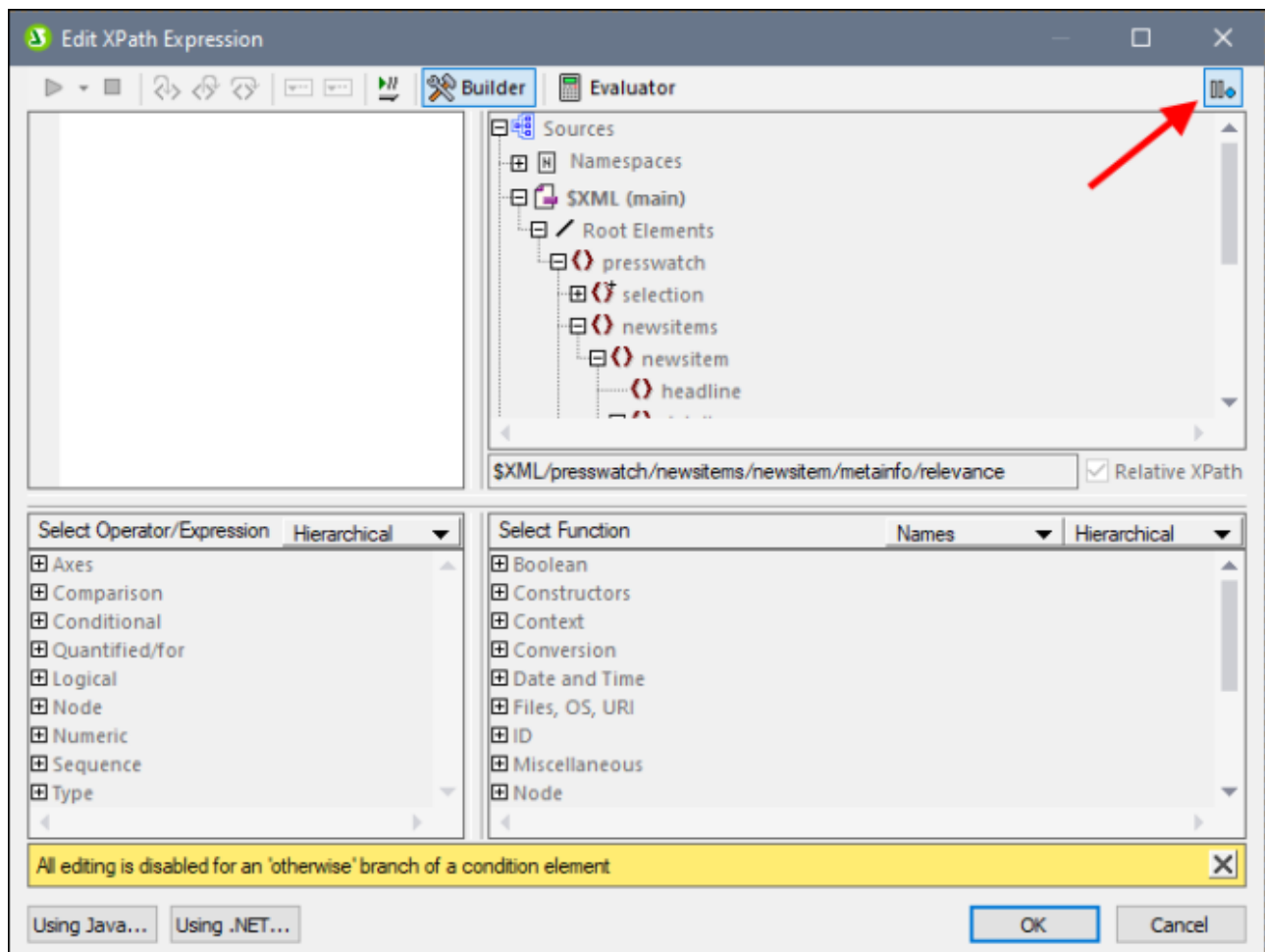
Si vous saisissez une expression directement dans la zone de texte *Expression*, les options disponibles à ce point sont affichées dans une fenêtre popup (voir la capture d'écran ci-dessous).



Celles-ci incluent les éléments, les fonctions XPath et les axes XPath. Faire défiler la liste des options en utilisant les touches **Haut** et **Bas**, et appuyer sur **Entrée** si vous souhaitez sélectionner une option et la saisir dans l'expression.

La zone à cocher Otherwise

Le toggle *Otherwise* (voir la flèche rouge dans la capture d'écran ci-dessous) ajoute une branche Otherwise au modèle conditionnel comme sa dernière branche. Uniquement une branche Otherwise peut être présente dans un modèle conditionnel. Lorsqu'un modèle conditionnel est évalué, la première branche à évaluer comme `true` est exécutée. Si aucune branche n'évalue comme `true`, alors, la branche Otherwise est exécutée si présente, autrement le modèle conditionnel est quitté sans ses branches étant exécutées. Puisque la branche Otherwise est déclenchée uniquement dans le cas où aucune branche précédente est évaluée comme `true`, elle n'a pas besoin d'avoir une condition qui est définie pour elle. En conséquence, lorsque la case à cocher est sélectionnée, le champ de saisie de l'expression XPath est désactivée.



Pour les détails et savoir comment utiliser la condition Otherwise, voir les [Modèles conditionnels](#) ²⁶⁶.

11.3 Barres d'outils

Un grand nombre de commandes StyleVision est disponible comme raccourcis de la barre d'outils, organisé en les barres d'outils suivantes :

- [Formatage](#) ⁴⁴⁵
- [Table](#) ⁴⁴⁶
- [Insérer Éléments de design](#) ⁴⁴⁷
- [Filtre de design](#) ⁴⁴⁹
- [Standard](#) ⁴⁵⁰

L'icône dans chaque barre d'outils est recensée dans les sous-sections de cette section, chacune avec une description succincte de la commande correspondante.

Positionner les barres d'outils

Une barre d'outils peut flotter librement à l'écran ou peut être placée dans une zone de barre d'outils le long de tout bord de la GUI. Les barres d'outils les plus communément placées le long du bord supérieur de la GUI juste en-dessous de la barre de menu. Toutefois, elles peuvent aussi être placées le long des bords sur le côté ou en bas de la GUI

Pour positionner une barre d'outils dans la zone de la barre d'outils, suivez les étapes suivantes :

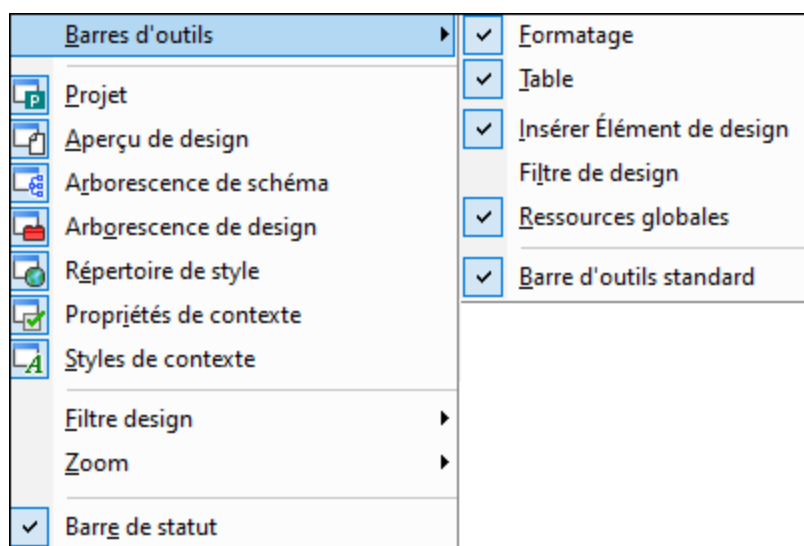
1. Saisissez la barre d'outils par sa poignée (si la barre d'outils est déjà dans la zone de la barre d'outils) ou par sa barre d'outils (si la barre d'outils est flottante).
2. Glissez la barre d'outils dans la zone de barre d'outils souhaitée, si elle existe, et déposez-la à l'emplacement désiré dans cette zone de la barre d'outils. Si aucune barre d'outils n'existe sur le bord le long duquel vous voulez placer la barre d'outils, glisser la barre d'outils vers ce bord créera automatiquement une zone de barre d'outils là où cette barre d'outils est déposée.

Pour faire flotter une barre d'outils librement, saisissez-la par sa poignée, glissez-la en dehors de la zone de la barre d'outils, et déplacez-la partout à l'écran à l'exception d'un bord ou d'une zone de la barre d'outils existante.

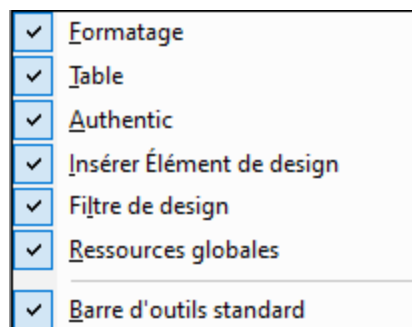
Activer et désactiver l'affichage des barres d'outils

L'affichage des barres d'outils individuel peut être activé et désactivé utilisant une des trois méthodes suivantes :

- Dans le menu **Affichage | Barres d'outils** (*capture d'écran ci-dessous*), sélectionnez ou décochez une barre d'outils, respectivement, affichez ou masquez cette barre d'outils.



- Cliquez avec la touche de droite sur une zone de la barre d'outils pour afficher un menu contextuel (*capture d'écran ci-dessous*) qui vous permet d'activer et de désactiver l'affichage des barres d'outils individuelles.

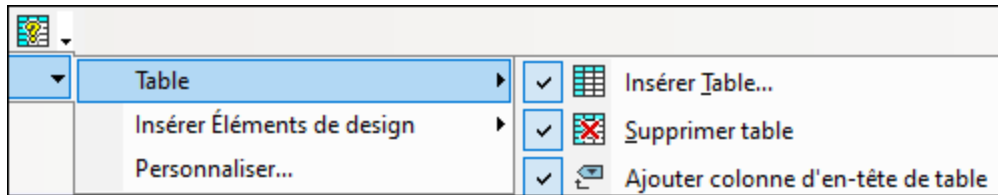


- Dans l'onglet Barres d'outils du [dialogue Personnaliser](#) ⁵³⁰ ([Outils | Personnaliser](#) ⁵³⁰), activez ou désactivez l'affichage des barres d'outils individuelles en cliquant sur la case à cocher d'une barre d'outils. Une fois terminé, cliquez sur le bouton **Fermer** pour fermer le dialogue.

Ajouter et supprimer les boutons de la barre d'outils

Les boutons de la barre d'outils individuels peuvent être ajoutés à ou supprimés de la barre d'outils, à savoir, ils peuvent être rendus visibles ou masqués. Pour ajouter ou supprimer un bouton de la barre d'outils, procédez comme suit :

1. Dans la barre d'outils où le bouton à ajouter ou à supprimer se trouve, cliquez sur **Plus de boutons** (si la barre d'outils est une zone de la barre d'outils) ou sur le bouton **Options de la barre d'outils** (si la barre d'outils est une barre d'outils flottante). Le bouton **Plus de boutons** est une tête de flèche située à droite de la barre d'outils (dans les zones de la barre d'outils) ou un bouton de la barre d'outils (dans les zones de la barre d'outils verticales). Le bouton des Options de la barre d'outils est une tête de flèche situé à droite de la barre d'outils flottante.
2. Dans le menu **Ajouter ou Supprimer les boutons** qui s'ouvre, placez le curseur au-dessus de l'item de menu **Ajouter ou Supprimer les boutons** (*capture d'écran ci-dessous*). Ceci fait apparaître un menu qui contient des noms des barres d'outils dans la zone de la barre plus l'item de menu **Personnaliser** (*capture d'écran ci-dessous*).



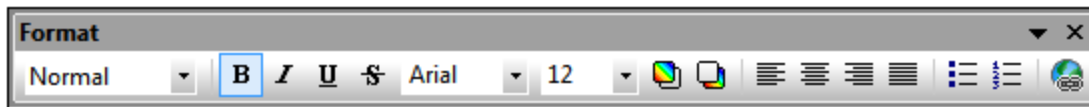
3. Placez le curseur au-dessus de la barre d'outils qui contient le bouton de la barre d'outils à ajouter ou à supprimer (*capture d'écran ci-dessus*).
4. Dans le menu qui se déroule (*capture d'écran ci-dessus*), cliquez sur le nom du bouton de la barre d'outils pour ajouter ou supprimer ce bouton de la barre d'outils.
5. Cliquer sur l'item Personnaliser fait que le [dialogue Personnaliser](#)⁵³⁰ s'ouvre.

L'item **Réinitialiser Barre d'outils** ci-dessous de la liste des boutons dans chaque menu de la barre d'outils réinitialise la barre d'outils à l'état auquel elle était lors du téléchargement StyleVision. Dans cet état, tous les boutons pour la barre d'outils sont affichés.

Note : les boutons d'une barre d'outils sont prédéfinis et ne peuvent pas être dissociés de la barre d'outils. Le processus décrit affiche ou masque le bouton dans la barre d'outils qui est affichée dans la GUI.

11.3.1 Format

La **barre d'outils Format** (*capture d'écran ci-dessous*) est activée dans le Mode Design et contient des commandes qui attribuent des propriétés de formatage bloc et inline utilisées généralement à l'/aux item/s sélectionné/s SPS dans le design.



Formats HTML prédéfinis

Le format HTML sélectionné depuis la liste déroulante est appliqué à la sélection dans le Mode Design. Par exemple, une sélection de `div` applique l'élément de HTML `Block (div)` autour de la sélection actuelle dans le Mode Design.

Propriétés de texte

Les propriétés de texte inline en gras, italique, souligné et barré peuvent être directement appliquées dans la sélection actuelle dans le Mode Design en cliquant sur le bouton approprié. Les style de police, taille de police, couleur d'avant-plan et d'arrière-plan peuvent aussi être appliqués par le biais des boutons de la barre d'outils.

Alignement

Les propriétés d'alignement (aligné à gauche, centré, aligné à droite, et justifié) peuvent être appliquées directement à la sélection dans le Mode Design.

Listes

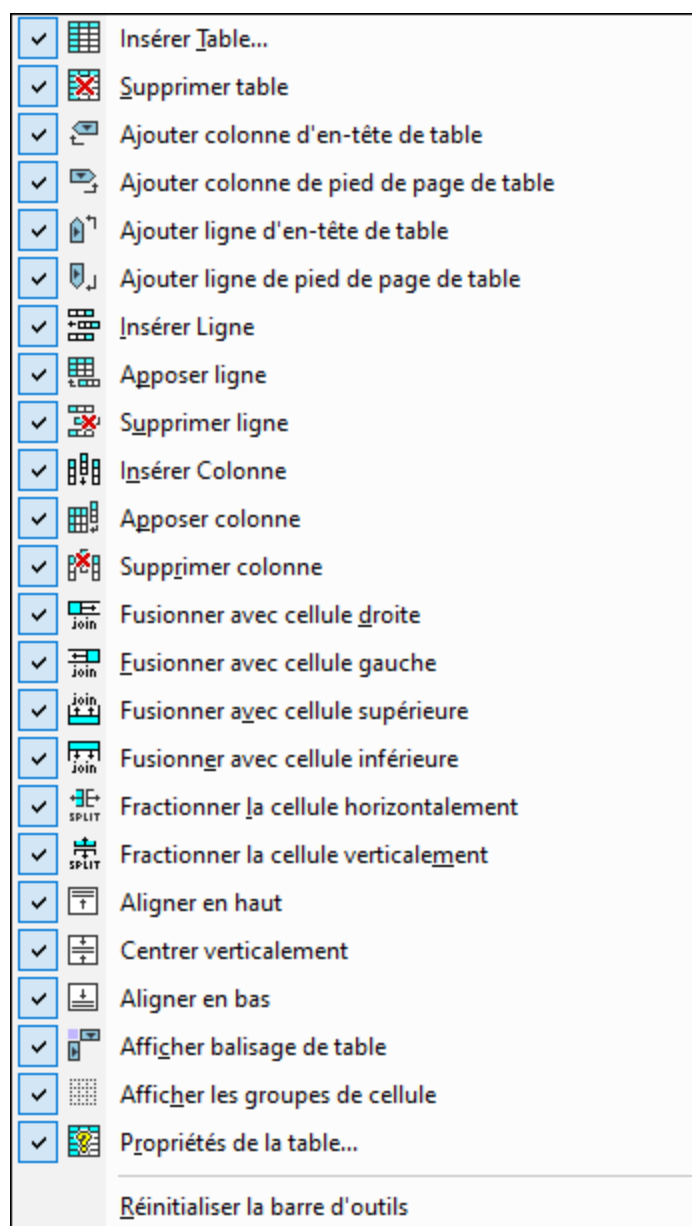
Les listes peuvent être insérées au point d'insertion du curseur, ou la sélection dans la SPS peut être convertie en une liste.

Lien hypertexte

Insère un hyperlien au niveau du point d'insertion du curseur. Voir [Hyperlien](#) ⁵⁰⁰ pour une description illustrant comment utiliser cette commande.

11.3.2 Table

La **barre d'outils Table** contient des commandes pour structurer et formater des tables statiques et dynamiques dans le Mode Design. Ces commandes sont affichées dans la capture d'écran ci-dessous (qui est celle du menu Personnaliser de la barre d'outils de la Table disponible lorsque vous cliquez sur le bouton **Personnaliser** à droite de la barre d'outils).



Opérations Ligne et Colonne

Les lignes et colonnes dans toute table SPS (statique ou dynamique) peuvent être insérées, ajoutées ou supprimées en faisant référence à l'emplacement du curseur. Les lignes et colonnes sont insérées avant l'emplacement du curseur actuel ou ajoutées après toutes les lignes/colonnes. La ligne/colonne dans laquelle le curseur peut aussi être supprimée. Ces opérations sont obtenues avec les boutons **Insérer Ligne/colonne**, **Apposer Ligne/colonne**, ou **Supprimer Ligne/colonne**. Vous pouvez aussi ajouter des en-têtes et bas de page à la table soit comme colonnes, soit comme lignes **Ajouter en-tête/colonne/ligne de pied de page de table**.

Opérations de cellule

Une table SPS dans laquelle le curseur est situé peut être fusionnée à toute cellule des quatre cellules qui l'entoure. L'opération de fusionnement est semblable à celle de l'extension des cellules de table dans HTML. Les boutons à utiliser pour ces opérations sont **Fusionner avec cellule droite/gauche/supérieure/inférieure**. Aussi, une cellule de table SPS dans laquelle le curseur est situé peut être fractionnée, soit de manière horizontale ou verticale, utilisant les boutons **Fractionner la cellule horizontalement** et **Fractionner la cellule verticalement**, respectivement. Le contenu de la cellule de table SPS peut être aligné verticalement en haut, au centre, et en bas. L'affichage des bords de la cellule peut être activé et désactivé avec la bascule **Afficher les limites de la cellule**.

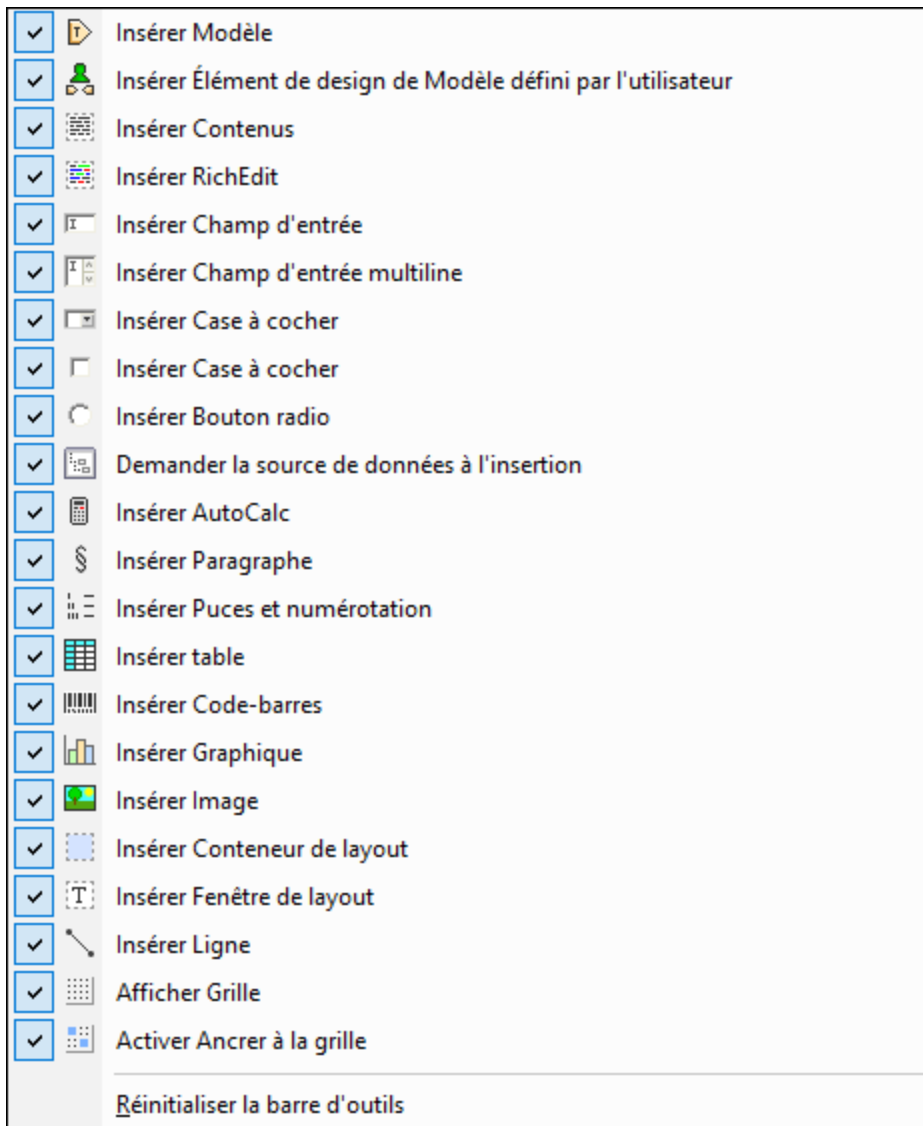
Opérations de table, propriétés, aperçu

Placez le curseur dans une table statique ou dynamique et cliquez sur **Supprimer table** ⁵¹⁶ supprime la table. Le markup de table peut être activé et désactivé avec la commande **Afficher balisage de table**. La commande **Propriétés de la table** s'ouvre dans le dialogue des **Propriétés de la table**, dans lequel les propriétés de la table peuvent être définies.

11.3.3 Insérer Éléments de design

La **barre d'outils des Éléments de design** contient des icônes pour les commandes qui doivent insérer des éléments de design dans le SPS, et pour des commandes associées. Les divers éléments de design qui peuvent être insérés via ces icônes de barre d'outils sont affichés dans la capture d'écran ci-dessous. Il existe trois types d'éléments dans la barre d'outils :

1. Les **éléments de design** ⁴⁴⁸, qui sont sensibles au 'context-node' (la majorité des éléments dans la barre d'outils),
2. Les **éléments de layout** ⁴⁴⁹, qui sont indépendants du nœud contextuel, et
3. Les **balises associées à la grille** ⁴⁴⁹ pour assister le design.



Éléments de design

Les éléments de design sont les éléments sensibles au 'context-node' qui sont disponibles dans le menu **Insérer**. Pour insérer un élément de design utilisant son icône de la barre d'outils, suivez les étapes suivantes :

1. Sélectionnez l'icône de la barre d'outils pour l'élément que vous souhaitez insérer.
2. Cliquez dans l'emplacement dans le design où l'élément doit être inséré. Un 'Insérer Élément de design' pour l'élément de design sélectionné s'ouvre. Ceci affiche l'arborescence de schéma avec le nœud contextuel en surbrillance. Le nœud contextuel est le nœud à l'intérieur duquel le curseur a été placé pour l'insertion de l'élément de design.
3. Si vous souhaitez insérer l'élément de design à l'intérieur du nœud contextuel sélectionné, cliquez sur **OK**. Si vous souhaitez sélectionner un autre nœud contextuel, faites-le dans l'arborescence de schéma, puis cliquez sur **OK**.

4. Dans le cas de certains éléments de design, tels que les Calculs automatiques, une étape supplémentaire est requise comme la définition d'un Calcul automatique. Dans d'autres cas, comme l'insertion d'un modèle défini par l'utilisateur, le dialogue Insérer Élément de design est sauté. Dans de tels cas, un autre dialogue, comme le [dialogue Éditer Expression XPath](#)⁴²⁶ apparaîtra Exécutez l'étape requise et appuyez sur le bouton du dialogue **OK**.

L'élément de design sera inséré à la fin de l'étape 3 ou 4, dépendant du genre d'élément de design étant inséré.

Éléments de layout

Il existe trois commandes d'élément de layout dans la barre d'outils Insérer Élément de design : pour insérer (i) un conteneur de layout; (ii) une case de layout ; et (iii) une ligne. Notez que les cases de layout et les lignes ne peuvent être insérées qu'à l'intérieur d'un conteneur de layout.

Pour insérer un conteneur de layout, sélectionnez l'icône **Insérer Conteneur de layout** , puis cliquez sur l'emplacement dans le design où vous souhaitez insérer le conteneur de layout. Vous serez invité à préciser la taille du conteneur de layout, en sélectionnant quel conteneur de layout sera inséré. Pour insérer une case de layout, cliquez sur l'icône **Insérer une Case de layout**, puis déplacez le curseur à l'emplacement à l'intérieur du conteneur de layout auquel vous souhaitez insérer la case de layout, puis cliquez. La case de layout est insérée. Cliquez à l'intérieur de la case de layout pour commencer à saisir. Pour insérer une ligne, cliquez sur l'icône **Insérer ligne** , puis déplacez le curseur à l'emplacement à l'intérieur du conteneur de layout dans lequel vous souhaitez dessiner la ligne. Cliquez pour définir le point de début de la ligne, puis glissez le curseur au point final désiré. Déposez le curseur au point final. La ligne est insérée et s'élargit à partir du point de début indiqué jusqu'au point final indiqué.

Pour redimensionner les conteneurs de layout et les cases de de layout, placez le curseur au-dessus du bord droit ou en bas du conteneur de layout ou de la case de layout, puis glissez le bord afin d'obtenir la dimension désirée. Pour déplacer la case de layout, placez le curseur au-dessus du bord du haut ou à gauche de la case de layout et, lorsque le curseur se transforme en croix, glissez la case de layout vers le nouvel emplacement.

Balises associées à la grille

La commande **Afficher grille** active ou désactive l'affichage de la grille de dessin. Lorsque la commande **Ancrer à la grille** est activée, les éléments créés à l'intérieur du conteneur de layout, tels les cases de layout et lignes, ancrent aux lignes de grille et intersections de la ligne de grille. Les propriétés de la grille peuvent être définies dans l'onglet de Design du dialogue des Options (**Outils | Options**).

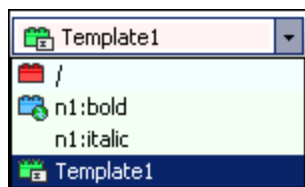
11.3.4 Filtre de design

La **barre d'outils Filtre de design** (*capture d'écran ci-dessous*) contient les commandes qui vous permettent de filtrer quels modèles sont affichés dans le design. Chaque icône dans la barre d'outils est expliquée ci-dessous.



Iconne	Commande	Description
	Afficher uniquement un modèle	Affiche uniquement le modèle sélectionné. Placer le curseur dans un modèle et cliquer pour afficher ce modèle uniquement.
	Afficher tous les types de modèles	Affiche tous les modèles dans SPS (principal, global, dénommé et layout) .
	Affiche les modèles importés	Bascule l'affichage des modèles importés sur activé/désactivé.
	Affiche/masque le modèle principal	Bascule l'affichage du modèle principal sur activé/désactivé.
	Affiche/masque les modèles globaux	Bascule l'affichage des modèles globaux sur activé/désactivé.
	Affiche/masque les fragments de design	Bascule l'affichage des fragments de design sur activé/désactivé.

La liste déroulante Filtre de design (*capture d'écran ci-dessous*) affiche une liste de tous les modèles dans la SPS.



Sélectionnez un modèle dans la liste déroulante fait que le modèle est sélectionné dans le design. La liste déroulante, vous permet donc de naviguer rapidement vers le modèle désiré dans le design, qui est utile si le design a plusieurs modèles, dont certains pourraient être masqués actuellement.

11.3.5 Standard

La **barre d'outils standard** contient des boutons pour les commandes qui fournissent une fonction associée au fichier et d'édition importante. Ces icônes sont recensées ci-dessous avec une brève description. Pour une description plus ample d'une commande, cliquez sur la commande pour aller à sa description dans la section Référence.

Btn	Commande	Raccourci	Description
	Nouveauté du Schéma XML /	Ctrl+N	Crée un nouveau document SPS basé sur un schéma. Cliquer sur la flèche de déroulement vous permet de

Btn	Commande	Raccourci	Description
	DTD ⁴⁵²		créer la SPS depuis une BD ou un document HTML, ou une SPS vide.
	Ouvrir ⁴⁵⁸	Ctrl+O	Ouvre un document SPS existant.
	Enregistrer le design ⁴⁶³	Ctrl+S	Enregistre le document SPS actif.
	Tout enregistrer ⁴⁶³	Ctrl+Shift+ S	Enregistre tous les documents SPS ouverts.
	Imprimer ⁴⁷⁴	Ctrl+P	Imprime Authentic View et le fichier de travail XML.
	Aperçu avant impression ⁴⁷⁴		Affiche un aperçu d'impression de Authentic View du fichier de travail XML.
	Couper ⁴⁷⁶	Shift+Del	Coupe la sélection et la place dans le presse-papier.
	Copier ⁴⁷⁶	Ctrl+C	Copie la sélection dans le presse-papier.
	Coller ⁴⁷⁶	Ctrl+P	Colle l'item du presse-papier dans l'emplacement du curseur.
	Supprimer ⁴⁷⁶	Del	Supprime la sélection.
	Annuler ⁴⁷⁶	Alt+ Backspace	Annule un changement d'édition. Un nombre illimité d'actions Annuler peuvent être réalisées à tout moment.
	Rétablir ⁴⁷⁶	Ctrl+Y	Rétablit une annulation.
	Recherche ⁴⁷⁶	Ctrl+F	Recherche du texte dans Authentic View et les aperçus de sortie.
	Trouver suivant ⁴⁷⁶	F3	Recherche la prochaine occurrence du texte recherché.
	XSLT 1.0 ⁹⁹		Définit XSLT 1.0 comme langage de feuille de style.
	XSLT 2.0 ⁹⁹		Définit XSLT 2.0 comme langage de feuille de style.
	XSLT 3.0 ⁹⁹		Définit XSLT 3.0 comme langage de feuille de style.
	Orthographe ⁵²⁵		Vérifie l'orthographe dans le document SPS.

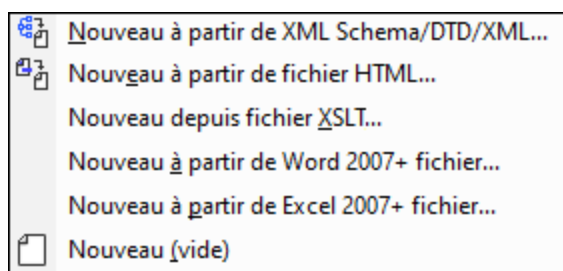
11.4 Menu Fichier

Le menu **Fichier** contient des commandes permettant de travailler avec des fichiers SPS et des fichiers associés. Les commandes suivantes sont disponibles :

- [Nouveau](#) ⁴⁵², pour créer un nouveau fichier SPS à partir de diverses sources.
- [Ouvrir. Recharger. Fermer. Fermer tout](#) ⁴⁵⁸, pour ouvrir et fermer le fichier actif, et pour recharger le fichier actif.
- [Enregistrer design. Enregistrer sous. Tout enregistrer](#) ⁴⁶³, commandes permettant d'enregistrer le fichier SPS actif et tous les fichiers SPS ouverts.
- [Exporter en tant que fichier MobileTogether Design](#) ⁴⁶⁹, pour générer un design MobileTogether à partir du fichier SPS actif.
- [Enregistrer les fichiers générés](#) ⁴⁶⁹, pour enregistrer les fichiers de sortie qui peuvent être générés à l'aide du SPS.
- [Design web](#) ⁴⁷², génère tous les fichiers nécessaires à l'exécution d'une application ASPX, dans l'emplacement du dossier que vous spécifiez.
- [Propriétés](#) ⁴⁷², pour définir l'encodage des documents de sortie, le mode de compatibilité CSS du navigateur, la manière dont les chemins d'image relatifs dans Authentic View doivent être résolus et si les images doivent être intégrées ou liées dans les sorties RTF (*éditions Enterprise et Professional*) et Word 2007+ (*édition Enterprise uniquement*).
- [Aperçu avant impression. Imprimer](#) ⁴⁷⁴, activés dans les vues de sortie, ces commandes impriment ce qui est affiché dans les aperçus.
- [Fichiers récemment utilisés. Quitter](#) ⁴⁷⁵, respectivement, pour ouvrir un fichier récemment utilisé et pour quitter le programme.

11.4.1 Nouveau

En plaçant le curseur sur la commande **Nouveau**, un sous-menu (*capture d'écran ci-dessous*) s'affiche, vous permettant de créer un nouveau document SPS de différents types :



- Un nouveau fichier SPS basé sur un schéma XML ou DTD ou un schéma XML généré à partir d'un fichier XML (**Nouveau à partir de XML Schema/DTD/XML**). Le schéma sélectionné est ajouté à la [barre latérale Aperçu de design](#) ³⁵ et une représentation graphique arborescente est ajoutée à l'arborescence du schéma (dans la [barre latérale Arborescence de schéma](#) ³⁷). Dans la [vue Design](#) ²⁸, le fichier SPS est créé avec un modèle principal vide. Un nouveau SPS peut également être créé à partir d'un fichier (schéma ou XML) via une URL ou une ressource globale (*voir ci-dessous*).
- Un nouveau SPS basé sur un schéma défini par l'utilisateur que vous créez nœud par nœud à partir d'un [fichier HTML](#) ³⁹⁴ (**Nouveau à partir de fichier HTML**). Le schéma défini par l'utilisateur est ajouté à la [barre latérale Aperçu de design](#) ³⁵ et à la [barre latérale Arborescence de schéma](#) ³⁷. Dans

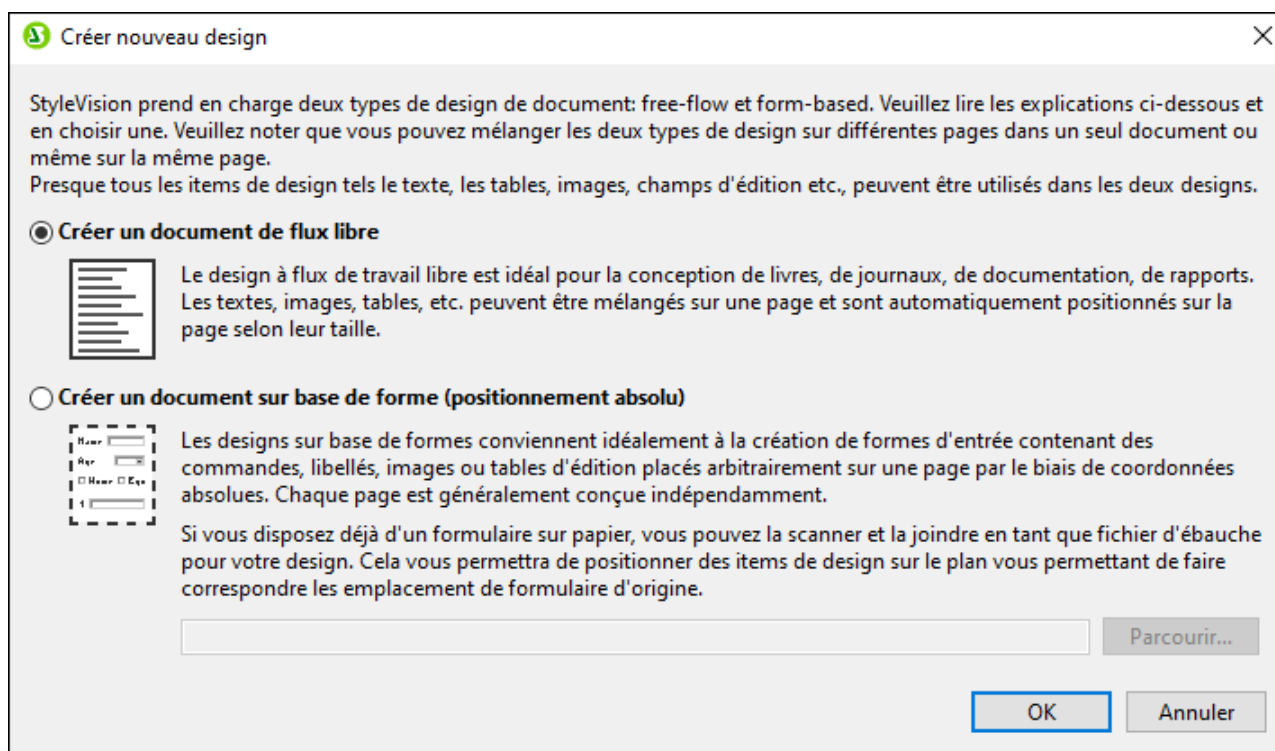
l'arborescence du schéma, il comportera un seul élément de document (élément racine) et le fichier HTML est chargé dans la [vue Design](#) ²⁸.

- Un SPS peut être créé à partir d'un fichier XSLT-for-HTML, XSLT-for-FO ou FO. La structure et le style du modèle dans le XSLT seront créés dans le SPS. Vous pouvez ensuite modifier les composants SPS et ajouter du contenu et des formats au SPS. Pour plus d'informations, voir [Nouveau à partir de XSLT](#) ³⁶⁵.
- Un nouveau SPS contenant le [contenu d'un document MS Word en tant que texte statique du design](#) ¹¹⁴.
- Un nouveau SPS contenant le [contenu d'un document MS Excel en tant que texte statique du design](#) ¹¹⁷.
- Un nouveau SPS vide (**Nouveau (vide)**). Aucun schéma n'est ajouté à la barre latérale Aperçu de design ni à l'arborescence de schéma. Un modèle principal vide sera créé dans la [vue Design](#) ²⁸.

Sélectionner le type de design

Après avoir sélectionné les fichiers sources (XSD et XML), si nécessaire, le dialogue Créer nouveau design s'affiche.

Le dialogue Créer nouveau design (*capture d'écran ci-dessous*) vous invite à sélectionner soit : (i) un design de document libre, soit (ii) un design de document basé sur un formulaire (dans lequel les composants sont positionnés de manière absolue, comme dans un programme de mise en page).



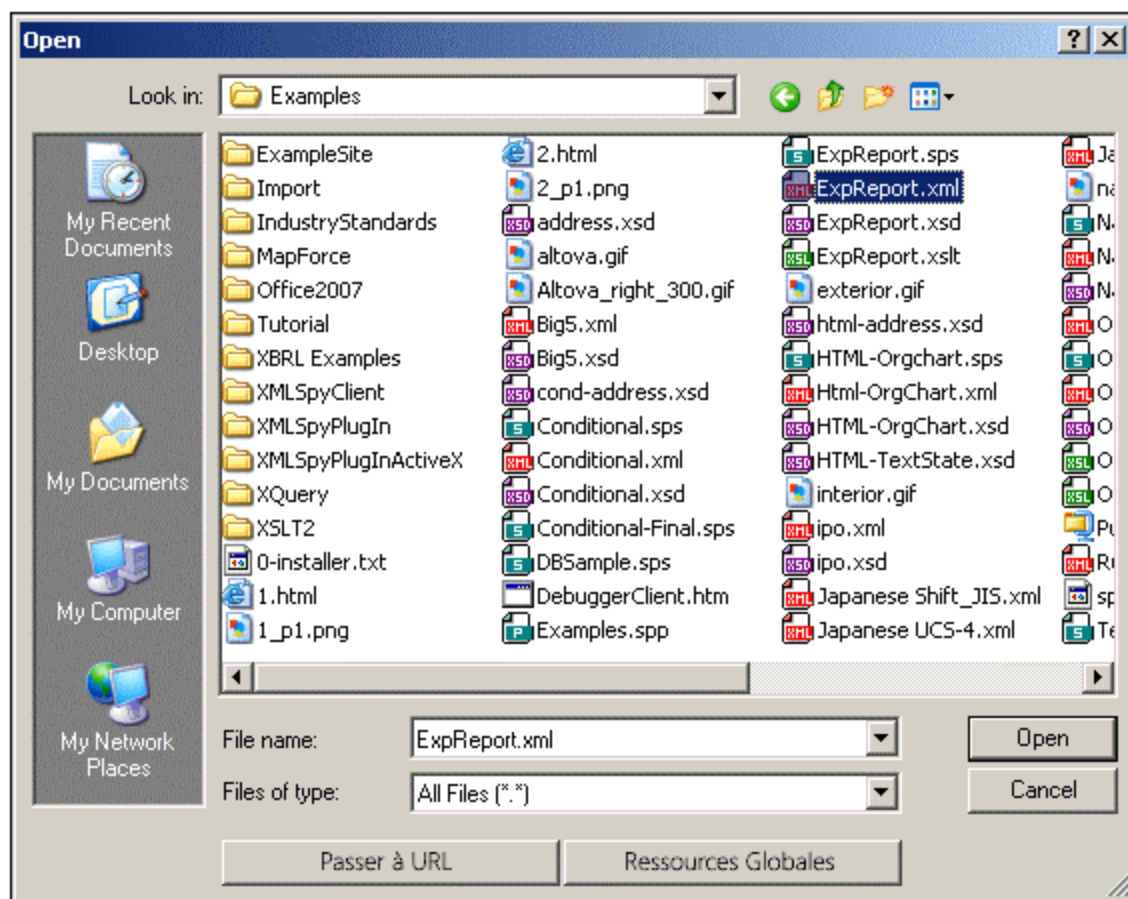
Dans une conception de document libre, le contenu du document est mis en page pour s'adapter à l'objet ou au visualiseur du support de sortie (papier ou écran). Les éléments du contenu du document ne peuvent être placés que les uns par rapport aux autres, et non de manière absolue. Ce type de conception convient aux documents tels que les rapports, les articles et les livres.

Dans un document basé sur un formulaire, un seul [conteneur de mise en page](#) ¹⁷¹ est créé, dans lequel les

composants de conception peuvent être positionnés de manière absolue. Les dimensions du conteneur de mise en page sont définies par l'utilisateur, et les boîtes de mise en page peuvent être positionnées de manière absolue dans le conteneur de mise en page et le contenu du document peut être placé dans des boîtes de mise en page individuelles. Si vous souhaitez que la conception de votre SPS reproduise un design spécifique basé sur un formulaire, vous pouvez utiliser une image du formulaire original comme [image de référence](#) ¹⁷¹. L'image de référence peut ensuite être incluse comme image d'arrière-plan du conteneur de mise en page. L'image de référence est utilisée pour vous aider à concevoir votre formulaire ; elle ne sera pas incluse dans la sortie.

Sélectionner des fichiers via URL et ressources globales

Dans plusieurs dialogues Ouvrir un fichier et Enregistrer un fichier, vous pouvez choisir de sélectionner le fichier requis ou d'enregistrer un fichier via une URL ou une ressource globale (*voir capture d'écran ci-dessous*). Sélectionnez **Passer à URL** ou **Ressources Globales** pour accéder à l'un de ces processus de sélection.



Sélectionner des fichiers via URL

Pour sélectionner un fichier via une URL, procédez comme suit :

1. Cliquez sur la commande **Passer à URL**. Cela permet de passer au mode URL du dialogue Ouvrir (*capture d'écran ci-dessous*).

Ouvrir

URL de fichier:

Ouvrir comme: ☒ Auto ☐ XML ☐ DTD

Chargement du fichier ☒ Utiliser cache/proxy ☐ Recharger

Identification

Utilisateur: Mot de passe:

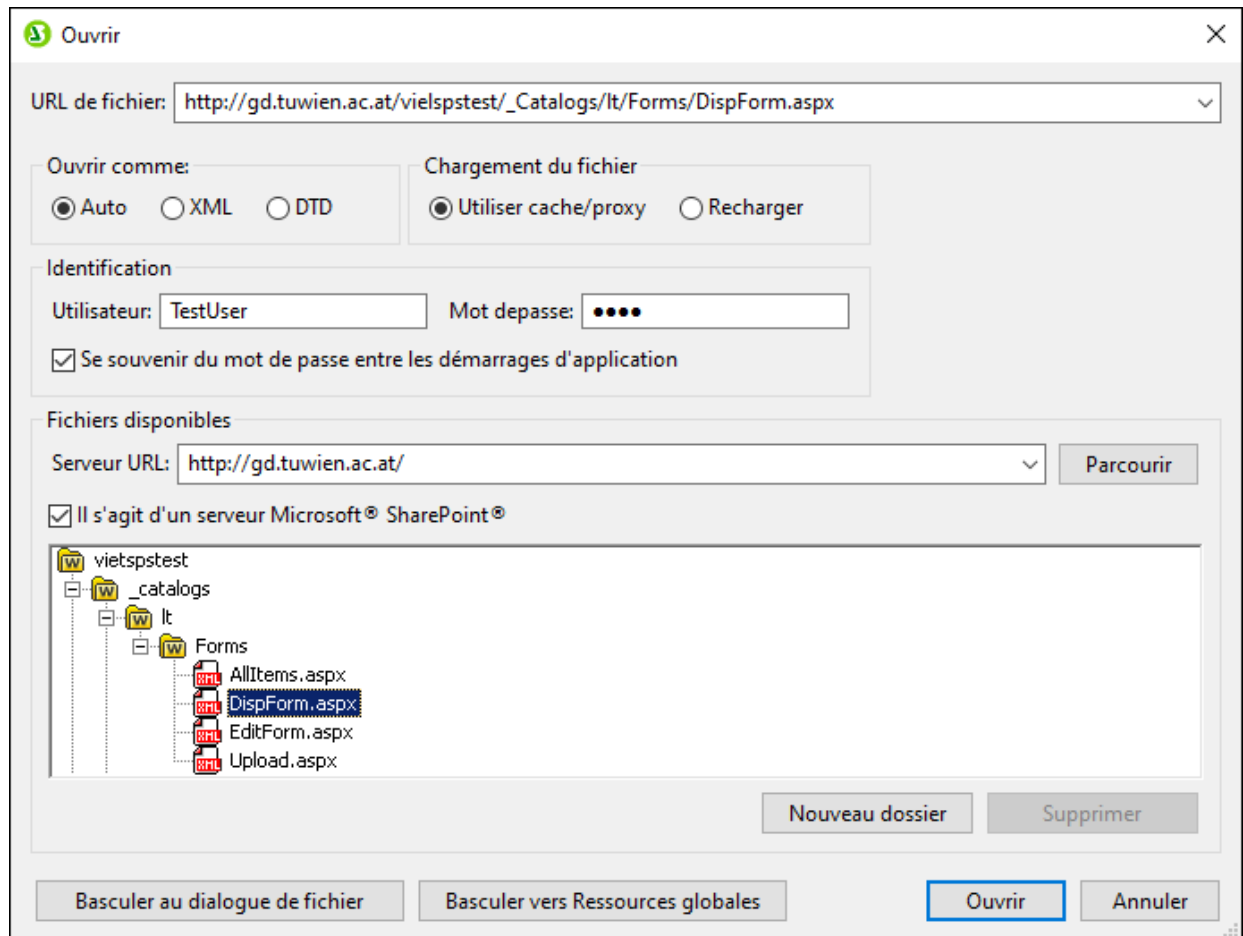
☐ Se souvenir du mot de passe entre les démarrages d'application

Fichiers disponibles

Serveur URL:

☒ Il s'agit d'un serveur Microsoft® SharePoint®

2. Entrez l'URL à laquelle vous souhaitez accéder dans le champ *Serveur URL* (capture d'écran ci-dessus). Si le serveur est un serveur Microsoft® SharePoint®, cochez la case *Il s'agit d'un serveur Microsoft® SharePoint®*. Consultez les notes sur le serveur Microsoft® SharePoint® ci-dessous pour plus d'informations sur l'utilisation des fichiers sur ce type de serveur.
3. Si le serveur est protégé par un mot de passe, entrez votre identifiant utilisateur et votre mot de passe dans les champs *Utilisateur* et *Mot de passe*.
4. Cliquez sur **Parcourir** pour afficher et naviguer dans la structure des répertoires du serveur.
5. Dans l'arborescence des dossiers, recherchez le fichier que vous souhaitez charger et cliquez dessus.



L'URL du fichier apparaît dans le champ URL de fichier (*capture d'écran ci-dessus*). Le bouton **Ouvrir** ne devient actif qu'à ce stade.

6. Cliquez sur le bouton **Ouvrir** pour charger le fichier. Le fichier que vous ouvrez apparaît dans la fenêtre principale.

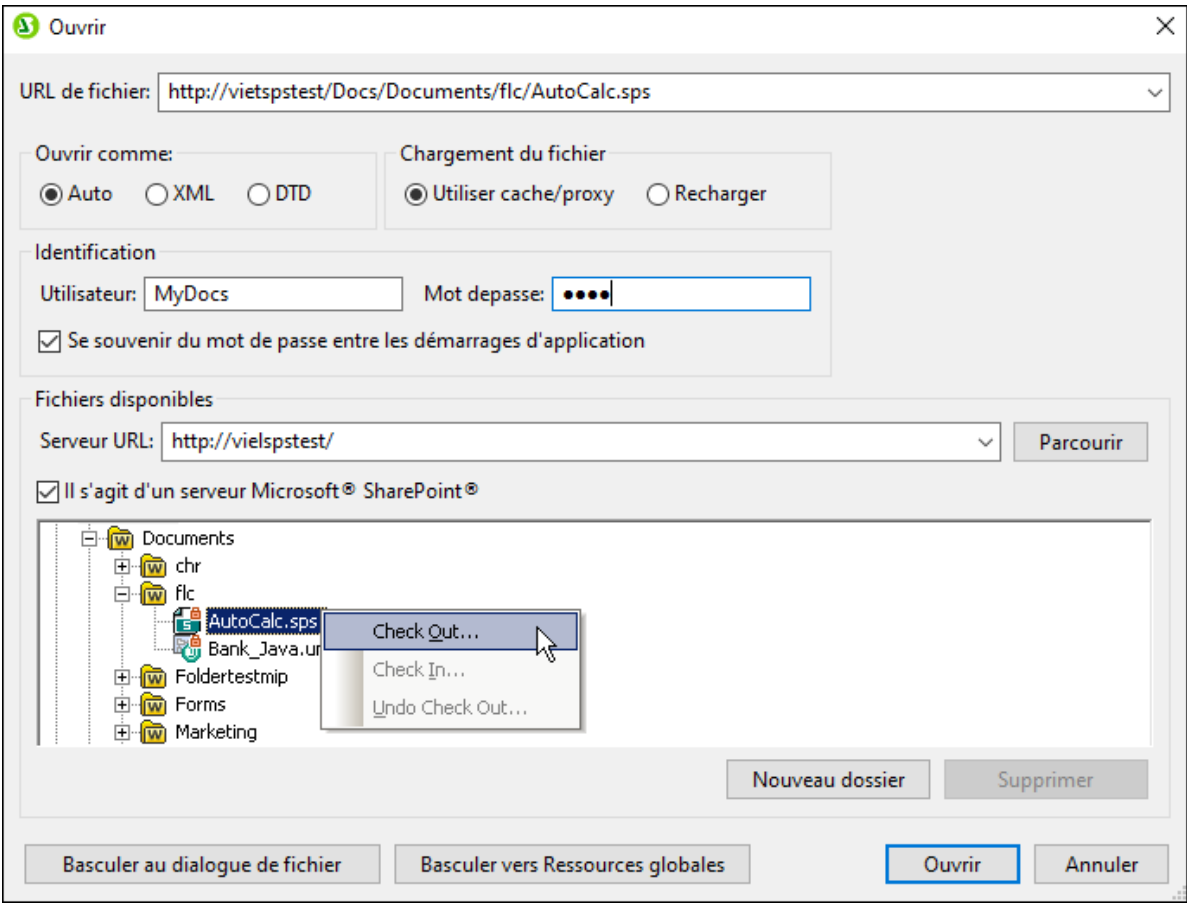
Note : la fonction Parcourir n'est disponible que sur les serveurs prenant en charge WebDAV et sur les serveurs Microsoft SharePoint. Les protocoles pris en charge sont FTP, HTTP et HTTPS.

Note : pour vous permettre de mieux contrôler le processus de chargement, vous pouvez choisir de charger le fichier via le cache local ou un serveur proxy (ce qui accélère considérablement le processus si le fichier a déjà été chargé auparavant). Vous pouvez également recharger le fichier si vous travaillez, par exemple, avec un système de publication électronique ou une base de données ; dans ce cas, sélectionnez l'option **Recharger**.

Notes sur le serveur Microsoft® SharePoint®

Notez les points suivants concernant les fichiers sur les serveurs Microsoft® SharePoint® :

- dans la structure de répertoires qui apparaît dans le volet Fichiers disponibles (*capture d'écran ci-dessus*), les icônes de fichiers comportent des symboles qui indiquent l'état d'archivage/d'extraction des fichiers.

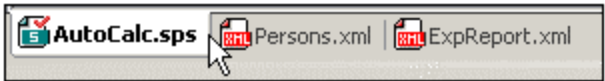


Un clic droit sur un fichier fait apparaître un menu contextuel contenant les commandes disponibles pour ce fichier (*capture d'écran ci-dessus*).

- Les différentes icônes de fichier sont affichées ci-dessous :

	Archivé. Disponible pour extraction.
	Extrait par un autre utilisateur. Non disponible pour l'extraction.
	Extrait localement. Peut être édité et archivé.

- 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 modifié via le menu contextuel dans le dialogue Ouvrir URL (*voir capture d'écran ci-dessus*), ou via le menu contextuel qui s'affiche lorsque vous cliquez sur l'onglet du fichier de la fenêtre principale de votre application (*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.

- 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**.

11.4.2 Ouvrir, Recharger, Fermer, Fermer tout

La commande **Ouvrir** (**Ctrl+O**)  vous permet d'ouvrir un fichier SPS ou PXF existant. Le [dialogue Ouvrir](#) ⁴⁵⁴ familier des systèmes Windows s'ouvre et vous permet de sélectionner un fichier avec une extension `.sps`.

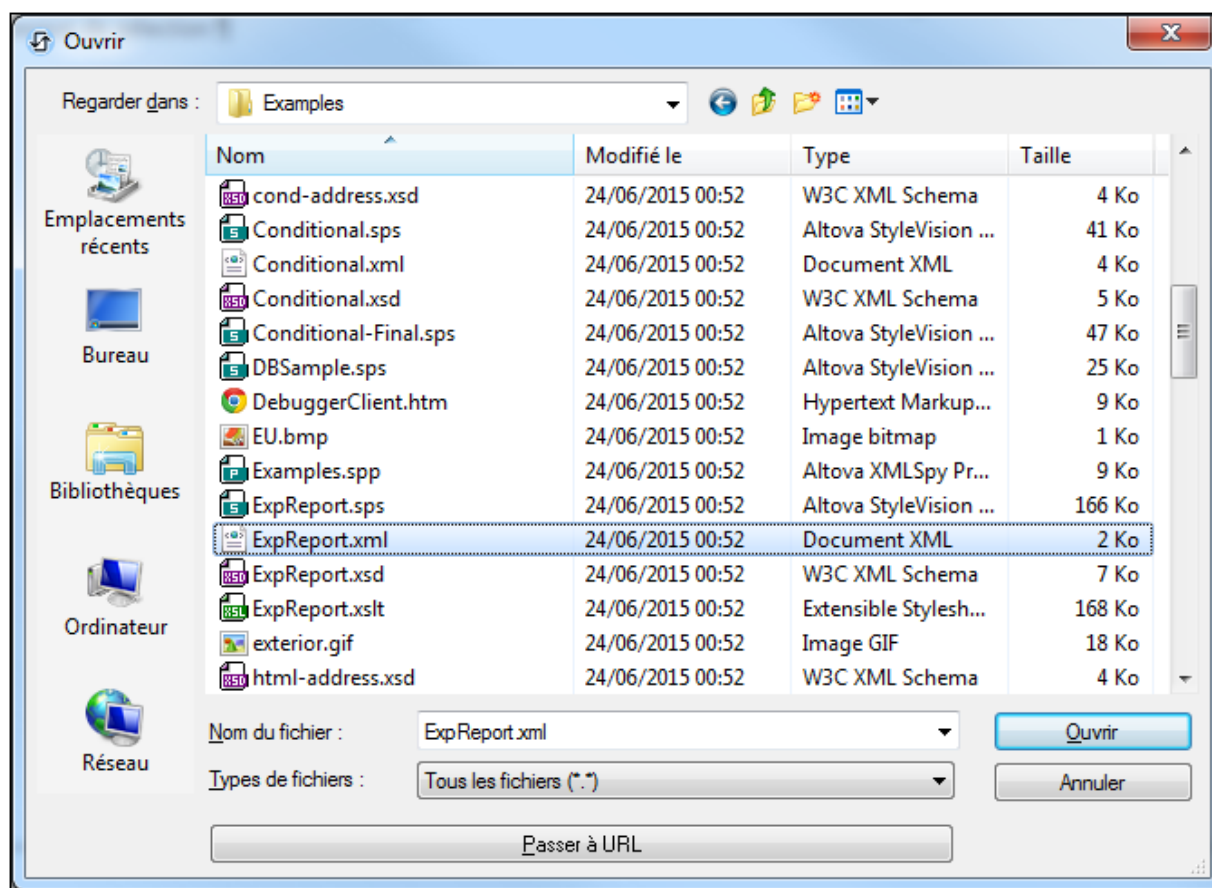
La commande **Recharger** recharge le fichier SPS à partir du fichier enregistré sur le disque. Toutes les modifications apportées depuis la dernière sauvegarde du fichier seront perdues. Le fichier XML de travail sera également rechargé, ce qui vous permettra de le mettre à jour s'il a été modifié en externe.

La commande **Fermer** ferme le document SPS actuellement actif. Notez que plusieurs fichiers peuvent être ouverts, mais qu'un seul est actif. Le document actif peut également être fermé en cliquant sur le bouton **Fermer** en haut à droite de la [fenêtre principale](#) ²⁷. Si vous avez apporté des modifications non enregistrées au document, vous serez invité à les enregistrer.

La commande **Fermer tout** ferme tous les documents SPS ouverts. Si vous avez apporté des modifications non enregistrées à un document ouvert, vous serez invité à les enregistrer.

▼ Sélectionner et enregistrer les fichiers via des URL et des Ressources globales

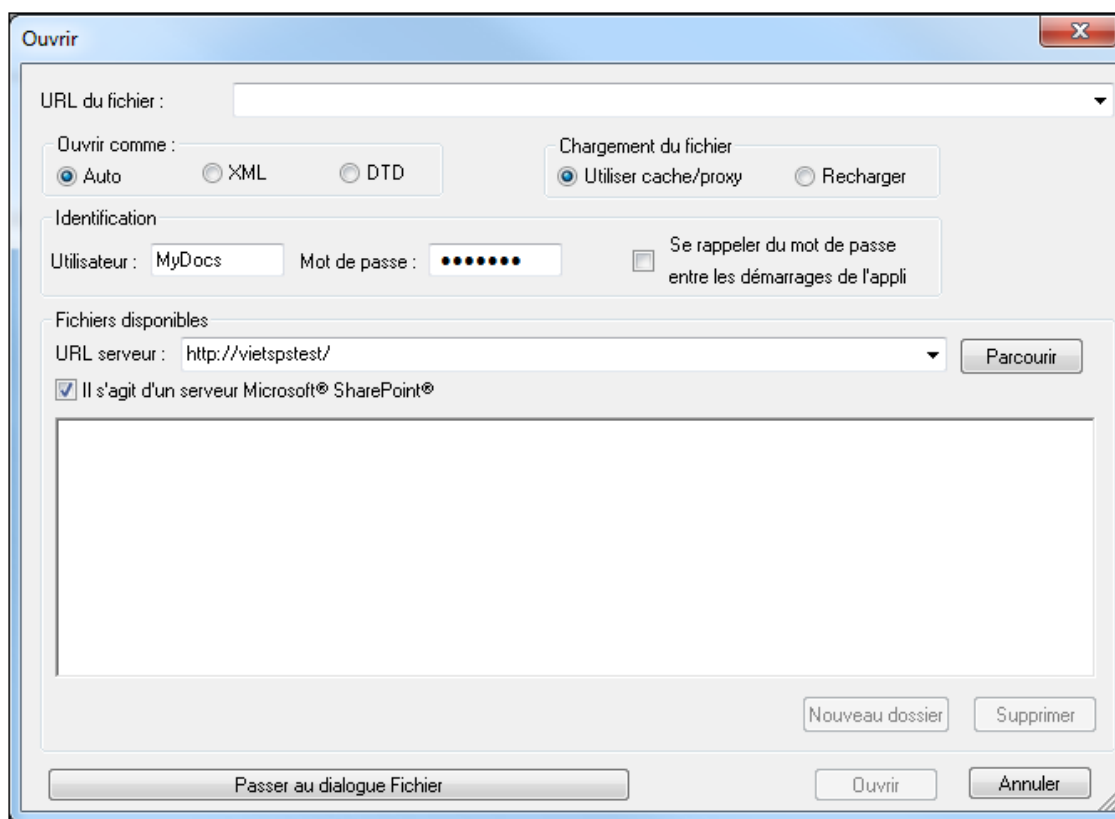
Dans plusieurs dialogues Fichiers Ouvrir et Fichier Enregistrer, vous pouvez choisir de sélectionner le fichier nécessaire ou d'enregistrer un fichier via une URL ou une Ressource globale (*voir capture d'écran ci-dessous*). Cliquer sur **Passer à URL** ou sur **Passer à Ressources globales** pour vous rendre sur un de ces processus de sélection.



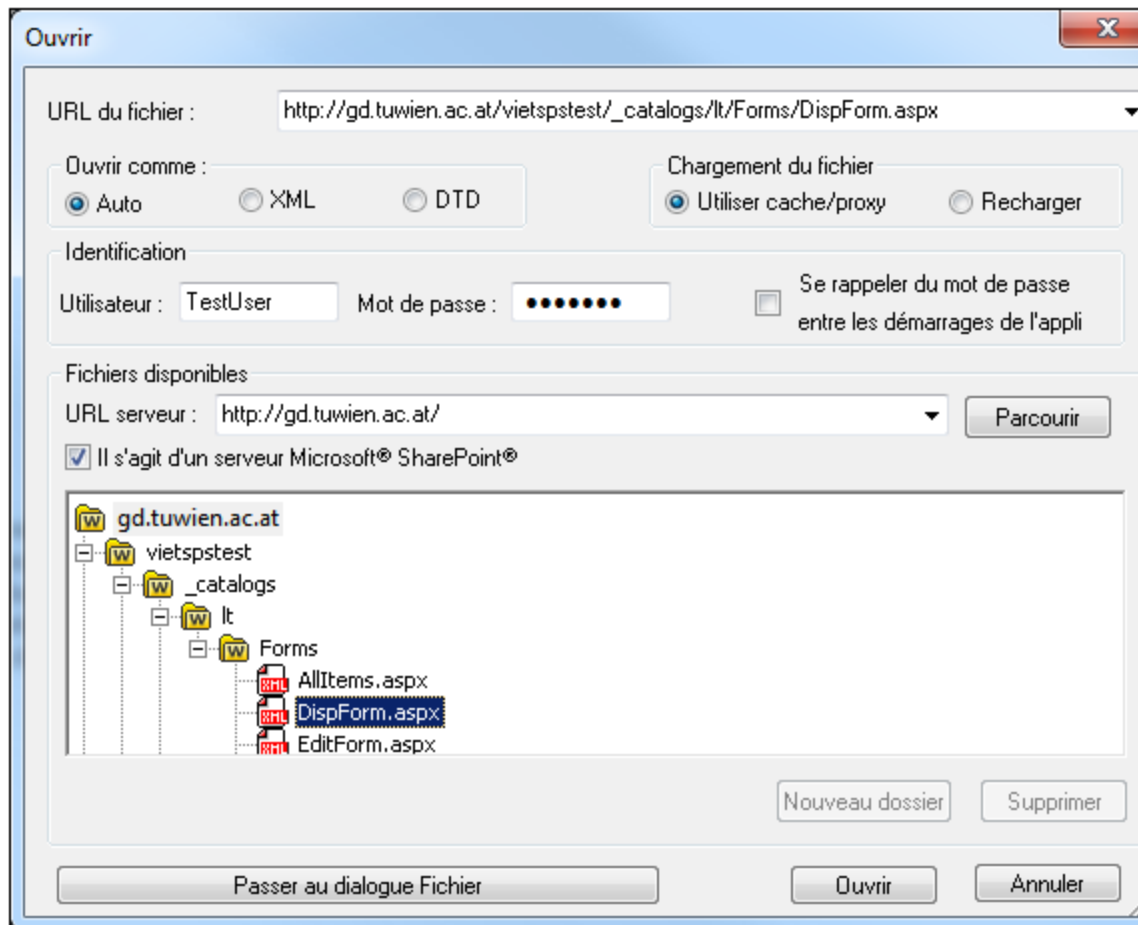
Sélectionner les fichiers via des URL

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).



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.

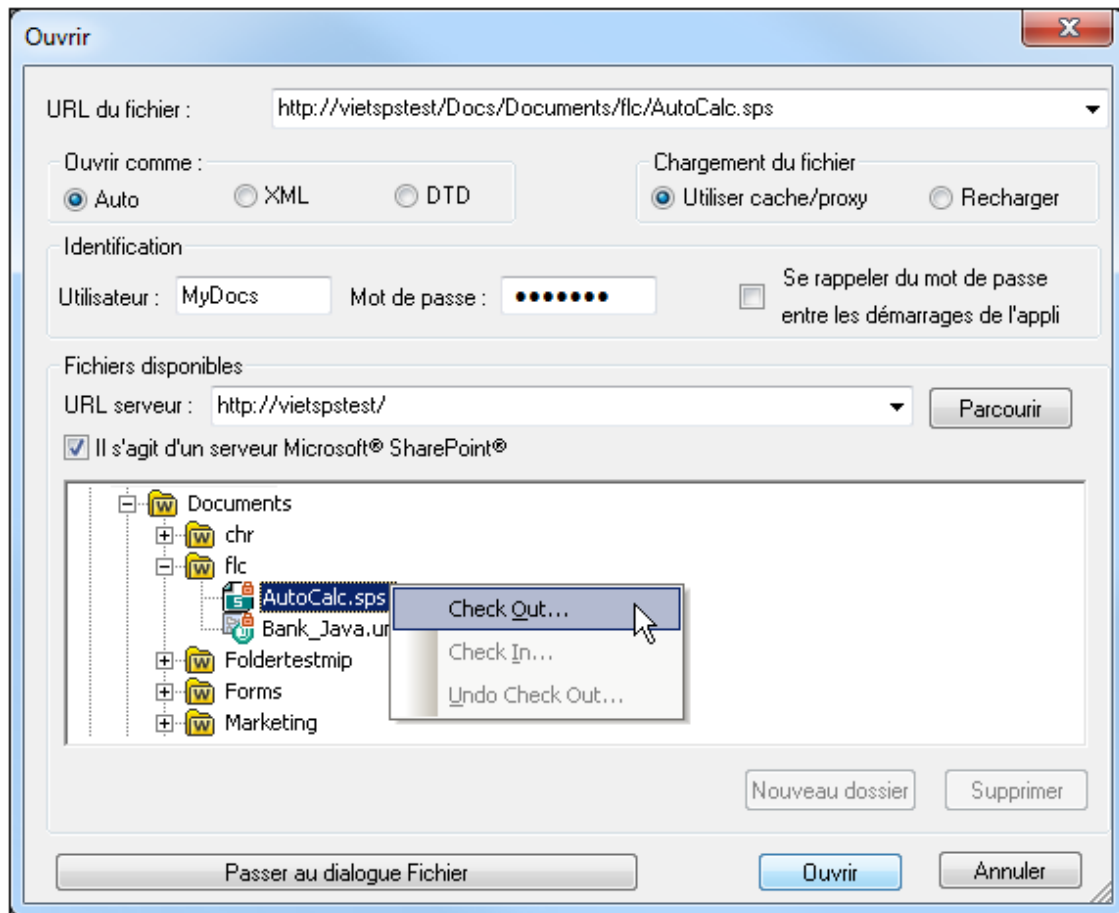
Notez les points suivants :

- 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

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.



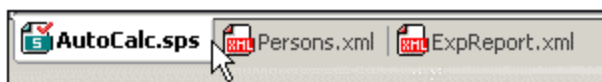
Cliquer avec la touche de droite sur un fichier, pour ouvrir un menu contextuel contenant les commandes disponibles pour ce fichier (*capture d'écran ci-dessus*).

- 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. Pour une description générale des Ressources globales, consulter la section dans cette documentation.

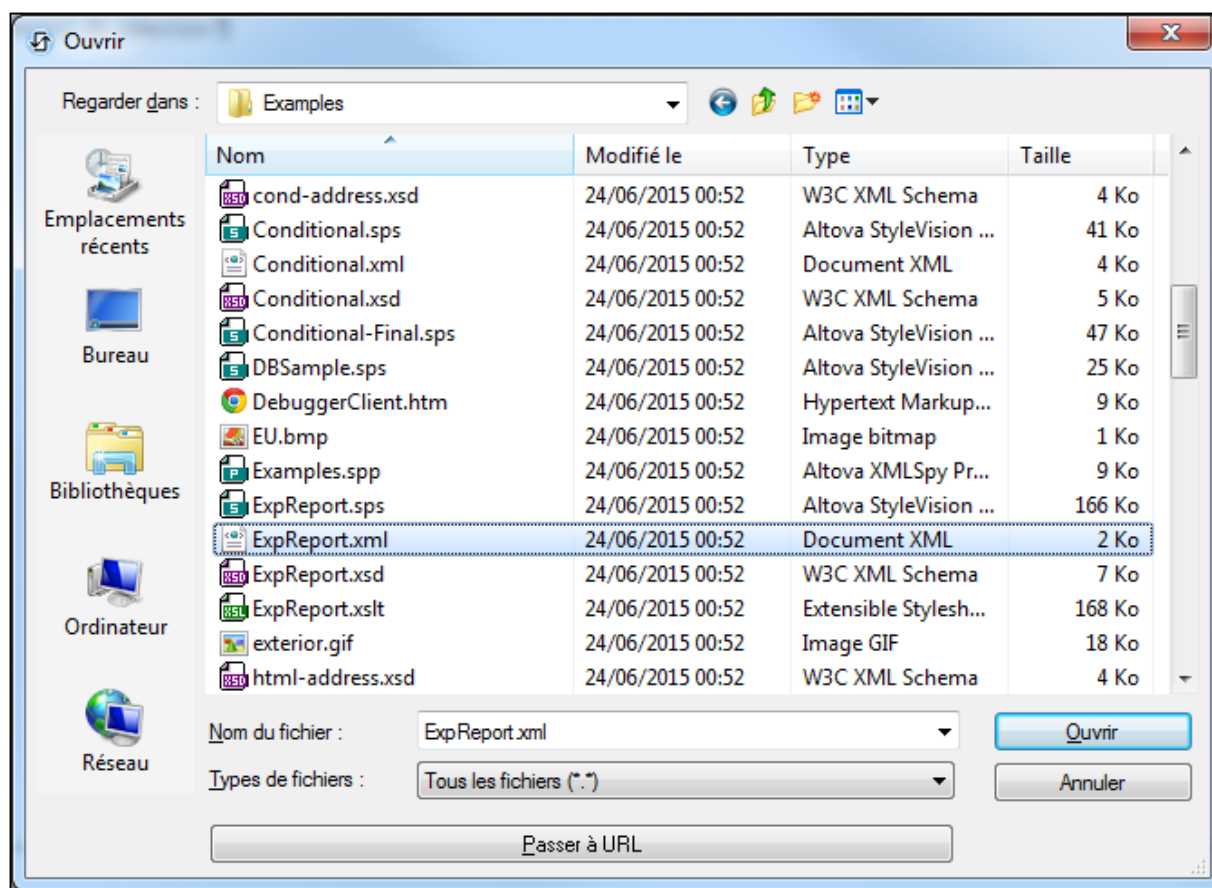
11.4.3 Enregistrer design, Tout enregistrer

La commande **Enregistrer design (Ctrl+S)**  enregistre le document actuellement ouvert sous forme de fichier SPS (avec l'extension `.sps`).

La commande **Tout enregistrer (Ctrl+Maj+S)**  enregistre tous les documents SPS ouverts.

▼ Sélectionner et enregistrer les fichiers via des URL et des Ressources globales

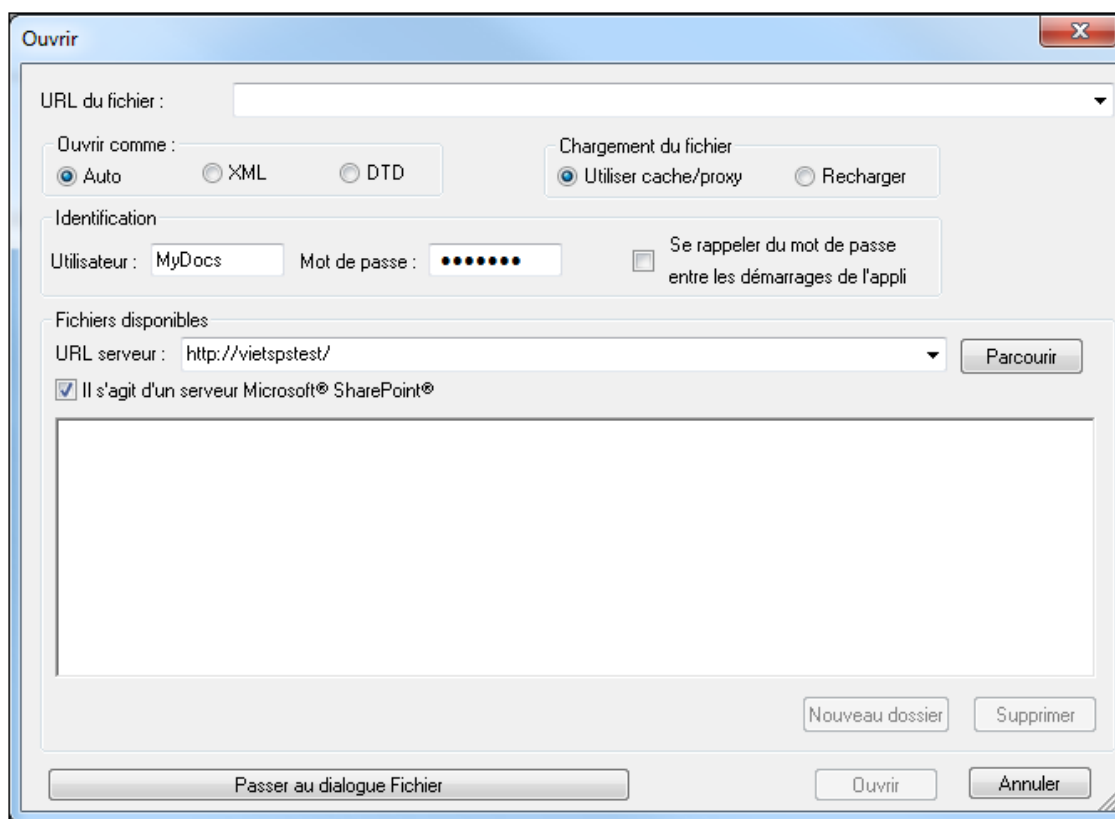
Dans plusieurs dialogues Fichiers Ouvrir et Fichier Enregistrer, vous pouvez choisir de sélectionner le fichier nécessaire ou d'enregistrer un fichier via une URL ou une Ressource globale (voir capture d'écran ci-dessous). Cliquer sur **Passer à URL** ou sur **Passer à Ressources globales** pour vous rendre sur un de ces processus de sélection.



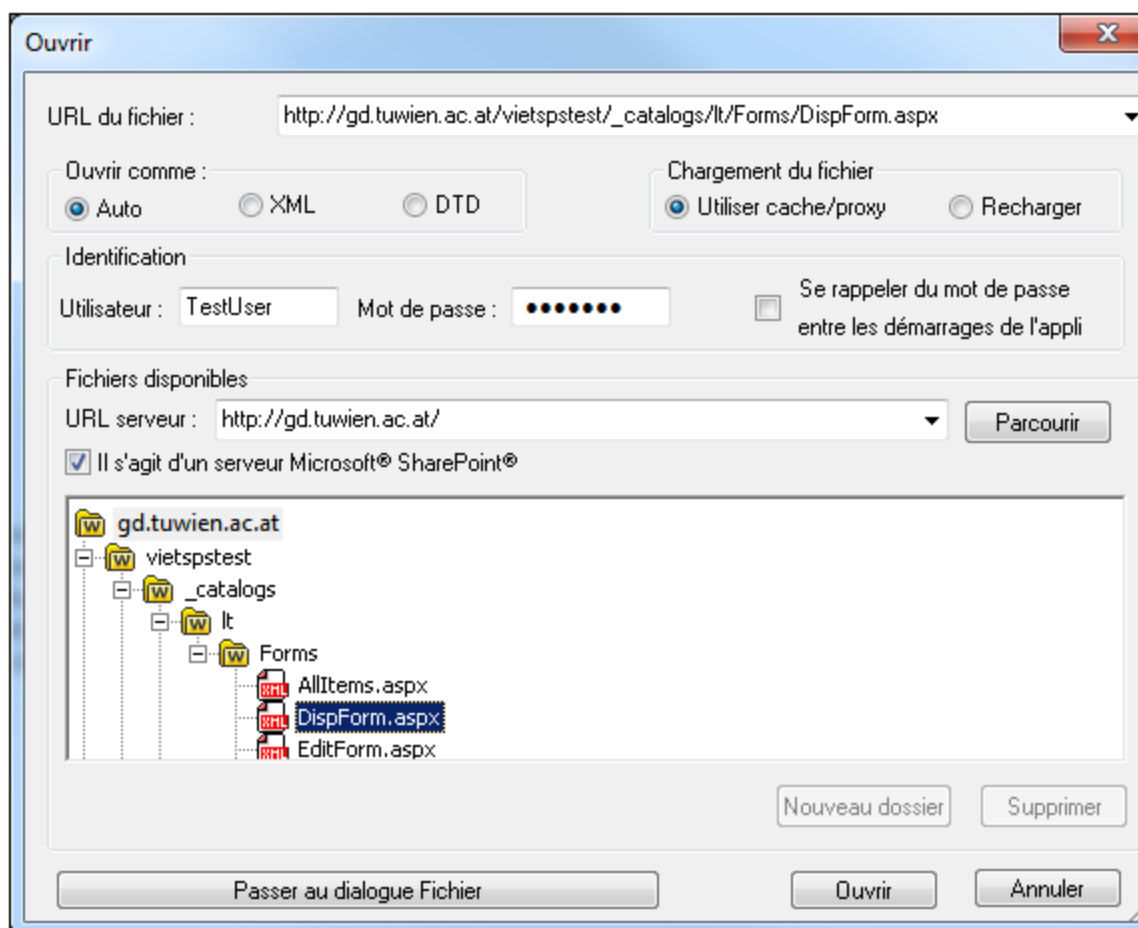
Sélectionner les fichiers via des URL

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).



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.

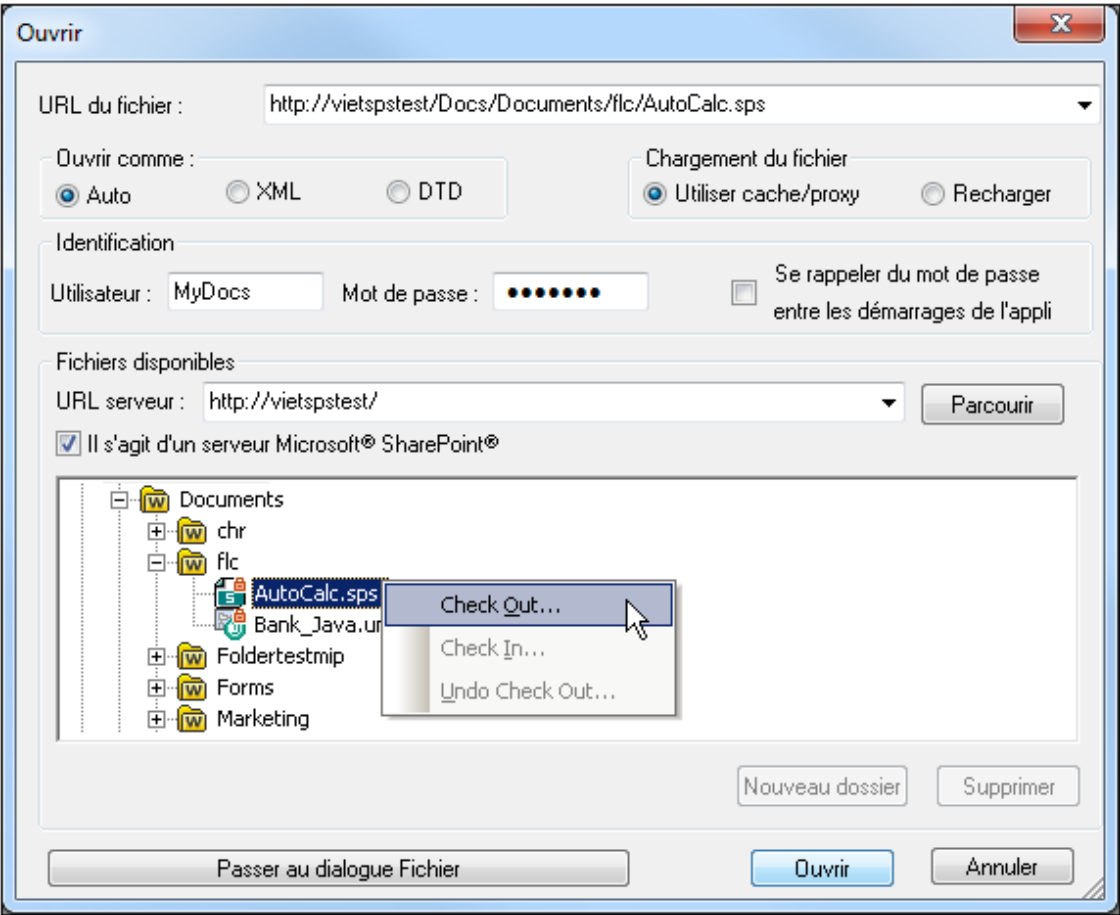
Notez les points suivants :

- 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

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.



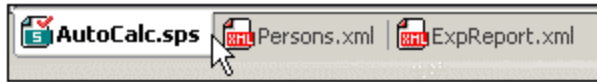
Cliquer avec la touche de droite sur un fichier, pour ouvrir un menu contextuel contenant les commandes disponibles pour ce fichier (*capture d'écran ci-dessus*).

- 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).



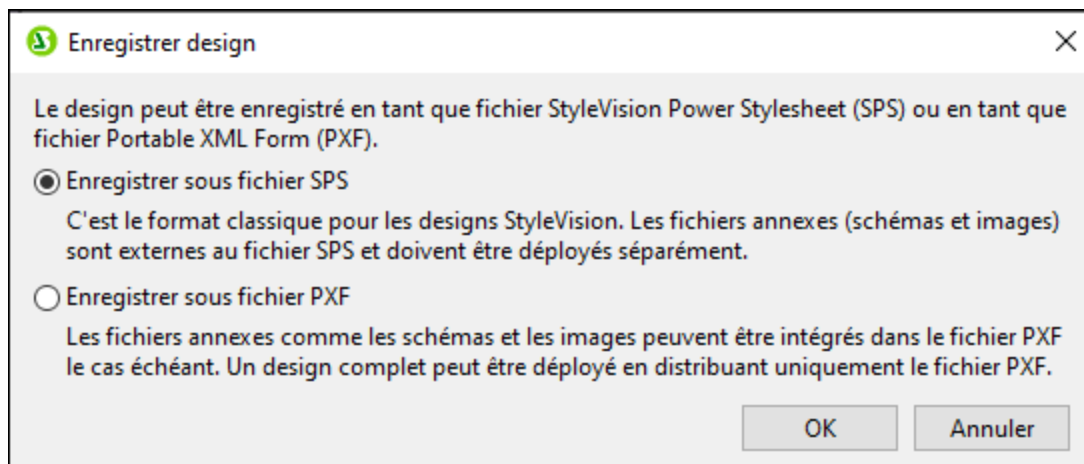
- Lorsqu'un fichier est récupéré par un autre utilisateur, il n'est plus disponible pour une récupération.
- Lorsqu'un fichier est récupéré localement par vous, vous pouvez annuler la récupération avec la commande Undo Check-Out (Annuler récupération) dans le menu contextuel. En conséquence, le fichier 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. Pour une description générale des Ressources globales, consulter la section dans cette documentation.

11.4.4 Enregistrer sous

La commande **Enregistrer sous** permet d'enregistrer le design : (i) en tant que fichier SPS ou (ii) en tant que fichier PXF (fichier Portable XML Form). Cliquer sur la commande affiche le dialogue Enregistrer design (capture d'écran ci-dessous). Sélectionnez le format souhaité et cliquez sur **OK**.



Le **format SPS** est le format Altova standard pour les designs StyleVision. Le **format PXF** est un format Altova qui permet d'intégrer à la conception tous les fichiers associés à celle-ci (fichiers de schéma, fichiers XML, fichiers image, feuilles de style XSLT générées, etc.). Ce format est très utile pour transporter tous les fichiers nécessaires à l'ouverture du design dans Authentic View et/ou pour générer des sorties HTML basées sur la conception.

Enregistrer sous SPS

La sélection de l'option SPS fait apparaître le dialogue Enregistrer sous familier des systèmes Windows.

L'enregistrement fonctionne exactement comme décrit pour la [commande](#) ⁴⁶³ **Enregistrer design** ⁴⁶³.

L'avantage de la commande **Enregistrer sous** est que les fichiers déjà enregistrés sous un nom peuvent être enregistrés sous un autre nom.

11.4.5 Exporter en tant que fichier MobileTogether Design

Cette commande génère un fichier de conception Altova MobileTogether à partir du design SPS actif. Un fichier de conception MobileTogether est utilisé pour exécuter des solutions dans l'application MobileTogether pour appareils mobiles. Par exemple, une solution MobileTogether peut être ouverte sur un appareil mobile, tel qu'un smartphone, pour afficher et modifier le contenu d'une base de données. Une solution MobileTogether est conçue dans Altova MobileTogether Designer. Cette commande vous permet de convertir un design SPS en un design MobileTogether qui peut être modifié dans MobileTogether Designer. Pour plus d'informations, consultez la [page web MobileTogether](#) et la [documentation MobileTogether Designer](#). Les options de conversion sont disponibles dans l'onglet Design du dialogue Options ([Outils | Options](#) ⁵³⁵).

Note : toutes les fonctionnalités de conception SPS n'ont pas de correspondance dans les designs MobileTogether. Après avoir exécuté cette commande, il est donc recommandé d'ouvrir le fichier généré dans MobileTogether Designer afin de le vérifier et, si nécessaire, de le corriger. Les fonctionnalités de conception suivantes ne sont pas exportées vers les designs MobileTogether : (i) les sources autres que la première répertoriée dans le design StyleVision (une fonctionnalité Enterprise) ; (ii) [modèles globaux](#) ²³⁴ ; (ii) [modules](#) ²²⁰.

11.4.6 Enregistrer les fichiers générés

La commande **Enregistrer les fichiers générés** affiche un sous-menu contenant des options permettant d'enregistrer les fichiers suivants (*capture d'écran ci-dessous*). Pour comprendre comment les fichiers générés s'intègrent dans la procédure d'utilisation générale, voir [Procédure d'utilisation | Fichiers générés](#) ¹⁰².

Enregistrer le(s) fichier(s) XSLT-HTML généré(s)

La commande Enregistrer le(s) fichier(s) XSLT-HTML généré(s) génère un fichier XSLT pour la sortie HTML à partir de votre SPS. Vous pouvez utiliser ce fichier XSLT par la suite pour transformer un document XML en HTML.

Enregistrer le(s) fichier(s) HTML généré(s)

La commande Enregistrer le(s) fichier(s) HTML généré(s) génère un ou plusieurs fichiers HTML. Plusieurs fichiers HTML seront générés si la [sortie multiple de document](#) ²⁵¹ a été spécifiée dans le design. Cette opération nécessite deux fichiers d'entrée :

- Le fichier XML de travail attribué au fichier SPS actuellement actif. Si aucun fichier XML de travail n'a été attribué, la commande **Enregistrer le fichier HTML généré** est désactivée.
- Un fichier XSLT, qui est généré automatiquement à partir du fichier SPS actuellement actif.

Enregistrer le schéma défini par l'utilisateur généré

Cette commande est activée lorsque le SPS implique un schéma défini par l'utilisateur. Le schéma que vous créez dans la barre latérale Arborescence de schéma est enregistré en tant que schéma XML avec l'extension `.xsd`.

Enregistrer les données XML définies par l'utilisateur générées

Les données du fichier HTML importé qui correspondent au schéma défini par l'utilisateur sont enregistrées sous forme de fichier XML. Les données correspondantes sont les nœuds du document HTML (dans la vue Design) qui ont été créés en tant que nœuds de schéma XML.

11.4.7 Déployer vers FlowForce

La commande **Déployer vers FlowForce** vous permet de déployer un fichier `.transformation` sur votre serveur Altova FlowForce Server. Le fichier `.transformation` contient tous les fichiers et informations nécessaires pour effectuer les transformations telles qu'elles ont été conçues dans le SPS. Une fois le fichier `.transformation` déployé sur le serveur FlowForce, vous pouvez créer des tâches dans Altova FlowForce qui utilisent le fichier `.transformation` pour générer des transformations en fonction des déclencheurs spécifiés dans la définition de la tâche. Pour plus d'informations sur la création de tâches FlowForce, consulter la documentation FlowForce.

Un fichier `.transformation` est généré à partir d'un fichier Portable XML Format (PXF). La commande **Déployer vers FlowForce** peut donc être utilisée lorsqu'un fichier PXF est actif. (Si un fichier SPS est actif, la commande **Déployer vers FlowForce** sera active, mais en cliquant dessus, vous serez invité à enregistrer le fichier SPS en tant que fichier PXF. Pour créer un fichier PXF à partir d'un fichier SPS, utilisez la commande **Fichier | Enregistrer sous** et sélectionnez PXF comme format d'enregistrement.)

Notez les points suivants :

- Lorsqu'un fichier PXF est enregistré, une option permet d'y inclure des fichiers externes (tels que des fichiers image). Si un fichier externe n'est pas inclus dans le fichier PXF mais est nécessaire pour la transformation, il doit être enregistré sur le serveur FlowForce. Étant donné que les fichiers externes seront accessibles à partir du répertoire de travail (spécifié dans la définition de la tâche FlowForce), ils doivent être placés par rapport au répertoire de travail, de manière à ce que les liens provenant du répertoire de travail puissent y accéder correctement.
- Lorsqu'une tâche FlowForce nécessitant une transformation StyleVision est exécutée, celle-ci est transmise à StyleVision Server, qui extrait le contenu du fichier PXF vers le répertoire de travail spécifié dans les paramètres de la tâche. Afin d'éviter tout conflit de noms de fichier lors de cette extraction, aucun fichier du répertoire de travail ne doit porter le même nom qu'un fichier contenu dans le fichier PXF.

Avant d'exécuter la commande **Déployer vers FlowForce**, assurez-vous qu'Altova FlowForce Server et Altova StyleVision Server sont correctement licenciés et fonctionnent. Consultez la documentation Altova FlowForce pour plus d'informations sur la configuration de FlowForce Server. (StyleVision Server est fourni avec FlowForce Server.)

La commande Déployer

La commande **Déployer vers FlowForce** fait apparaître le dialogue Déployer la transformation (*capture d'écran ci-dessous*).

Déployer la transformation

Saisir le nom de l'hôte et le port d'une interface d'administration FlowForce pour déployer la transformation actuelle.

Serveur: 127.0.0.1 Port: 8082

Utilisateur: root ☐ Utiliser SSL

Mot de passe: ●●●●

Connexion: <Défaut>

Déployer en tant que

Chemin: /public/YearlySales.transformation

Le chemin doit débuter avec une barre oblique.

☐ Enregistrer les modifications du design avant de le déployer

☒ Ouvrir le navigateur web pour créer une nouvelle tâche

Dans ce dialogue, vous spécifiez les éléments suivants :

- L'adresse et le numéro de port du serveur Web FlowForce (et non du serveur FlowForce), ainsi que les détails d'accès (utilisateur et mot de passe) pour le serveur FlowForce.
- Le nom du fichier de transformation et l'emplacement sur le serveur FlowForce où il doit être enregistré. Le chemin d'accès au fichier doit commencer par une barre oblique, qui représente le répertoire racine du serveur FlowForce.
- Si des modifications ont été apportées au design depuis la dernière sauvegarde du fichier, la case à cocher *Enregistrer les modifications du design avant de le déployer* sera activée. Cochez cette case si vous souhaitez enregistrer ces modifications ; sinon, décochez-la.
- Pour déployer le mappage via une connexion cryptée SSL, cochez la case **Utiliser SSL**. Cela suppose que FlowForce Server est déjà configuré pour accepter les connexions SSL. Pour plus d'informations, veuillez vous référer à la documentation FlowForce Server

En cliquant sur **OK**, le fichier `.transformation` est déployé sur FlowForce Server à l'emplacement spécifié. Si vous avez coché la case *Ouvrir le navigateur web pour créer une nouvelle tâche* (voir *capture d'écran ci-dessus*), l'interface FlowForce Web Server s'ouvre dans un navigateur web et la tâche créée lors de l'étape de déploiement peut être éditée directement dans le navigateur.

Plusieurs versions de StyleVision Server

Si le serveur sur lequel vous déployez le fichier `.transformation` dispose de plusieurs versions de StyleVision Server fonctionnant sous la gestion de FlowForce Server (applicable uniquement aux serveurs Windows), le dialogue *Sélectionner StyleVision Server* s'affiche et vous invite à spécifier la version de StyleVision Server avec laquelle vous souhaitez exécuter ce mappage. Vous pouvez sélectionner manuellement la version que vous souhaitez utiliser ou laisser le serveur sélectionner automatiquement la version la plus appropriée.

Ce dialogue apparaît lorsque le répertoire d'installation de FlowForce Server contient un fichier `.tool` pour chaque version de StyleVision Server qui s'exécute sous la gestion de FlowForce Server. Par défaut, un fichier `.tool` de StyleVision Server est ajouté automatiquement à ce répertoire lorsque vous installez StyleVision Server dans le cadre de l'installation de FlowForce Server. Le chemin d'accès où les fichiers `.tool` sont stockés dans FlowForce est le suivant : `C:\Program Files\Altova\FlowForceServer2027\tools`. Si vous disposez d'autres versions de StyleVision Server que vous souhaitez exécuter sous la gestion de FlowForce Server, leurs fichiers `.tool` devront peut-être être copiés manuellement dans le répertoire ci-dessus. Le fichier `.tool` de StyleVision Server se trouve à l'emplacement suivant : `C:\Program Files\Altova\StyleVisionServer2027\etc`.

Note : pour plus d'informations sur l'utilisation de FlowForce Server, consultez la [documentation FlowForce](#).

11.4.8 Design web

La commande **Design web** affiche un sous-menu contenant la commande **Générer application web ASPX**. Cette dernière commande génère tous les fichiers nécessaires à l'exécution d'une application ASPX, dans le dossier que vous avez spécifié. Un navigateur web lit le fichier ASPX qui est le document de sortie. Le code C# de ce fichier lance un processus dans lequel les données de la base de données source ou du fichier XML sont transformées de manière dynamique à l'aide d'un fichier XSLT dans le package ASPX. Le fichier ASPX (le document de sortie du processus de transformation) est mis à jour avec les dernières données de la base de données source ou du fichier XML.

Pour plus d'informations, voir [Interface ASPX pour les applications web](#) ⁴⁰¹.

11.4.9 Propriétés

La commande **Propriétés** ouvre le dialogue Propriétés (*la capture d'écran ci-dessous montre le dialogue dans l'édition Enterprise*), dans lequel vous pouvez définir diverses propriétés pour le SPS actif.

Sortie

Les propriétés suivantes peuvent être définies dans l'onglet Sortie :

- **Encodage de sortie :** Dans le volet Encodage de sortie, vous pouvez sélectionner l'encodage de vos documents de sortie. Si vous modifiez l'encodage dans ce dialogue, l'encodage du SPS actuellement actif est modifié. Vous pouvez également spécifier pour l'ensemble de l'application pour tous les documents SPS dans l'onglet Encodage du dialogue Options.
- **Mode de sortie HTML :** vous pouvez choisir de générer l'intégrité d'un document HTML ou uniquement les éléments enfants de l'élément HTML `body`. Les éléments enfants sont générés parallèlement les

uns aux autres, c'est-à-dire au même niveau, et contiennent tous les descendants de manière récursive. Par conséquent, les documents générés peuvent être des fragments de code HTML..

- **Mode de sortie HTML (DOCTYPE)** : vous pouvez choisir de générer un document HTML5, HTML 4.01 Transitional ou XHTML 1.0 Transitional pour la sortie HTML. Ce paramètre peut être modifié à tout moment lors de la création ou de la modification du document SPS.
- **Compatibilité avec Internet Explorer et prise en charge CSS** : la prise en charge CSS dans les versions d'Internet Explorer (IE) antérieures à IE 6 était incomplète et, à certains égards, mal interprétée. La prise en charge CSS a été améliorée et corrigée dans IE 6, puis encore améliorée et étendue dans IE 7, IE 9 et les versions ultérieures.

Dans un SPS, vous pouvez sélectionner le mode de compatibilité souhaité dans le dialogue Propriétés (*capture d'écran ci-dessus*). Vous pouvez sélectionner IE 5, IE 7 ou IE 9. (Notez que pour que la compatibilité IE 9 s'applique, IE 9 ou une version ultérieure doit être installée.) Le niveau de prise en charge IE spécifié est immédiatement disponible dans l'aperçu HTML. Notez que les nouveaux documents SPS sont créés avec la compatibilité IE7 sélectionnée. Les documents SPS créés dans des versions antérieures d'Altova StyleVision peuvent être réenregistrés dans le mode de compatibilité requis (sélectionné dans le dialogue [Propriétés](#)⁴⁷²).

XSD/XSLT

Dans cet onglet (*la capture d'écran ci-dessous montre le dialogue dans l'édition Enterprise*), vous pouvez spécifier quel validateur XSD utiliser pour la validation XML et quelle version XSLT utiliser dans le SPS.

StyleVision dispose à la fois d'un validateur XSD 1.0 et d'un validateur XSD 1.1. Vous pouvez choisir parmi les options suivantes :

- Utilisez le validateur XSD 1.1 si l'attribut `/xs:schema/@vc:minVersion` du document XSD est défini sur 1.1 ; sinon, utilisez le validateur XSD 1.0.
- Utilisez toujours le validateur XSD 1.1.
- Utilisez toujours le validateur XSD 1.0.

Sélectionnez la version XSLT pour le document actif dans cet onglet. Cochez l'option relative à l'instruction `xsl:import-schema` pour que l'élément `xsl:import-schema` des spécifications XSLT 2.0 et 3.0 soit inclus dans le document XSLT généré par StyleVision. Il est recommandé de sélectionner cette option afin que les types de données puissent être lus à partir du schéma s'il n'y a pas d'attribut `xsi:schemaLocation` dans le document XML.

Incorporation d'images dans la sortie HTML

Les graphiques, codes-barres et images en ligne peuvent être intégrés dans la sortie HTML. Si cette case est cochée, les données d'image sont converties en une chaîne Base64 qui est stockée dans l'attribut `src` de l'élément `img` HTML à l'aide du schéma URI de données. Ce type d'intégration n'est disponible que lorsque la version XSLT du SPS est 2.0 ou 3.0.

Chemins

Les chemins d'accès par défaut pour divers fichiers créés par le fichier SPS et pour les chemins d'accès enregistrés dans le fichier SPS sont spécifiés dans les paramètres de cet onglet (*la capture d'écran ci-dessous montre le dialogue dans l'édition Enterprise*).

Les valeurs par défaut suivantes sont définies dans l'onglet Chemins :

- Si les fichiers d'aperçu sont créés dans le répertoire du fichier SPS ou dans le fichier XML de travail de la source de schéma principale.
- Où les autres fichiers générés (fichiers image, fichiers image de codes-barres, fichiers image de graphiques, etc.) sont créés.
- Si les chemins d'accès dans le fichier SPS sont relatifs uniquement lorsque le dossier cible se trouve dans l'arborescence du fichier SPS, ou relatifs même lorsque le dossier cible se trouve en dehors de l'arborescence du fichier SPS.

11.4.10 Aperçu avant impression, Imprimer

Aperçu avant impression

La commande **Aperçu avant impression**  est activée dans le mode Design et Authentic View (Authentic View est pris en charge uniquement dans les éditions Enterprise et Professional). La commande **Aperçu avant impression** ouvre une fenêtre contenant un aperçu du design SPS (lorsque le mode Design est actif) ou de Authentic View du fichier XML de travail lorsque Authentic View est actif. L'aperçu affiche le design avec ou sans balises, selon ce qui est affiché à l'écran.



Dans la fenêtre Aperçu avant impression, vous pouvez effectuer les opérations suivantes à l'aide des commandes de la barre d'outils située en haut de la page (*capture d'écran ci-dessus*) ainsi que des icônes de navigation en bas de la fenêtre.. Les commandes de la barre d'outils Aperçu avant impression, de gauche à droite, permettent de :

- imprimer la page à l'aide du bouton Imprimer ;
- définir l'orientation du papier en sélectionnant Portrait ou Paysage ;
- définir les propriétés de la page en cliquant sur **Mise en page** pour ouvrir le dialogue correspondant ;
- activer ou désactiver l'affichage et l'impression des en-têtes et des pieds de page ;
- ajuster l'affichage afin que la largeur ou la hauteur de la page occupe respectivement toute la largeur ou toute la hauteur de l'écran ;
- définir le nombre de pages affichées à l'écran ;
- modifier le facteur de zoom à l'aide des boutons Zoom avant et Zoom arrière ou en sélectionnant une valeur dans la liste déroulante.

Pour naviguer entre les pages de l'aperçu, utilisez les boutons de navigation situés en bas de l'aperçu ou saisissez le numéro de page dans le champ Page.

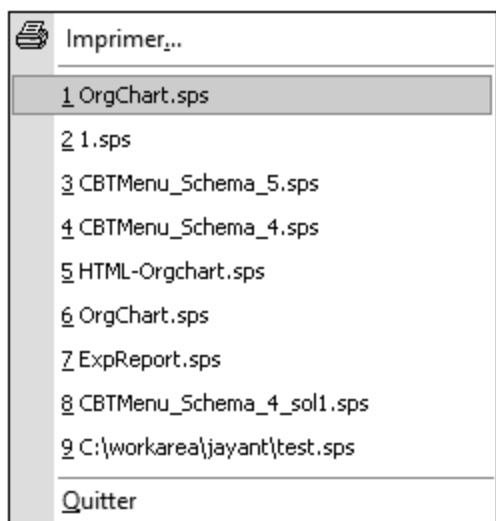
Imprimer

La commande **Imprimer**  est activée dans les onglets Authentic View et de prévisualisation de sortie. Elle imprime la vue sélectionnée du fichier XML de travail en fonction de la mise en page de cette vue. Notez que la mise en page de Authentic View peut être modifiée dans le dialogue Mise en page, accessible via la fenêtre Aperçu avant impression.

Note : pour activer les couleurs et les images d'arrière-plan dans l'aperçu avant impression, procédez comme suit : (i) dans le menu **Outils** d'Internet Explorer, cliquez sur Options Internet, puis sur l'onglet *Avancé* ; (ii) dans la zone Paramètres, sous Impression, cochez la case *Imprimer les couleurs et les images d'arrière-plan*, puis cliquez sur **OK**.

11.4.11 Fichiers récemment utilisés, Quitter

La liste des fichiers récemment utilisés affiche le nom et le chemin d'accès des neuf fichiers les plus récemment utilisés. En cliquant sur l'une de ces entrées, le fichier s'ouvre dans un nouvel onglet de la fenêtre principale.



Pour accéder à ces fichiers à l'aide du **clavier**, appuyez sur **ALT+F** pour ouvrir le menu Fichier, puis sur le numéro du fichier que vous souhaitez ouvrir. Par exemple, appuyez sur **1** pour ouvrir le premier fichier de la liste, sur **2** pour ouvrir le deuxième fichier, et ainsi de suite.

La commande **Quitter** permet de quitter StyleVision. Si vous avez un fichier ouvert contenant des modifications non enregistrées, vous serez invité à enregistrer ces modifications.

11.5 Menu Éditer

Le menu **Éditer** contient des commandes qui facilitent l'édition des documents SPS. Outre les commandes d'édition standard, telles que **Couper** (Maj+Suppr ou Ctrl+X), **Copier** (Ctrl+C), **Coller** (Ctrl+V) et **Supprimer** (Suppr), qui ne sont pas décrites dans cette section, les commandes suivantes sont disponibles :

- [Annuler, Rétablir, Tout sélectionner](#)⁴⁷⁶, pour annuler ou rétablir vos actions précédentes et pour sélectionner tout le contenu du SPS.
- [Rechercher, Trouver suivant, Remplacer](#)⁴⁷⁶, pour rechercher du texte dans les aperçus des feuilles de style SPS et XSLT.
- [Paramètres de feuille de style](#)⁴⁸², pour modifier les paramètres déclarés globalement pour le SPS.
- [Réduire/agrandir balisage](#)⁴⁸², pour réduire et développer les balises des composants de conception SPS.

Les commandes sont également disponibles via le menu contextuel qui s'affiche lorsque vous cliquez avec le bouton droit sur un composant ou sur le point d'insertion du curseur. De plus, certaines commandes sont disponibles sous forme de raccourcis clavier et/ou d'icônes de la barre d'outils. Notez toutefois que les commandes qui ne sont pas applicables dans une vue de document particulière ou à un emplacement donné sont grisées dans le menu.

11.5.1 Annuler, Rétablir, Sélectionner tout

La commande **Annuler** (Ctrl+Z)  vous permet d'annuler une modification apportée lors de l'édition. Un nombre illimité d'actions d'annulation est pris en charge. Chaque action peut être annulée et il est possible d'annuler une commande après l'autre jusqu'à la première action effectuée depuis l'ouverture du document.

La commande **Rétablir** (Ctrl+Y)  vous permet de rétablir un nombre illimité de commandes précédemment annulées. En utilisant les commandes Annuler et Rétablir, vous pouvez revenir en arrière et avancer dans l'historique des commandes.

La commande **Sélectionner tout** sélectionne tout le contenu de la fenêtre du document de Design.

11.5.2 Rechercher, Trouver suivant, Remplacer

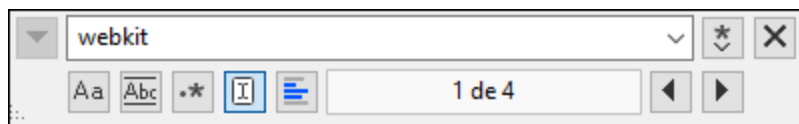
La commande **Rechercher** (Ctrl+F)  vous permet de rechercher des chaînes de texte dans le mode Design, l'éditeur JavaScript et les feuilles de style XSLT.

Le dialogue Rechercher & remplacer qui s'affiche dépend de la vue actuellement active.

- Lorsque le mode Design est actif, cliquer sur la commande **Rechercher** ou **Remplacer** place le focus sur la [barre latérale Rechercher & remplacer](#)⁵². La fonctionnalité de recherche et de remplacement du mode Design est décrite dans la rubrique [Barre latérale Rechercher & remplacer](#)⁵².
- La fonctionnalité de recherche et de remplacement dans l'éditeur JavaScript et les feuilles de style XSLT est décrite dans cette rubrique.

Feuilles de style XSLT et éditeur JavaScript

Cliquer sur la commande **Rechercher** dans l'onglet XSLT pour HTML ou Éditeur JavaScript affiche le dialogue suivant :



Vous pouvez choisir parmi les options suivantes :

- *Respecter la casse* : recherche sensible à la casse lorsque cette option est activée (*Address* n'est pas identique à *address*).
- *Correspondance avec le mot entier* : seuls les mots exacts du texte seront pris en compte. Par exemple, pour la chaîne de caractères *fit*, avec l'option *Correspondance avec le mot entier* activée, seul le mot *fit* correspondra à la chaîne de recherche ; le mot *fit* dans *fitness*, par exemple, ne sera pas pris en compte.
- *Expression régulière* : si cette option est activée, le terme recherché sera lu comme une expression régulière. Voir [Expressions régulières](#)⁴⁷⁸ ci-dessous pour obtenir une description de l'utilisation des expressions régulières.
- *Trouver ancre* : lorsqu'un terme de recherche est saisi, les correspondances dans le document sont mises en surbrillance et l'une d'entre elles est marquée comme sélection actuelle. L'option *Trouver ancre* détermine si cette première sélection actuelle est effectuée par rapport à la position du curseur ou non. Si l'option *Trouver ancre* est activée, la première correspondance sélectionnée sera celle située après la position actuelle du curseur. Si l'option *Trouver ancre* est désactivée, la première correspondance sélectionnée sera celle située en haut du document.
- *Chercher dans la sélection* : si cette option est activée, la sélection de texte actuelle est verrouillée et la recherche est limitée à cette sélection. Sinon, la recherche porte sur l'ensemble du document. Avant de sélectionner une nouvelle plage de texte, déverrouillez la sélection actuelle en désactivant l'option *Chercher dans la sélection*.
- *Remplacer* : dans l'onglet Éditeur JavaScript, cliquez sur le bouton **Flèche vers le bas**, situé en haut à gauche du dialogue, pour ouvrir le champ *Remplacer*. Vous pouvez y saisir la chaîne que vous souhaitez substituer à la chaîne trouvée.

Aperçu HTML

Cliquez sur la commande **Rechercher** dans Aperçu HTML pour ouvrir un dialogue simple [Rechercher et remplacer](#)⁵².

Veuillez noter les points suivants :

- Pour faire correspondre l'entrée à des mots entiers, cochez la case "Faire correspondre mot entier uniquement". Par exemple, une entrée *soft* ne trouvera que le mot entier *soft* ; elle ne trouvera pas, par exemple, le mot *soft* dans *software*.
- Pour faire correspondre l'entrée à des fragments de mots, laissez la case "Faire correspondre mot entier uniquement" décochée. Cela vous permettra, par exemple, de saisir *soft* et *software*.
- Pour que la recherche ne tienne pas compte de la casse, laissez la case "Respecter la casse" décochée. Cela vous permettra de trouver, par exemple, *Soft* avec une entrée *soft*.

Commande Trouver suivant

La commande Trouver suivant (F3)  répète la dernière commande Rechercher pour rechercher la prochaine occurrence du texte demandé. Voir [Rechercher](#) ⁴⁷⁶ pour obtenir une description de l'utilisation de la fonction de recherche.

Utiliser des expressions régulières

Vous pouvez utiliser des expressions régulières (regex) pour rechercher une chaîne de texte. Pour ce faire, activez d'abord l'option *Expression régulière* (voir ci-dessus). Cela indique que le texte saisi dans le champ de recherche doit être évalué comme une expression régulière. Ensuite, saisissez l'expression régulière dans le champ de recherche. Pour obtenir de l'aide sur la création d'une expression régulière, cliquez sur le bouton **Générateur d'expression régulière**, situé à droite du champ de recherche. Cliquez sur un élément dans le générateur pour saisir le ou les métacaractères regex correspondants dans le champ de recherche. Vous trouverez ci-dessous une brève description des métacaractères.

Métacaractères d'expression régulière

Ci-dessous, vous trouverez une liste des métacaractères 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 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\1YYY, 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 générera 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 <i>ensemble de caractères</i> . 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'expressions régulières

Cet exemple montre comment rechercher 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>
```

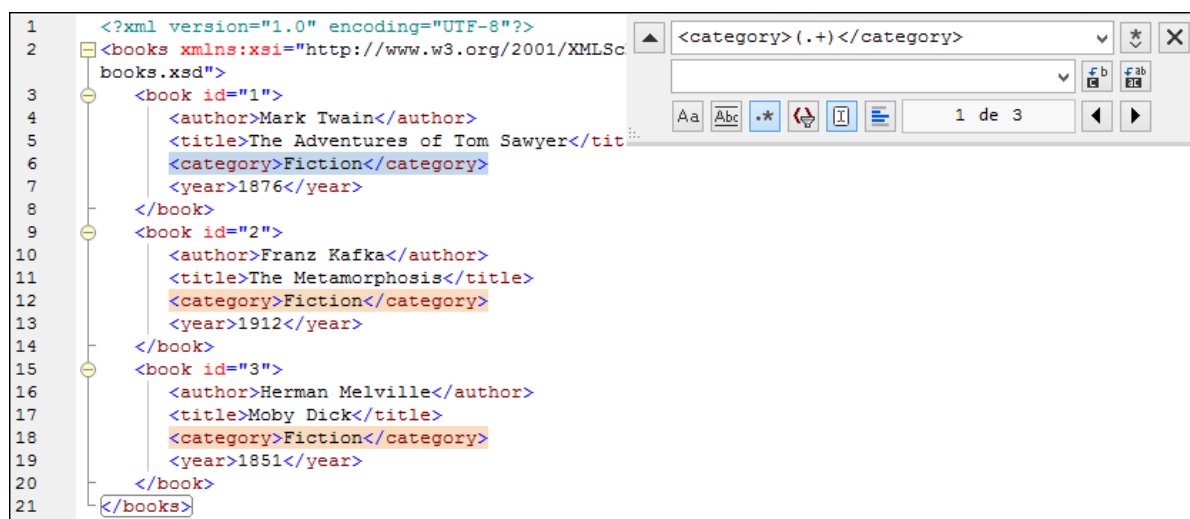
```

<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 rechercher 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.



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-référence `\1` à l'expression balisée précédemment du champ Rechercher. Dans d'autres 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`.

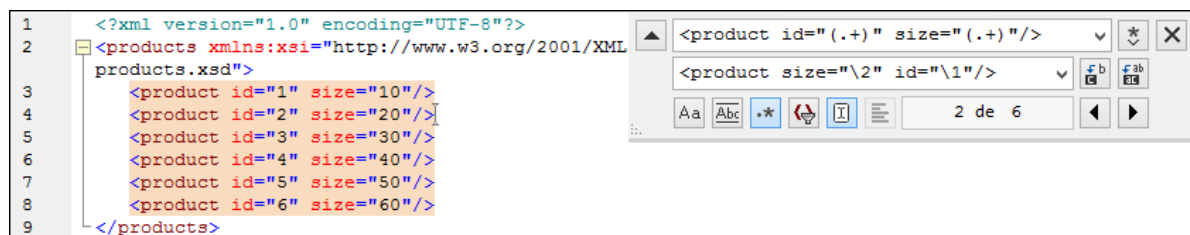
Exemple 2 : changer l'ordre des attributs

L'échantillon de code XML ci-dessous contient une liste de produits. Chaque élément de produit contient deux attributs : `id` et `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 moyens 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 :

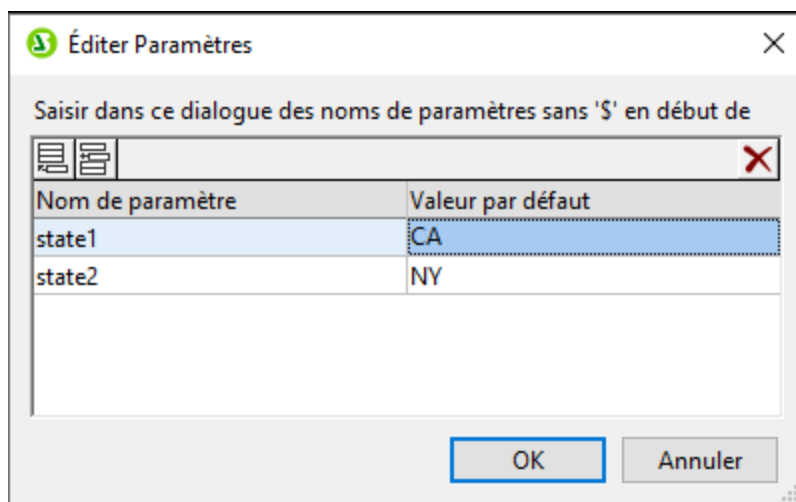
1. Appuyer sur **Ctrl+H** pour ouvrir la fenêtre Rechercher et remplacer.
2. Cliquer sur **Utiliser expression régulière** .
3. Dans le champ Rechercher, saisir le texte suivant : `<product id="(.)" size="(.)"/>`. Cette expression régulière correspond à un élément de produit dans le document XML. Veuillez 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 (présupposé 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 correspondent aux expressions balisées du champ Rechercher. 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`.

11.5.3 Paramètres de feuille de style

La commande **Paramètres de feuille de style**  vous permet de déclarer et de modifier les paramètres et leurs valeurs par défaut. Cette commande est disponible dans la vue document Design et dans la vue éditeur Authentic. Lorsque vous cliquez sur cette commande, le dialogue Éditer paramètres (*illustré ci-dessous*) s'affiche.



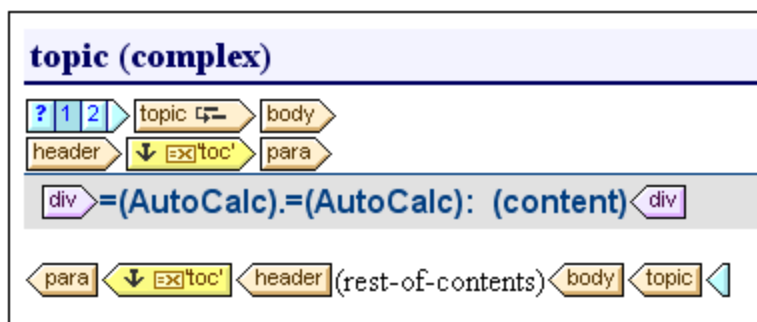
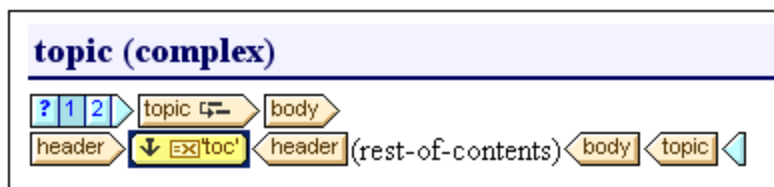
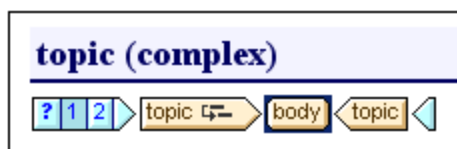
Veillez noter les points suivants :

- Vous pouvez insérer, ajouter, modifier et supprimer des paramètres pour l'ensemble de la feuille de style.
- Les noms de paramètres doivent commencer par une lettre et peuvent contenir les caractères A à Z, a à z, 0 à 9 et le trait de soulignement.
- Le dialogue Éditer paramètres contient tous les paramètres définis par l'utilisateur dans un SPS.
- Les paramètres peuvent également être déclarés dans la [barre latérale de l'Aperçu du design](#) ³⁵.

11.5.4 Réduire/agrandir balisage

La commande **Réduire/agrandir balisage** est une commande à bascule qui réduit et développe la balise sélectionnée. Elle peut être appliquée à tout type de balise : nœud, format prédéfini, mécanisme SPS, etc. Pour réduire/agrandir une balise, double-cliquez sur celle-ci ; vous pouvez également double-cliquer sur la balise de fin d'une balise développée pour la réduire.

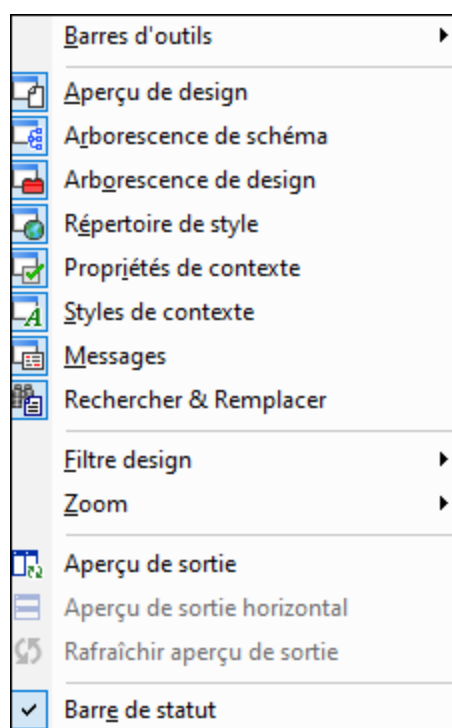
Les captures d'écran ci-dessous montrent comment une série de balises est réduite. Double-cliquez sur une balise réduite pour la développer.



La réduction d'une balise peut être utile pour optimiser l'affichage en fonction de vos besoins d'édition.

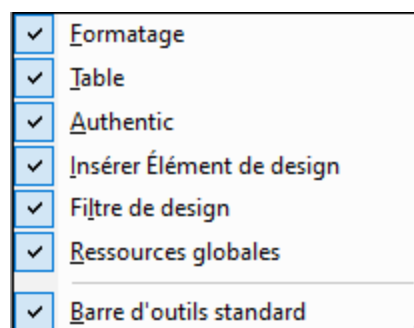
11.6 Menu Affichage

Le menu **Affichage** (*capture d'écran ci-dessous*) vous permet de modifier l'apparence de l'interface graphique et d'activer ou de désactiver l'affichage des composants de l'interface graphique. Vous pouvez activer ou désactiver l'affichage des barres d'outils individuelles, des barres latérales de design individuelles, des filtres de design et de la barre d'état.



11.6.1 Barre d'outils et Barre de statut

En plaçant le curseur sur la commande **Barre d'outils**, un sous-menu s'affiche (*capture d'écran ci-dessous*), qui vous permet d'activer et de désactiver l'affichage des différentes barres d'outils.



Lorsqu'une barre d'outils est cochée, elle s'affiche. Dans la capture d'écran ci-dessus, toutes les barres d'outils sont affichées. Pour activer ou désactiver l'affichage d'une barre d'outils, cliquez sur la barre d'outils.

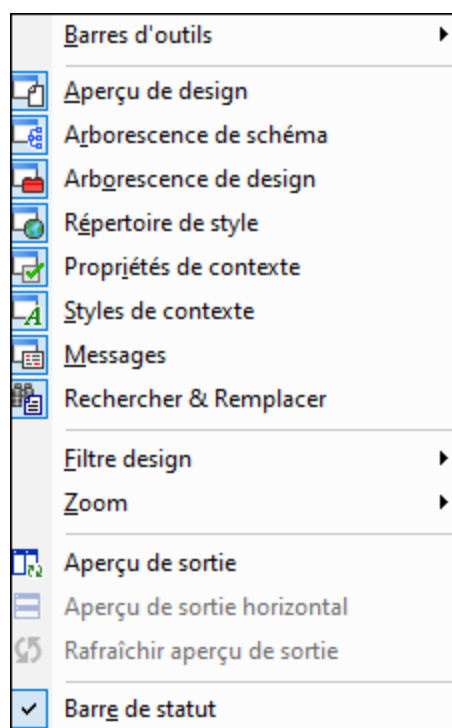
correspondante. Pour une description complète des barres d'outils, voir la section [Référence | Barres d'outils](#) ⁴⁴³.

Barre de statut

L'affichage de la barre de statut, située au bas de la fenêtre de l'application, peut être activé ou désactivé en cochant la commande **Barre de statut**.

11.6.2 Barres latérales de design

Le menu **Affichage** contient des commandes permettant d'activer et de désactiver l'affichage de chaque barre latérale (*capture d'écran ci-dessous*).



Lorsqu'une barre latérale est activée (l'icône de la commande est encadrée), elle s'affiche dans l'interface graphique. Cliquez sur une barre latérale pour activer ou désactiver son affichage, selon vos besoins. Cette commande permet également de rendre visible une barre latérale masquée. Le paramètre d'affichage spécifié pour une barre latérale est spécifique au mode : un paramètre défini dans un mode particulier (mode Design, mode de sortie, aucun document ouvert) est conservé pour ce mode particulier jusqu'à ce qu'il soit modifié.

11.6.3 Filtre design, Zoom

Filtre de design

L'élément de menu **Filtre design** déploie un sous-menu contenant des commandes qui vous permettent de filtrer les modèles affichés dans le mode Design. Cette fonction est utile si votre design est très long ou contient plusieurs modèles. Le mécanisme de filtre de design vous permet de spécifier les types de modèles à afficher. Les options de filtre suivantes sont disponibles :

Icon e	Commande	Description
	Afficher uniquement un modèle	Affiche uniquement le modèle sélectionné. Placer le curseur dans un modèle et cliquer pour afficher ce modèle uniquement.
	Afficher tous les types de modèles	Affiche tous les modèles dans SPS (principal, global, dénommé et layout) .
	Affiche les modèles importés	Bascule l'affichage des modèles importés sur activé/désactivé.
	Affiche/masque le modèle principal	Bascule l'affichage du modèle principal sur activé/désactivé.
	Affiche/masque les modèles globaux	Bascule l'affichage des modèles globaux sur activé/désactivé.
	Affiche/masque les fragments de design	Bascule l'affichage des fragments de design sur activé/désactivé.

Notez que ces commandes sont également disponibles sous forme d'icônes dans la barre d'outils [Filtres de design](#) ⁴⁴⁹.

Zoom

La commande **Zoom** vous permet de sélectionner un facteur de zoom dans le sous-menu qui s'affiche. Vous pouvez également zoomer ou dézoomer en modifiant le facteur de zoom dans la liste déroulante Zoom (dans la barre d'outils Standard) ou en appuyant sur la touche **Ctrl** et en faisant défiler avec la souris.

11.6.4 Aperçu de sortie

Il existe trois commandes d'aperçu de sortie :

- **Aperçu de sortie**, qui active ou désactive l'aperçu de sortie
- **Aperçu de sortie horizontal**, qui est activé lorsque l'aperçu de sortie est activé. Cette commande permet de passer d'une vue divisée verticalement à une vue divisée horizontalement. Lorsque l'aperçu de la sortie est disponible dans une vue divisée horizontale, la commande passe automatiquement à **Aperçu de sortie vertical**, qui rétablit la vue divisée verticale.

- **Rafraîchir aperçu de sortie**, qui actualise immédiatement l'aperçu de la sortie avec toutes les modifications apportées au design.

11.7 Menu Insérer

Le menu **Insérer** fournit des commandes vous permettant d'insérer divers composants de conception dans le SPS. Certaines de ces commandes sont disponibles sous forme d'[icônes de barre d'outils](#)⁴⁴³. De plus, les commandes du menu **Insérer** sont également disponibles via les **menus contextuels** qui apparaissent lorsque vous cliquez avec le bouton droit de la souris sur un point d'insertion du curseur dans le design SPS. Dans les menus contextuels, les commandes qui ne sont pas disponibles à cet emplacement dans le SPS sont désactivées.

Note : les commandes **Insérer** étant utilisées pour construire le SPS, elles ne sont disponibles que dans le mode Design.

11.7.1 Contenus

La commande **Contenus** insère un espace réservé (contenu) à l'emplacement du curseur. Cet espace réservé (contenu) peut être inséré dans deux types de nœuds, **élément** et **attribut**, et indique que tous les enfants du nœud actuel seront traités.

- Si le nœud actuel est un nœud élément, les nœuds éléments enfants et les nœuds texte du nœud seront traités. Pour le traitement des nœuds éléments enfants, les modèles globaux seront utilisés s'ils existent. Sinon, la règle de modèle intégrée pour les éléments sera utilisée. Pour le traitement des nœuds de texte, la règle de modèle intégrée pour les nœuds de texte sera utilisée, ce qui aura pour effet d'afficher le texte. En effet, la règle de modèle intégrée pour les éléments affiche le texte de tous les nœuds de texte descendants. Il est important de noter que les valeurs des attributs ne seront pas affichées lorsque l'espace réservé (contenu) est utilisé, sauf si un modèle global est défini pour l'élément parent de l'attribut ou l'un de ses ancêtres et que l'attribut est explicitement affiché, à l'aide de l'espace réservé (contenu) ou de tout autre composant de rendu de contenu.
- Si le nœud actuel est un nœud d'attribut, la règle de modèle intégrée pour le nœud de texte enfant de l'attribut sera utilisée. Ce modèle copie le texte du nœud de texte dans la sortie, affichant ainsi la valeur de l'attribut.

L'espace réservé (contenu) peut également être inséré pour un nœud en plaçant le curseur à l'intérieur des balises du nœud, en cliquant avec le bouton droit de la souris et en sélectionnant **Insérer | Contenus** ou en cliquant sur l'icône **Insérer Contenus** dans la [barre d'outils Insérer éléments de design](#)⁴⁴⁷, puis en cliquant à l'emplacement du design où l'élément doit être inséré.

Styler le contenu

L'espace réservé (contenu) peut être formaté en le sélectionnant et en utilisant un format et/ou des propriétés prédéfinis dans la barre latérale Styles. Ce formatage est visible dans le design et, dans la sortie, il sera appliqué au contenu du nœud.

Remplacer le contenu

Si un autre nœud de l'arborescence du schéma est déposé dans un nœud contenant un espace réservé (contenu), l'espace réservé (contenu) existant est remplacé par le nouveau nœud.

Supprimer le contenu

L'espace réservé (contenu) peut être supprimé en le sélectionnant et en appuyant sur la touche **Supprimer** du clavier.

Note : vous pouvez créer une **règle de modèle vide** en supprimant l'espace réservé (contenu) d'un nœud. Une règle de modèle vide est utile si vous souhaitez définir qu'aucun modèle ne s'applique à certains nœuds, c'est-à-dire qu'ils ne produisent aucun résultat.

11.7.2 Reste des contenus

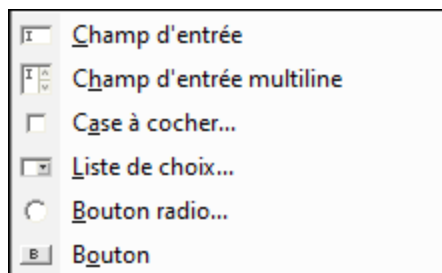
La commande **Reste des contenus** insère l'espace réservé (reste-des-contenus) pour ce nœud. Cet espace réservé représente le contenu des **nœuds enfants inutilisés** du nœud actuel ; il correspond à la règle `xsl:apply-templates` de XSLT appliquée aux éléments et nœuds de texte inutilisés de l'élément actuel. Notez que les modèles ne sont pas appliqués aux attributs enfants. L'espace réservé (reste-des-contenus) peut également être inséré pour un élément en plaçant le curseur à l'intérieur des balises de l'élément, en cliquant avec le bouton droit de la souris et en sélectionnant **Insérer le reste des contenus**.

Utilisez l'espace réservé (reste-des-contenus) lorsque vous souhaitez traiter un élément enfant d'une manière spécifique et appliquer des modèles à ses frères. Il est important d'appliquer des modèles aux frères afin d'éviter qu'ils ne soient pas traités. Cela vous permet d'atteindre des éléments situés plus bas dans la hiérarchie du document.

L'espace réservé (reste-des-contenus) peut être supprimé en le sélectionnant et en appuyant sur la touche **Supprimer** du clavier.

11.7.3 Commandes de formulaire

En passant la souris sur la commande **Commandes de formulaire**, un sous-menu (*capture d'écran ci-dessous*) contenant des commandes permettant d'insérer divers contrôles de formulaire ([dispositifs de saisie de données](#)¹⁵⁸) s'affiche.



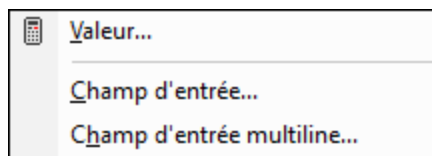
La procédure de création de chacun de ces contrôles de formulaire est décrite dans la section [Utiliser des dispositifs de saisie de données](#)¹⁵⁸. Une fois qu'un contrôle de formulaire a été créé, ses propriétés peuvent être modifiées en le sélectionnant, puis en modifiant la propriété requise dans la [barre latérale Propriétés](#)⁴⁸.

Les contrôles de formulaire peuvent également être insérés dans le design en cliquant avec le bouton droit de la souris sur le point d'insertion et en sélectionnant **Insérer | Commandes de formulaire**, ou en cliquant sur l'icône Commande de formulaire correspondante dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷, puis en cliquant à l'emplacement de la conception où l'élément doit être inséré.

Note : les sections CDATA ne peuvent pas être insérées dans les champs de saisie (c'est-à-dire dans les zones de texte et les zones de texte multilignes). Les sections CDATA ne peuvent être saisies que dans les éléments affichés dans Authentic View en tant que composants de contenu textuel.

11.7.4 Calcul automatique

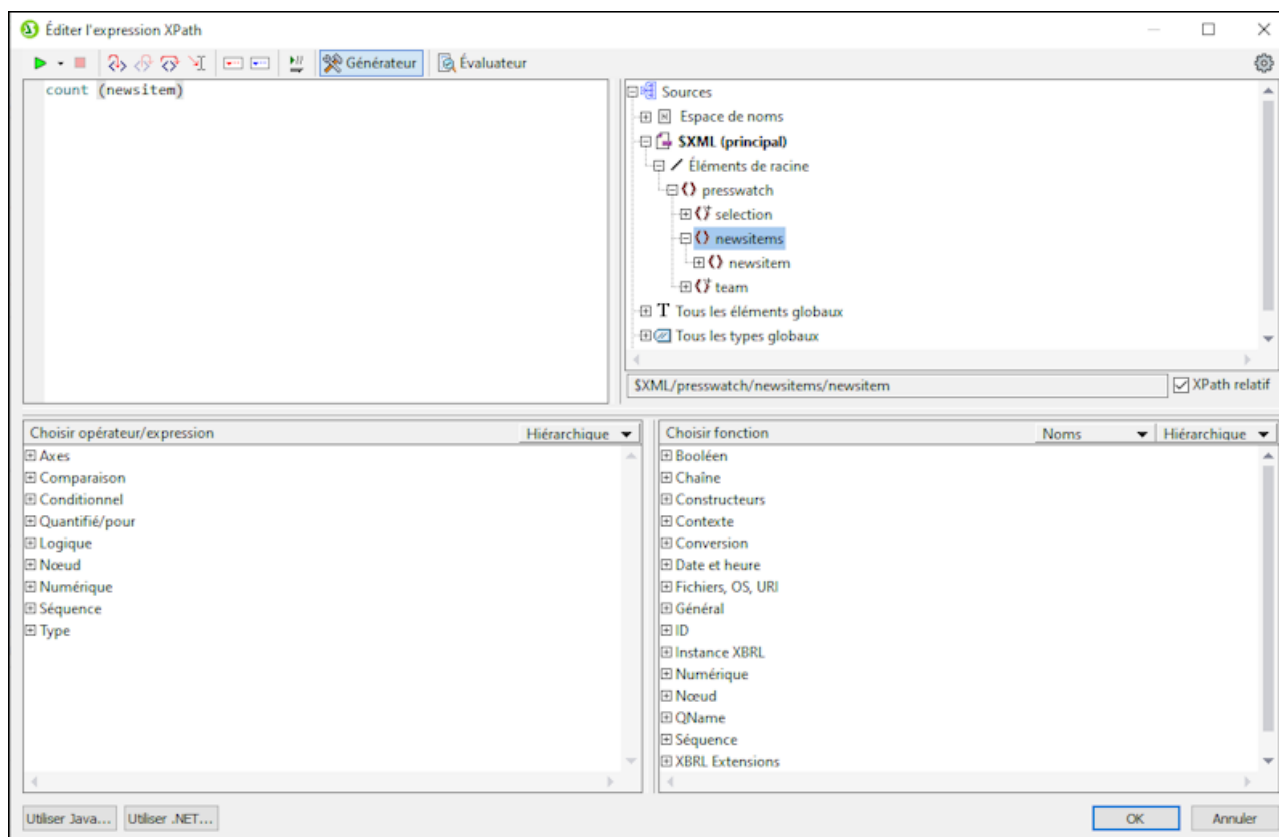
Un **calcul automatique** utilise une expression XPath pour calculer une valeur. Cette valeur s'affiche à l'endroit où le calcul automatique est inséré. Un calcul automatique peut être inséré dans le SPS sous forme de valeur textuelle, de champ de saisie ou de champ de saisie multiligne. Placez le curseur à l'endroit où le calcul automatique doit être inséré, puis cliquez avec le bouton droit de la souris ou utilisez la commande du menu **Insérer**. Lorsque le curseur est placé sur **Insérer | Calcul automatique**, un menu contextuel s'affiche (*capture d'écran ci-dessous*), vous permettant de choisir comment le calcul automatique doit être inséré. Vous pouvez également utiliser l'icône Calcul automatique dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷.



La valeur du calcul automatique s'affichera en conséquence dans le document de sortie.

L'expression XPath pour le calcul automatique

Lorsque vous sélectionnez le mode de représentation du calcul automatique, le dialogue [Éditer l'expression XPath](#) ⁴²⁶ (*capture d'écran ci-dessous*) s'affiche.



Le nœud de contexte de l'expression en cours de construction est mis en surbrillance dans l'arborescence du schéma dans le volet situé à l'extrême gauche. Vous pouvez saisir l'expression XPath directement dans la zone de texte ou double-cliquer sur un élément (dans l'un des trois volets) pour l'insérer. Les nœuds insérés à partir de l'arborescence du schéma dans le volet de gauche sont insérés par rapport au nœud de contexte (si la case *Relative XPath* est cochée) ou sous forme d'expression absolue à partir du nœud du document (si la case *XPath relatif* n'est pas cochée).

Une fois l'expression XPath terminée, cliquez sur **OK** pour terminer l'insertion du calcul automatique.

11.7.5 Paragraphe, Paragraphe spécial

La commande **Paragraphe**  insère un élément HTML <p> autour du composant sélectionné. Un composant est considéré comme sélectionné à cette fin lorsque le nœud entier est sélectionné (en cliquant sur l'une de ses balises) ou lorsque du texte statique est sélectionné. Si le curseur est placé dans du texte statique, l'élément de paragraphe est inséré (balises de début et de fin) à cet endroit. Un paragraphe peut également être inséré à l'aide de l'icône **Insérer paragraphe** dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷.

La commande **Paragraphe spécial** vous permet d'attribuer un format prédéfini au nœud sélectionné. Les formats prédéfinis disponibles peuvent également être sélectionnés dans la liste déroulante de la barre d'outils.

Chaque type de paragraphe possède des caractéristiques de formatage particulières qui peuvent être utilisées à votre avantage. Notez que le type de format `pre` permet d'afficher les retours chariot tels quels au lieu de les

normaliser en espaces blancs.

11.7.6 Image

La commande **Image** affiche le dialogue Insérer Image (*voir captures d'écran ci-dessous*), dans lequel vous pouvez spécifier l'image à insérer. L'icône **Insérer image** dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷ affiche également le dialogue Insérer Image.

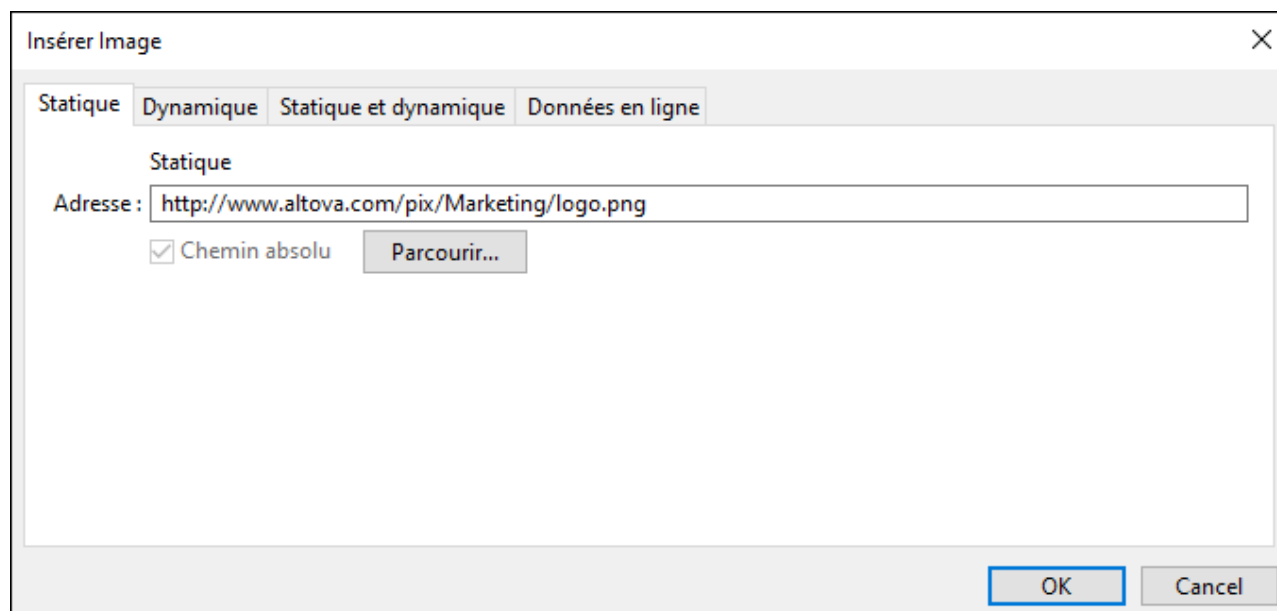
Le dialogue Insérer Image comporte quatre onglets, chacun offrant une méthode différente pour spécifier l'emplacement de l'image. Ces onglets sont les suivants :

- *Statique* : pour saisir directement l'URI de l'image
- *Dynamique* : pour obtenir l'URI de l'image à partir du document XML ou le générer à l'aide d'une expression XPath
- *Statique et dynamique* : pour combiner les méthodes statique et dynamique
- *Données en ligne* : pour sélectionner une image stockée dans un fichier XML sous forme de texte encodé en Base-16 ou Base-64

Les onglets sont décrits en détail ci-dessous.

Statique

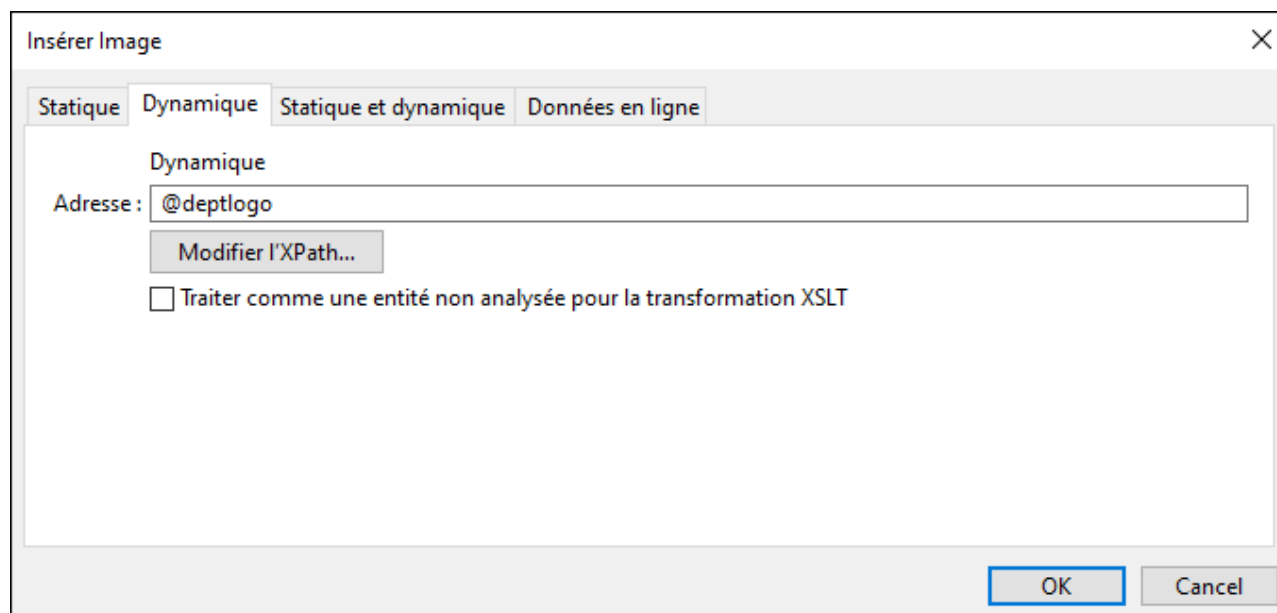
L'URI de l'image est saisi directement dans le champ *Adresse* (*voir capture d'écran ci-dessous*). Dans la capture d'écran ci-dessous, l'URI de l'image est : `http://www.altova.com/pix/Marketing/logo.png`.



Vous pouvez spécifier si l'URI est absolu (case *Chemin absolu* cochée) ou relatif (case *Chemin absolu* décochée). Si un URI relatif est saisi, il sera résolu par rapport à l'emplacement du fichier SPS. Pour saisir automatiquement l'URI (absolu ou relatif), cliquez sur **Parcourir** et recherchez le fichier image.

Dynamique

Une expression XPath renvoie l'URI de l'image. Dans la capture d'écran ci-dessous, l'expression XPath est `@deptlogo`. Cela suppose que l'URI de l'image est stocké dans l'attribut `deptlogo` du nœud de contexte. Le nœud de contexte est le nœud dans lequel l'image est créée.

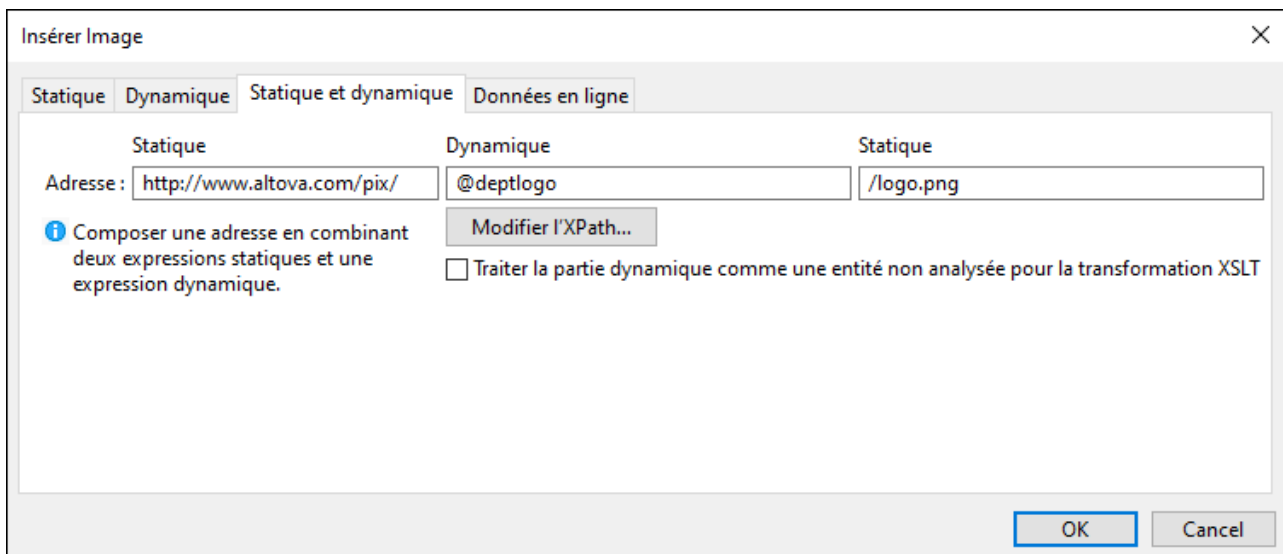


Cliquez sur le bouton **Modifier l'XPath** pour afficher le [générateur d'expressions XPath](#) ⁴³⁸. Dans l'arborescence du générateur d'expressions XPath, le nœud de contexte sera mis en surbrillance.

Si le SPS est basé sur DTD et utilise des **entités non parsées**, une entité non parsée qui fait référence à l'URI de l'image peut être utilisée. Cochez d'abord la case *Traiter comme une entité non analysée*. Entrez ensuite une expression XPath qui sélectionne le nœud contenant l'entité non parsée. Pour plus d'informations sur l'utilisation des entités non parsées, voir [URI d'entités non parsées](#) ³⁶³.

Statique et dynamique

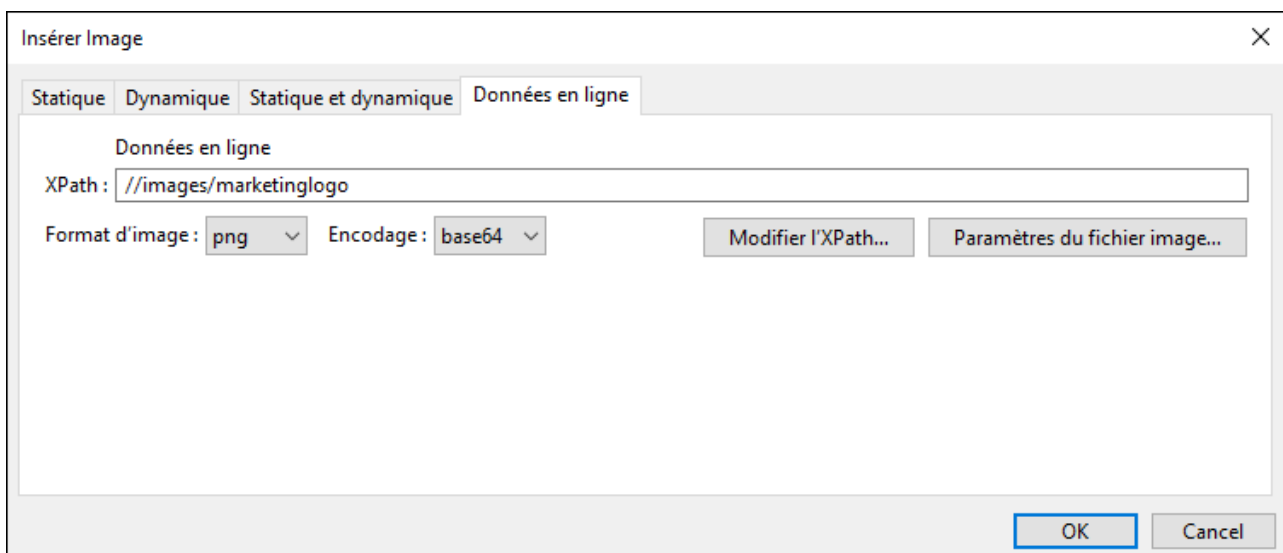
Utilisez conjointement les mécanismes statique et dynamique pour générer l'URI.



Si l'attribut `deptname` du nœud Contexte a la valeur `Marketing`, l'URI de l'image composée dans la capture d'écran ci-dessus sera : `http://www.altova.com/pix/Marketing/logo.png`. Notez que vous pouvez utiliser le [Générateur d'expressions XPath](#) ⁴³⁸ pour la partie dynamique.

Données en ligne

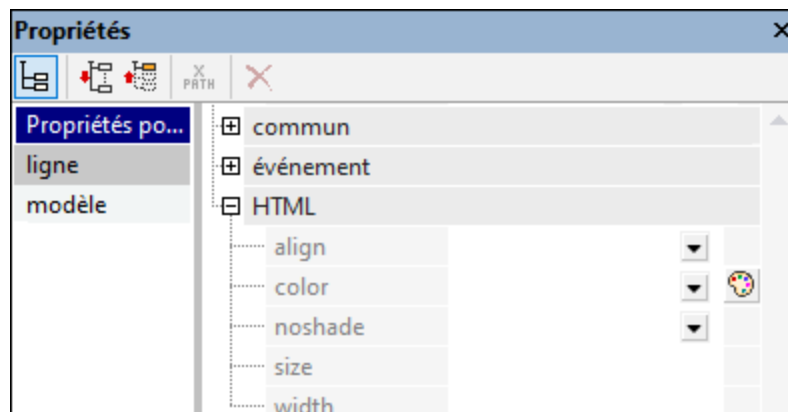
Une image peut être stockée dans un fichier XML sous forme de texte encodé en Base-16 ou Base-64. L'expression XPath dans le dialogue Insérer Image (voir *capture d'écran ci-dessous*) sélectionne le nœud contenant le texte encodé. La liste déroulante Encodage spécifie l'encodage utilisé dans le XML source afin que StyleVision puisse lire correctement le texte encodé. La liste déroulante Format d'image indique le format dans lequel le fichier image doit être généré. (Un fichier d'image est généré à partir des données de texte encodées, puis ce fichier est utilisé dans le document de sortie.)



Le dialogue Paramètres du fichier image généré (accessible en cliquant sur le bouton **Paramètres du fichier image**) vous permet de donner un nom au fichier image qui sera créé. Vous pouvez choisir de ne pas fournir de nom, auquel cas StyleVision en générera un.

11.7.7 Ligne horizontale

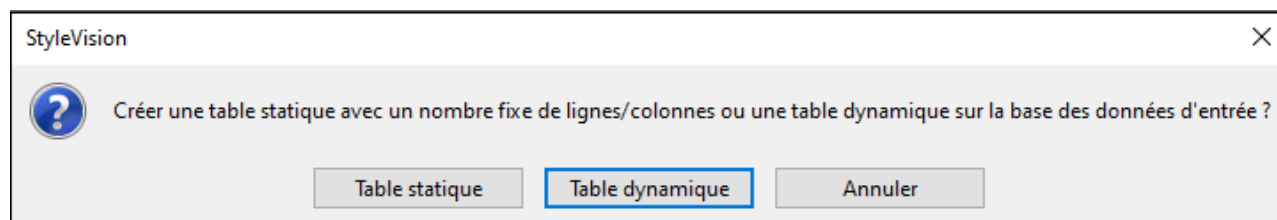
La commande **Ligne horizontale** insère une ligne horizontale à l'emplacement du curseur. Cette commande n'est pas disponible lorsqu'un composant SPS est sélectionné. Pour définir les propriétés de la ligne horizontale, sélectionnez la ligne dans le design, puis dans la barre latérale Propriétés, sélectionnez *ligne* et spécifiez les valeurs des propriétés dans le groupe *HTML* (voir *capture d'écran ci-dessous*).



Vous pouvez spécifier les propriétés suivantes pour la ligne : sa couleur, sa taille (épaisseur), sa largeur (dans le design), son alignement et la propriété *noshade*.

11.7.8 Table

La commande **Insérer Table** affiche le dialogue Stylevision (*capture d'écran ci-dessous*).



Selon que vous souhaitez créer un tableau statique ou dynamique, sélectionnez le bouton approprié. La procédure à suivre pour chaque type de tableau est décrite dans la section : [Tables SPS statiques](#)¹²⁷ et [Tables SPS dynamiques](#)¹²⁸.

Notez que les tables peuvent également être créés à l'aide de la commande de menu **Table | Insérer Table** et de l'icône **Insérer Table**  dans la barre d'outils Insérer éléments de design.

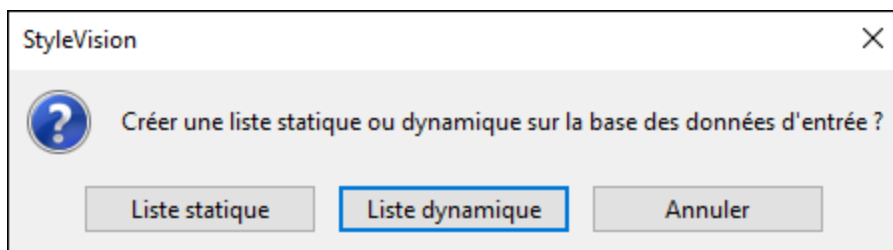
11.7.9 Puces et numérotation



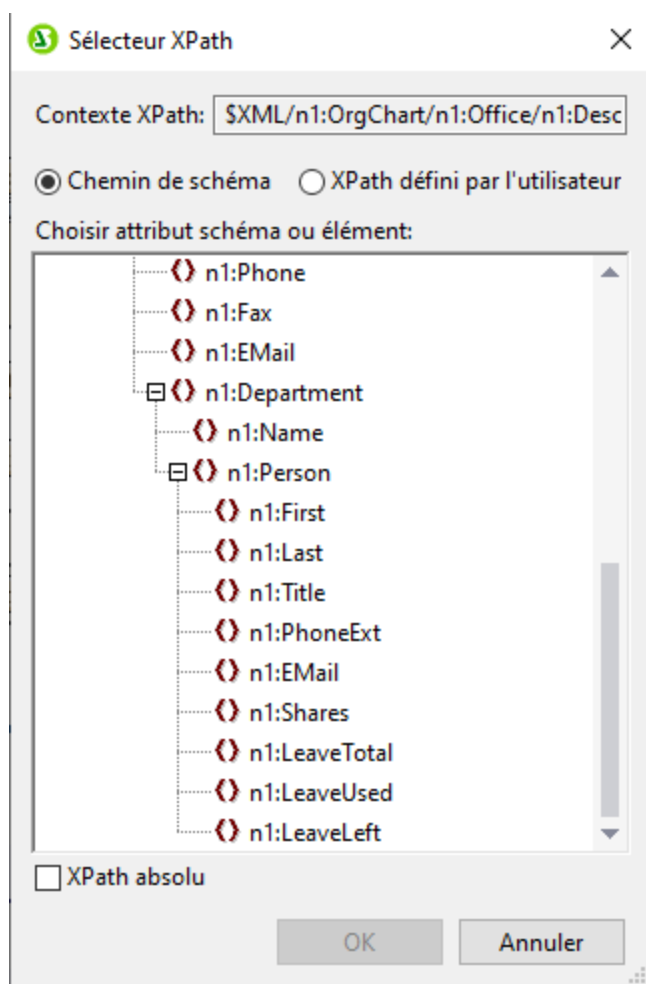
La commande **Puces et numérotation** vous permet de créer une liste, statique ou dynamique. Les éléments d'une liste statique sont saisis dans le SPS, tandis que ceux d'une liste dynamique sont les valeurs des nœuds frères dans le document XML.

Pour créer une liste, procédez comme suit :

1. Placez le curseur à l'endroit où vous souhaitez insérer la liste et cliquez sur la commande **Puces et numérotation**. Ceci fait apparaître un dialogue vous demandant si vous souhaitez créer une liste statique ou une liste dynamique (*capture d'écran ci-dessous*).

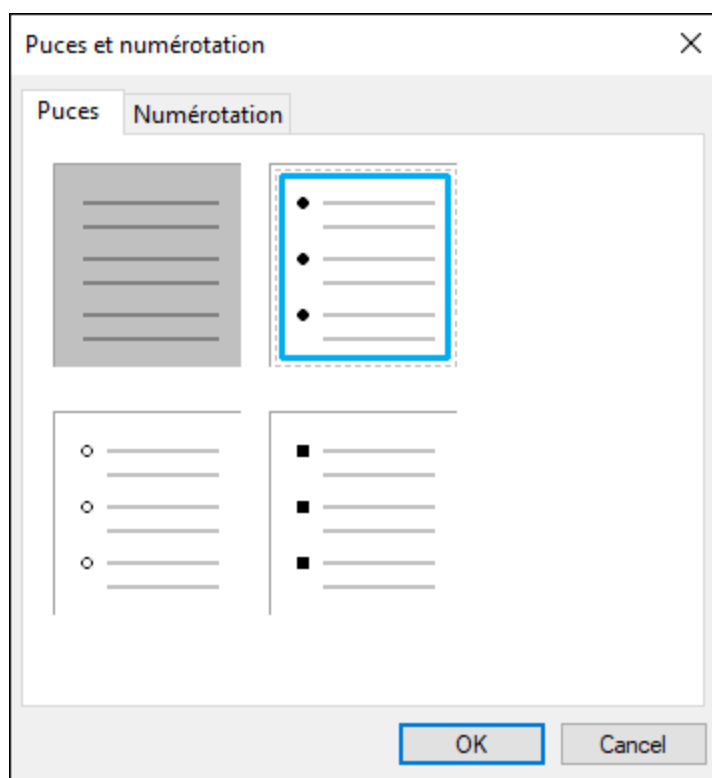


- Si vous cliquez sur **Liste statique**, le dialogue Puces et numérotation décrit à l'étape 3 s'affiche. Si vous cliquez sur **Liste dynamique**, le dialogue Sélecteur XPath s'affiche (*capture d'écran ci-dessous*).
2. Dans le dialogue Sélecteur XPath, notez que le Contexte XPath est le contexte de l'emplacement d'insertion, et qu'il ne peut pas être modifié dans le dialogue. Sélectionnez le nœud qui doit être créé en tant que liste dynamique. Dans la capture d'écran ci-dessous, le nœud contextuel est `n1:Department`, et le nœud `n1:Person` a été sélectionné comme nœud à créer en tant que liste. Cela signifie que le contenu de chaque nœud `n1:Person` sera créé en tant qu'élément de la liste.



Si vous sélectionnez l'option XPath défini par l'utilisateur, vous pouvez alors saisir une expression XPath pour sélectionner le nœud à créer en tant que table dynamique. Cliquez sur **OK** pour afficher le dialogue Puces et numérotation décrit à l'étape suivante.

3. Dans le dialogue Puces et numérotation, sélectionnez le type de liste que vous souhaitez créer. Vous pouvez choisir entre une liste à puces (avec une puce, un cercle ou un carré comme marqueur d'élément de liste) ou une liste numérotée. Cliquez sur **OK** pour créer la liste avec le type de marqueur d'élément de liste que vous avez sélectionné.



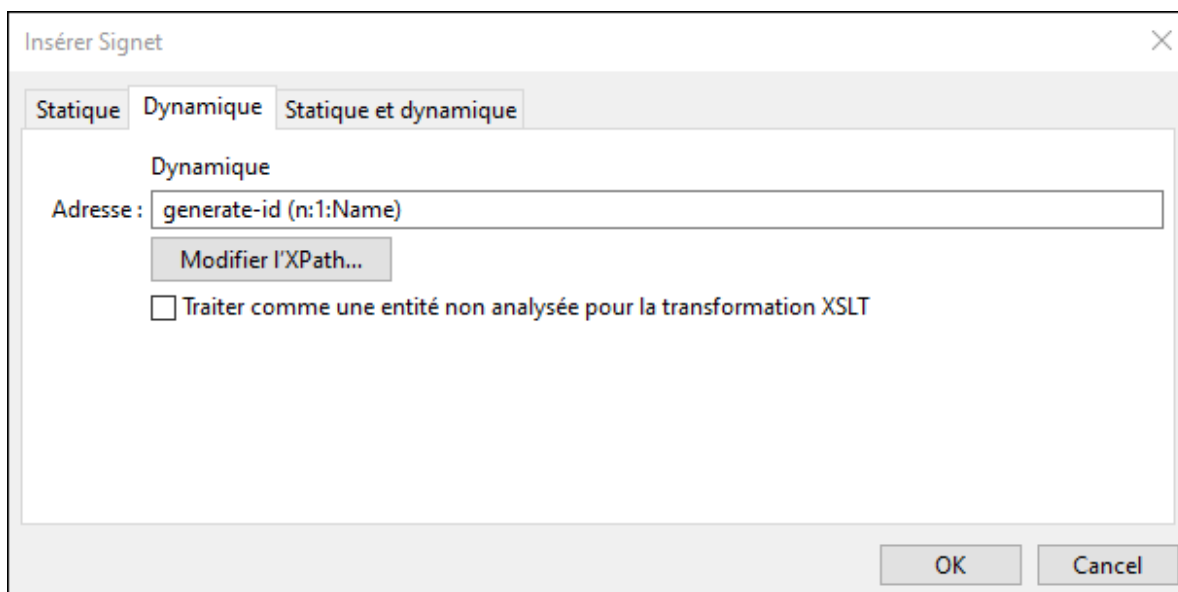
Note : vous pouvez également créer une liste statique en plaçant le curseur à l'emplacement où la liste doit être créée, puis en cliquant sur l'icône Liste à puces ou Liste numérotée dans la [barre d'outils Insérer éléments de design](#)⁴⁴⁷, selon vos besoins. Vous pouvez également créer une liste dynamique en faisant glisser un nœud de l'arborescence du schéma vers le design.

11.7.10 Signet

La commande **Signet** vous permet d'insérer un signet (ou une ancre) à n'importe quel endroit dans le fichier SPS. Un signet peut être référencé par un [lien hypertexte](#)⁵⁰⁰.

Pour insérer un signet, faites comme suit :

1. Placez le curseur à l'emplacement où vous souhaitez créer le signet.
2. Sélectionnez **Insérer | Signet**, ou cliquez avec le bouton droit de la souris et sélectionnez **Insérer | Signet**. Le dialogue Insérer Signet apparaît.



3. Dans le [dialogue Insérer Signet](#) ³²¹, sélectionnez un onglet selon que le nom du signet doit être statique (onglet Statique), obtenu de manière dynamique depuis le document XML (Dynamique) ou composé à la fois de parties statiques et dynamiques (Statique et dynamique). Dans la capture d'écran ci-dessus, un signet dynamique est créé, dont le nom est un identifiant unique pour chaque enfant `Nom` du nœud contextuel.
4. Cliquez sur **OK**. Le signet est défini.

Note : les signets sont créés à l'emplacement spécifié dans le design. Si cet emplacement se trouve à l'intérieur d'un élément répétitif, un signet est créé à l'intérieur de chaque instance de l'élément répétitif. Si un nom statique est donné, chaque signet aura le même nom. Pour cette raison, il est mieux, dans de tels cas (éléments répétitifs), de donner un nom dynamique, qui peut être, par exemple, le nom d'un élément enfant du nœud contextuel (l'élément à l'intérieur duquel le signet est créé). Si le nœud sélectionné pour le nom dynamique pourrait avoir le même contenu dans les multiples instances, alors le caractère unique du nom du signet peut être assuré en utilisant la fonction `generate-id()` pour générer le nom (*voir la capture d'écran ci-dessus*). Pour référencer un tel signet, la même ID peut être générée comme valeur `href` d'un [hyperlien](#) ³²⁶. Dans ce cas, assurez-vous d'utiliser l'identificateur de fragment `#` devant la fonction `generate-id()`. L'expression XPath serait : `concat('#', generate-id(nodeXXX))`.

Vous pouvez modifier le nom d'un signet après sa création. Pour ce faire, cliquez avec le bouton droit de la souris sur le signet et sélectionnez la commande **Éditer nom de signet** dans le menu contextuel qui s'affiche. Dans la barre latérale Propriétés du groupe de propriétés *Signet*, vous pouvez également cliquer sur le bouton **Éditer** de l'attribut Nom du signet et apporter les modifications souhaitées.

Supprimer un signet

Pour supprimer un signet, sélectionnez-le dans le design et appuyez sur la touche **Supprimer**.

11.7.11 Hyperlien



La commande **Hyperlien** vous permet d'insérer un lien à partir de n'importe quelle partie du document de sortie (HTML) vers une ancre dans le document de sortie ou vers un document externe ou un fragment de document.

Pour insérer un hyperlien, procédez comme suit :

1. Un hyperlien peut être créé autour d'un composant de design existant ou inséré à n'importe quel endroit du document (le texte du lien étant inséré ultérieurement). Sélectionnez le composant SPS ou le fragment de texte à transformer en hyperlien ou placez le curseur à l'endroit où le lien doit être inséré.
2. Cliquez sur l'icône Hyperlien dans la barre d'outils, ou sélectionnez **Insérer | Hyperlien**, ou cliquez avec le bouton droit de la souris et sélectionnez **Insérer | Hyperlien** (lorsqu'aucun composant de design n'est sélectionné) ou **Contenir dans | Lien hypertexte** (lorsqu'un composant de design est sélectionné). Un hyperlien peut également être inséré à l'aide de l'icône **Insérer hyperlien** dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷.
3. Dans le dialogue [Insérer Hyperlien](#) ³²³ qui s'affiche, spécifiez le document ou le fragment de document vers lequel vous souhaitez créer un lien. Si vous créez un lien vers un fragment de document (c'est-à-dire vers un signet dans un document), n'oubliez pas d'inclure le symbole #. L'URI de l'hyperlien est spécifié sous l'une des formes suivantes :
 - Sous forme d'adresse statique (saisie directement ; vous pouvez sélectionner un fichier HTML à l'aide du bouton **Parcourir** et un fragment dans le document actuel à l'aide du bouton **Signet**).
Exemples : `http://www.altova.com` (URI de page Web statique) ; `U: \documentation\index.html` (via le bouton Parcourir) ; ou `#top_of_page` (via le bouton Signet).
 - Sous forme d'adresse dynamique (provenant d'un nœud du document XML ; vous spécifiez le nœud). Un exemple serait un nœud tel que `//otherdocs/doc1`. Si le nom d'un signet a été généré à l'aide de la fonction `generate-id()`, alors l'attribut `href` du lien hypertexte doit être généré à l'aide de la même fonction `generate-id()`. Pour plus d'informations, voir [Définir des hyperliens](#) ³²⁶.
 - En tant que combinaison de texte statique et dynamique pour une adresse (vous spécifiez le texte statique et le nœud du document XML). Un exemple serait `www.altova.com -- department/name -- #intropara`.
4. Cliquez sur **OK**. L'hyperlien est créé.

Note : lorsque vous spécifiez le nœud d'une entrée d'hyperlien dynamique, vous pouvez entrer l'expression XPath en tant qu'expression XPath absolue en cochant la case *Chemin d'accès absolu*. Si cette case n'est pas cochée, l'expression XPath sera relative au nœud de contexte, qui est le nœud dans lequel le lien hypertexte est inséré.

Utiliser des entités non parsées

Si vous utilisez un DTD comme schéma, vous pouvez utiliser l'URI déclaré pour une entité non parsée dans le DTD dans la partie dynamique d'une adresse d'hyperlien. Pour plus d'informations sur l'utilisation des entités non parsées, voir [Utiliser des URI d'entités non parsées](#) ³⁶³.

Éditer un hyperlien

Vous pouvez modifier l'attribut `href` d'un hyperlien après sa création. Pour ce faire, cliquez avec le bouton droit sur l'hyperlien et sélectionnez la commande **Éditer URL**. Dans la barre latérale Propriétés du groupe de

propriétés *Lien*, vous pouvez également cliquer sur le bouton **Éditer** de l'attribut URL et apporter les modifications nécessaires.

Supprimer un hyperlien

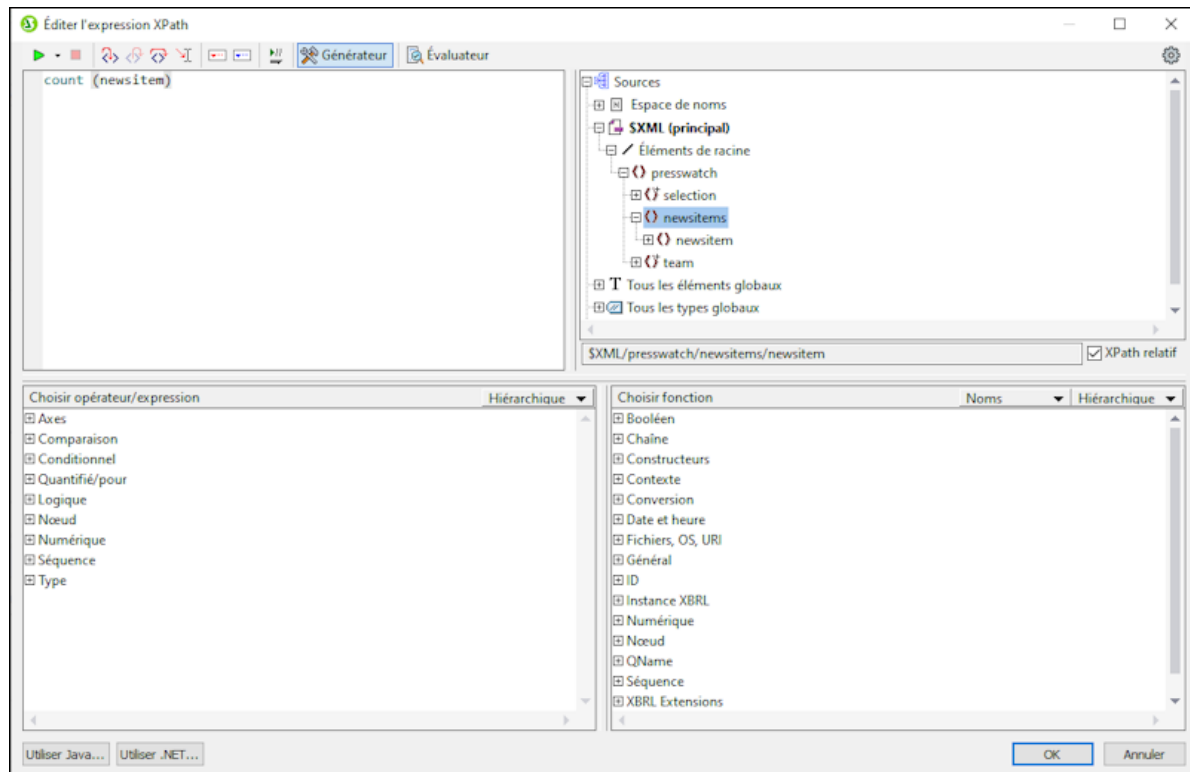
Pour supprimer un hyperlien, sélectionnez-le dans le design et appuyez sur la touche **Supprimer**.

11.7.12 Condition, Condition sur base de sortie

La commande **Condition** vous permet d'insérer une condition à l'emplacement du curseur ou autour de la sélection. Une condition se compose d'une ou plusieurs branches, chacune contenant un ensemble spécifique de règles de traitement. De cette manière, différents ensembles de règles de traitement peuvent être spécifiés pour différentes branches. Par exemple, si le contenu d'un nœud est la chaîne `Stop`, la branche peut tester cela et spécifier que le contenu du nœud doit être coloré en rouge ; une deuxième branche peut tester si le contenu du nœud est la chaîne `Go` et, si oui, colorer le contenu du nœud en vert ; une troisième branche peut spécifier que si le contenu du nœud n'est ni la chaîne `Stop` ni la chaîne `Go`, le contenu du nœud doit être coloré en noir.

Pour insérer une condition, procédez comme suit :

1. Placez le curseur à l'emplacement souhaité dans le design ou sélectionnez le composant autour duquel la condition doit être insérée.
2. Sélectionnez la commande de menu **Insérer | Condition** ou cliquez avec le bouton droit de la souris et sélectionnez la commande du menu contextuel **Insérer | Condition**.
3. Dans le dialogue [Éditer l'expression XPath](#) ⁴²⁶ qui s'affiche (*capture d'écran ci-dessous*), entrez l'expression XPath.



Le nœud de contexte de l'expression en cours de construction est mis en surbrillance dans l'arborescence du schéma dans le volet situé à l'extrême gauche. Vous pouvez saisir l'expression XPath directement dans la zone de texte ou double-cliquer sur un élément (dans l'un des trois volets) pour l'insérer. Les nœuds insérés à partir de l'arborescence du schéma dans le volet de gauche sont insérés par rapport au nœud de contexte (si la case *Relative XPath* est cochée) ou sous forme d'expression absolue à partir du nœud du document (si la case *XPath relatif* n'est pas cochée).

4. Cliquez sur **OK** pour terminer l'insertion de la condition. La condition est créée avec une branche, dont le test est l'expression XPath que vous avez saisie.

Modifier des expressions XPath de branches

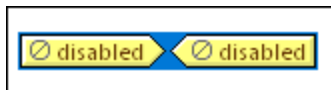
Pour modifier l'expression XPath d'une branche, sélectionnez la branche dans la vue Design. Ensuite, dans la barre latérale Propriétés, sélectionnez *branche de condition* | quand. Cliquez sur le bouton **Éditer**  pour l'élément XPath. Le dialogue *Éditer l'expression XPath (capture d'écran ci-dessus)* s'affiche, dans lequel vous pouvez modifier l'expression. Cliquez sur **OK** lorsque vous avez terminé.

Ajouter des branches, modifier l'ordre des branches et supprimer des branches

Pour ajouter de nouvelles branches, modifier l'ordre des branches et supprimer des branches, cliquez avec le bouton droit sur la branche souhaitée et sélectionnez l'élément correspondant dans le menu contextuel.

11.7.13 Désactivé

La commande **Désactivé** insère un composant Désactivé à l'emplacement du curseur (*capture d'écran ci-dessous*). (Pour mettre la composante désactivée autour du contenu sélectionné, utilisez la commande [Encadrer avec | Désactivé](#) ⁵¹⁴.)



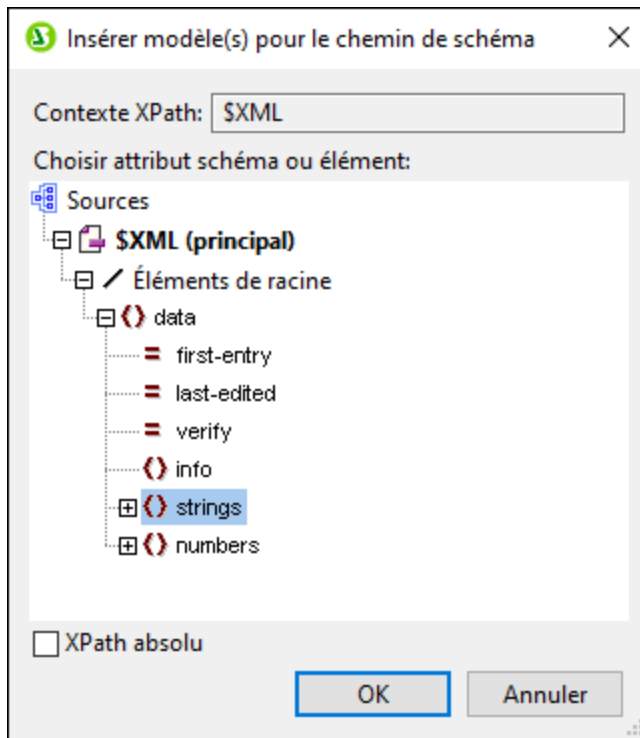
Le contenu d'un composant Désactivé est ignoré dans la sortie. Donc vous pouvez ajouter du contenu que vous souhaitez ignorer dans le composant Désactivé. Le composant Désactivé sert donc de moyen de commenter le contenu.

Pour rétablir le contenu désactivé, supprimez simplement les balises Désactivé autour du contenu. Pour ce faire, sélectionnez le composant Désactivé, cliquez avec le bouton droit de la souris et sélectionnez **Supprimer balise uniquement**.

11.7.14 Modèle

La commande **Modèle** insère, à l'emplacement du curseur, un modèle vide pour le nœud de l'arborescence de schéma que vous sélectionnez. Pour insérer un modèle, procédez comme suit.

1. Placez le curseur à l'emplacement où vous souhaitez insérer le modèle dans le design.
2. Cliquez sur la commande **Insérer | Modèle**. Le dialogue Insérer modèle(s) pour le chemin de schéma s'affiche (*capture d'écran ci-dessous*).



3. Le champ Contexte XPath contient le nœud de contexte du point d'insertion du curseur. Ce nœud sera le nœud de contexte du modèle lorsqu'il sera créé. Dans le dialogue, sélectionnez le nœud pour lequel vous souhaitez créer le modèle. Dans la capture d'écran ci-dessus, le nœud `headline` est sélectionné. Nous allons donc créer un modèle pour le nœud `headline` dans le contexte de l'élément `newsitem`.
4. Cliquez sur **OK** pour terminer.

Un modèle vide sera créé pour le nœud sélectionné. Dans la capture d'écran ci-dessous, un modèle vide a été créé pour le nœud `headline`.



11.7.15 Modèle défini par l'utilisateur

La commande **Modèle défini par l'utilisateur** insère, à l'emplacement du curseur, un modèle vide qui sélectionne un nœud spécifié par l'utilisateur dans une expression XPath. Pour insérer un modèle défini par l'utilisateur, procédez comme suit.

1. Placez le curseur à l'emplacement où vous souhaitez insérer le modèle dans le design.

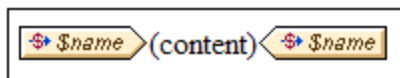
2. Cliquez sur la commande **Insérer | Modèle défini par l'utilisateur**. Le [dialogue Éditer l'expression XPath](#)⁴²⁶ s'affiche..
3. Entrez l'expression XPath pour sélectionner le nœud souhaité. Il convient de noter plusieurs points à cet égard : (i) L'expression XPath sera évaluée dans le contexte du nœud dans lequel le modèle défini par l'utilisateur est créé ; (ii) L'expression XPath peut sélectionner n'importe quel nœud partout dans le document ainsi que dans un autre document XML.
4. Après avoir saisi l'expression XPath, cliquez sur **OK** pour terminer.

Un modèle défini par l'utilisateur vide sera créé pour le nœud ciblé.

Pour plus d'informations, voir la section [Fichier SPS : Contenus | Modèles définis par l'utilisateur](#)¹¹⁹.

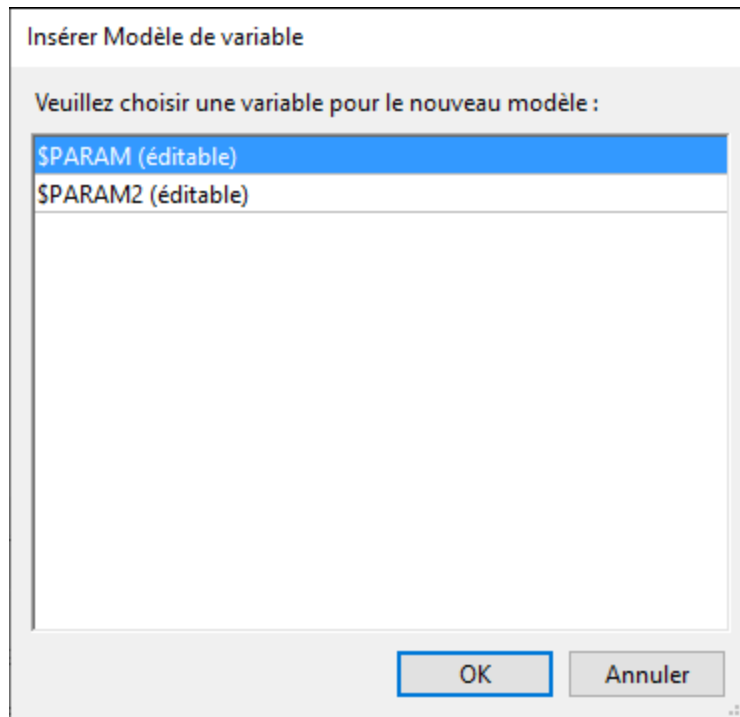
11.7.16 Modèle de variable

Un **Modèle de variable** est un modèle qui cible une variable et, par défaut, affiche son contenu. Il est inséré avec la commande **Insérer | Modèle de variable** ou **Contenir dans | Variable**, qui insère, au point d'insertion du curseur, un modèle pour une variable définie dans SPS. Le modèle variable (*capture d'écran ci-dessous*) contient par défaut un espace réservé `contenu`, qui sert à afficher le contenu de la variable. Vous pouvez insérer du contenu supplémentaire (statique ou dynamique) dans le modèle variable selon vos besoins, ou le modifier comme vous le feriez avec tout autre modèle. Un modèle variable est indiqué par le symbole dollar dans ses balises de début et de fin.



Pour insérer un modèle variable, procédez comme suit :

1. Placez le curseur à l'emplacement où vous souhaitez insérer le modèle dans le design.
2. Cliquez avec le bouton droit de la souris et sélectionnez la commande **Insérer | Modèle de variable**. Le dialogue Insérer Modèle de variable s'affiche (*capture d'écran ci-dessous*).



3. Le dialogue contient une liste de tous les [paramètres et variables déclarés par l'utilisateur](#)²⁸⁵ et définis dans le SPS. Sélectionnez la variable à laquelle vous souhaitez ajouter un modèle.
4. Cliquez sur **OK** pour terminer.

11.7.17 Fragment de design

En passant la souris sur la commande **Fragment de design**, un sous-menu contenant tous les fragments de design actuellement présents dans la conception s'affiche. Cliquer sur un fragment de design dans le sous-menu l'insère à l'emplacement du curseur.

11.7.18 Conteneur de layout, Fenêtre de layout, Ligne

La commande **Insérer | Conteneur de layout** permet d'insérer un conteneur de mise en page à n'importe quel endroit du design. Une fenêtre de layout et une ligne peuvent être insérées dans un conteneur de layout, et ces deux commandes ne sont activées que lorsqu'un conteneur de mise en page est sélectionné.

Les conteneurs de layout, les fenêtres de layout et les lignes peuvent également être insérés à l'aide des icônes correspondantes dans la [barre d'outils Insérer éléments de design](#)⁴⁴⁷. Pour insérer à l'aide des icônes de la barre d'outils, vous devez d'abord sélectionner l'icône appropriée, puis cliquer à l'emplacement où vous souhaitez insérer l'élément de mise en page dans le design.

Pour une description détaillée des modules de mise en page et de leur insertion et utilisation dans le design, voir la section [Modules de layout](#)¹⁷¹.

11.7.19 Table des matières

En passant la souris sur la commande **Table des matières**, un sous-menu s'affiche, contenant des commandes permettant d'insérer diverses commandes liées à la création d'un modèle de table des matières (TdM), de signets TdM et d'une structure de document de conception pour la table des matières.

La liste des commandes est la suivante. Pour plus de détails sur leur utilisation, cliquez sur les liens correspondants, qui vous dirigeront vers la section consacrée à l'utilisation de ce composant particulier de la table des matières.

- [Insérer Table des matières](#) ³⁰³
- [Signet TdM](#) ³⁰⁰
- [Signet TdM \(Assistant\)](#) ²⁹⁶
- [Référence TM](#) ³⁰⁶
- [Référence TM | Texte d'entrée/Points de suite/Référence de page](#) ³⁰⁷
- [Numérotation hiérarchique](#) ³⁰⁷
- [Numérotation séquentielle](#) ³⁰⁷
- [Niveau](#) ²⁹⁸
- [Référence de niveau](#) ³⁰⁵
- [Le modèle sert de niveau](#) ²⁹⁸

Note : ces commandes sont également disponibles dans un menu contextuel, selon l'endroit où vous cliquez avec le bouton droit dans la conception.

11.7.20 Nouveau document

La commande **Insérer nouveau document** insère un modèle Nouveau document (*capture d'écran ci-dessous*) à l'emplacement du curseur.



Le modèle Nouveau document contient une Section initiale du document vide. Vous pouvez maintenant saisir du contenu dans la section initiale du document. Si vous le souhaitez, vous pouvez ajouter des sections de document supplémentaires à la Section initiale du document à l'aide de la commande **Insérer | Insérer page/colonne/section de document**.

Un modèle Nouveau document crée un nouveau document dans la sortie. Par conséquent, la sortie comprendra plusieurs documents de sortie.

Pour une description détaillée de la manière de travailler avec plusieurs documents de sortie, voir la section [Sortie multiple de document](#)²⁵¹.

11.7.21 Item défini par l'utilisateur

En passant la souris sur la commande **Insérer | Item défini par l'utilisateur**, un sous-menu s'affiche, contenant des commandes permettant d'insérer un [élément défini par l'utilisateur](#)¹²² ou un [bloc de texte XML défini par l'utilisateur](#)¹²³. L'utilisation de ces deux composants est décrite dans la section [Fichier SPS](#)¹²² : [Contenu | Éléments définis par l'utilisateur, blocs de texte XML](#)¹²².

11.8 Menu Contenir dans

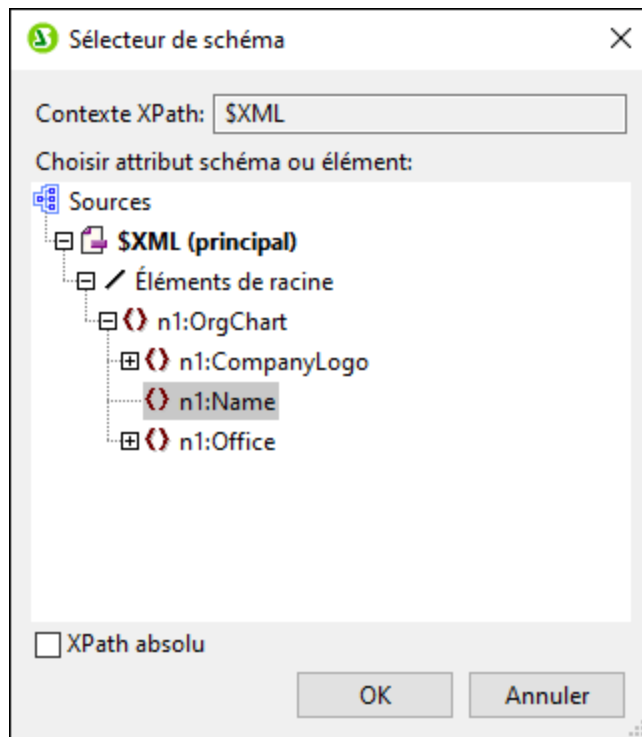
Le menu **Inclure dans** fournit des commandes vous permettant d'inclure une sélection dans la conception à l'aide de divers composants de conception. Certaines de ces commandes sont disponibles sous forme d'[icônes dans la barre d'outils](#) ⁴⁴³, vous permettant d'insérer le composant dans la conception (des commandes équivalentes sont disponibles dans le [menu Insérer](#) ⁴⁸⁸). De plus, les commandes du menu **Inclure dans** sont également disponibles via les **menus contextuels** qui apparaissent lorsque vous cliquez avec le bouton droit de la souris sur une sélection dans la conception SPS. Dans les menus et les menus contextuels, les commandes qui ne sont pas disponibles à cet emplacement dans SPS sont désactivées.

Note : étant donné que les commandes **Inclure dans** sont utilisées pour construire le SPS, elles ne sont disponibles que dans le mode Design.

11.8.1 Modèle

La commande **Contenir dans | Modèle** inclut le composant de conception ou le texte sélectionné dans un modèle pour le nœud de l'arborescence de schéma que vous sélectionnez. Procédez comme suit.

1. Sélectionnez le composant de conception ou le texte que vous souhaitez inclure dans un modèle.
2. Cliquez sur la commande **Contenir dans | Modèle**. Le dialogue Sélecteur de schéma s'affiche (*capture d'écran ci-dessous*).



3. Le champ Contexte XPath contient le nœud de contexte de la sélection et sera le nœud de contexte du modèle lorsqu'il sera créé. Sélectionnez le nœud pour lequel vous souhaitez créer le modèle. Dans

la capture d'écran ci-dessus, le nœud `n1:Name` est sélectionné comme nœud pour lequel le modèle est créé.

4. Cliquez sur **OK** pour terminer.

Un modèle pour le nœud sélectionné sera créé autour de la sélection.

11.8.2 Modèle défini par l'utilisateur

La commande **Contenir dans | Modèle défini par l'utilisateur** ajoute à la sélection un modèle pour un nœud spécifié par l'utilisateur dans une expression XPath. Pour insérer un modèle défini par l'utilisateur, procédez comme suit.

1. Sélectionnez le composant dans le design que vous souhaitez inclure dans un modèle défini par l'utilisateur.
2. Cliquez sur la commande **Contenir dans | Modèle défini par l'utilisateur**. Le dialogue [Éditer l'expression XPath](#) ⁴²⁶ s'affiche.
3. Entrez l'expression XPath pour sélectionner le nœud souhaité. Il convient de noter plusieurs points à cet égard : (i) L'expression XPath sera évaluée dans le contexte du nœud dans lequel le modèle défini par l'utilisateur est créé ; (ii) L'expression XPath peut sélectionner n'importe quel nœud partout dans le document ainsi que dans un autre document XML.
4. Après avoir saisi l'expression XPath, cliquez sur **OK** pour terminer.

Un modèle défini par l'utilisateur pour le nœud ciblé sera créé autour de la sélection.

Pour plus d'informations, voir la section [Fichier SPS : Structure | Modèles et fragments de design | Modèles définis par l'utilisateur](#) ²³⁹.

11.8.3 Modèle de variable

La commande **Contenir dans | Modèle variable** inclut la sélection dans un modèle pour une variable définie dans la conception SPS.

1. Sélectionnez le composant de la conception que vous souhaitez inclure dans un modèle de variable.
2. Cliquez sur la commande **Contenir dans | Modèle variable**. Le [dialogue Joindre avec le modèle variable](#) ²⁴¹ s'affiche.
3. Dans la liste du dialogue, sélectionnez la variable pour laquelle vous souhaitez créer le modèle.
4. Cliquez sur **OK** pour terminer.

Un modèle de variable sera créé autour de la sélection.

Pour plus d'informations, voir la section [Fichier SPS : Structure | Modèles et fragments de design | Modèles variable](#) ²⁴¹.

11.8.4 Paragraphe, paragraphe spécial

La commande **Paragraphe**  insère un élément HTML `<p>` autour du composant sélectionné. Un composant est considéré comme sélectionné à cette fin lorsque le nœud entier est sélectionné (en cliquant sur l'une de ses balises) ou lorsque du texte statique est sélectionné. Si le curseur est placé dans du texte statique, l'élément de paragraphe est inséré (balises de début et de fin) à cet endroit. Un paragraphe peut également être inséré à l'aide de l'icône **Insérer paragraphe** dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷.

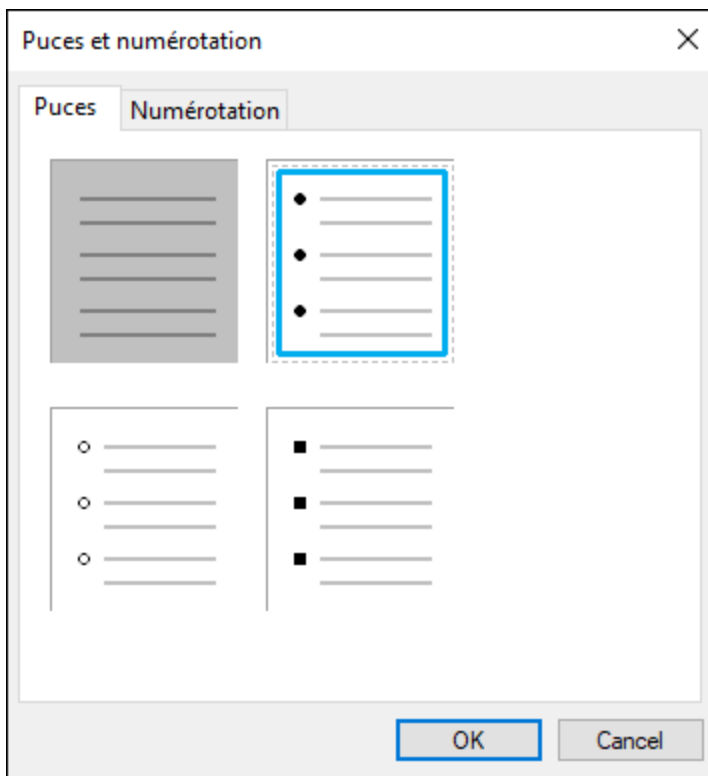
La commande **Paragraphe spécial** vous permet d'attribuer un format prédéfini au nœud sélectionné. Les formats prédéfinis disponibles peuvent également être sélectionnés dans la liste déroulante de la barre d'outils.

Chaque type de paragraphe possède des caractéristiques de formatage particulières qui peuvent être utilisées à votre avantage. Notez que le type de format `pre` permet d'afficher les retours chariot tels quels au lieu de les normaliser en espaces blancs.

11.8.5 Puces et numérotation

La commande **Inclure dans | Puces et numérotation** crée une liste statique et des éléments de liste autour de la sélection. Si la sélection contient un CR-LF, un retour chariot et/ou un saut de ligne (inséré en appuyant sur la touche **Entrée**), des éléments de liste distincts sont créés pour chaque fragment de texte séparé par un CR-LF.

Lorsque cette commande est sélectionnée, le dialogue Puces et numérotation (capture d'écran ci-dessous) s'affiche.



Sélectionnez le marqueur d'élément de liste souhaité et cliquez sur **OK**. Une liste est créée. Le nombre d'éléments de la liste correspond au nombre de CR-LF (retours chariot et/ou sauts de ligne) dans la sélection. Vous pouvez ajouter d'autres éléments à la liste en appuyant sur **Entrée**.

Note : vous pouvez obtenir les mêmes résultats en sélectionnant le contenu statique, puis en cliquant sur les icônes Liste à puces ou Liste numérotée dans la [barre d'outils Insérer éléments de design](#) ⁴⁴⁷.

11.8.6 Signet et lien hypertexte

Les commandes **Inclure dans | Signet** et **Inclure dans | Hyperlien** sont activées lorsque du texte ou un composant de la conception SPS est sélectionné. Ces commandes permettent respectivement de créer un signet et un hyperlien autour de la sélection. Pour plus d'informations sur le fonctionnement des signets et des hyperliens et sur leur création, voir la section [Fonctions avancées | Table des matières. Référencement. Signets](#) ³²¹.

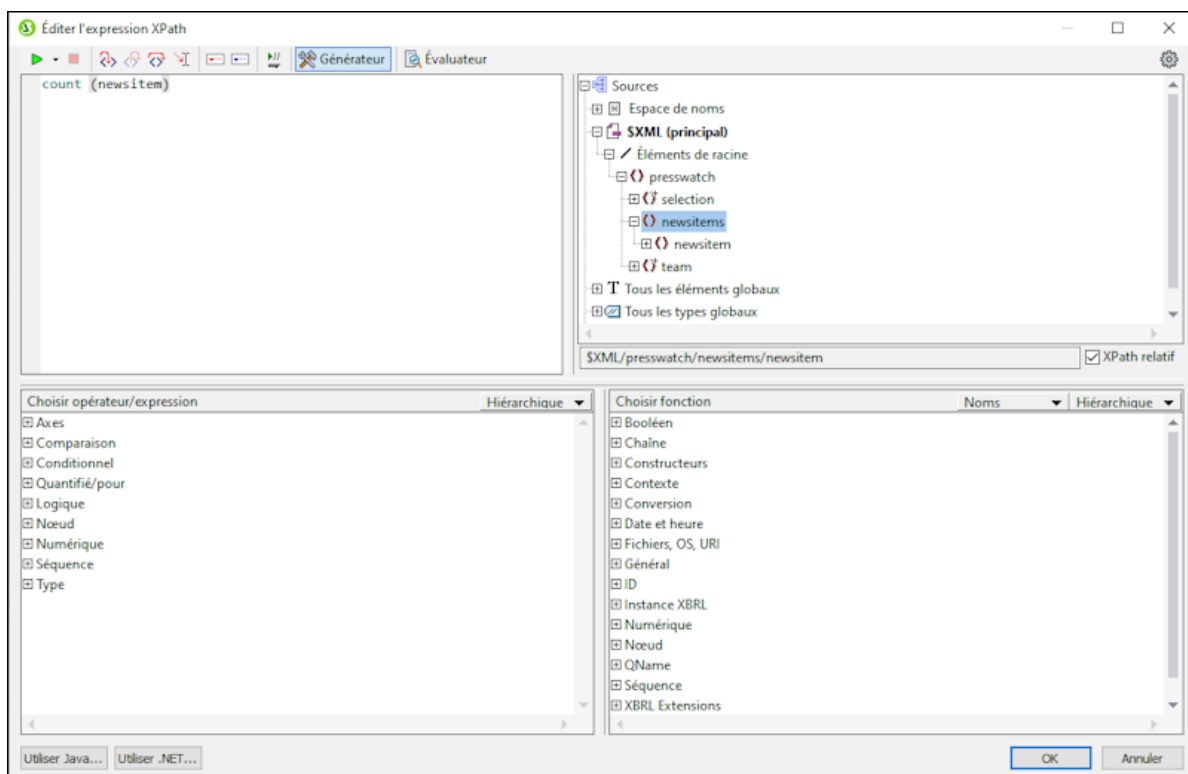
11.8.7 Condition, condition sur base de sortie

La commande **Condition** vous permet d'insérer une condition à l'emplacement du curseur ou autour de la sélection. Une condition se compose d'une ou plusieurs branches, chacune contenant un ensemble spécifique de règles de traitement. De cette manière, différents ensembles de règles de traitement peuvent être spécifiés pour différentes branches. Par exemple, si le contenu d'un nœud est la chaîne `stop`, la branche peut tester cela et spécifier que le contenu du nœud doit être coloré en rouge ; une deuxième branche peut tester si le contenu du nœud est la chaîne `Go` et, si oui, colorer le contenu du nœud en vert ; une troisième branche peut

spécifier que si le contenu du nœud n'est ni la chaîne `stop` ni la chaîne `Go`, le contenu du nœud doit être coloré en noir.

Pour insérer une condition, procédez comme suit :

1. Placez le curseur à l'emplacement souhaité dans le design ou sélectionnez le composant autour duquel la condition doit être insérée.
2. Sélectionnez la commande de menu **Insérer | Condition** ou cliquez avec le bouton droit de la souris et sélectionnez la commande du menu contextuel **Insérer | Condition**.
3. Dans le dialogue [Éditer l'expression XPath](#) ⁴²⁶ qui s'affiche (*capture d'écran ci-dessous*), entrez l'expression XPath.



Le nœud de contexte de l'expression en cours de construction est mis en surbrillance dans l'arborescence du schéma dans le volet situé à l'extrême gauche. Vous pouvez saisir l'expression XPath directement dans la zone de texte ou double-cliquer sur un élément (dans l'un des trois volets) pour l'insérer. Les nœuds insérés à partir de l'arborescence du schéma dans le volet de gauche sont insérés par rapport au nœud de contexte (si la case *Relative XPath* est cochée) ou sous forme d'expression absolue à partir du nœud du document (si la case *XPath relatif* n'est pas cochée).

4. Cliquez sur **OK** pour terminer l'insertion de la condition. La condition est créée avec une branche, dont le test est l'expression XPath que vous avez saisie.

Modifier des expressions XPath de branches

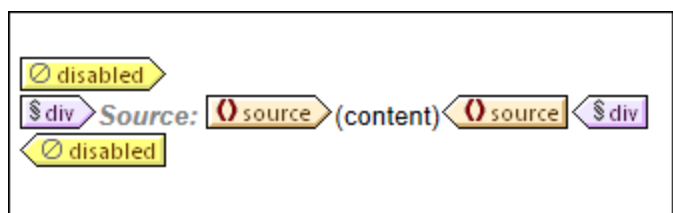
Pour modifier l'expression XPath d'une branche, sélectionnez la branche dans la vue Design. Ensuite, dans la barre latérale Propriétés, sélectionnez `branche de condition | quand`. Cliquez sur le bouton **Éditer**  pour l'élément XPath. Le dialogue *Éditer l'expression XPath* (*capture d'écran ci-dessus*) s'affiche, dans lequel vous pouvez modifier l'expression. Cliquez sur **OK** lorsque vous avez terminé.

Ajouter des branches, modifier l'ordre des branches et supprimer des branches

Pour ajouter de nouvelles branches, modifier l'ordre des branches et supprimer des branches, cliquez avec le bouton droit sur la branche souhaitée et sélectionnez l'élément correspondant dans le menu contextuel.

11.8.8 Désactivé

La commande **Désactivé** inclut le contenu sélectionné (y compris les composants de conception) dans un composant Désactivé (voir la capture d'écran ci-dessous).



Le contenu d'un composant Désactivé est ignoré dans la sortie. Vous pouvez donc inclure le contenu que vous souhaitez ignorer dans un composant Désactivé.

Pour rétablir le contenu désactivé, supprimez simplement les balises Désactivé autour du contenu. Pour ce faire, sélectionnez le composant Désactivé, cliquez avec le bouton droit de la souris et sélectionnez **Supprimer uniquement balise**.

11.8.9 Signet TdM et niveau TM

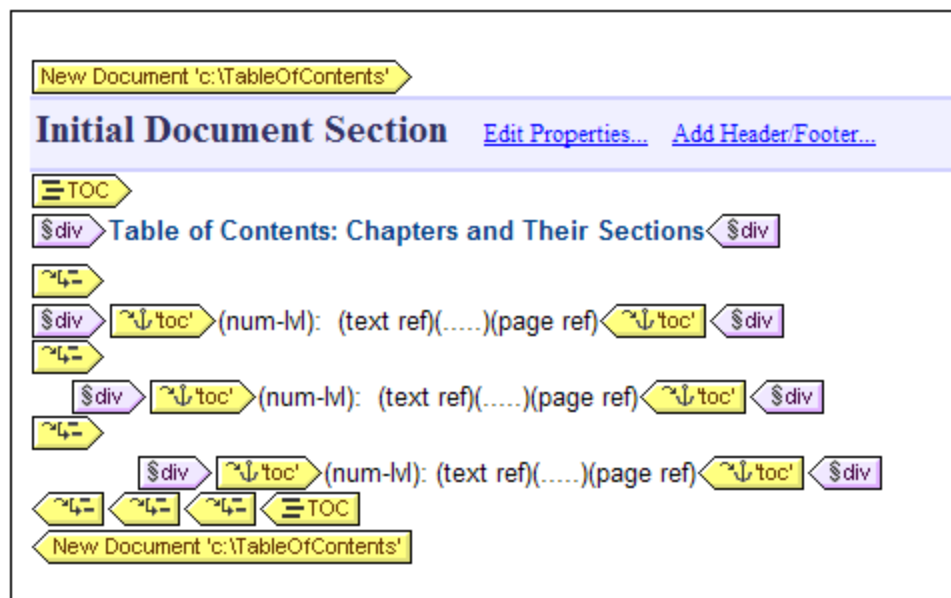
Lorsqu'un composant de la conception est sélectionné, il peut être inclus dans un ou plusieurs composants pertinents de la table des matières (TdM). La liste des commandes TdM est la suivante. Pour plus de détails sur leur utilisation, cliquez sur les liens correspondants ; ils vous mèneront à la section dédiée à l'utilisation de chaque composant TdM.

- [Signet TdM](#) ³⁰⁰
- [Signet TdM \(Assistant\)](#) ²⁹⁶
- [Niveau](#) ²⁹⁸
- [Référence de niveau](#) ³⁰⁵

Note : ces commandes sont également disponibles dans un menu contextuel, selon l'endroit où vous cliquez avec le bouton droit dans la conception.

11.8.10 Nouveau document

La commande **Inclure dans nouveau document** inclut la sélection actuelle avec un modèle Nouveau document (capture d'écran ci-dessous).



Le modèle Nouveau document contient une section Document initial qui contient la sélection de conception qui était mise en surbrillance lorsque la commande **Inclure dans nouveau document** a été sélectionnée. Dans la capture d'écran ci-dessus, le composant de conception TOC a été sélectionné et inclut dans un modèle Nouveau document. Le contenu peut désormais être saisi dans la Section initiale du document. Si vous le souhaitez, vous pouvez ajouter des Sections supplémentaires à la Section initiale du document à l'aide de la commande **Insérer | Insérer page/colonne/section de document**.

Un modèle Nouveau document crée un nouveau document dans la sortie. Par conséquent, la sortie comprendra plusieurs documents.

Pour une description détaillée de l'utilisation de plusieurs documents de sortie, voir la section [Sortie multiple de document](#) ²⁵¹.

11.8.11 Élément défini par l'utilisateur

La commande **Contenir dans | Élément défini par l'utilisateur** crée un [Élément défini par l'utilisateur](#) ¹²² autour de la sélection dans la conception. L'utilisation des éléments définis par l'utilisateur est décrite dans la section [Fichier SPS : Contenu | Éléments définis par l'utilisateur](#) ¹²².

11.9 Menu Table

Le menu **Table** fournit des commandes vous permettant d'insérer un tableau statique ou dynamique et de modifier la structure et les propriétés des tableaux statiques et dynamiques. Vous pouvez modifier la structure du tableau en ajoutant, insérant, supprimant, joignant et divisant des lignes et des colonnes. Les propriétés du tableau ainsi que celles des colonnes, lignes et cellules individuelles sont définies à l'aide de [styles CSS](#) ¹³⁵ et de [propriétés HTML pour les tableaux et leurs sous-composants](#) ¹³⁵.

Les commandes Table sont disponibles dans le menu **Table** (*voir la liste ci-dessous*) et sous forme d'icônes dans la [barre d'outils Table](#) ⁴⁴⁶. La disponibilité des différentes commandes de table dépend de la position actuelle du curseur. Un tableau peut être inséré à n'importe quel endroit dans le SPS en cliquant sur la commande [Insérer Table](#) ⁵¹⁶. Pour modifier la structure du tableau, placez le curseur dans la cellule, la colonne ou la ligne appropriée, puis sélectionnez la commande de modification requise. Pour modifier une propriété de formatage, placez le curseur dans la cellule, la colonne, la ligne ou le tableau approprié, puis, dans la [barre latérale Styles](#) ¹³⁵ et/ou [Propriétés](#) ¹³⁵, définissez la propriété requise pour ce composant du tableau.

Les commandes suivantes sont disponibles dans le menu Table :

- [Insérer table, Supprimer table](#) ⁵¹⁶
- [Ajouter en-têtes, pieds de page de table](#) ⁵¹⁷
- [Apposer/Insérer ligne/colonne](#) ⁵¹⁷
- [Supprimer ligne, colonne](#) ⁵¹⁸
- [Fusionner avec cellule gauche, droite, inférieure, supérieure](#) ⁵¹⁸
- [Fractionner la cellule horizontalement, verticalement](#) ⁵¹⁹
- [Afficher bordures de cellule, balisage de table](#) ⁵¹⁹
- [Propriétés de la table](#) ⁵¹⁹
- [Alignement vertical du contenu des cellules](#) ⁵²¹

En-têtes et pieds de page

Lorsque vous créez une table dynamique, vous pouvez spécifier si vous souhaitez inclure des en-têtes et/ou des pieds de page. (Les pieds de page ne sont autorisés que lorsque la table s'étend de haut en bas.) Vous pouvez créer un en-tête et un pied de page dans une table statique en insérant manuellement une ligne supérieure et une ligne inférieure, respectivement. La structure des en-têtes et des pieds de page dans les tableaux statiques et dynamiques peut être modifiée en divisant et en fusionnant des cellules.

Naviguer dans les tableaux

Utilisez les touches Maj et les touches fléchées pour naviguer dans les cellules du tableau.

Ajouter du contenu dans une cellule

Tout type de composant SPS peut être inséré comme contenu d'une cellule. Le composant doit être formaté à l'aide des outils de formatage standard.

11.9.1 Insérer table, Supprimer table

La commande **Insérer table**  permet d'insérer un tableau vide dans l'onglet Design. Sélectionner cette commande ouvre un dialogue dans lequel vous pouvez choisir de créer un tableau statique ou dynamique.

- Si vous choisissez de créer un tableau statique, un dialogue vous invite à indiquer la taille du tableau (en termes de lignes et de colonnes).
- Si vous choisissez de créer un tableau dynamique, le dialogue Sélecteur XPath s'affiche, dans lequel vous pouvez sélectionner le nœud qui sera converti en tableau dynamique. Lorsque vous cliquez sur **OK**, le dialogue Créer une table dynamique s'affiche, dans lequel vous pouvez sélectionner les nœuds enfants que vous souhaitez afficher comme champs de chaque élément du tableau. Pour plus de détails, voir [Créer des tables dynamiques](#) ¹²⁸.

Vous pouvez modifier la structure d'un tableau ultérieurement en ajoutant, insérant et supprimant des lignes et/ou des colonnes.

La commande **Supprimer table**  supprime le tableau statique ou dynamique sur lequel se trouve le curseur.

11.9.2 Ajouter en-têtes, pieds de page de table

Les en-têtes de tableau peuvent apparaître sous forme de ligne d'en-tête (au-dessus du corps du tableau) ou de colonne d'en-tête (à gauche du corps du tableau, bien que d'un point de vue balisage, une colonne d'en-tête puisse être placée à l'intérieur du corps du tableau). De même, les pieds de page du tableau peuvent apparaître sous forme de ligne de pied de page (sous le corps du tableau) ou de colonne de pied de page (à droite du corps du tableau, bien que, d'un point de vue balisage, un pied de page puisse être placé à l'intérieur du corps du tableau).

Note : dans le résultat HTML, les en-têtes de tableau étant encadrés par des éléments `th`, ils apparaissent en gras (car le formatage en gras est inhérent à l'élément `th`).

Les commandes Ajouter en-tête de table et Ajouter pied de page de table ajoutent des en-têtes et des pieds de page de tableau sous forme de colonnes et de lignes, comme suit :

	Ajouter colonne d'en-tête de table :	ajoute une colonne d'en-tête à gauche du corps du tableau.
	Ajouter colonne de pied de page de table :	ajoute une colonne de pied de page à droite du corps du tableau.
	Ajouter ligne d'en-tête de table :	ajoute une ligne d'en-tête au-dessus du corps du tableau.
	Ajouter ligne de pied de page de table :	ajoute une ligne de pied de page sous le corps du tableau.

11.9.3 Apposer/Insérer ligne/colonne

La commande **Apposer ligne**  ajoute une ligne au tableau statique ou dynamique dans lequel se trouve le curseur.

La commande **Insérer ligne**  insère une ligne au-dessus de la ligne dans laquelle se trouve le curseur. Cette commande s'applique aux tables statiques et dynamiques.

La commande **Apposer colonne**  ajoute une colonne à la table statique ou dynamique dans laquelle se trouve le curseur.

La commande **Insérer colonne**  insère une colonne à gauche de la colonne dans laquelle se trouve le curseur. Cette commande s'applique aux tables statiques et dynamiques.

11.9.4 Supprimer ligne, colonne

La commande **Supprimer ligne**  supprime la ligne sur laquelle se trouve le curseur. Cette commande s'applique aux tables statiques et dynamiques.

La commande **Supprimer colonne**  supprime la colonne dans laquelle se trouve le curseur. Cette commande s'applique aux tables statiques et dynamiques.

11.9.5 Fusionner avec cellule gauche, droite, inférieure, supérieure

La commande **Fusionner avec cellule gauche**  joint la cellule dans laquelle se trouve le curseur à la cellule adjacente à gauche. Le contenu des deux cellules est concaténé dans la nouvelle cellule. Toutes les valeurs de propriété de la cellule de gauche sont transférées vers la nouvelle cellule. Cette commande s'applique aux tables statiques et dynamiques.

La commande **Fusionner avec cellule droite**  joint la cellule dans laquelle se trouve le curseur à la cellule de droite. Le contenu des deux cellules est concaténé dans la nouvelle cellule. Toutes les valeurs de propriété de la cellule de gauche sont transférées vers la nouvelle cellule. Cette commande s'applique aux tables statiques et dynamiques.

La commande **Fusionner avec cellule inférieure**  joint la cellule dans laquelle se trouve le curseur à la cellule située en dessous. Le contenu des deux cellules est concaténé dans la nouvelle cellule. Toutes les valeurs de propriété de la cellule du haut sont transférées vers la nouvelle cellule. Cette commande s'applique aux tables statiques et dynamiques.

La commande **Fusionner avec cellule supérieure**  joint la cellule dans laquelle se trouve le curseur à la cellule située au-dessus. Le contenu des deux cellules est concaténé dans la nouvelle cellule. Toutes les valeurs de propriété de la cellule du haut sont transférées vers la nouvelle cellule. Cette commande s'applique aux tables statiques et dynamiques.

11.9.6 Fractionner la cellule horizontalement, verticalement

La commande **Fractionner la cellule horizontalement**  crée une nouvelle cellule à droite de la cellule dans laquelle se trouve le curseur. Le contenu de la cellule d'origine reste dans la cellule d'origine. Toutes les propriétés de la cellule d'origine sont transférées à la nouvelle cellule. Cette commande s'applique aux tables statiques et dynamiques.

La commande **Fractionner la cellule verticalement**  crée une nouvelle cellule sous la cellule dans laquelle se trouve le curseur. Le contenu de la cellule d'origine reste dans la cellule supérieure. Toutes les propriétés de la cellule d'origine sont transférées à la nouvelle cellule. Cette commande s'applique aux tables statiques et dynamiques.

11.9.7 Afficher bordures de cellule, Afficher balisage de table

Les commandes **Afficher bordures de cellule** et **Afficher balisage de table** affichent respectivement les bordures des cellules et le balisage des colonnes et des lignes du tableau. Lorsque ces deux options sont activées, vous pouvez mieux comprendre la structure du tableau. Cependant, lorsqu'elles sont désactivées, vous pouvez visualiser le tableau avec plus de précision.



La commande **Afficher bordures de cellule** active ou désactive l'affichage des limites (bordures) des tableaux dont la valeur de bordure est 0.

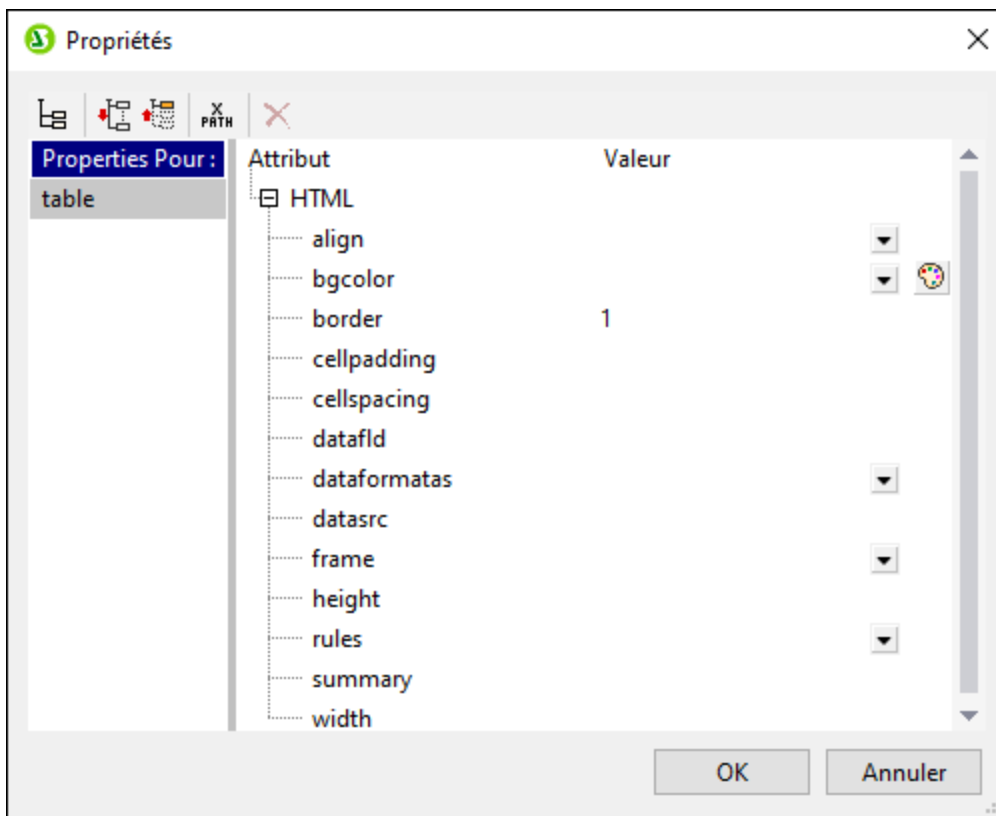


La commande **Afficher balisage de table** active ou désactive l'affichage des marqueurs bleus des colonnes et des lignes.

11.9.8 Propriétés de la table



La commande **Propriétés de la table** est activée lorsque le curseur est placé à l'intérieur d'un [tableau statique ou dynamique](#) ¹²⁵. En cliquant sur la commande, la barre latérale Propriétés s'affiche, avec le composant *Table* sélectionné (*capture d'écran ci-dessous*).

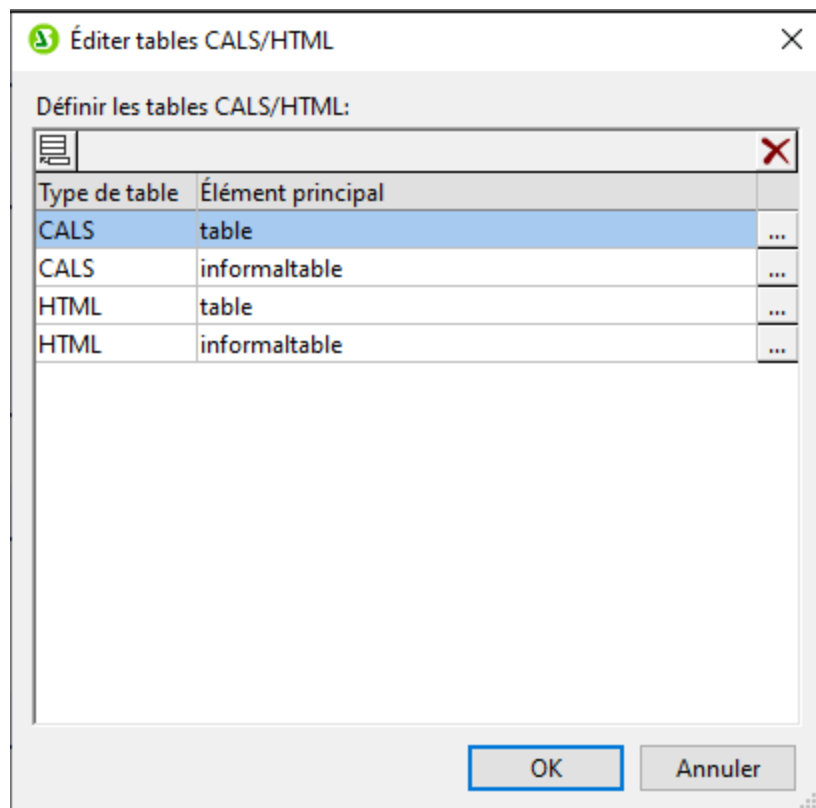


Vous pouvez maintenant modifier les propriétés du tableau. Cliquez sur **OK** lorsque vous avez terminé.

11.9.9 Éditer tables CALS/HTML

Si le schéma sous-jacent du SPS comporte un élément avec une sous-structure qui suit le modèle de table CALS ou HTML, vous pouvez générer automatiquement cet élément XML sous forme de table dans les sorties. Il est nécessaire de spécifier quel élément du schéma SPS correspond à quel élément de table CALS ou HTML. Le balisage du tableau dans les formats de sortie est alors dérivé directement du document XML.

La commande **Éditer tables CALS/HTML** affiche le dialogue Éditer tables CALS/HTML (*capture d'écran ci-dessous*). Dans le dialogue, sélectionnez le nom de l'élément de schéma SPS qui correspond à un élément de tableau CALS ou HTML. Vous pouvez spécifier les styles de formatage de base des tableaux dans le dialogue et ajouter des styles ultérieurement dans le design.



Pour plus de détails sur les tableaux CALS/HTML, voir la section [Tables](#) ¹²⁵.

11.9.10 Alignement vertical du contenu des cellules

Les commandes permettant de définir l'alignement vertical du contenu des cellules sont disponibles sous forme d'icônes dans la barre d'outils Table. Placez le curseur n'importe où dans la cellule, puis cliquez sur l'icône souhaitée.



Aligner en haut verticalement aligne verticalement le contenu de la cellule avec le haut de la cellule.



Centrer verticalement aligne verticalement le contenu de la cellule avec le milieu de la cellule.



Aligner en bas verticalement aligne verticalement le contenu de la cellule avec le bas de la cellule.

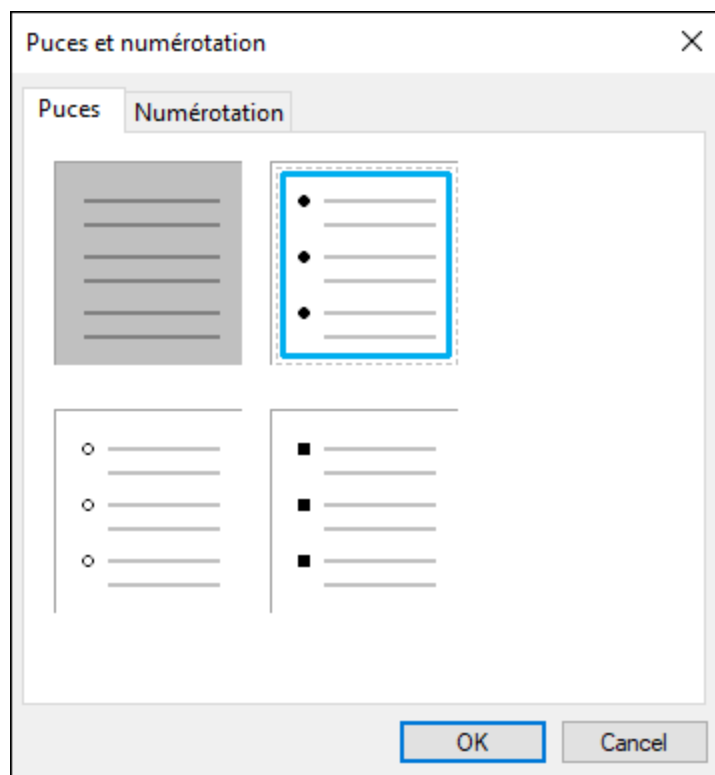
11.10 Menu Propriétés

Le menu **Propriétés** contient des commandes qui vous permettent d'insérer des listes et de définir des formats de type de données pour la fonctionnalité de [formatage des entrées](#) ³³⁴. La description des commandes est organisée dans les sous-sections suivantes :

- Commande [Éditer les puces et numérotation](#) ⁵²², pour insérer des listes.
- Commande [Chaîne de formatage de valeur prédéfinie](#) ⁵²³, pour définir des formats de type de données numériques pour un SPS donné.

11.10.1 Éditer les puces et numérotation

La commande **Éditer les puces et numérotation** vous permet d'insérer une liste à l'emplacement du curseur. En cliquant sur la commande, le dialogue Puces et numérotation s'affiche (*capture d'écran ci-dessous*), dans lequel vous pouvez sélectionner le style de liste ; dans le cas d'une liste numérotée, le numéro initial peut également être spécifié.



11.10.2 Chaîne de formatage de valeur prédéfinie

Tout espace réservé (contenu), champ de saisie ou calcul automatique de type `numeric`, `date`, `time`, `dateTime` ou `duration` peut se voir attribuer un format personnalisé à l'aide du dialogue [Formatage des valeurs](#)³³⁴. Dans le dialogue Formatage des valeurs, vous pouvez soit créer un format directement, soit sélectionner un format prédéfini dans une liste déroulante.

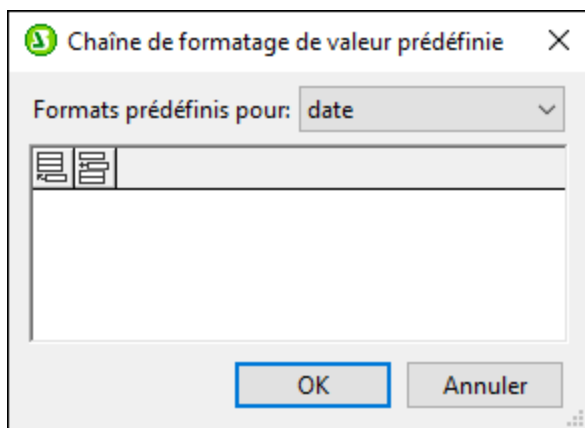
Les formats prédéfinis disponibles dans la liste déroulante sont de deux types :

- les formats prédéfinis fournis avec StyleVision et
- les formats prédéfinis créés par l'utilisateur à l'aide de la commande **Chaînes de formatage de valeur prédéfinie** (cette commande). Lorsqu'un utilisateur crée des formats de valeur prédéfinis, ceux-ci sont créés pour le fichier SPS actuellement ouvert, et non pour l'ensemble de l'application. Une fois que l'utilisateur a créé des formats de valeur prédéfinis, le fichier SPS doit être enregistré afin que les formats soient disponibles lors de la prochaine ouverture du fichier.

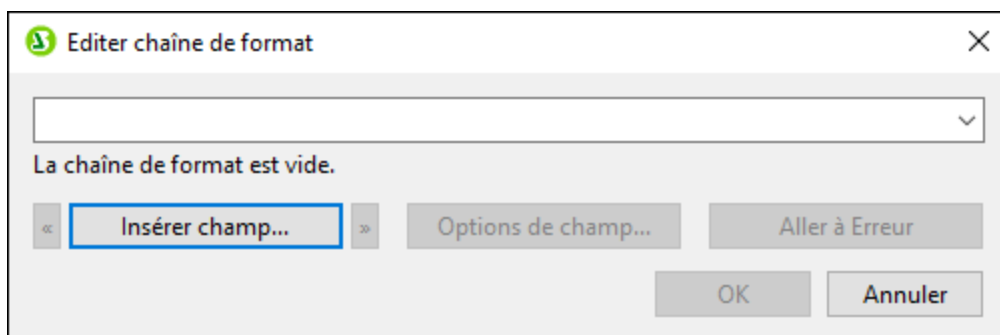
Créer une chaîne de formatage de valeur prédéfinie

Une chaîne de formatage de valeur prédéfinie est spécifique à un type de données. Pour créer une chaîne de formatage de valeur prédéfinie, procédez comme suit :

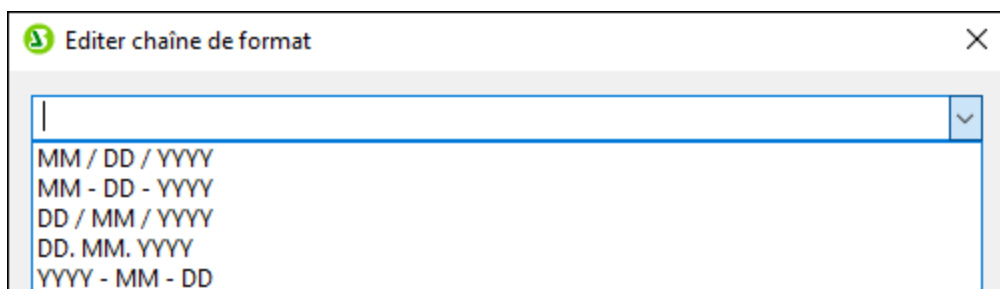
1. Cliquez sur **Propriétés | Chaîne de formatage de valeur prédéfinie**. Le dialogue suivant s'affiche :



2. Sélectionnez un type de données dans la liste déroulante de la zone de liste déroulante, puis cliquez sur l'icône **Ajouter** ou **Insérer** selon vos besoins. Le dialogue Éditer chaîne de format s'affiche :



Si vous cliquez sur la flèche vers le bas de la liste déroulante, une liste déroulante contenant les formats prédéfinis fournis par StyleVision pour ce type de données s'affiche (comme illustré dans la capture d'écran ci-dessous).



Vous pouvez soit sélectionner un format dans la liste et le modifier, soit saisir un format directement dans le champ de saisie. La syntaxe permettant de définir un format est expliquée dans la section [Formatage des valeurs](#) ³³⁴. Si vous avez besoin d'aide concernant la syntaxe, utilisez les boutons **Insérer champ** et **Options de champ**.

- Une fois le format défini, cliquez sur **OK** et enregistrez le fichier SPS. La chaîne de formatage est ajoutée à la liste des formats prédéfinis pour ce type de données et apparaîtra comme option dans le dialogue Formatage des valeurs (du fichier SPS actuel) lorsque l'élément sélectionné est du type de données correspondant.

Notez les points suivants :

- Vous pouvez ajouter autant de chaînes de formatage personnalisées que vous le souhaitez pour différents types de données.
- L'ordre séquentiel des chaînes de format dans le dialogue Chaînes de formatage prédéfinies détermine l'ordre dans lequel ces chaînes de format apparaissent dans le dialogue Formatage des valeurs. Les chaînes de formatage personnalisées apparaissent au-dessus des formats prédéfinis fournis.
- Pour modifier une chaîne de formatage personnalisée, double-cliquez sur l'entrée dans le dialogue Chaînes de format prédéfinies.
- Pour supprimer une chaîne de format personnalisée, sélectionnez-la et cliquez sur l'icône **Supprimer** dans le dialogue Chaîne de formatage de valeur prédéfinie.

11.11 Menu Outils

Le menu **Outils** contient la commande de vérification orthographique et les commandes qui vous permettent de personnaliser StyleVision.

La description des commandes du menu Outils est organisée en plusieurs sous-sections :

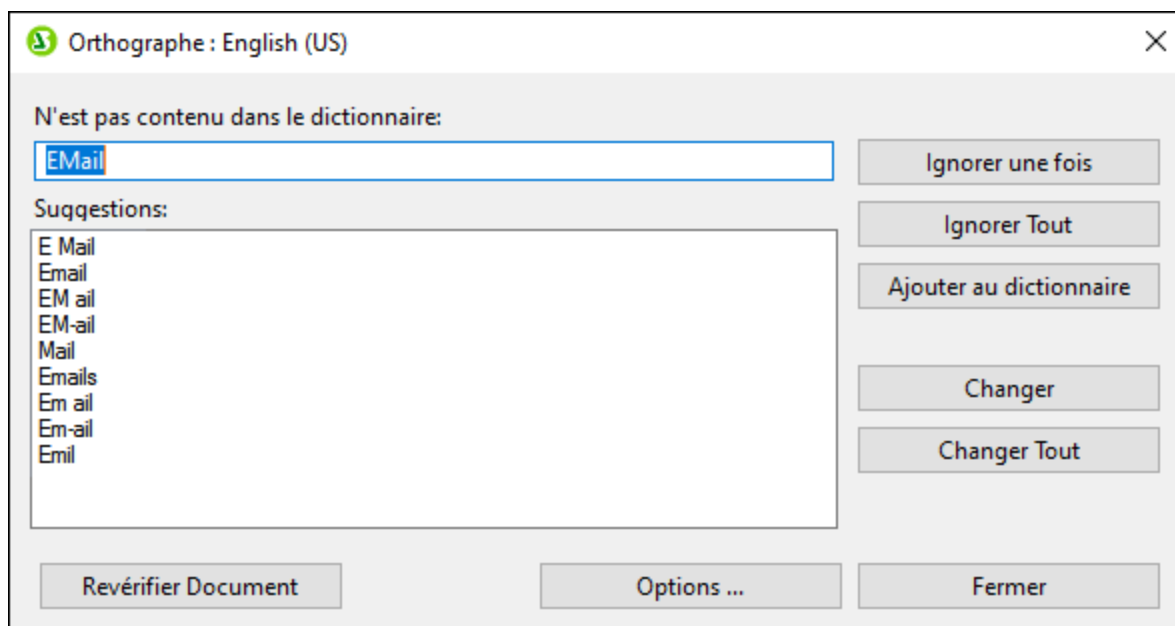
- [Orthographe](#) ⁵²⁵
- [Options du vérificateur d'orthographe](#) ⁵²⁶
- [Gestionnaire de schéma XML](#) ⁵²⁹
- [Personnaliser](#) ⁵³⁰

11.11.1 Orthographe

La commande **Orthographe** effectue une vérification orthographique sur le fichier SPS (dans le mode Design). Vous pouvez sélectionner la langue à utiliser parmi les dictionnaires linguistiques intégrés au vérificateur orthographique (*voir note ci-dessous*).

Note : la sélection des dictionnaires intégrés fournis avec les logiciels Altova ne reflète pas les préférences linguistiques d'Altova, mais repose principalement sur la disponibilité des dictionnaires dont la redistribution avec des logiciels commerciaux est autorisée, tels que les licences [MPL](#), [LGPL](#) ou [BSD](#). Il existe de nombreux autres dictionnaires open source, mais ils sont distribués sous des licences plus restrictives, telles que la licence [GPL](#). La plupart de ces dictionnaires sont disponibles dans un programme d'installation séparé, accessible à l'adresse <http://www.altova.com/dictionaries>. Il vous appartient de décider si vous acceptez les conditions de la licence applicable au dictionnaire et si celui-ci est adapté à l'utilisation que vous faites du logiciel sur votre ordinateur.

Lorsque vous cliquez sur cette commande, le dialogue ci-dessous s'affiche. Les mots qui ne figurent pas dans le dictionnaire sélectionné s'affichent, dans l'ordre du document et un par un, dans le champ « N'est pas contenu dans le dictionnaire » du dialogue et sont mis en évidence dans le document de design.

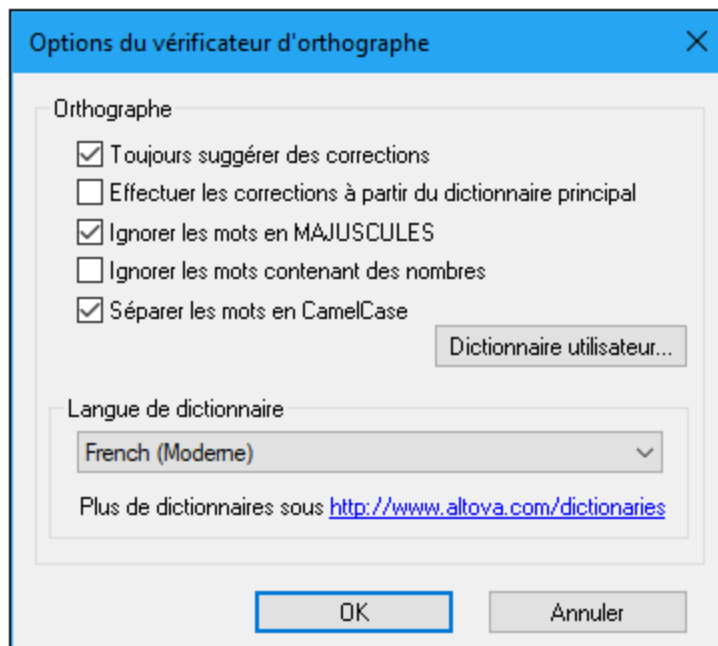


Vous pouvez ensuite sélectionner une entrée dans la liste du volet Suggestions et cliquer sur **Changer** ou **Changer Tout** pour modifier respectivement l'occurrence mise en évidence de cette orthographe ou toutes ses occurrences. (Un double-clic sur un mot dans la liste Suggestions le remplace par le mot inconnu.) Vous pouvez également ignorer *cette occurrence* du mot inconnu (**Ignorer une fois**) ; ou ignorer *toutes les occurrences* de ce mot inconnu (**Ignorer Tout**) ; ou ajouter ce mot inconnu au dictionnaire utilisateur (**Ajouter au dictionnaire**). L'ajout du mot inconnu au dictionnaire permet au correcteur orthographique de traiter le mot comme correct et de passer au mot suivant qui ne figure pas dans le dictionnaire. Vous pouvez revérifier le document depuis le début (**Revérifier Document**) ou fermer le dialogue (**Fermer**) à tout moment.

Le bouton **Options** ouvre le dialogue [Options du vérificateur d'orthographe](#)⁵²⁶, dans lequel vous pouvez spécifier les options de vérification orthographique.

11.11.2 Options du vérificateur d'orthographe

La commande **Options du vérificateur d'orthographe** ouvre un dialogue (illustré ci-dessous) dans lequel vous pouvez définir les options de vérification orthographique.



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](http://www.altova.com/dictionaries).

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 :

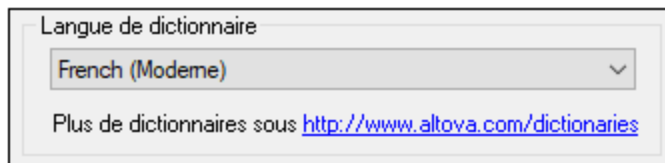
```
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (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 :

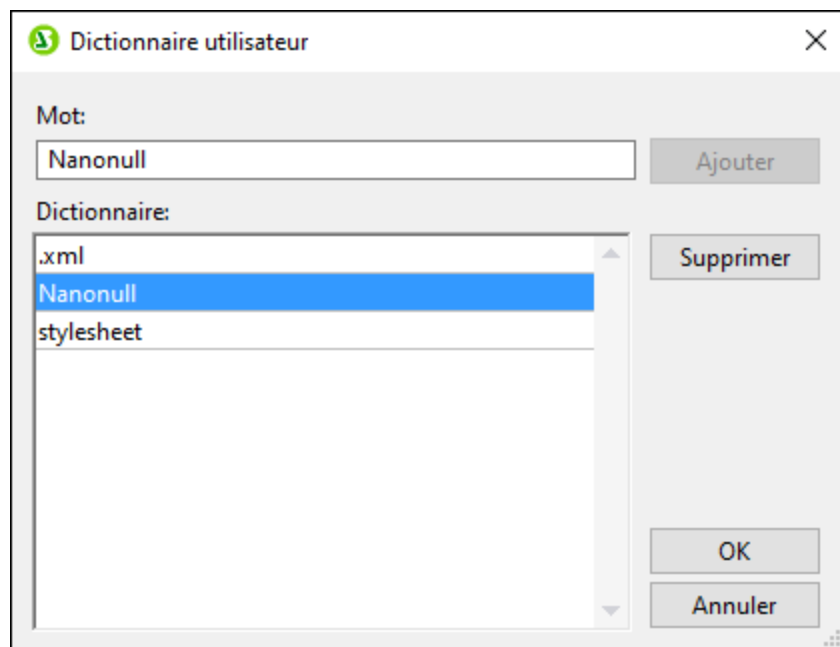
- En ajoutant les dictionnaires Hunspell dans un nouveau sous-dossier du dossier `Lexicons`. Les dictionnaires Hunspell peuvent être téléchargés, par exemple depuis <http://wiki.services.openoffice.org/wiki/Dictionaryes> ou <http://extensions.services.openoffice.org/en/dictionaries>. (Veuillez noter 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](#)⁶²⁵ 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

11.11.3 Gestionnaire de schéma XML

Cette commande ouvre le dialogue Gestionnaire de schéma XML, qui vous permet de gérer vos paquets de schémas XML.

Pour installer un schéma XML, cochez la case située à côté du schéma que vous souhaitez installer, puis cliquez sur **Appliquer**. Vous pouvez également désinstaller des schémas, mettre à niveau des schémas, rechercher de nouveaux schémas et, de manière générale, gérer tous vos schémas à partir d'un emplacement centralisé.

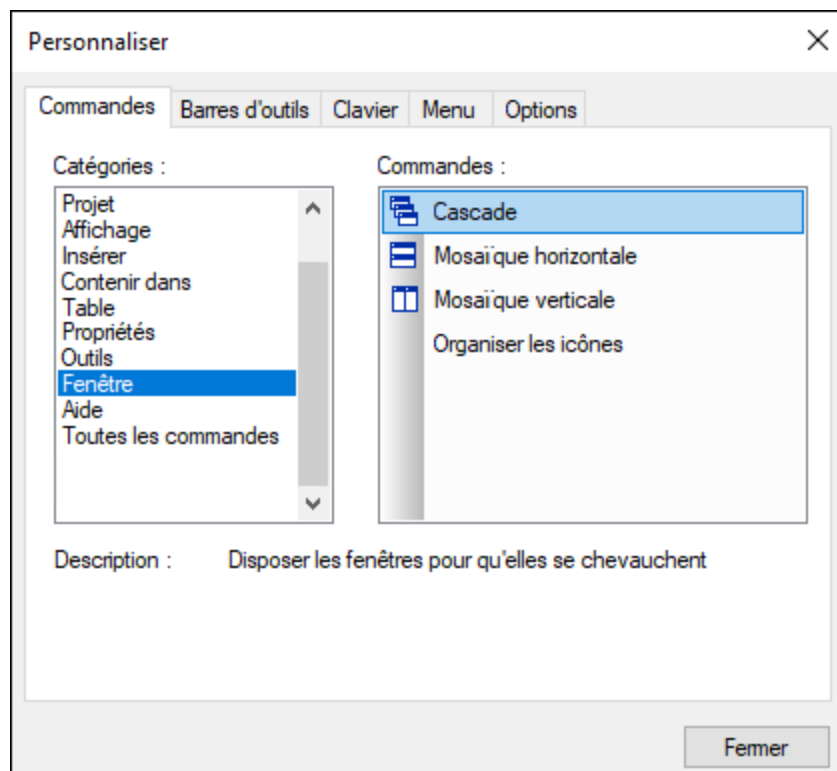
Pour plus d'informations, voir [Gestionnaire de schéma](#) ¹⁹⁶.

11.11.4 Personnaliser

La commande Personnaliser vous permet de personnaliser StyleVision en fonction de vos besoins personnels.

Onglet Commandes

L'onglet **Commandes** du dialogue Personnaliser vous permet de placer des commandes individuelles dans la barre de menus et la barre d'outils.



Pour **ajouter une commande** à la barre de menus ou à la barre d'outils, sélectionnez la commande dans le volet Commandes de l'onglet Commandes, puis faites-la glisser vers la barre de menus ou la barre d'outils. Lorsque le curseur est placé sur une position valide, un I apparaît et la commande peut être déposée à cet emplacement. Si l'emplacement n'est pas valide, une coche apparaît. Lorsque vous déposez la commande, elle est créée sous forme d'icône si elle est déjà associée à une icône ; sinon, la commande est créée sous forme de texte. Après avoir ajouté une commande à la barre de menus ou à la barre d'outils, vous pouvez modifier son apparence en cliquant dessus avec le bouton droit de la souris, puis en sélectionnant l'action souhaitée.

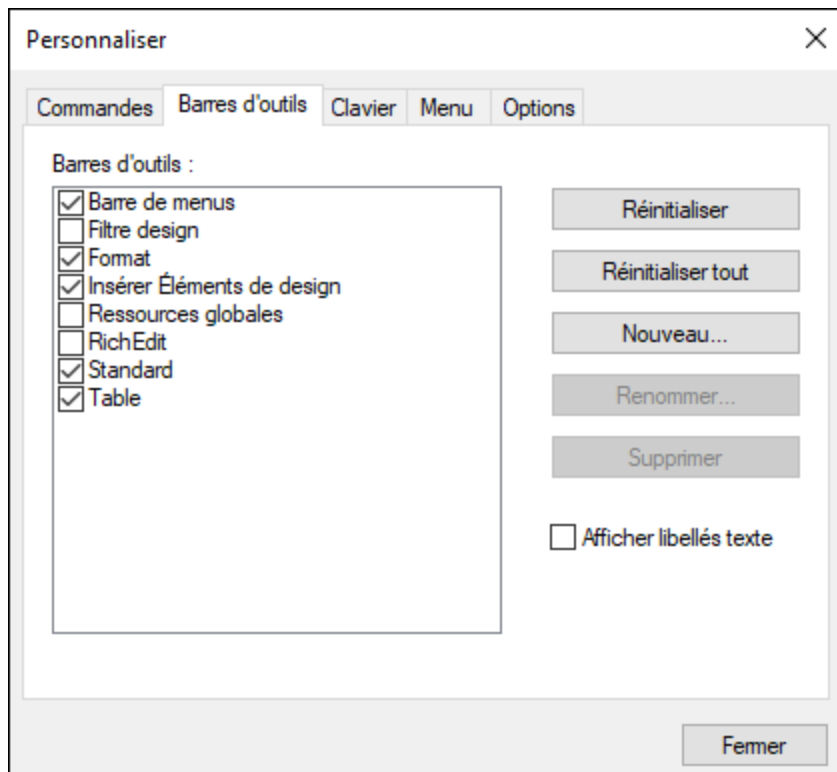
Pour **supprimer** un élément de la barre de menus ou de la barre d'outils, ouvrez le dialogue Personnaliser, cliquez avec le bouton droit de la souris sur l'élément à supprimer et sélectionnez **Supprimer**.

Note :

- La personnalisation décrite ci-dessus s'applique à l'application, que le document soit ouvert dans StyleVision ou non.
- Pour rétablir les menus et les barres d'outils tels qu'ils étaient lors de l'installation de StyleVision, accédez à l'onglet Barres d'outils et cliquez sur le bouton **Réinitialiser** approprié.

Onglet Barres d'outils

L'onglet **Barres d'outils** vous permet d'activer ou de désactiver des barres d'outils spécifiques, d'afficher les libellés des éléments de la barre d'outils et de rétablir l'état d'installation de la barre de menus et des barres d'outils.



L'interface StyleVision affiche une barre de menus fixe et plusieurs barres d'outils facultatives (Filtre design, Format, Standard, Table et Table des matières).

Chaque barre d'outils peut être divisée en groupes de commandes. Les commandes peuvent être ajoutées à une barre d'outils via l'onglet Commandes. Une barre d'outils peut être déplacée de sa position d'ancrage vers n'importe quel emplacement de l'écran. Un double-clic sur la barre de titre d'une barre d'outils (agrandie ou réduite) permet d'ancrer ou de désancrer la barre d'outils.

Dans l'onglet Barres d'outils du dialogue Personnaliser, vous pouvez activer ou désactiver une barre d'outils en cochant ou décochant sa case. Lorsqu'une barre d'outils est sélectionnée (dans l'onglet Barres d'outils), vous pouvez afficher les libellés de texte des éléments de cette barre d'outils en cochant la case **Afficher libellés texte**. Vous pouvez également réinitialiser une barre d'outils sélectionnée à l'état dans lequel elle se trouvait

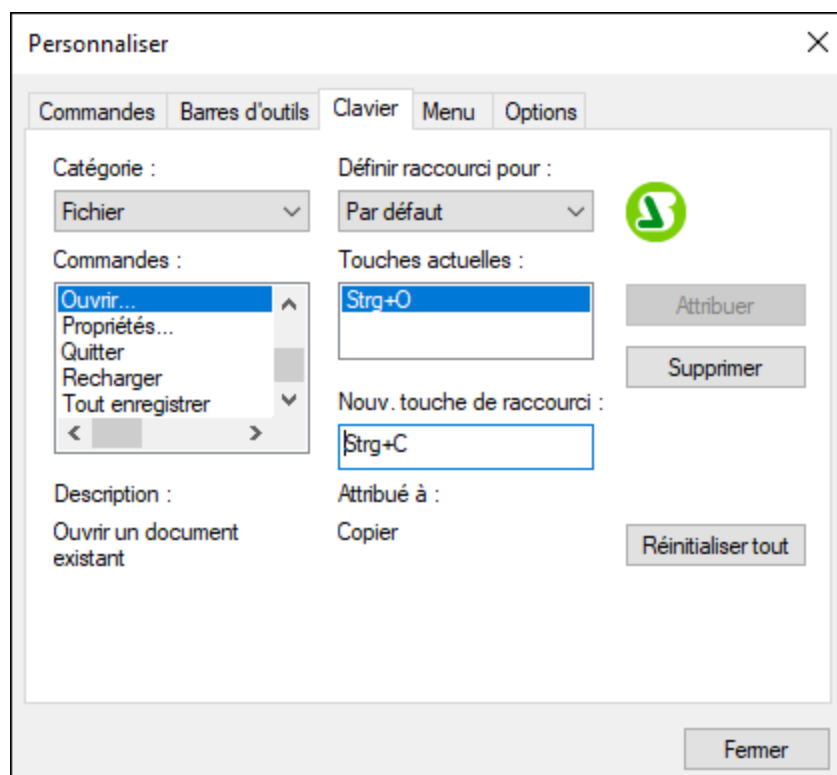
lors de l'installation de StyleVision en cliquant sur le bouton **Réinitialiser**. Vous pouvez réinitialiser toutes les barres d'outils et la barre de menus en cliquant sur le bouton **Tout réinitialiser**.

Note concernant la barre de menus

Vous pouvez ajouter des commandes à la barre de menus et supprimer des éléments de celle-ci : voir Commandes ci-dessus. Pour réinitialiser la barre de menus à son état lors de l'installation de StyleVision, sélectionnez Barre de menus dans l'onglet Barres d'outils du dialogue Personnaliser, puis cliquez sur le bouton **Réinitialiser**. (Cliquez sur le bouton **Réinitialiser tout** pour réinitialiser également les barres d'outils.)

Onglet Clavier

L'onglet **Clavier** vous permet de définir (ou de modifier) les raccourcis clavier pour toute commande StyleVision.



Pour attribuer un raccourci à une commande

1. Sélectionnez la catégorie dans laquelle se trouve la commande à l'aide de la liste déroulante Catégorie.
2. Sélectionnez la commande à laquelle vous souhaitez attribuer un raccourci dans la liste Commandes.
3. Cliquez dans le champ de saisie Nouv. touche de raccourci, puis appuyez sur les touches de raccourci qui doivent activer la commande. Le raccourci apparaît immédiatement dans le champ de saisie Nouv. touche de raccourci. Si ce raccourci a déjà été attribué à une commande, celle-ci s'affiche sous le champ de saisie. (Par exemple, dans la capture d'écran ci-dessus, **Ctrl+C** a déjà été attribué à la commande **Copier** et ne peut pas être attribué à la commande **Ouvrir fichier**). Pour effacer le champ de saisie Nouv. touche de raccourci, appuyez sur l'une des touches de contrôle, **Ctrl**, **Alt** ou **Maj**.
4. Cliquez sur le bouton **Attribuer** pour attribuer définitivement le raccourci. Le raccourci apparaît désormais dans la liste Touches actuelles.

Pour désattribuer (ou supprimer) un raccourci

1. Sélectionnez la commande pour laquelle le raccourci doit être supprimé.
2. Cliquez sur le raccourci que vous souhaitez supprimer dans la liste Touches actuelles.
3. Cliquez sur le bouton **Supprimer** (qui est désormais actif).

Pour réinitialiser toutes les affectations du clavier

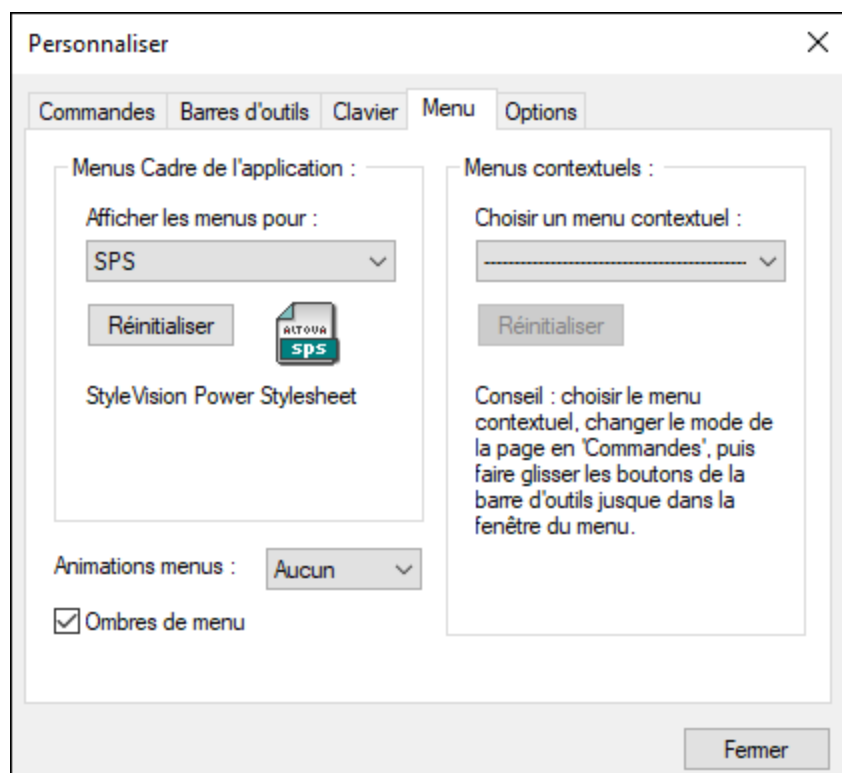
1. Cliquez sur le bouton **Réinitialiser tout** pour revenir aux raccourcis d'origine, tels qu'ils étaient au moment de l'installation. Un dialogue apparaît pour vous demander de confirmer si vous souhaitez réinitialiser toutes les affectations du clavier.
2. Cliquez sur **Oui** si vous souhaitez réinitialiser toutes les affectations du clavier.

Définir raccourci pour

Aucune fonction n'est actuellement disponible.

Onglet Menu

L'onglet **Menu** vous permet de personnaliser la barre de menu principale ainsi que les menus contextuels (menus accessibles par un clic droit). Il existe deux types de barres de menus principales : *Menu par défaut* (qui s'affiche lorsqu'aucun document n'est ouvert) et *SPS* (qui s'affiche lorsqu'un document SPS est ouvert).

Pour personnaliser un menu

1. Sélectionnez la barre de menus que vous souhaitez personnaliser (*menu SPS dans la capture d'écran ci-dessus*).
2. Cliquez sur l'onglet **Commandes** et faites glisser les commandes vers la barre de menus de votre choix.

Pour supprimer des commandes d'un menu

1. Dans le volet Menus Cadre de l'application, sélectionnez soit *Menu par défaut* (qui affiche les menus disponibles lorsqu'aucun document n'est ouvert), soit *SPS* (qui affiche les menus disponibles lorsqu'un ou plusieurs documents sont ouverts).
2. Une fois le dialogue Personnaliser ouvert, sélectionnez (i) le menu que vous souhaitez supprimer de la barre de menus de l'application, ou (ii) la commande que vous souhaitez supprimer de l'un de ces menus.
3. Soit (i) faites glisser le menu depuis la barre de menus ou la commande de menu depuis le menu, soit (ii) cliquez avec le bouton droit sur le menu ou la commande de menu et sélectionnez **Supprimer**.

Pour réinitialiser l'une des barres de menus

1. Sélectionnez l'entrée de menu que vous souhaitez réinitialiser dans la zone de liste déroulante du volet Menus Cadre de l'application.
2. Cliquez sur le bouton **Réinitialiser** situé juste en dessous du nom du menu. Un message apparaît pour vous demander si vous êtes certain de vouloir réinitialiser la barre de menus.

Pour personnaliser un menu contextuel (menu accessible par un clic droit)

1. Sélectionnez le menu contextuel dans la liste déroulante.
2. Cliquez sur l'onglet **Commandes** et faites glisser les commandes vers le menu contextuel qui est maintenant ouvert.

Pour supprimer des commandes d'un menu contextuel

1. Cliquez avec le bouton droit sur la commande ou l'icône représentant la commande.
2. Sélectionnez l'option **Supprimer** dans le menu contextuel ou faites glisser la commande hors du menu contextuel et déposez-la dès que l'icône en forme de coche apparaît sous le pointeur de la souris.

Pour réinitialiser un menu contextuel

1. Sélectionnez le menu contextuel dans la liste déroulante, puis
2. Cliquez sur le bouton **Réinitialiser** situé juste en dessous du nom du menu contextuel. Un message s'affiche pour vous demander si vous êtes certain de vouloir réinitialiser le menu contextuel.

Pour fermer une fenêtre de menu contextuel

- Cliquez sur l'icône **Fermer** en haut à droite de la barre de titre, ou
- Cliquez sur le bouton **Fermer** du dialogue Personnaliser.

Animations de menu

L'option Animations menus spécifie la manière dont un menu s'affiche lorsqu'on clique dessus. Sélectionnez une option dans la liste déroulante des animations du menu.

Ombres de menu

Si vous souhaitez que les menus s'affichent avec une ombre autour, sélectionnez cette option. Tous les menus auront alors une ombre.

Onglet Options

L'onglet **Options** vous permet de personnaliser des fonctionnalités supplémentaires de la barre d'outils.

Les info-bulles des éléments de la barre d'outils s'affichent si l'option Afficher les info-bulles est cochée. L'option Info-bulles comporte une sous-option permettant de choisir d'afficher ou non les raccourcis (lorsqu'ils sont disponibles) dans les info-bulles.

11.11.5 Restaurer barres d'outils et fenêtres

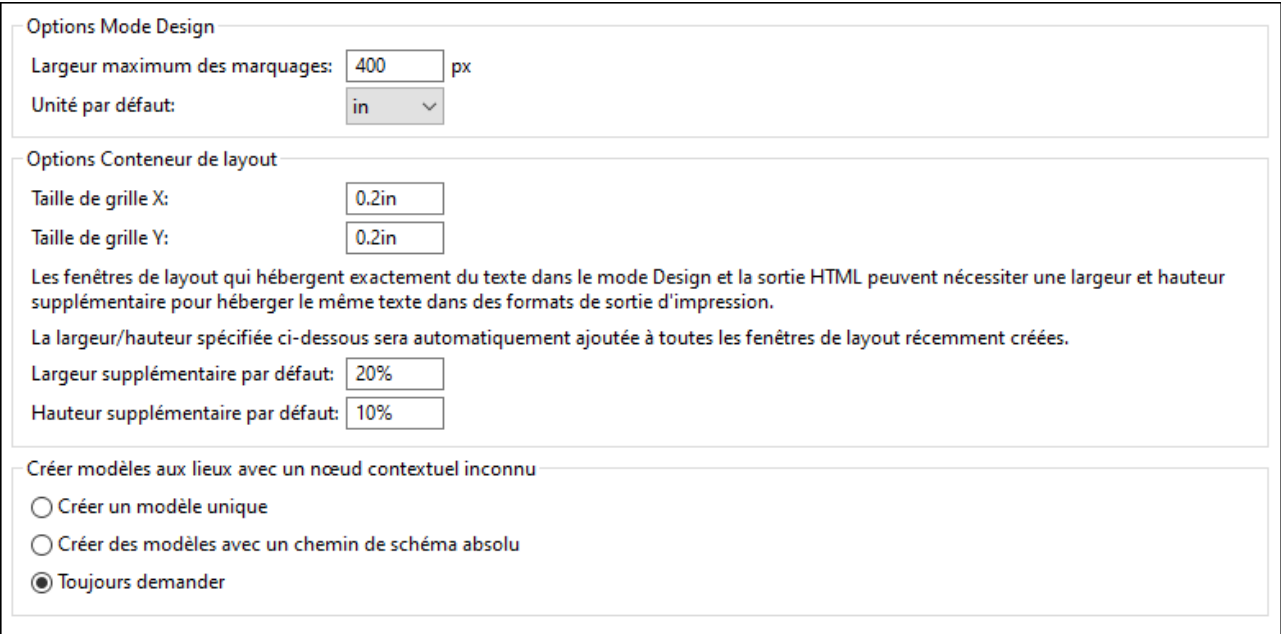
Cette commande rétablit l'état par défaut des barres d'outils, des fenêtres, des aides à la saisie et des autres composants de l'interface graphique. Il est nécessaire de redémarrer StyleVision pour que les modifications prennent effet.

11.11.6 Options

La commande **Options** ouvre un dialogue (*capture d'écran ci-dessous*) dans lequel vous pouvez spécifier l'encodage du fichier de sortie HTML.

Options de design

Dans l'onglet Design (*capture d'écran ci-dessous*), vous pouvez définir les options générales de conception pour l'ensemble de l'application.



Options Mode Design

Largeur maximum des marquages: 400 px

Unité par défaut: in

Options Conteneur de layout

Taille de grille X: 0.2in

Taille de grille Y: 0.2in

Les fenêtres de layout qui hébergent exactement du texte dans le mode Design et la sortie HTML peuvent nécessiter une largeur et hauteur supplémentaire pour héberger le même texte dans des formats de sortie d'impression.

La largeur/hauteur spécifiée ci-dessous sera automatiquement ajoutée à toutes les fenêtres de layout récemment créées.

Largeur supplémentaire par défaut: 20%

Hauteur supplémentaire par défaut: 10%

Créer modèles aux lieux avec un nœud contextuel inconnu

☐ Créer un modèle unique

☐ Créer des modèles avec un chemin de schéma absolu

☒ Toujours demander

Les options suivantes peuvent être définies :

- Largeur maximale (en pixels) des balises de marquage. Entrez le nombre entier positif correspondant au nombre de pixels requis.
- Taille de la grille des conteneurs de mise en page en unités de longueur absolues. Les longueurs spécifiées correspondent aux distances entre deux points sur l'axe de la grille respectif.
- Largeur et hauteur supplémentaires par défaut des boîtes de mise en page. Ces longueurs supplémentaires sont ajoutées à toutes les boîtes de mise en page afin de fournir la longueur supplémentaire souvent nécessaire pour s'adapter aux rendus de texte plus grands des formats d'impression. Ces valeurs peuvent être spécifiées sous forme de pourcentages ou d'unités de longueur

absolues.

- Comportement par défaut lorsqu'un modèle de nœud est créé à un emplacement où le nœud de contexte n'est pas connu. Cette option s'applique généralement aux modèles définis par l'utilisateur dans lesquels le modèle a été créé pour des éléments qui ne peuvent pas être placés dans le contexte de la source de schéma du design. Si un nœud est créé dans un tel modèle défini par l'utilisateur, il peut être créé avec (i) uniquement son nom ou (ii) avec le chemin d'accès complet à partir de la racine du schéma. Vous pouvez définir l'une de ces options comme comportement par défaut ou demander à être invité à chaque fois que cette situation se présente. La sélection par défaut pour cette option est *Toujours demander*.

Aperçus

Dans l'onglet Aperçu (*capture d'écran ci-dessous*), vous pouvez sélectionner les options pour les aperçus.

- *Sortie HTML* : sélectionnez si vous souhaitez utiliser Edge/Webview2 ou Internet Explorer comme navigateur pour les aperçus HTML.
- *Sortie PDF* : sélectionnez le lecteur PDF à utiliser pour l'aperçu PDF : (i) PDF.js (un lecteur PDF open source pour navigateurs) dans Edge ; (ii) Edge natif (le lecteur PDF intégré de Microsoft Edge) ; ou (iii) Adobe Acrobat Reader.
- *Fractionner les paramètres d'aperçu* : vous pouvez définir un délai pour les aperçus de sortie générés lors des actualisations automatiques. Les actualisations automatiques se produisent lorsque le design est modifié. Si vous définissez, par exemple, un délai nul, l'aperçu de sortie sera regénéré avec un délai nul à chaque fois que la conception seront modifiées. Si le délai est court, la génération de sortie sera déclenchée à intervalles courts et se produira donc à une fréquence élevée. Cela peut entraîner non seulement un ralentissement de la génération de l'aperçu de sortie, mais aussi des sorties intermédiaires, telles que des messages d'erreur qui apparaissent si une modification en cours entraîne une conception ou des données temporairement invalides. D'autre part, un délai long peut entraîner une longue attente pour l'aperçu de la sortie. Sélectionnez une valeur optimale en fonction de la taille de votre design et de vos données. Notez que le délai spécifié dans ce paramètre ne s'applique pas lorsque vous passez pour la première fois à l'aperçu fractionné ou lorsque vous effectuez un rafraîchissement manuel en cliquant sur le bouton **Rafraîchir** du volet Aperçu de sortie. Dans ces deux derniers cas, la sortie est générée immédiatement.

Options de schéma

Dans l'arborescence du schéma, les éléments et les attributs peuvent être classés par ordre alphabétique croissant. Pour ce faire, cochez les cases correspondantes dans l'onglet Schéma. Par défaut, les attributs sont classés par ordre alphabétique et les éléments sont classés dans un ordre correspondant à la structure du schéma, dans la mesure du possible.

Encodage par défaut

Pour définir l'encodage par défaut du fichier HTML de sortie, ouvrez le menu déroulant de la zone de liste déroulante, sélectionnez l'option souhaitée dans la liste des options d'encodage, puis cliquez sur **OK**. Chaque nouveau SPS que vous créerez à partir de ce moment définira l'encodage de sortie HTML tel que défini dans cet onglet.

Dans le XSLT-for-HTML, les informations d'encodage de sortie sont enregistrées aux emplacements suivants :

- Dans l'attribut `encoding` de l'élément `xsl:output` de la feuille de style :

```
<xsl:output version="1.0" encoding="UTF-8" indent="no" omit-xml-declaration="no" media-type="text/html" />
```


- Dans l'attribut `charset` de l'élément meta `content-type` situé dans l'en-tête HTML :
`<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`

Note : Ces paramètres sont les encodages par défaut et seront utilisés pour les nouveaux SPS. Il n'est pas possible de modifier l'encodage du SPS actuellement ouvert à l'aide de ce dialogue. Pour modifier l'encodage du fichier SPS actuellement ouvert, utilisez la commande [Fichier | Propriétés](#) ⁴⁷².

Options XSL

Dans les métadonnées des fichiers de sortie HTML, la ligne « Généré par StyleVision » sera générée par défaut. Les versions achetées du produit offrent la possibilité de désactiver la génération de cette ligne.

Options réseau

La section **Réseau** (voir la capture d'écran ci-dessous) vous permet de configurer les paramètres de proxy de réseau.

Réseau

Adresse IP

☐ Préférer adresses IPv6

Délai d'expiration

☒ Délai d'expiration du transfert: 40 s

☒ Délai d'expiration de la phase de connexion: 300 s

Certificat

☒ Vérifier le certificat serveur

☒ Vérifier l'identité du serveur TLS/SSL

Adresses IP

Lorsque les noms d'hôte résolvent plus d'une adresse dans les réseaux mixtes IPv4/IPv6, sélectionner cette option fait que les adresses IPv6 sont utilisées. Si l'option n'est pas sélectionnée dans de tels environnements, alors les adresses IPv4 sont utilisées.

Délai d'expiration

- *Délai d'expiration du transfert* : Si cette limite est atteinte pour deux packs de données consécutifs d'un transfert (envoyés ou reçus), alors le transfert entier est abandonné. Les valeurs peuvent être spécifiées en secondes [s] ou millisecondes [ms], avec le défaut étant 40 secondes. Si l'option n'est pas sélectionnée, alors il n'y a pas de limite de temps pour abandonner un transfert.
- *Délai d'expiration de la phase de connexion* : Il s'agit de la limite de temps au sein de laquelle la connexion doit être établie, y compris le temps pris pour les négociations de sécurité. Les valeurs peuvent être spécifiées en secondes [s] ou millisecondes [ms], avec le défaut étant 300 secondes. Ce délai d'expiration ne peut être désactivé :

Certificat

- *Vérifier le certificat de serveur TLS/SSL* : Si sélectionné, alors l'authenticité du certificat de serveur est

vérifiée en contrôlant la chaîne de signatures numériques jusqu'à ce qu'un certificat racine de confiance est atteint. Cette option est activée par défaut. Si cette option n'est pas sélectionnée, alors la communication n'est pas sécurisée, et les attaques (par exemple, une attaque « man-in-the-middle ») ne serait pas détectée. Notez que cette option ne vérifie pas que le certificat est en fait pour le serveur avec lequel il communique. Pour permettre une sécurité entière, les deux certificats et l'identité doivent être vérifiés (*voir la prochaine option*).

- *Vérifier l'identité de serveur TLS/SSL* : Si sélectionné, alors le certificat de serveur est vérifié pour appartenir au serveur avec lequel nous voulons communiquer. Ceci est fait en vérifiant que le nom de serveur dans l'URL est le même que le nom dans le certificat. Cette option est activée par défaut. Si cette option n'est pas sélectionnée, alors l'identité du serveur n'est pas vérifiée. Notez que cette option n'active pas la vérification du certificat du serveur. Pour permettre une sécurité entière, les deux certificats et l'identité doivent être vérifiés (*voir l'option précédente*).

Options 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 schéma 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.

Proxy de réseau

Configuration du proxy **Manuel** ▼

Proxy HTTP Port

☐ Utiliser ce serveur proxy pour tous les protocoles

Proxy SSL Port

Pas de Proxy pour

☐ Ne pas utiliser le serveur de proxy pour les adresses locales

Paramètres de proxy actuels

Tester URL

(en utilisant l'URL test http://www.example.com)
Sans utiliser de proxy.

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 (`.`).

Note : Si un serveur proxy a été défini et que vous voulez déployer une de mappage sur [Altova FlowForce Server](#), vous devez sélectionner l'option *Ne pas utiliser le serveur de proxy pour les adresses locales*.

Paramètres de proxy actuels

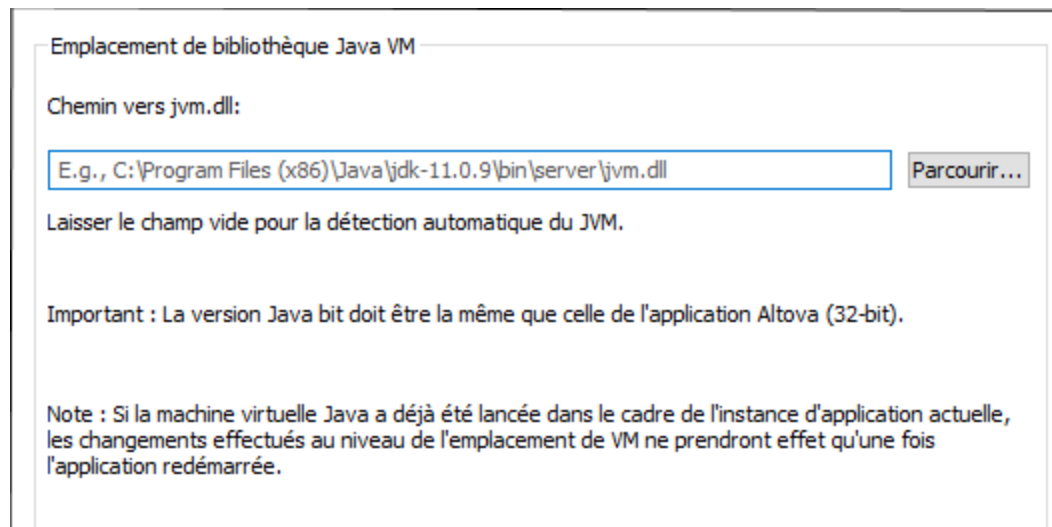
Fournit un journal verbeux de la détection de proxy. Il peut être réinitialisé avec la touche **Réinitialiser** située à droite du champ *Tester URL* (par exemple, en changeant l'URL de test, ou lorsque les paramètres de proxy ont été modifiés).

- *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*).

Options 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, StyleVision 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 StyleVision.



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.

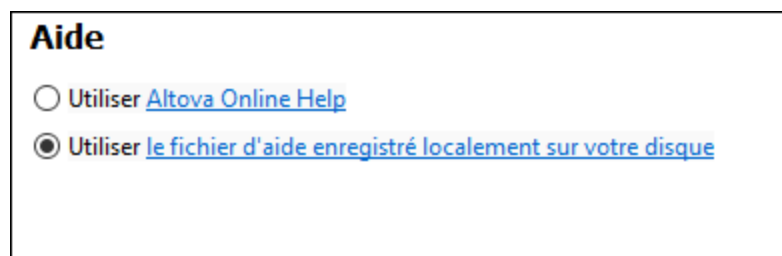
- Le chemin doit pointer vers le fichier `jvm.dll` provenant du répertoire `\bin\server` ou `\bin\client`, par rapport au répertoire sur lequel le JDK a été installé.
- La plateforme StyleVision (32-bit, 64-bit) doit être la même que celle du JDK.
- Une fois avoir modifié le chemin Java VM, vous devrez éventuellement redémarrer StyleVision pour que les nouveaux paramètres prennent effet.

Aide

StyleVision fournit de l'Aide (le manuel utilisateur) en deux formats :

- Aide en ligne, sous format HTML, qui est disponible sur le site web d'Altova. Afin d'accéder à l'Aide en ligne, vous aurez besoin d'un accès Internet.
- Un fichier Aide sous format PDF, est installé sur votre machine quand vous installez StyleVision. Il est appelé `StyleVision.pdf` et est situé dans le dossier d'application (dans le dossier de fichiers de programme). Si vous n'avez pas accès à Internet, vous pouvez toujours ouvrir ce fichier Aide enregistré localement.

L'option Aide (*capture d'écran ci-dessous*) vous permet de sélectionner lequel des deux formats s'ouvre quand vous cliquez sur la commande **Aide (F1)** dans le menu **Aide**.



Vous pouvez modifier cette option à tout moment pour la nouvelle sélection. Les liens dans cette section (*voir capture d'écran ci-dessus*) ouvre le format Aide respectif.

11.12 Menu Fenêtre

Le **menu Fenêtre** contient des commandes permettant de spécifier comment les fenêtres StyleVision doivent être affichées dans l'interface graphique.

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.

Pour maximiser une fenêtre, cliquez sur le bouton Maximiser de cette fenêtre.

Thèmes

StyleVision vous offre un choix des trois thèmes recensés ci-dessous. Lorsque vous sélectionnez un thème, il est appliqué immédiatement.

- Classique (le défaut)
- Clair
- Sombre

Liste de fenêtres actuellement ouvertes

Toutes les fenêtres de document actuellement ouvertes sont répertoriées au bas du menu **Fenêtre** par nom de document, la fenêtre active étant cochée. Pour activer une autre fenêtre, cliquez sur le nom de la fenêtre que vous souhaitez activer. Cette liste affiche toutes les fenêtres actuellement ouvertes et vous permet de passer rapidement de l'une à l'autre. Vous pouvez également utiliser les raccourcis clavier **CTRL+Tab** ou **CTRL+F6** pour parcourir les fenêtres ouvertes.

Dialogue Fenêtre

Tout en bas de la liste des fenêtres ouvertes se trouve une entrée pour le dialogue **Fenêtres**. Cliquez sur cette entrée pour ouvrir le dialogue Fenêtres, qui affiche une liste de toutes les fenêtres ouvertes par nom de fichier et fournit des commandes de boutons qui peuvent être appliquées à la ou aux fenêtres sélectionnées. (Une fenêtre est sélectionnée en cliquant sur son nom de fichier. Utilisez la touche **Ctrl** ou la touche **Maj** pour sélectionner plusieurs fichiers.)

Les commandes suivantes sont disponibles :

- *Activer* : rend le fichier sélectionné actif et ferme le dialogue.
- *OK* : ferme le dialogue.
- *Enregistrer* : enregistre le ou les fichiers sélectionnés.
- *Fermer la ou les fenêtres* : ferme le ou les fichiers sélectionnés.
- *Cascade, Mosaïque horizontale, Mosaïque verticale* : organise les fenêtres sélectionnées selon la commande choisie. Ces commandes seront désactivées si un seul fichier est sélectionné.
- *Réduire* : réduit la ou les fenêtres du ou des fichiers sélectionnés.

11.13 Menu Aide

Le menu **Aide** contient des commandes permettant d'accéder au manuel d'aide à l'écran de StyleVision, des commandes fournissant des informations sur StyleVision et des liens vers les pages d'assistance du site web d'Altova. Le menu Aide contient également le [dialogue Inscription](#) ⁵⁴³ qui vous permet de saisir votre clé de licence une fois que vous avez acheté le produit.

La description des commandes du menu **Aide** est organisée en plusieurs sous-sections :

- [Aide](#) ⁵⁴³
- [Activation, Bon de commande, Inscription, Mises à jour](#) ⁵⁴³
- [Autres commandes](#) ⁵⁴⁷

11.13.1 Aide

La commande **Aide (F1)** ouvre la documentation d'Aide de l'application (son manuel utilisateur). Par défaut, l'Aide en ligne sous le format HTML sur le site web d'Altova sera ouvert.

Si vous n'avez pas d'accès Internet ou ne voulez pas, pour une raison ou une autre, accéder à l'Aide en ligne, vous pouvez utiliser la version du manuel utilisateur stockée localement. La version locale est un fichier PDF appelé `StyleVision.pdf` qui est stocké dans le dossier d'application (dans le dossier des fichiers de programme).

Si vous voulez changer le format par défaut pour ouvrir (Aide en ligne ou PDF local), faites-le dans la section Aide du dialogue des Options (commande de menu **Outils | Options**).

11.13.2 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. Cliquez dessus pour obtenir votre licence. Quand vous cliquez sur ce bouton, votre ID de l'appareil sera hashé et envoyé à Altova via HTTPS. L'information liée à la licence sera envoyée à l'appareil via une réponse HTTP. Si la licence est créée avec succès, un dialogue à cet effet apparaîtra dans votre application d'Altova. En cliquant **OK** dans ce dialogue, le logiciel sera activé pour une période de 30 jours **sur cet appareil particulier**.
- **Clé de licence permanente.** Le dialogue **Activation du logiciel** contient un bouton pour acheter une clé de licence permanente. Cliquez 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 10 N 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 contiendra votre fichier de licence en pièce jointe. Le fichier de licence a une extension de fichier `.altova_licenses`.

Pour activer votre produit Altova, vous pouvez choisir une des étapes suivantes :

- Enregistrez le fichier 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**. Chercher ou saisir le champ vers le fichier de licence, et cliquer 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, saisissez 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.

Altova XMLSpy Enterprise Edition 2018 rel. 2 Activation du logiciel

Merci d'avoir choisi Altova XMLSpy Enterprise Edition 2018 rel. 2 et bienvenue dans le processus d'activation du logiciel. Vous pouvez visionner votre licence acquise ou sélectionner un Altova LicenseServer qui vous fournira une licence. (NOTE : Pour utiliser ce logiciel vous devez disposer d'une licence via Altova LicenseServer ou d'un code-clé de licence valide d'Altova.)

Si vous ne souhaitez pas utiliser Altova LicenseServer, cliquer ici pour saisir un code-clé Saisir un code-clé

Pour activer votre logiciel, veuillez saisir ou sélectionner le nom du Altova LicenseServer sur votre réseau.

Altova LicenseServer : ➡

☒ Une licence vous a déjà été attribuée sur LicenseServer sous VIEPDEV02.vie.altova.com.

Nom	
Société	Altova GmbH
Nbr d'utilisateurs	50
Type de licence	simultané
Expire dans	199
KSM	199 jours restants

Retourner Licence
Extraire la licence
Copier code de support
Enregistrer
Fermer

Connecté à Altova LicenseServer sous VIEPDEV02.vie.altova.com

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 annuler l'attribution à tout moment d'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'**Activation 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.

Si la licence étant extraite est une licence Utilisateur Installée ou licence Utilisateur Concurrent, alors la licence est extraite vers l'appareil et disponible à l'utilisateur qui a extrait la licence. Si la licence extraite est une Licence Utilisateur Nommée, alors la licence est extraite dans le compte Windows de l'utilisateur nommé. Le check-out de licence fonctionnera pour les appareils virtuels, mais pas pour le desktop virtuel (dans un VDI). Notez que quand une licence Utilisateur Nommée est extraite, les données à identifier ce check-out de licence sont stockées dans le profil utilisateur. Pour que le check-out de licence fonctionne, le profil d'utilisateur doit être stocké sur un appareil local qui sera utilisé pour le travail hors ligne. Si le profil de l'utilisateur est stocké à un emplacement non-local (tel que « file-share »), alors le checkout sera rapporté comme invalide quand l'utilisateur tente de démarrer l'application Altova.

Les check-in de licence doivent être de la même version majeure du produit d'Altova pour lequel la licence a été extraite. Donc assurez-vous d'archiver une licence avant que vous ne mettiez à jour votre produit d'Altova à 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.

11.13.3 Autres commandes

☐ Centre de support

Ouvre le formulaire d'assistance sur le site Web d'Altova, où vous pouvez soumettre une demande d'assistance.

☐ 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.

☐ StyleVision sur Internet

Un lien menant au [site web Altova](#) sur Internet. Vous pouvez en apprendre plus sur StyleVision, les technologies et produits liés le le [site web Altova](#).

☐ À propos de StyleVision

Affiche la fenêtre d'accueil et le numéro de version de votre produit. Si vous utilisez la version 64-bit de StyleVision, 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.

12 Annexes

Ces annexes contiennent (i) l'information sur les Moteurs XSLT utilisées dans StyleVision; (ii) l'information sur la conversion des types de données DB vers les types de données de schéma XML ; (iii) l'information technique sur StyleVision; et (iv) l'information de licence pour StyleVision. Chaque annexe contient des sous-sections recensées ci-dessous :

[Information sur le Moteur XSLT](#) ⁵⁴⁹

Fournit une information spécifique de mise en œuvre sur les Moteurs XSLT d'Altova, qui sont utilisés par StyleVision pour générer la sortie.

- Moteur Altova XSLT 1.0
- Moteur Altova XSLT 2.0
- Moteur Altova XSLT 3.0
- Fonctions XSLT et XPath/XQuery

[Données techniques](#) ⁶⁶¹

Fournit une information technique sur StyleVision.

- OS et exigences de mémoire
- Parseur XML Altova
- Moteurs XSLT et XQuery Altova
- Prise en charge Unicode
- Utilisation Internet

[Information de licence](#) ⁶⁶⁴

Contient des informations sur la manière dont StyleVision est distribué et sur sa mise sous licence.

- Distribution électronique de logiciel
- Mesure des licences
- Copyright
- Contrat de licence de l'utilisateur final

12.1 Informations concernant le moteur XSLT et XQuery

Les moteurs XSLT et XQuery de StyleVision 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 StyleVision.

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](#) ⁵⁴⁹
- [XSLT 2.0](#) ⁵⁴⁹
- [XSLT 3.0](#) ⁵⁵²
- [XQuery 1.0](#) ⁵⁵³
- [XQuery 3.1](#) ⁵⁵⁷

12.1.1 XSLT 1.0

Le moteur XSLT 1.0 de StyleVision 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, ` ` ; .

12.1.2 XSLT 2.0

Cette section :

- [Conformité du moteur](#) ⁵⁵⁰
- [Rétrocompatibilité](#) ⁵⁵⁰
- [Espaces de nom](#) ⁵⁵⁰

- [Compatibilité avec le schéma](#) ⁵⁵¹
- [Comportement spécifique à la mise en œuvre](#) ⁵⁵¹

Conformité

Le moteur XSLT 2.0 de StyleVision 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. Généralement, la compatibilité rétroactive du moteur XSLT 2.0 entre en jeu si vous utilisez le moteur XSLT 2.0 pour traiter une feuille de style XSLT 1.0 ou une instruction. 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'argument `href` accepte (i) les chemins relatifs pour les fichiers dans le dossier base-uri et (ii) les chemins absolus avec ou sans `file://` protocol. Les encodages pris en charge de manière supplémentaire sont (spécifiques à Altova): `x-binarytobase16` et `x-binarytobase64`. Exemple : `xs:base64Binary(unparsed-text('chart.png', 'x-binarytobase64'))`.

unparsed-text-available

L'argument `href` accepte (i) les chemins relatifs pour les fichiers dans le dossier base-uri et (ii) les chemins absolus avec ou sans `file://` protocol. Les encodages pris en charge de manière supplémentaire sont (spécifiques à Altova): `x-binarytobase16` et `x-binarytobase64`.

Note : les valeurs d'encodage suivantes, qui sont mises en œuvre dans des versions antérieures du produit prédécesseur de RaptorXML, AltovaXML, sont dépréciées à présent : `base16tobinary`, `base64tobinary`, `binarytobase16` and `binarytobase64`.

12.1.3 XSLT 3.0

Le moteur XSLT 3.0 de StyleVision 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](#)⁵⁴⁹. 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.

Espaces de noms

Votre feuille de style XSLT 3.0 devrait déclarer les espaces de noms pour que vous puissiez utiliser tous les types de constructeurs et fonctions disponibles dans XSLT 3.0. Les préfixes ci-dessous sont utilisés de manière conventionnelle ; vous pourriez utiliser des préfixes alternatifs si vous le souhaitez.

Nom Espaces de noms	Préfixe	Espace de noms URI
Types de schéma XML	xs:	http://www.w3.org/2001/XMLSchema
Fonctions XPath/XQuery 3.1	fn:	http://www.w3.org/2005/xpath-functions
Fonctions Math	math:	http://www.w3.org/2005/xpath-functions/math
Fonctions Map	map:	http://www.w3.org/2005/xpath-functions/map
Fonctions Array	array:	http://www.w3.org/2005/xpath-functions/array
Codes d'erreur XQuery, XSLT et XPath	err:	http://www.w3.org/2005/xpath-functions/xqt-errors
Fonctions de sérialisation	output	http://www.w3.org/2010/xslt-xquery-serialization

Généralement, ces espaces de noms seront déclarés dans l'élément `xsl:stylesheet` ou `xsl:transform`, tel que recensé dans la liste suivante :

```
<xsl:stylesheet version="3.0"
  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
```

Les points suivants doivent être notés :

- Le moteur XSLT 3.0 utilise les Fonctions XPath et XQuery et l'espace de noms de Operators 3.1 (recensé dans la table ci-dessus) comme son **espace de noms des fonctions par défaut**. Donc vous pouvez utiliser les fonctions de cet espace de noms dans votre feuille de style sans préfixe, alors vous pouvez utiliser le préfixe en plus assigné à la déclaration.
- Quand vous utilisez les constructeurs de type et les types depuis l'espace de noms du schéma XML, le préfixe utilisé dans la déclaration de l'espace de noms doit être utilisé lorsque le constructeur de type est appelé (par exemple, `xs:date`).
- Certaines fonctions XPath/XQuery ont le même nom que les types de base de données de schéma XML. Par exemple, pour les fonctions XPath `fn:string` et `fn:boolean`, il existe des types de données de schéma XML avec les mêmes noms locaux : `xs:string` et `xs:boolean`. Donc si vous deviez utiliser le string de l'expression XPath (`'Hello'`), l'expression évaluée comme `fn:string('Hello')` - et non comme `xs:string('Hello')`.

12.1.4 XQuery 1.0

Cette section :

- [Conformité du moteur](#) ⁵⁵³
- [Compatibilité du schéma \(Schema awareness\)](#) ⁵⁵³
- [Encodage](#) ⁵⁵³
- [Espaces de nom](#) ⁵⁵⁰
- [XML source et validation](#) ⁵⁵⁵
- [Contrôle de type statique et dynamique](#) ⁵⁵⁵
- [Modules bibliothèque](#) ⁵⁵⁵
- [Modules externes](#) ⁵⁵⁵
- [Collations](#) ⁵⁵⁶
- [Précision des données numériques](#) ⁵⁵⁶
- [Prise en charge des instructions XQuery](#) ⁵⁵⁶
- [Comportement spécifique à la mise en œuvre](#) ⁵⁵⁶

Conformité

Le moteur XQuery 1.0 de StyleVision 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 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 l'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.

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'éditio n 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](#)⁵⁵⁸. Pour utiliser une collation spécifique, fournir son URI tel que donné dans la [liste des collations prises en charge](#)⁵⁵⁸. 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.

Comportement spécifique à la mise en œuvre

CI-dessous, vous trouverez une description pour savoir comment les moteurs XQuery et XQuery Update 1.0 gèrent les aspects de certaines fonctions spécifiques à la mise en œuvre.

unparsed-text

L'argument `href` accepte (i) les chemins relatifs pour les fichiers dans le dossier base-uri et (ii) les chemins absolus avec ou sans `file://` protocol. Les encodages pris en charge de manière supplémentaire sont (spécifiques à Altova): `x-binarytobase16` et `x-binarytobase64`. Exemple : `xs:base64Binary(unparsed-text('chart.png', 'x-binarytobase64'))`.

unparsed-text-available

L'argument `href` accepte (i) les chemins relatifs pour les fichiers dans le dossier base-uri et (ii) les chemins absolus avec ou sans `file://` protocol. Les encodages pris en charge de manière supplémentaire sont (spécifiques à Altova): `x-binarytobase16` et `x-binarytobase64`.

Note : les valeurs d'encodage suivantes, qui sont mises en œuvre dans des versions antérieures du produit prédécesseur de RaptorXML, AltovaXML, sont dépréciées à présent : `base16tobinary`, `base64tobinary`, `binarytobase16` and `binarytobase64`.

12.1.5 XQuery 3.1

Le moteur XQuery 3.1 de StyleVision 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](#) ⁵⁵³.

12.2 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.

Cette section décrit les fonctions d'extension XPath/XQuery qui ont été créées par Altova pour fournir des opérations supplémentaires ainsi que d'[autres fonctions d'extension](#)⁶³⁵. [Ces fonctions d'extension](#)⁵⁵⁹ peuvent être calculées par les moteurs XSLT et XQuery d'Altova selon les règles décrites dans cette section. Pour information sur les fonctions XPath/XQuery régulières, voir la [Fonction de référence XPath/XQuery d'Altova](#).

Points généraux

Les points généraux suivants devraient être notés :

- Les fonctions des bibliothèques de fonction core définies dans les spécifications W3C peuvent être appelées sans préfixe. La raison étant que les moteurs XSLT et XQuery d'Altova lisent les fonctions sans préfixe comme appartenant à l'espace de nom <http://www.w3.org/2005/xpath-functions>, qui est l'espace de nom des fonctions par défaut spécifiées dans les spécifications des fonctions XPath/XQuery. 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](#) recensées ci-dessous ; pour en utiliser une, fournissez son URI tel qu'énoncé 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.

Langage	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 informations de l'espace de nom, avec des mécanismes XPath 2.0, utilisez les fonctions `in-scope-prefixes()`, `namespace-uri()` et `namespace-uri-for-prefix()`.

12.2.1 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 bibliothèque 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 présumé 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 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 bibliothèques de fonction XPath/XQuery et XSLT sont recensées dans un préfixe. Les fonctions d'extension provenant d'autres bibliothèques, 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

Usage de Fonctions d'extension Altova

Pour pouvoir utiliser les fonctions d'extension d'Altova, vous devez déclarer l'espace de nom des fonctions d'extension d'Altova (*d'abord mettre en surbrillance dans la liste de codes ci-dessous*) puis utiliser les fonctions d'extension pour qu'elles soient résolues comme appartenant à cet espace de noms (*voir deuxième mise en surbrillance*). L'exemple ci-dessous utilise la fonction d'extension d'Altova appelée **âge**.

```
<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:altova="http://www.altova.com/xslt-extensions">
  <xsl:output method="text" encoding="ISO-8859-1"/>
  <xsl:template match="Persons">
    <xsl:for-each select="Person">
      <xsl:value-of select="concat(Name, ': ')" />
      <xsl:value-of select="altova:age(xs:date(BirthDate))" />
      <xsl:value-of select="' ' years&#x0A;' " />
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Fonctions XSLT ⁵⁶¹

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). Les fonctions XBRL Altova peuvent uniquement être utilisées avec des éditions des produits Altova qui présentent 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](#) ⁵⁶⁴
- [Géolocalisation](#) ⁵⁸¹
- [Liée à l'image](#) ⁵⁹¹
- [Numérique](#) ⁵⁹⁵
- [Schéma](#) ⁵⁹⁸
- [Séquence](#) ⁶¹⁸
- [String](#) ⁶²⁶
- [Divers](#) ⁶³³

12.2.1.1 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 générales

▼ **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 contenus 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*`.

☐ 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](#) ⁵⁶¹]

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](#) ⁵⁶¹]

12.2.1.2 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 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 présumé 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](#) ⁵⁶⁵
- [Ajouter une durée à xs:date et retourner xs:date](#) ⁵⁶⁷
- [Ajouter une durée à xs:time et retourner à xs:time](#) ⁵⁶⁹
- [Formater et récupérer des durées](#) ⁵⁶⁸
- [Supprimer des fuseaux horaires de fonctions qui génèrent des date/heures actuels](#) ⁵⁷⁰
- [Retourne un jour de la semaine en tant qu'un entier à partir de la date](#) ⁵⁷¹
- [Retourne un jour de la semaine en tant qu'entier à partir de la date](#) ⁵⁷³
- [Retourne nombre de semaine en tant qu'entier à partir de la date](#) ⁵⁷³
- [Construire le type de date, d'heure ou de durée à partir des composants lexicaux de chaque type](#) ⁵⁷⁵
- [Construire le type de date, dateHeure ou heure à partir de l'entrée de chaîne](#) ⁵⁷⁶
- [Fonctions liées à l'âge](#) ⁵⁷⁸
- [Fonctions Epoch time \(heure Unix\)](#) ⁵⁷⁹

▼ Liste alphabétique

[altova:add-days-to-date](#) ⁵⁶⁷
[altova:add-days-to-dateTime](#) ⁵⁶⁵
[altova:add-hours-to-dateTime](#) ⁵⁶⁵
[altova:add-hours-to-time](#) ⁵⁶⁹
[altova:add-minutes-to-dateTime](#) ⁵⁶⁵
[altova:add-minutes-to-time](#) ⁵⁶⁹
[altova:add-months-to-date](#) ⁵⁶⁷
[altova:add-months-to-dateTime](#) ⁵⁶⁵
[altova:add-seconds-to-dateTime](#) ⁵⁶⁵
[altova:add-seconds-to-time](#) ⁵⁶⁹
[altova:add-years-to-date](#) ⁵⁶⁷
[altova:add-years-to-dateTime](#) ⁵⁶⁵
[altova:age](#) ⁵⁷⁸
[altova:age-details](#) ⁵⁷⁸
[altova:build-date](#) ⁵⁷⁵
[altova:build-duration](#) ⁵⁷⁵
[altova:build-time](#) ⁵⁷⁵
[altova:current-dateTime-no-TZ](#) ⁵⁷⁰
[altova:current-date-no-TZ](#) ⁵⁷⁰
[altova:current-time-no-TZ](#) ⁵⁷⁰
[altova:date-no-TZ](#) ⁵⁷⁰
[altova:dateTime-from-epoch](#) ⁵⁷⁹
[altova:dateTime-from-epoch-no-TZ](#) ⁵⁷⁹
[altova:dateTime-no-TZ](#) ⁵⁷⁰
[altova:days-in-month](#) ⁵⁷¹
[altova:epoch-from-dateTime](#) ⁵⁷⁹
[altova:hours-from-dateTimeDuration-accumulated](#) ⁵⁷¹
[altova:minutes-from-dateTimeDuration-accumulated](#) ⁵⁷¹
[altova:seconds-from-dateTimeDuration-accumulated](#) ⁵⁷¹
[altova:format-duration](#) ⁵⁶⁸
[altova:parse-date](#) ⁵⁷⁶
[altova:parse-dateTime](#) ⁵⁷⁶
[altova:parse-duration](#) ⁵⁶⁸
[altova:parse-time](#) ⁵⁷⁶
[altova:time-no-TZ](#) ⁵⁷⁰
[altova:weekday-from-date](#) ⁵⁷³
[altova:weekday-from-dateTime](#) ⁵⁷³
[altova:weeknumber-from-date](#) ⁵⁷⁴
[altova:weeknumber-from-dateTime](#) ⁵⁷⁴

[[Haut](#) ⁵⁶⁴]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) est optionnel.

▼ `add-years-to-dateTime` [`altova:`]

```
altova:add-years-to-dateTime (DateTime as xs:dateTime, Years as xs:integer)
as xs: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`.

▣ 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`

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](#) ⁵⁶⁴]

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

[[Haut](#) ⁵⁶⁴]

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) as xs:duration **XP3.1 XQ3.1**

Takes a patterned string as the first argument, and a picture string as the second argument. The input string is parsed on the basis of the picture string, and an xs:duration is returned.

▢ Exemples

- **altova:parse-duration**("Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"),

- ```
"Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") returns
"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]") returns "P3M2DT2H53M"

[ [Haut](#) <sup>564</sup> ]

## 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*) **asxs: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*) **asxs: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*) **asxs: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](#) <sup>564</sup> ]

## 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()` **asxs: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`:

- `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 valeur `xs: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) as xs: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) as xs:time **XP3.1 XQ3.1**

This function takes an xs:time argument, removes the timezone part from it, and returns an xs:time value. Note that the time is not modified.

☐ Exemples

- **altova:time-no-TZ**(xs:time("14:00:00+01:00")) returns 14:00:00

[ [Haut](#) <sup>564</sup> ]

## 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) as xs: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) as xs:integer **XP3.1 XQ3.1**

Returns the total number of hours in the duration submitted by the DayAndTime argument (which is of type xs:duration). The hours in the Day and Time components are added together to give a result that is an integer. A new hour is counted only for a full 60 minutes. Negative durations result in a negative hour value.

☐ Exemples

- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5D")) returns 120, which is the total number of hours in 5 days.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H")) returns 122, which is the total number of hours in 5 days plus 2 hours.

- `altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H60M"))` returns 123, which is the total number of hours in 5 days plus 2 hours and 60 mins.
- `altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H119M"))` returns 123, which is the total number of hours in 5 days plus 2 hours and 119 mins.
- `altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H120M"))` returns 124, which is the total number of hours in 5 days plus 2 hours and 120 mins.
- `altova:hours-from-dayTimeDuration-accumulated(xs:duration("-P5DT2H"))` returns -122

#### ▼ minutes-from-dayTimeDuration-accumulated

`altova:minutes-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) as xs:integer` [XP3.1](#) [XQ3.1](#)

Returns the total number of minutes in the duration submitted by the `DayAndTime` argument (which is of type `xs:duration`). The minutes in the `Day` and `Time` components are added together to give a result that is an integer. Negative durations result in a negative minute value.

##### ▢ Examples

- `altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT60M"))` returns 60
- `altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT1H"))` returns 60, which is the total number of minutes in 1 hour.
- `altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT1H40M"))` returns 100
- `altova:minutes-from-dayTimeDuration-accumulated(xs:duration("P1D"))` returns 1440, which is the total number of minutes in 1 day.
- `altova:minutes-from-dayTimeDuration-accumulated(xs:duration("-P1DT60M"))` returns -1500

#### ▼ seconds-from-dayTimeDuration-accumulated

`altova:seconds-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) as xs:integer` [XP3.1](#) [XQ3.1](#)

Returns the total number of seconds in the duration submitted by the `DayAndTime` argument (which is of type `xs:duration`). The seconds in the `Day` and `Time` components are added together to give a result that is an integer. Negative durations result in a negative seconds value.

##### ▢ Examples

- `altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1M"))` returns 60, which is the total number of seconds in 1 minute.
- `altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1H"))` returns 3600, which is the total number of seconds in 1 hour.
- `altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1H2M"))` returns 3720
- `altova:seconds-from-dayTimeDuration-accumulated(xs:duration("P1D"))` returns 86400, which is the total number of seconds in 1 day.
- `altova:seconds-from-dayTimeDuration-accumulated(xs:duration("-P1DT1M"))` returns -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](#) <sup>564</sup> ]

## Retourne le nombre de la semaine à partir de `xs:dateTime` ou `xs:date` XP2 XQ1 XP3.1 XQ3.1

Ces fonctions retournent le nombre de la semaine (en tant qu'entier) depuis `xs:dateTime` ou `xs:date`. La numérotation des semaines est disponible dans les formats de calendrier US, ISO/Européen 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).

### 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](#) <sup>564</sup> ]

## 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 premiers, 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 partie 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 fournit 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*., 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](#) <sup>564</sup> ]

## 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) asxs: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.

|          |       |
|----------|-------|
| <b>D</b> | Jour  |
| <b>M</b> | Mois  |
| <b>Y</b> | 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** spécifie 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.

|          |          |
|----------|----------|
| <b>D</b> | Date     |
| <b>M</b> | Mois     |
| <b>Y</b> | Année    |
| <b>H</b> | Heure    |
| <b>m</b> | Minutes  |
| <b>s</b> | 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.

|          |          |
|----------|----------|
| <b>H</b> | Heure    |
| <b>m</b> | minutes  |
| <b>s</b> | 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
- **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

[ [Haut](#) <sup>564</sup> ]

## 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)

[ [Haut](#) <sup>564</sup> ]

## 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 le 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`

#### ▼ epoch-from-dateTime [altova:]

`altova:epoch-from-dateTime`(dateTimeValue as `xs:dateTime`) as `xs:decimal` 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 `epoch-from-dateTime` 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. Veuillez noter que vous devrez explicitement construire la valeur `xs:dateTime`. La valeur soumise `xs:dateTime` peut ou ne peut pas contenir la partie

optionnelle `tz` (fuseau horaire).

Que la partie du fuseau horaire soit soumise en tant que partie de l'argument ou non, le décalage du fuseau horaire local (pris de l'horloge système) est retiré de l'argument soumis `dateTimeValue`. Ceci porduit l'heure UTC équivalente de laquelle l'Epoch time équivalent est calculé. 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), une heure sera retirée de la valeur soumise `dateTimeValue` avant de calculer la valeur Epoch. Consultez également la fonction `dateTime-from-epoch`.

 Exemples

Les exemples ci-dessous supposent un fuseau horaire local UTC +01:00. En conséquence, une heure sera soustraite à `dateTime` avant de calculer l'Epoch time.

- `altova:epoch-from-dateTime(xs:dateTime("1970-01-01T01:00:34+01:00"))` returns 34
- `altova:epoch-from-dateTime(xs:dateTime("1970-01-01T01:00:34"))` returns 34
- `altova:epoch-from-dateTime(xs:dateTime("2021-04-01T11:22:33"))` returns 1617272553

[ [Haut](#) <sup>564</sup> ]

### 12.2.1.3 Fonctions XPath/XQuery : Géolocalisation

Les fonctions d'extension de géolocalisation XPath/XQuery suivantes sont prises en charge dans la version actuelle de StyleVision 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 présumé 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) :     | <code>XP1</code> <code>XP2</code> <code>XP3.1</code>     |
| Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :      | <code>XSLT1</code> <code>XSLT2</code> <code>XSLT3</code> |
| Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) : | <code>XQ1</code> <code>XQ3.1</code>                      |

 `format-geolocation [altova:]`

`altova:format-geolocation(Latitude as xs:decimal, Longitude as xs:decimal,`

**GeolocationOutputStringFormat** as xs:integer) asxs: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](#)<sup>591</sup> et les attributs de métadonnées Exif peuvent être utilisés pour fournir les chaînes d'entrée.

#### Exemples

- **altova:format-geolocation**(33.33, -22.22, 4) retourne xs:string "33.33 -22.22"
- **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)<br>D°M'S.SS"N/S D°M'S.SS"E/W<br><b>Exemple</b> : 33°55'11.11"N 22°44'66.66"W                       |
| 2 | Degrés décimaux, avec orientation suffixée (N/S, E/O)<br>D.DDN/S D.DDE/W<br><b>Exemple</b> : 33.33N 22.22W                                                                   |
| 3 | Degrés, minutes, secondes décimales, avec signe préfixé (+/-); le signe plus (N/E) est optionnel<br>+/-D°M'S.SS" +/-D°M'S.SS"<br><b>Exemple</b> : 33°55'11.11" -22°44'66.66" |
| 4 | Degrés décimaux, avec signe préfixé (+/-); le signe plus (N/E) est optionnel<br>+/-D.DD +/-D.DD<br><b>Exemple</b> : 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 : GPSTLatitude, GPSTLatitudeRef, GPSTLongitude, GPSTLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

| GPSTLatitude | GPSTLatitudeRef | GPSTLongitude | GPSTLongitudeRef | Geolocation                  |
|--------------|-----------------|---------------|------------------|------------------------------|
|              | f               |               | f                |                              |
| 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) as xs:decimal+ **XP3.1 XQ3.1**  
Parses the supplied GeolocationInputString argument and returns the geolocation's latitude and longitude (in that order) as a sequence two xs:decimal items. The formats in which the geolocation input string can be supplied are listed below.

**Note:** The [image-exif-data](#)<sup>591</sup> function and the Exif metadata's [@Geolocation](#)<sup>591</sup> attribute can be used to supply the geolocation input string (see example below).

#### ▣ Examples

- **altova:parse-geolocation**("33.33 -22.22") returns the sequence of two xs:decimals (33.33, 22.22)
- **altova:parse-geolocation**("48°51'29.6"N 24°17'40.2"W") returns the sequence of two xs:decimals (48.858222222222, 24.2945)
- **altova:parse-geolocation**("48°51'29.6"N 24°17'40.2"W") returns the sequence of two xs:decimals (48.858222222222, 24.2945)
- **altova:parse-geolocation**( **image-exif-data**(//MyImages/Image20141130.01)/**@Geolocation** ) returns a sequence of two 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'`

*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 :

GPSPLatitude, GPSPLatitudeRef, GPSPLongitude, GPSPLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

| GPSPLatitude | GPSPLatitudeRef | GPSPLongitude | GPSPLongitudeRef | 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) asxs: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](#)<sup>591</sup> et l'attribut [@Geolocation](#)<sup>591</sup> 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"W")`  
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 : GPSTLatitude, GPSTLatitudeRef, GPSTLongitude, GPSTLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

| GPSTLatitude | GPSTLatitudeRef | GPSTLongitude | GPSTLongitudeRef | Geolocation                  |
|--------------|-----------------|---------------|------------------|------------------------------|
|              | f               |               | f                |                              |
| 33 51 21.91  | S               | 151 13 11.73  | E                | 33°51'21.91"S 151°13'11.73"E |

### ▼ **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](#)<sup>591</sup> et l'attribut [@Geolocation](#)<sup>591</sup> 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"W")  
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](#)<sup>591</sup> et l'attribut [@Geolocation](#)<sup>591</sup> 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"E"))` 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](#)<sup>591</sup> et l'attribut [@Geolocation](#)<sup>591</sup> 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"E") 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)  
 $D^{\circ}M'S.SS''N/S$   $D^{\circ}M'S.SS''W/E$   
*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  
 $+/-D^{\circ}M'S.SS''$   $+/-D^{\circ}M'S.SS''$   
*Exemple* : 33°55'11.11" -22°44'55.25"
- Degrés, minutes décimales, avec orientation suffixée (N/S, E/W)  
 $D^{\circ}M.MM'N/S$   $D^{\circ}M.MM'W/E$   
*Exemple* : 33°55.55'N 22°44.44'W
- Degrés, minutes décimales, avec signe préfixé (+/-) ; le signe plus pour (N/E) est optionnel  
 $+/-D^{\circ}M.MM'$   $+/-D^{\circ}M.MM'$   
*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 |

### 12.2.1.4 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 StyleVision 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 présumé 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) as string? XP3.1 XQ3.1

Takes the Base64 encoding of an image file as its argument and returns the file extension of the image as recorded in the Base64-encoding of the image. The returned value is a suggestion based on the image type information available in the encoding. If this information is not available, then an empty string is returned. This function is useful if you wish to save a Base64 image as a file and wish to dynamically retrieve an appropriate file extension.

#### ▢ Examples

- **altova:suggested-image-file-extension** (/MyImages/MobilePhone/Image20141130.01) returns 'jpg'
- **altova:suggested-image-file-extension** (\$XML1/Staff/Person/@photo) returns ''

In the examples above, the nodes supplied as the argument of the function are assumed to contain a Base64-encoded image. The first example retrieves `jpg` as the file's type and extension. In the second example, the submitted Base64 encoding does not provide usable file extension information.

#### ▼ image-exif-data [altova:]

**altova:image-exif-data**(Base64BinaryString as string) as element? XP3.1 XQ3.1

Takes a Base64-encoded JPEG image as its argument and returns an element called **Exif** that contains the Exif metadata of the image. The Exif metadata is created as attribute-value pairs of the **Exif** element. The attribute names are the Exif data tags found in the Base64 encoding. The list of Exif-specification tags

is given below. If a vendor-specific tag is present in the Exif data, this tag and its value will also be returned as an attribute-value pair. Additional to the standard Exif metadata tags (*see list below*), Altova-specific attribute-value pairs are also generated. These Altova Exif attributes are listed below.

#### ▣ Examples

- To access any one attribute, use the function like this:  
`image-exif-data (//MyImages/Image20141130.01) /@GPSLatitude`  
`image-exif-data (//MyImages/Image20141130.01) /@Geolocation`
- To access all the attributes, use the function like this:  
`image-exif-data (//MyImages/Image20141130.01) /@*`
- To access the names of all the attributes, use the following expression:  
`for $i in image-exif-data (//MyImages/Image20141130.01) /@* return name($i)`  
 This is useful to find out the names of the attributes returned by the function.

#### ▣ 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).

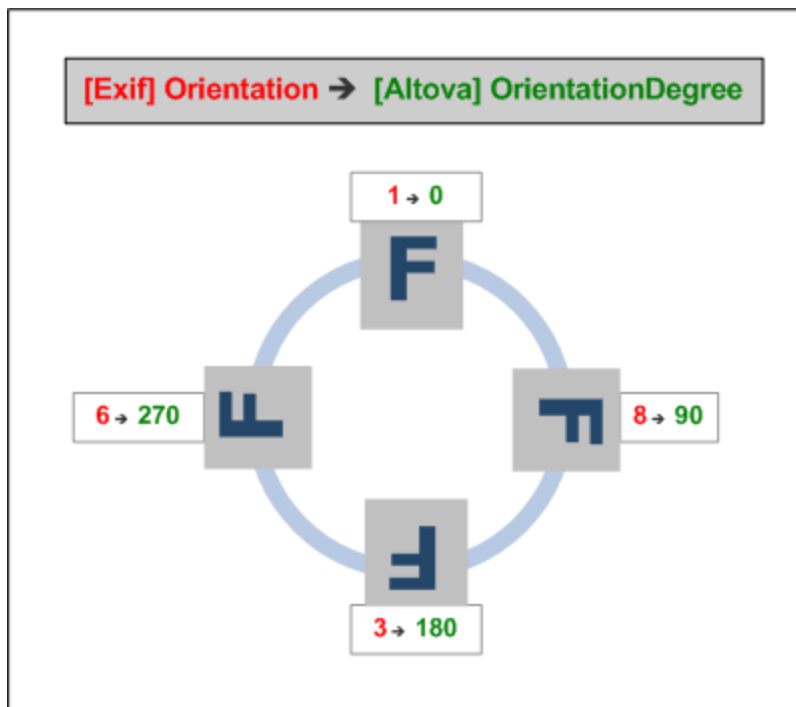
| GPSLatitude | GPSLatitudeRef | GPSLongitude | GPSLongitudeRef | Geolocation                  |
|-------------|----------------|--------------|-----------------|------------------------------|
| 33 51 21.91 | S              | 151 13 11.73 | E               | 33°51'21.91"S 151°13'11.73"E |

#### ▣ 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

- 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](#) <sup>591</sup> ]

### 12.2.1.5 Fonctions XPath/XQuery : Numérique

Les fonctions d'extension numériques d'Altova peuvent être utilisées dans les expressions XPath et XQuery et fournissent 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 présumé 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`.

[ [Haut](#) <sup>595</sup> ]

## 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) as xs:string XP3.1 XQ3.1

Takes an integer argument and returns its Base-16 equivalent as a string.

#### ▢ Exemples

- `altova:integer-to-hex-string(1)` returns '1'
- `altova:integer-to-hex-string(9)` returns '9'
- `altova:integer-to-hex-string(10)` returns 'A'
- `altova:integer-to-hex-string(11)` returns 'B'
- `altova:integer-to-hex-string(15)` returns 'F'
- `altova:integer-to-hex-string(16)` returns '10'
- `altova:integer-to-hex-string(32)` returns '20'
- `altova:integer-to-hex-string(33)` returns '21'
- `altova:integer-to-hex-string(90)` returns '5A'

[ [Haut](#) <sup>595</sup> ]

[ [Haut](#) <sup>595</sup> ]

### 12.2.1.6 Fonctions XPath/XQuery : Schéma

Les fonctions d'extension Altova recensées ci-dessous retournent l'information de 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 des expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple, dans XSLT avec l'instruction [xslt:import-schema](#). Dans des expressions XQuery, le schéma doit être importé explicitement utilisant un [schema import](#).

#### Information de schéma depuis les nœuds XML

La fonction `altova:type` soumet le nœud à un document XML et retourne 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 présumé 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) :     | <code>XP1</code> <code>XP2</code> <code>XP3.1</code>     |
| Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :      | <code>XSLT1</code> <code>XSLT2</code> <code>XSLT3</code> |
| Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) : | <code>XQ1</code> <code>XQ3.1</code>                      |

#### ▼ Schema (zero arguments)

`altova:schema() as (function(xs:string) as item()*)? XP3.1 XQ3.1`

Returns the `schema` component as a whole. You can navigate further into the `schema` component by selecting one of the `schema` component's properties.

- If this property is a component, you can navigate another step deeper by selecting one of this component's properties. This step can be repeated to navigate further into the schema.
- If the component is an atomic value, the atomic value is returned and you cannot navigate any deeper.

The properties of the `schema` component are:

```
"type definitions"
"attribute declarations"
"element declarations"
"attribute group definitions"
"model group definitions"
"notation declarations"
"identity-constraint definitions"
```

The properties of all other component kinds (besides `schema`) are listed below.

**Note:** In XQuery expressions, the schema must be explicitly imported. In XPath expressions, the schema must have been imported into the processing environment, for example, into XSLT with the `xslt:import` instruction.

#### Examples

- `import schema "" at "C:\Test\ExpReport.xsd"; for $typedef in altova:schema() ("type definitions") return $typedef ("name")` returns the names of all simple types or complex types in the schema
- `import schema "" at "C:\Test\ExpReport.xsd"; altova:schema() ("type definitions")[1] ("name")` returns the name of the first of all simple types or complex types in the schema

### Components and their properties

#### 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" |
| nom              | string            | Nom local de l'attribut |

|                      |                                                                                                                                                                                                                                                      |                                 |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| espace de nom cible  | string                                                                                                                                                                                                                                               | Espace de nom Uri de l'attribut |
| type definition      | Type simple ou Type complexe                                                                                                                                                                                                                         |                                 |
| étendue              | Une fonction avec des propriétés ("class": "Scope", "variety": "global" ou "local", "parent": le complexe Type contenant ou Groupe d'attribut)                                                                                                       |                                 |
| contrainte de valeur | Si présent, une fonction avec des propriétés ("class": "Value Constraint", "variety": "fixed" ou "default", "value": valeur atomique, "lexical form": string. Notez que la propriété "value" n'est pas disponible pour les types namespace-sensitive |                                 |
| peut être hérité     | booléen                                                                                                                                                                                                                                              |                                 |

#### ☐ Déclaration de groupe d'attribut

| Nom de propriété               | Type de propriété              | Valeur de propriété                    |
|--------------------------------|--------------------------------|----------------------------------------|
| kind                           | string                         | "Attribute Group Definition"           |
| nom                            | string                         | Nom local du groupe d'attribut         |
| espace de nom cible            | string                         | Espace de nom URI du groupe d'attribut |
| utilisations d'attribut        | Séquence de ('Attribute Use')  |                                        |
| caractère générique d'attribut | Caractère d'attribut optionnel |                                        |

#### ☐ Utilisation d'attribut

| Nom de propriété     | Type de propriété           | Valeur de propriété                             |
|----------------------|-----------------------------|-------------------------------------------------|
| kind                 | string                      | "Attribute Use"                                 |
| requis               | booléen                     | true si attribut est requis, false si optionnel |
| contrainte de valeur | Voir Déclaration d'attribut |                                                 |
| peut être hérité     | booléen                     |                                                 |

#### ☐ Caractère générique d'attribut

| Nom de propriété           | Type de propriété                                                                                                                     | Valeur de propriété |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| kind                       | string                                                                                                                                | "Wildcard"          |
| contrainte d'espace de nom | Fonction avec propriétés ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, |                     |



|                       |                                                                                             |  |
|-----------------------|---------------------------------------------------------------------------------------------|--|
|                       | "disallowed names": liste contenant QNames et/ou les strings "defined" et "definedSiblings" |  |
| contenus du processus | string ("strict" "lax" "skip")                                                              |  |

#### Type complexe

| Nom de propriété               | Type de propriété                                                                                                                                                                                                                                                                                                         | Valeur de propriété                            |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| kind                           | string                                                                                                                                                                                                                                                                                                                    | "Complex Type"                                 |
| nom                            | string                                                                                                                                                                                                                                                                                                                    | Nom local du type (vide si anonyme)            |
| espace de nom cible            | string                                                                                                                                                                                                                                                                                                                    | URI d'espace de noms du type (vide si anonyme) |
| définition de type de base     | Définition de type complexe                                                                                                                                                                                                                                                                                               |                                                |
| final                          | Séquence de strings ("restriction" "extension")                                                                                                                                                                                                                                                                           |                                                |
| contexte                       | Séquence vide (non implémentée)                                                                                                                                                                                                                                                                                           |                                                |
| méthode de dérivation          | string ("restriction" "extension")                                                                                                                                                                                                                                                                                        |                                                |
| abstrait                       | booléen                                                                                                                                                                                                                                                                                                                   |                                                |
| utilisations d'attribut        | Séquence de l'utilisation d'attribut                                                                                                                                                                                                                                                                                      |                                                |
| caractère générique d'attribut | Caractère d'attribut optionnel                                                                                                                                                                                                                                                                                            |                                                |
| type de contenu                | Fonction avec propriétés : ("class": "Content Type", "variety": string ("element-only" "empty" "mixed" "simple"), particle: particule optionnel, "open content": fonction avec propriétés ("class": "Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Type simple) |                                                |
| substitutions interdites       | Séquence de strings ("restriction" "extension")                                                                                                                                                                                                                                                                           |                                                |
| assertions                     | Séquence d'assertion                                                                                                                                                                                                                                                                                                      |                                                |

#### Déclaration d'élément

| Nom de propriété | Type de propriété | Valeur de propriété        |
|------------------|-------------------|----------------------------|
| kind             | string            | "Complex Type"             |
| nom              | string            | Nom local du type (vide si |

|                                   |                                                                                                                                                         |                                                |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
|                                   |                                                                                                                                                         | anonyme)                                       |
| espace de nom cible               | string                                                                                                                                                  | URI d'espace de noms du type (vide si anonyme) |
| définition type                   | Type simple ou Type complexe                                                                                                                            |                                                |
| type table                        | fonction avec propriétés ("class": "Type Table", "alternatives": séquence de Type Alternative, "default type definition": Type simple ou Type complexe) |                                                |
| étendue                           | fonction avec propriétés ("class": "Scope", "variety": ("global" "local"), "parent": complexe Type optionnel)                                           |                                                |
| contrainte de valeur              | voir Déclaration d'attribut                                                                                                                             |                                                |
| nillable                          | booléen                                                                                                                                                 |                                                |
| définitions identity-constraint   | Séquence de contrainte d'identité                                                                                                                       |                                                |
| substitution groupe affiliations  | Séquence de Déclaration d'élément                                                                                                                       |                                                |
| substitution exclusions de groupe | Séquence de strings ("restriction" "extension")                                                                                                         |                                                |
| substitutions refusées            | Séquence de strings ("restriction" "extension" "substitution")                                                                                          |                                                |
| abstrait                          | booléen                                                                                                                                                 |                                                |

☐ Caractère générique d'élément

| Nom de propriété           | Type de propriété                                                                                                                                                                                                                 | Valeur de propriété |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| kind                       | string                                                                                                                                                                                                                            | "Wildcard"          |
| contrainte d'espace de nom | Fonction avec propriétés ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": liste contenant QNames et/ou les strings "defined" et "definedSiblings" |                     |
| contenus du processus      | string ("strict" "lax" "skip")                                                                                                                                                                                                    |                     |

☐ Facette

| Nom de propriété | Type de propriété | Valeur de propriété                                            |
|------------------|-------------------|----------------------------------------------------------------|
| kind             | string            | Le nom de la facette, par exemple "minLength" ou "enumeration" |

| valeur      | dépend de la facette                                               | La valeur de la facette                                                                                                                                                                                                                     |
|-------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fixé        | booléen                                                            |                                                                                                                                                                                                                                             |
| typed-value | Pour la facette d'énumération uniquement, array(xs:anyAtomicType*) | Une array contenant les valeurs d'énumération, dont chacune peut être en général une séquence de valeurs atomiques. (Note : pour la facette d'énumération, la propriété "value" est une séquence de strings, indépendamment du type actuel) |

#### ☐ Contrainte d'identité

| Nom de propriété              | Type de propriété                               | Valeur de propriété                  |
|-------------------------------|-------------------------------------------------|--------------------------------------|
| kind                          | string                                          | "Identity-Constraint Definition"     |
| nom                           | string                                          | Nom local de la contrainte           |
| espace de nom cible           | string                                          | URI d'espace de nom de la contrainte |
| catégorie identity-constraint | string ("key" "unique" "keyRef")                |                                      |
| selector                      | XPath Property Record                           |                                      |
| champs                        | Séquence de XPath Property Record               |                                      |
| Clé référencée                | (Pour keyRef uniquement): Contrainte d'identité | La contrainte de clé correspondante  |

#### ☐ Groupe de modèle

| Nom de propriété | Type de propriété                  | Valeur de propriété |
|------------------|------------------------------------|---------------------|
| kind             | string                             | "Model Group"       |
| compositor       | string ("sequence" "choice" "all") |                     |
| particules       | Séquence de particule              |                     |

#### ☐ Définition de groupe de modèle

| Nom de propriété    | Type de propriété | Valeur de propriété              |
|---------------------|-------------------|----------------------------------|
| kind                | string            | "Model Group Definition"         |
| nom                 | string            | Nom local du groupe de modèle    |
| espace de nom cible | 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 | "Notation Declaration"             |
| nom                    | string | Nom local de la notation           |
| espace de nom cible    | string | URI d'espace de nom de la notation |
| identifiant de système | anyURI |                                    |
| identifiant public     | string |                                    |

▣ Particule

| Nom de propriété | Type de propriété                                                  | Valeur de propriété |
|------------------|--------------------------------------------------------------------|---------------------|
| kind             | string                                                             | "Particule"         |
| min occurs       | integer                                                            |                     |
| 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"                    |
| nom                         | string                                                       | Nom local du type (vide si anonyme)            |
| espace de nom cible         | string                                                       | URI d'espace de noms du type (vide si anonyme) |
| final                       | Séquence de string("restriction" "extension" "list" "union") |                                                |
| contexte                    | composant contenant                                          |                                                |
| définition de type de base  | Type simple                                                  |                                                |
| facettes                    | Séquence de Facette                                          |                                                |
| facettes fondamentales      | Séquence vide (non implémentée)                              |                                                |
| variety                     | string ("atomic" "list" "union")                             |                                                |
| définition de type primitif | Type simple                                                  |                                                |
| définition de type item     | (uniquement pour les types de liste) Type simple             |                                                |
| définitions de type membre  | (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        |                     |
| définition type  | Type simple ou Type complexe |                     |

#### XPath Property Record

| Nom de propriété          | Type de propriété                                                                  | Valeur de propriété                          |
|---------------------------|------------------------------------------------------------------------------------|----------------------------------------------|
| Liaisons d'espace de noms | Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI) |                                              |
| espace de nom par défaut  | anyURI                                                                             |                                              |
| base URI                  | anyURI                                                                             | L'URI de base statique de l'expression XPath |
| expression                | string                                                                             | L'expression XPath en tant que string        |

#### Schema (two arguments)

`altova:schema(ComponentKind as xs:string, Name as xs:QName) as (function(xs:string) as item(*)?)? XP3.1 XQ3.1`

Returns the component kind that is specified in the first argument which has a name that is the same as the name supplied in the second argument. You can navigate further by selecting one of the component's properties.

- If this property is a component, you can navigate another step deeper by selecting one of this component's properties. This step can be repeated to navigate further into the schema.
- If the component is an atomic value, the atomic value is returned and you cannot navigate any deeper.

**Note:** In XQuery expressions, the schema must be explicitly imported. In XPath expressions, the schema must have been imported into the processing environment, for example, into XSLT with the `xslt:import` instruction.

#### Examples

- `import schema "" at "C:\Test\ExpReport.xsd";`  
`altova:schema("element declaration", xs:QName("OrgChart"))("type definition")`  
`("content type")("particles")[3]!.("term")("kind")`  
returns the kind property of the term of the third particles component. This particles component is a descendant of the element declaration having a QName of 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")]
returns, for each facet of each emailType component, an array containing that facet's kind and
value

```

### Components and their properties

#### ▣ 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"         |
| nom                  | string                                                                                                                                                                                                                                               | Nom local de l'attribut         |
| espace de nom cible  | string                                                                                                                                                                                                                                               | Espace de nom Uri de l'attribut |
| type definition      | Type simple ou Type complexe                                                                                                                                                                                                                         |                                 |
| étendue              | Une fonction avec des propriétés ("class": "Scope", "variety": "global" ou "local", "parent": le complexe Type contenant ou Groupe d'attribut)                                                                                                       |                                 |
| contrainte de valeur | Si présent, une fonction avec des propriétés ("class": "Value Constraint", "variety": "fixed" ou "default", "value": valeur atomique, "lexical form": string. Notez que la propriété "value" n'est pas disponible pour les types namespace-sensitive |                                 |
| peut être hérité     | booléen                                                                                                                                                                                                                                              |                                 |

#### ▣ Déclaration de groupe d'attribut

| Nom de propriété        | Type de propriété              | Valeur de propriété                    |
|-------------------------|--------------------------------|----------------------------------------|
| kind                    | string                         | "Attribute Group Definition"           |
| nom                     | string                         | Nom local du groupe d'attribut         |
| espace de nom cible     | string                         | Espace de nom URI du groupe d'attribut |
| utilisations d'attribut | Séquence de ('Attribute Use')  |                                        |
| caractère générique     | Caractère d'attribut optionnel |                                        |

|            |  |  |
|------------|--|--|
| d'attribut |  |  |
|------------|--|--|

#### Utilisation d'attribut

| Nom de propriété     | Type de propriété           | Valeur de propriété                             |
|----------------------|-----------------------------|-------------------------------------------------|
| kind                 | string                      | "Attribute Use"                                 |
| requis               | booléen                     | true si attribut est requis, false si optionnel |
| contrainte de valeur | Voir Déclaration d'attribut |                                                 |
| peut être hérité     | booléen                     |                                                 |

#### Caractère générique d'attribut

| Nom de propriété           | Type de propriété                                                                                                                                                                                                                 | Valeur de propriété |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| kind                       | string                                                                                                                                                                                                                            | "Wildcard"          |
| contrainte d'espace de nom | Fonction avec propriétés ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": liste contenant QNames et/ou les strings "defined" et "definedSiblings" |                     |
| contenus du processus      | string ("strict" "lax" "skip")                                                                                                                                                                                                    |                     |

#### Type complexe

| Nom de propriété               | Type de propriété                               | Valeur de propriété                            |
|--------------------------------|-------------------------------------------------|------------------------------------------------|
| kind                           | string                                          | "Complex Type"                                 |
| nom                            | string                                          | Nom local du type (vide si anonyme)            |
| espace de nom cible            | string                                          | URI d'espace de noms du type (vide si anonyme) |
| définition de type de base     | Définition de type complexe                     |                                                |
| final                          | Séquence de strings ("restriction" "extension") |                                                |
| contexte                       | Séquence vide (non implémentée)                 |                                                |
| méthode de dérivation          | string ("restriction" "extension")              |                                                |
| abstrait                       | booléen                                         |                                                |
| utilisations d'attribut        | Séquence de l'utilisation d'attribut            |                                                |
| caractère générique d'attribut | Caractère d'attribut optionnel                  |                                                |

|                             |                                                                                                                                                                                                                                                                                                                                                       |  |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| type de contenu             | Fonction avec propriétés :<br>("class": "Content Type", "variety": string<br>("element-<br>only" "empty" "mixed" "simple"), particle:<br>particule optionnel, "open content":<br>fonction avec propriétés ("class": "Open<br>Content", "mode": string<br>("interleave" "suffix"), "wildcard":<br>Wildcard), "simple type definition": Type<br>simple) |  |
| substitutions<br>interdites | Séquence de strings<br>("restriction" "extension")                                                                                                                                                                                                                                                                                                    |  |
| assertions                  | Séquence d'assertion                                                                                                                                                                                                                                                                                                                                  |  |

#### Déclaration d'élément

| Nom de propriété                     | Type de propriété                                                                                                                                                | Valeur de propriété                            |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| kind                                 | string                                                                                                                                                           | "Complex Type"                                 |
| nom                                  | string                                                                                                                                                           | Nom local du type (vide si anonyme)            |
| espace de nom cible                  | string                                                                                                                                                           | URI d'espace de noms du type (vide si anonyme) |
| définition type                      | Type simple ou Type complexe                                                                                                                                     |                                                |
| type table                           | fonction avec propriétés ("class": "Type<br>Table", "alternatives": séquence de Type<br>Alternative, "default type definition": Type<br>simple ou Type complexe) |                                                |
| étendue                              | fonction avec propriétés<br>("class": "Scope", "variety":<br>("global" "local"), "parent": complexe<br>Type optionnel)                                           |                                                |
| contrainte de valeur                 | voir Déclaration d'attribut                                                                                                                                      |                                                |
| nilable                              | booléen                                                                                                                                                          |                                                |
| définitions identity-<br>constraint  | Séquence de contrainte d'identité                                                                                                                                |                                                |
| substitution groupe<br>affiliations  | Séquence de Déclaration d'élément                                                                                                                                |                                                |
| substitution<br>exclusions de groupe | Séquence de strings<br>("restriction" "extension")                                                                                                               |                                                |
| substitutions refusées               | Séquence de strings<br>("restriction" "extension" "substitution")                                                                                                |                                                |
| abstrait                             | booléen                                                                                                                                                          |                                                |

#### Caractère générique d'élément



| Nom de propriété           | Type de propriété                                                                                                                                                                                                                 | Valeur de propriété |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| kind                       | string                                                                                                                                                                                                                            | "Wildcard"          |
| contrainte d'espace de nom | Fonction avec propriétés ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": liste contenant QNames et/ou les strings "defined" et "definedSiblings" |                     |
| contenus du processus      | string ("strict" "lax" "skip")                                                                                                                                                                                                    |                     |

#### Facette

| Nom de propriété | Type de propriété                                                  | Valeur de propriété                                                                                                                                                                                                                         |
|------------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| kind             | string                                                             | Le nom de la facette, par exemple "minLength" ou "enumeration"                                                                                                                                                                              |
| valeur           | dépend de la facette                                               | La valeur de la facette                                                                                                                                                                                                                     |
| fixé             | booléen                                                            |                                                                                                                                                                                                                                             |
| typed-value      | Pour la facette d'énumération uniquement, array(xs:anyAtomicType*) | Une array contenant les valeurs d'énumération, dont chacune peut être en général une séquence de valeurs atomiques. (Note : pour la facette d'énumération, la propriété "value" est une séquence de strings, indépendamment du type actuel) |

#### Contrainte d'identité

| Nom de propriété              | Type de propriété                               | Valeur de propriété                  |
|-------------------------------|-------------------------------------------------|--------------------------------------|
| kind                          | string                                          | "Identity-Constraint Definition"     |
| nom                           | string                                          | Nom local de la contrainte           |
| espace de nom cible           | string                                          | URI d'espace de nom de la contrainte |
| catégorie identity-constraint | string ("key" "unique" "keyRef")                |                                      |
| selector                      | XPath Property Record                           |                                      |
| champs                        | Séquence de XPath Property Record               |                                      |
| Clé référencée                | (Pour keyRef uniquement): Contrainte d'identité | La contrainte de clé correspondante  |

#### Groupe de modèle

| Nom de propriété | Type de propriété                  | Valeur de propriété |
|------------------|------------------------------------|---------------------|
| kind             | string                             | "Model Group"       |
| compositor       | string ("sequence" "choice" "all") |                     |
| particules       | Séquence de particule              |                     |

☐ Définition de groupe de modèle

| Nom de propriété    | Type de propriété | Valeur de propriété              |
|---------------------|-------------------|----------------------------------|
| kind                | string            | "Model Group Definition"         |
| nom                 | string            | Nom local du groupe de modèle    |
| espace de nom cible | 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            | "Notation Declaration"             |
| nom                    | string            | Nom local de la notation           |
| espace de nom cible    | string            | URI d'espace de nom de la notation |
| identifiant de système | anyURI            |                                    |
| identifiant public     | string            |                                    |

☐ Particule

| Nom de propriété | Type de propriété                                                  | Valeur de propriété |
|------------------|--------------------------------------------------------------------|---------------------|
| kind             | string                                                             | "Particule"         |
| min occurs       | integer                                                            |                     |
| 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"                    |
| nom                 | string            | Nom local du type (vide si anonyme)            |
| espace de nom cible | string            | URI d'espace de noms du type (vide si anonyme) |

|                             |                                                              |  |
|-----------------------------|--------------------------------------------------------------|--|
| final                       | Séquence de string("restriction" "extension" "list" "union") |  |
| contexte                    | composant contenant                                          |  |
| définition de type de base  | Type simple                                                  |  |
| facettes                    | Séquence de Facette                                          |  |
| facettes fondamentales      | Séquence vide (non implémentée)                              |  |
| variety                     | string ("atomic" "list" "union")                             |  |
| définition de type primitif | Type simple                                                  |  |
| définition de type item     | (uniquement pour les types de liste) Type simple             |  |
| définitions de type membre  | (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        |                     |
| définition type  | Type simple ou Type complexe |                     |

#### XPath Property Record

| Nom de propriété          | Type de propriété                                                                  | Valeur de propriété                          |
|---------------------------|------------------------------------------------------------------------------------|----------------------------------------------|
| Liaisons d'espace de noms | Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI) |                                              |
| espace de nom par défaut  | 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`

The function `altova:type` submits an element or attribute node of an XML document and returns the node's type information from the PSVI.

**Note:** The XML document must have a schema declaration so that the schema can be referenced.

#### Examples

- ```

for $element in //Email
let $type := altova:type($element)
return $type

```

returns a function that contains the `Email` node's type information
- ```

for $element in //Email
let $type := altova:type($element)
return $type ("kind")

```

takes the `Email` node's type component (Simple Type or Complex Type) and returns the value of the component's `kind` property

The `"_props"` parameter returns the properties of the selected component. For example:

- ```

for $element in //Email
let $type := altova:type($element)
return ($type ("kind"), $type ("_props"))

```

takes the `Email` node's type component (Simple Type or Complex Type) and returns (i) the value of the component's `kind` property, and then (ii) the properties of that component.

Components and their properties

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"
nom	string	Nom local de l'attribut
espace de nom cible	string	Espace de nom Uri de l'attribut
type definition	Type simple ou Type complexe	
étendue	Une fonction avec des propriétés ("class": "Scope", "variety": "global" ou "local", "parent": le complexe Type contenant ou Groupe d'attribut)	
contrainte de valeur	Si présent, une fonction avec des propriétés ("class": "Value Constraint", "variety": "fixed" ou "default", "value": valeur atomique, "lexical form": string. Notez que la propriété "value" n'est pas	

	disponible pour les types namespace-sensitive	
peut être hérité	booléen	

☐ Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
nom	string	Nom local du groupe d'attribut
espace de nom cible	string	Espace de nom URI du groupe d'attribut
utilisations d'attribut	Séquence de ('Attribute Use')	
caractère générique d'attribut	Caractère d'attribut optionnel	

☐ Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
requis	booléen	true si attribut est requis, false si optionnel
contrainte de valeur	Voir Déclaration d'attribut	
peut être hérité	booléen	

☐ Caractère générique d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
contrainte d'espace de nom	Fonction avec propriétés ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": liste contenant QNames et/ou les strings "defined" et "definedSiblings"	
contenus du processus	string ("strict" "lax" "skip")	

☐ Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Complex Type"
nom	string	Nom local du type (vide si

		anonyme)
espace de nom cible	string	URI d'espace de noms du type (vide si anonyme)
définition de type de base	Définition de type complexe	
final	Séquence de strings ("restriction" "extension")	
contexte	Séquence vide (non implémentée)	
méthode de dérivation	string ("restriction" "extension")	
abstrait	booléen	
utilisations d'attribut	Séquence de l'utilisation d'attribut	
caractère générique d'attribut	Caractère d'attribut optionnel	
type de contenu	Fonction avec propriétés : ("class": "Content Type", "variety": string ("element-only" "empty" "mixed" "simple"), particle: particule optionnel, "open content": fonction avec propriétés ("class": "Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
substitutions interdites	Séquence de strings ("restriction" "extension")	
assertions	Séquence d'assertion	

■ Déclaration d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Complex Type"
nom	string	Nom local du type (vide si anonyme)
espace de nom cible	string	URI d'espace de noms du type (vide si anonyme)
définition type	Type simple ou Type complexe	
type table	fonction avec propriétés ("class": "Type Table", "alternatives": séquence de Type Alternative, "default type definition": Type simple ou Type complexe)	
étendue	fonction avec propriétés ("class": "Scope", "variety": ("global" "local"), "parent": complexe)	

	Type optionnel)	
contrainte de valeur	voir Déclaration d'attribut	
nillable	booléen	
définitions identity-constraint	Séquence de contrainte d'identité	
substitution groupe affiliations	Séquence de Déclaration d'élément	
substitution exclusions de groupe	Séquence de strings ("restriction" "extension")	
substitutions refusées	Séquence de strings ("restriction" "extension" "substitution")	
abstrait	booléen	

☐ Caractère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
contrainte d'espace de nom	Fonction avec propriétés ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": liste contenant QNames et/ou les strings "defined" et "definedSiblings"	
contenus du processus	string ("strict" "lax" "skip")	

☐ Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	Le nom de la facette, par exemple "minLength" ou "enumeration"
valeur	dépend de la facette	La valeur de la facette
fixé	booléen	
typed-value	Pour la facette d'énumération uniquement, array(xs:anyAtomicType*)	Une array contenant les valeurs d'énumération, dont chacune peut être en général une séquence de valeurs atomiques. (Note : pour la facette d'énumération, la propriété "value" est une séquence de strings, indépendamment du type actuel)

☐ Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
nom	string	Nom local de la contrainte
espace de nom cible	string	URI d'espace de nom de la contrainte
catégorie identity-constraint	string ("key" "unique" "keyRef")	
selector	XPath Property Record	
champs	Séquence de XPath Property Record	
Clé référencée	(Pour keyRef uniquement): Contrainte d'identité	La contrainte de clé correspondante

☐ Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Model Group"
compositor	string ("sequence" "choice" "all")	
particules	Séquence de particule	

☐ Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Model Group Definition"
nom	string	Nom local du groupe de modèle
espace de nom cible	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	"Notation Declaration"
nom	string	Nom local de la notation
espace de nom cible	string	URI d'espace de nom de la notation
identifiant de système	anyURI	
identifiant public	string	

☐ Particule

Nom de propriété	Type de propriété	Valeur de propriété
------------------	-------------------	---------------------

kind	string	"Particule"
min occurs	integer	
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"
nom	string	Nom local du type (vide si anonyme)
espace de nom cible	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction" "extension" "list" "union")	
contexte	composant contenant	
définition de type de base	Type simple	
facettes	Séquence de Facette	
facettes fondamentales	Séquence vide (non implémentée)	
variety	string ("atomic" "list" "union")	
définition de type primitif	Type simple	
définition de type item	(uniquement pour les types de liste) Type simple	
définitions de type membre	(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	
définition type	Type simple ou Type complexe	

■ XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
Liaisons d'espace	Séquence des fonctions avec les	

de noms	propriétés ("prefix": string, "namespace": anyURI)	
espace de nom par défaut	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

12.2.1.7 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 présumé 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.

<i>Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :</i>	XP1 XP2 XP3.1
<i>Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :</i>	XSLT1 XSLT2 XSLT3
<i>Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :</i>	XQ1 XQ3.1

▼ attributs [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`

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-available()` 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)) as item())* 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 `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))) **as item()*** **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).

▼ 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))) **as xs: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-available()` 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()) as item()* 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, 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 (jointes et disjointes), 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()) as item()*

XP3.1 XQ3.1

If FirstSequence is empty, returns SecondSequence. If FirstSequence is not empty, returns FirstSequence.

☐ Examples

- **altova:substitute-empty**((1,2,3), (4,5,6)) returns (1,2,3)
- **altova:substitute-empty**((), (4,5,6)) returns (4,5,6)

12.2.1.8 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 présumé 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(AsString 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 `AsString`. 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(AsString as xs:string) asxs:string` **XP2 XQ1 XP3.1 XQ3.1**

`altova:create-hash-from-string(AsString as xs:string, HashAlgo as xs:string)`

`asxs:string` **XP2 XQ1 XP3.1 XQ3.1**

Génère un string de hashage depuis `AsString` 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:]

`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*) as *xs: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()**) as *xs: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*) as *xs:string* **XP3.1 XQ3.1**

Returns a string containing the last *X-Number* of characters of the string obtained by converting the value of the context item to *xs:string*.

▢ Exemples

If the context item is 1234ABCD:

- **altova:last-chars**(2) returns CD
- **altova:last-chars**(5) returns 4ABCD
- **altova:last-chars**(9) returns 1234ABCD

altova:last-chars(*InputString* as *xs:string*, *X-Number* as *xs:integer*) as *xs:string* **XP3.1 XQ3.1**

Returns a string containing the last *X-Number* of characters of the string submitted as the *InputString* argument.

▢ Exemples

- **altova:last-chars**("2014-01-15", 5) returns 01-15

- `altova:last-chars("USA", 10)` returns `USA`

▼ `pad-string-left` [altova:]

`altova:pad-string-left(StringToPad as xs:string, StringLength as xs:integer, PadCharacter as xs:string) as xs:string` **XP3.1 XQ3.1**

The `PadCharacter` argument is a single character. It is padded to the left of the string to increase the number of characters in `StringToPad` so that this number equals the integer value of the `StringLength` argument. The `StringLength` argument can have any integer value (positive or negative), but padding will occur only if the value of `StringLength` is greater than the number of characters in `StringToPad`. If `StringToPad` has more characters than the value of `StringLength`, then `StringToPad` is left unchanged.

▢ Examples

- `altova:pad-string-left('AP', 1, 'Z')` returns `'AP'`
- `altova:pad-string-left('AP', 2, 'Z')` returns `'AP'`
- `altova:pad-string-left('AP', 3, 'Z')` returns `'ZAP'`
- `altova:pad-string-left('AP', 4, 'Z')` returns `'ZZAP'`
- `altova:pad-string-left('AP', -3, 'Z')` returns `'AP'`
- `altova:pad-string-left('AP', 3, 'YZ')` returns a `pad-character-too-long` error

▼ `pad-string-right` [altova:]

`altova:pad-string-right(StringToPad as xs:string, StringLength as xs:integer, PadCharacter as xs:string) as xs:string` **XP3.1 XQ3.1**

The `PadCharacter` argument is a single character. It is padded to the right of the string to increase the number of characters in `StringToPad` so that this number equals the integer value of the `StringLength` argument. The `StringLength` argument can have any integer value (positive or negative), but padding will occur only if the value of `StringLength` is greater than the number of characters in `StringToPad`. If `StringToPad` has more characters than the value of `StringLength`, then `StringToPad` is left unchanged.

▢ Examples

- `altova:pad-string-right('AP', 1, 'Z')` returns `'AP'`
- `altova:pad-string-right('AP', 2, 'Z')` returns `'AP'`
- `altova:pad-string-right('AP', 3, 'Z')` returns `'APZ'`
- `altova:pad-string-right('AP', 4, 'Z')` returns `'APZZ'`
- `altova:pad-string-right('AP', -3, 'Z')` returns `'AP'`
- `altova:pad-string-right('AP', 3, 'YZ')` returns a `pad-character-too-long` error

▼ `repeat-string` [altova:]

`altova:repeat-string(InputString as xs:string, Repeats as xs:integer) as xs:string` **XP2 XQ1 XP3.1 XQ3.1**

Génère une chaîne qui est composée du premier argument `InputString` répété `Repeats` nombre de fois.

▢ Examples

- `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) as xs:string **XP3.1 XQ3.1**

If CheckString is found in MainString, then the substring that occurs after CheckString in MainString is returned. If CheckString is not found in MainString, then the empty string is returned. If CheckString is an empty string, then MainString is returned in its entirety. If there is more than one occurrence of CheckString in MainString, then the substring after the last occurrence of CheckString is returned.

▢ Examples

- **altova:substring-after-last**('ABCDEFGH', 'B') returns 'DEFGH'
- **altova:substring-after-last**('ABCDEFGH', 'BC') returns 'DEFGH'
- **altova:substring-after-last**('ABCDEFGH', 'BD') returns ''
- **altova:substring-after-last**('ABCDEFGH', 'Z') returns ''
- **altova:substring-after-last**('ABCDEFGH', '') returns 'ABCDEFGH'
- **altova:substring-after-last**('ABCD-ABCD', 'B') returns 'CD'
- **altova:substring-after-last**('ABCD-ABCD-ABCD', 'BCD') returns ''

▼ substring-before-last [altova:]

altova:substring-before-last(MainString as xs:string, CheckString as xs:string) as xs:string **XP3.1 XQ3.1**

If CheckString is found in MainString, then the substring that occurs before CheckString in MainString is returned. If CheckString is not found in MainString, or if CheckString is an empty string, then the empty string is returned. If there is more than one occurrence of CheckString in MainString, then the substring before the last occurrence of CheckString is returned.

▢ Examples

- **altova:substring-before-last**('ABCDEFGH', 'B') returns 'A'
- **altova:substring-before-last**('ABCDEFGH', 'BC') returns 'A'
- **altova:substring-before-last**('ABCDEFGH', 'BD') returns ''
- **altova:substring-before-last**('ABCDEFGH', 'Z') returns ''
- **altova:substring-before-last**('ABCDEFGH', '') returns ''
- **altova:substring-before-last**('ABCD-ABCD', 'B') returns 'ABCD-A'
- **altova:substring-before-last**('ABCD-ABCD-ABCD', 'ABCD') returns 'ABCD-ABCD-'

▼ substring-pos [altova:]

altova:substring-pos(StringToCheck as xs:string, StringToFind as xs:string) as xs:integer **XP3.1 XQ3.1**

Returns the character position of the first occurrence of StringToFind in the string StringToCheck. The character position is returned as an integer. The first character of StringToCheck has the position 1. If StringToFind does not occur within StringToCheck, the integer 0 is returned. To check for the second or a later occurrence of StringToCheck, use the next signature of this function.

▢ Examples

- **altova:substring-pos**('Altova', 'to') returns 3
- **altova:substring-pos**('Altova', 'tov') returns 3
- **altova:substring-pos**('Altova', 'tv') returns 0
- **altova:substring-pos**('AltovaAltova', 'to') returns 3

altova:substring-pos(StringToCheck as xs:string, StringToFind as xs:string, Integer as

`xs:integer) as xs:integer XP3.1 XQ3.1`

Returns the character position of `StringToFind` in the string, `StringToCheck`. The search for `StringToFind` starts from the character position given by the `Integer` argument; the character substring before this position is not searched. The returned integer, however, is the position of the found string within the *entire* string, `StringToCheck`. This signature is useful for finding the second or a later position of a string that occurs multiple times with the `StringToCheck`. If `StringToFind` does not occur within `StringToCheck`, the integer 0 is returned.

▣ Examples

- `altova:substring-pos('Altova', 'to', 1)` returns 3
- `altova:substring-pos('Altova', 'to', 3)` returns 3
- `altova:substring-pos('Altova', 'to', 4)` returns 0
- `altova:substring-pos('Altova-Altova', 'to', 0)` returns 3
- `altova:substring-pos('Altova-Altova', 'to', 4)` returns 10

▼ `trim-string [altova:]`

`altova:trim-string(InputString as xs:string) as xs:string XP3.1 XQ3.1`

This function takes an `xs:string` argument, removes any leading and trailing whitespace, and returns a "trimmed" `xs:string`.

▣ Examples

- `altova:trim-string(" Hello World ")` returns "Hello World"
- `altova:trim-string("Hello World ")` returns "Hello World"
- `altova:trim-string(" Hello World")` returns "Hello World"
- `altova:trim-string("Hello World")` returns "Hello World"
- `altova:trim-string("Hello World")` returns "Hello World"

▼ `trim-string-left [altova:]`

`altova:trim-string-left(InputString as xs:string) as xs:string XP3.1 XQ3.1`

This function takes an `xs:string` argument, removes any leading whitespace, and returns a left-trimmed `xs:string`.

▣ Examples

- `altova:trim-string-left(" Hello World ")` returns "Hello World "
- `altova:trim-string-left("Hello World ")` returns "Hello World "
- `altova:trim-string-left(" Hello World")` returns "Hello World"
- `altova:trim-string-left("Hello World")` returns "Hello World"
- `altova:trim-string-left("Hello World")` returns "Hello World"

▼ `trim-string-right [altova:]`

`altova:trim-string-right(InputString as xs:string) as xs:string XP3.1 XQ3.1`

This function takes an `xs:string` argument, removes any trailing whitespace, and returns a right-trimmed `xs:string`.

▣ Examples

- `altova:trim-string-right(" Hello World ")` returns " Hello World"
- `altova:trim-string-right("Hello World ")` returns "Hello World"

- `altova:trim-string-right(" Hello World")` returns " Hello World"
- `altova:trim-string-right("Hello World")` returns "Hello World"
- `altova:trim-string-right("Hello World")` returns "Hello World"

12.2.1.9 Fonctions XPath/XQuery : Divers

L'objectif général suivant des fonctions d'extension XPath/XQuery sont prises en charge dans la version actuelle de StyleVision 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 présumé 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() **as** **xs: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() **as** **xs: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) as node() **XP3.1** **XQ3.1**

The `HTMLText` argument is a string that contains the text of an HTML document. The function creates an HTML tree from the string. The submitted string may or may not contain the HTML element. In either case, the root element of the tree is an element named `HTML`. It is best to make sure that the HTML code in the submitted string is valid HTML.

▢ Exemples

- **altova:parse-html** ("`<html><head/><body><h1>Header</h1></body></html>`") creates an HTML tree from the submitted string

▼ **sleep**[altova:]

altova:sleep(Millisecs as xs:integer) as empty-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](#) ⁶³³]

12.2.2 Fonctions d'extension diverses

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. Un bon exemple serait les fonctions maths disponibles dans Java, telles que `sin()` et `cos()`. Si ces fonctions étaient disponibles aux concepteurs des feuilles de style XSLT et des requêtes XQuery, ceci augmenterait la zone d'application des feuilles de style et des requêtes, et simplifierait grandement les tâches des créateurs de la feuille de style. Les moteurs XSLT et XQuery utilisés dans un nombre de produits Altova prennent en charge l'utilisation des fonctions d'extension dans [Java](#) ⁶³⁵ et [.NET](#) ⁶⁴⁴, ainsi que les [scripts MSXSL pour XSLT](#) ⁶⁵¹. Cette section décrit comment utiliser les fonctions d'extension et les scripts MSXSL dans vos feuilles de scripts XSLT. Les fonctions d'extension disponibles sont organisées dans les sections suivantes :

- [Fonctions d'extension Java](#) ⁶³⁵
- [Fonctions d'extension .NET](#) ⁶⁴⁴
- [Scripts MSXSL pour XSLT](#) ⁶⁵¹

Les deux problèmes principaux considérés pour les descriptions sont : (i) comment sont appelées les fonctions dans les bibliothèques respectives ; et (ii) quelles sont les règles à suivre pour la conversion d'arguments 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.

12.2.2.1 Fonctions d'extension Java

Une fonction d'extension Java peut être utilisée à l'intérieur d'une expression XPath ou XQuery pour invoquer un constructeur Java ou appeler une méthode Java (statique ou instance).

Un champ dans une classe Java est considéré être une méthode sans argument. Un champ peut être statique ou d'instance. L'accès aux champs est décrit dans les sous-sections respectives, statiques et d'instance.

Cette section est organisée dans les sous-sections suivantes :

- [Java : Constructeurs](#) ⁶⁴¹

- [Java : Méthodes statiques et champs statiques](#) ⁶⁴¹
- [Java : Méthodes d'instance et champs d'instance](#) ⁶⁴²
- [Types de données : XPath/XQuery en Java](#) ⁶⁴³
- [Types de données : Java en XPath/XQuery](#) ⁶⁴⁴

Veuillez noter les points suivants

- Si vous utilisez un produit Altova pour ordinateur de bureau, l'application Altova tente de détecter automatiquement le chemin d'accès à la machine virtuelle Java en lisant (dans cet ordre) : (i) le registre Windows, et (ii) la variable d'environnement `JAVA_HOME`. Vous pouvez également ajouter un chemin d'accès personnalisé dans le dialogue Options de l'application ; cette entrée aura priorité sur tout autre chemin d'accès à la machine virtuelle Java détecté automatiquement.
- Si vous exécutez un produit serveur Altova sur une machine Windows, le chemin d'accès à la machine virtuelle Java sera d'abord lu à partir du registre Windows ; si cela échoue, la variable d'environnement `JAVA_HOME` sera utilisée.
- Si vous utilisez un produit serveur Altova sur une machine Linux, assurez-vous que la variable d'environnement `JAVA_HOME` est correctement définie et que la bibliothèque Java Virtual Machines (sous Windows, le fichier `jvm.dll`) se trouve dans le répertoire `\bin\server` ou `\bin\client`.

Forme de la fonction d'extension

La fonction d'extension de l'expression XPath/XQuery doit avoir la forme `prefix:fname()`.

- La partie `prefix:` identifie la fonction d'extension en tant qu'une fonction Java. Elle procède en associant la fonction d'extension avec une déclaration d'espace de nom in-scope, dont l'URI doit commencer avec `java:` (*voir les exemples ci-dessous*). La déclaration d'espace de nom devrait identifier une classe Java, par exemple : `xmlns:myns="java:java.lang.Math"`. Néanmoins, elle pourrait être simplement : `xmlns:myns="java"` (sans les doubles points), l'identification de la classe Java se faisant dans la partie `fname()` de la fonction d'extension.
- La partie `fname()` identifie la méthode Java appelée, et fournit les arguments pour la méthode (*voir les exemples ci-dessous*). Néanmoins, si l'URI d'espace de nom identifié par la partie `prefix:` n'identifie pas une classe Java (*voir point précédent*), alors la classe Java devrait être identifiée dans la partie `fname()`, avant la classe et séparé de la classe par un point (*voir le second exemple XSLT ci-dessous*).

Note : La classe appelée doit être sur le classpath de la machine.

Exemple XSLT

Voici deux exemples pour démontrer comment appeler une méthode statique. Dans le premier exemple, le nom de classe (`java.lang.Math`) est inclus dans l'URI d'espace de nom et ne doit donc pas être dans la même partie `fname()`. Dans le second exemple, la partie `prefix:` fournit le préfixe `java:` alors que la partie `fname()` identifie la classe et la méthode.

```
<xsl:value-of xmlns:jMath="java:java.lang.Math"
  select="jMath:cos(3.14)" />

<xsl:value-of xmlns:jmath="java"
  select="jmath:java.lang.Math.cos(3.14)" />
```

La méthode nommée dans la fonction d'extension (`cos()` dans l'exemple ci-dessus) doit correspondre au nom d'une méthode statique publique dans la classe Java nommée (`java.lang.Math` dans l'exemple ci-dessus).

Exemple XQuery

Voici un exemple XQuery semblable à l'exemple XSLT ci-dessus:

```
<cosine xmlns:jMath="java:java.lang.Math">
  {jMath:cos(3.14)}
</cosine>
```

Classes Java définies par l'utilisateur

Si vous avez créé vos propres classes Java, les méthodes dans ces classes sont appelées différemment selon que : (i) les classes sont accédées par un fichier JAR ou un fichier classe et (ii) si ces fichiers (JAR ou classe) sont situés dans le répertoire actuel (le même répertoire que le document XSLT ou XQuery) ou pas. Consulter les sections [Fichiers de classe définis par l'utilisateur](#)⁶³⁷ et [Fichiers Jar définis par l'utilisateur](#)⁶⁴⁰ pour apprendre comment localiser ces fichiers. Veuillez noter que les chemins vers les fichiers de classe ne se trouvant pas dans le répertoire actuel et vers tous les fichiers JAR doivent être spécifiés.

Note : Si vous souhaitez ajouter un espace de nom à une feuille de style XSLT générée depuis un SPS créé dans StyleVision, l'espace de nom doit être ajouté à l'élément de niveau supérieur `schema` du schéma XML sur lequel le SPS se base. Veuillez noter que la déclaration d'espace de nom suivante `xmlns:java="java"` est créée automatiquement par défaut dans chaque SPS créé dans StyleVision.

12.2.2.1.1 Fichiers de classe définis par l'utilisateur

Si l'accès se produit par un fichier de classe, quatre possibilités s'ouvrent à vous :

- Le fichier de classe se trouve dans un package. Le fichier XSLT ou XQuery se trouve dans le même dossier que le package Java. ([Voir exemple ci-dessous](#)⁶³⁸.)
- Le fichier de classe ne se trouve pas dans un package. Le fichier XSLT ou XQuery se trouve dans le même dossier que le fichier de classe. ([Voir exemple ci-dessous](#)⁶³⁸.)
- Le fichier de classe se trouve dans un package. Le fichier XSLT ou XQuery se trouve dans un emplacement aléatoire. ([Voir exemple ci-dessous](#)⁶³⁸.)
- Le fichier de classe ne se trouve pas dans un package. Le fichier XSLT ou XQuery se trouve dans un emplacement aléatoire. ([Voir exemple ci-dessous](#)⁶³⁹.)

Prenons le cas où le fichier de classe ne se trouve pas dans un package et se trouve dans le même dossier que le document XSLT ou XQuery document. Dans ce cas, puisque toutes les classes dans le dossier sont trouvées, l'emplacement du fichier ne doit pas être spécifié. La syntaxe pour identifier une classe est :

```
java:classname
```

alors que

```
java: indique qu'une fonction Java définie par l'utilisateur est appelée ; (les classes Java dans le
répertoire actuelles seront chargées par défaut)
classname est le nom de la classe de la méthode requise
```

La classe est identifiée dans un URI d'espace de nom, et l'espace de nom est utilisé pour préfixer un appel de méthode.

Fichier de classe dans un package, le fichier XSLT/XQuery se trouve dans le même dossier que le package Java

L'exemple ci-dessous appelle la méthode `getVehicleType()` de la classe `Car` du package `com.altova.extfunc`. Le package `com.altova.extfunc` se trouve dans le dossier `JavaProject`. Le fichier XSLT se trouve également dans le dossier `JavaProject`.

```
<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:car="java:com.altova.extfunc.Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>

<xsl:template match="/">
  <a>
    <xsl:value-of select="car:getVehicleType()" />
  </a>
</xsl:template>

</xsl:stylesheet>
```

Fichier de classe est référencé, le fichier XSLT/XQuery se trouve dans le même dossier que le fichier de classe

L'exemple ci-dessous appelle la méthode `getVehicleType()` de la classe `Car`. Partons du principe que (i) le fichier de classe `Car` se trouve dans l'emplacement de dossier suivant : `JavaProject/com/altova/extfunc` et (ii) que ce dossier est le dossier actuel de l'exemple ci-dessous. Le fichier XSLT se trouve aussi dans le dossier `JavaProject/com/altova/extfunc`.

```
<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:car="java:Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>

<xsl:template match="/">
  <a>
    <xsl:value-of select="car:getVehicleType()" />
  </a>
</xsl:template>

</xsl:stylesheet>
```

Fichier de classe dans un package, le fichier XSLT/XQuery se trouve dans n'importe quel emplacement

L'exemple ci-dessous appelle la méthode `getCarColor()` de la classe `Car` du package `com.altova.extfunc`. Le package `com.altova.extfunc` se trouve dans le dossier `JavaProject`. Le fichier XSLT se trouve dans

n'importe quel emplacement. Dans ce cas, l'emplacement du package doit être spécifié dans l'URI en tant que chaîne de requête. La syntaxe est :

```
java:classname[?path=uri-of-package]
```

alors que

java : indique qu'une fonction Java définie par l'utilisateur est appelée
 uri-of-package est l'URI du package Java
 classname est le nom de la classe de méthode requise

La classe est identifiée dans un URI d'espace de nom et l'espace de nom est utilisé pour préfixer un appel de méthode. L'exemple ci-dessous montre comment accéder à un fichier de classe qui se situe dans un autre répertoire que le répertoire actuel.

```
<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:car="java:com.altova.extfunc.Car?path=file:///C:/JavaProject/" >

  <xsl:output exclude-result-prefixes="fn car xsl xs"/>

  <xsl:template match="/">
    <xsl:variable name="myCar" select="car:new('red') " />
    <a><xsl:value-of select="car:getCarColor($myCar)"/></a>
  </xsl:template>

</xsl:stylesheet>
```

Fichier de classe référencé, XSLT/XQuery se trouve dans n'importe quel emplacement

L'exemple ci-dessous appelle la méthode `getCarColor()` de la classe `Car`. Si par exemple le fichier de classe `Car` se trouve dans le dossier `C:/JavaProject/com/altova/extfunc` et que le fichier XSLT se trouve dans n'importe quel emplacement. L'emplacement du fichier de classe doit être spécifié dans le cadre de l'URI d'espace de nom en tant que chaîne de requête. La syntaxe est :

```
java:classname[?path==<uri-of-classfile>]
```

alors que

java : indique qu'une fonction Java définie par l'utilisateur est appelée
 uri-of-classfile est l'URI du dossier contenant le fichier de classe
 classname est le nom de la classe de méthode requise

La classe est identifiée dans un URI d'espace de nom et l'espace de nom est utilisé pour préfixer un appel de méthode. L'exemple ci-dessous montre comment accéder à un fichier de classe qui se situe dans un autre répertoire que le répertoire actuel.

```
<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:car="java:Car?path=file:///C:/JavaProject/com/altova/extfunc/" >

<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
  <xsl:variable name="myCar" select="car:new('red') " />
  <a><xsl:value-of select="car:getCarColor($myCar) "/></a>
</xsl:template>

</xsl:stylesheet>

```

Note : Lorsqu'un chemin d'accès est fourni par le biais de la fonction d'extension, le chemin est ajouté au ClassLoader.

12.2.2.1.2 Fichiers Jar définis par l'utilisateur

Si l'accès s'effectue par le biais d'un fichier JAR, l'URI du fichier JAR doit être spécifié à l'aide de la syntaxe suivante :

```
xmlns:classNS="java:classname?path=jar:uri-of-jarfile!/"
```

La méthode est ensuite appelée en utilisant le préfixe de l'URI d'espace de nom qui identifie la classe :

```
classNS:method()
```

Dans l'exemple ci-dessus :

```

java: indique qu'une fonction Java est appelée
classname est le nom de la classe définie par l'utilisateur
? est le séparateur entre le nom de classe et le chemin d'accès
path=jar: indique qu'un chemin d'accès vers un fichier JAR est donné
uri-of-jarfile est l'URI du fichier jar
!/ est le délimiteur de fin du chemin d'accès
classNS:method() est l'appel de la méthode

```

En alternative, le nom de classe peut être donné avec l'appel de la méthode. Voici deux exemples de la syntaxe :

```
xmlns:ns1="java:docx.layout.pages?path=jar:file:///c:/projects/docs/docx.jar!/"
ns1:main()
```

```
xmlns:ns2="java?path=jar:file:///c:/projects/docs/docx.jar!/"
ns2:docx.layout.pages.main()
```

Voici un exemple XSLT complet qui utilise un fichier JAR pour appeler une fonction d'extension Java :

```

<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:car="java?path=jar:file:///C:/test/Car1.jar!/" >
<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
    <xsl:variable name="myCar" select="car:Car1.new('red') " />
    <a><xsl:value-of select="car:Car1.getCarColor($myCar) "/></a>
</xsl:template>

<xsl:template match="car"/>

</xsl:stylesheet>

```

Note : Lorsqu'un chemin d'accès est fourni par le biais de la fonction d'extension, le chemin est ajouté au ClassLoader.

12.2.2.1.3 Java: Constructeurs

Une fonction d'extension peut être utilisée pour appeler un constructeur Java. Tous les constructeurs sont appelés avec la pseudo-fonction `new()`.

Si le résultat d'un appel de constructeur Java peut être [converti implicitement en des types de données XPath/XQuery](#) ⁶⁴⁴, alors la fonction d'extension Java retournera une séquence qui est un type de données XPath/XQuery. Si le résultat d'un appel de constructeur Java ne peut pas être converti en un type de données XPath/XQuery adéquat, le constructeur crée un objet Java encapsulé avec un type qui est le nom de la classe retournant cet objet Java. Par exemple, si un constructeur pour la classe `java.util.Date` est appelé (`java.util.Date.new()`), alors un objet ayant un type `java.util.Date` est retourné. Le format lexical de l'objet retourné peut ne pas être conforme au format lexical d'un type de données XPath et la valeur devrait donc être convertie dans le format lexical du type de données XPath requis, puis au type de données XPath requis.

Deux choses peuvent être réalisées avec un objet Java créé par un constructeur :

- Il peut être attribué à une variable :


```
<xsl:variable name="currentdate" select="date:new() "
xmlns:date="java:java.util.Date" />
```
- Il peut être passé à une fonction d'extension (voir [Méthode d'instance et Champs d'instance](#) ⁶⁴²) :


```
<xsl:value-of select="date:toString(date:new()) " xmlns:date="java:java.util.Date" />
```

12.2.2.1.4 Java: Méthodes statiques et Champs statiques

Une méthode est appelée directement par son nom Java et en fournissant les arguments pour la méthode. Les champs statiques (méthodes qui ne prennent aucun argument), comme les champs de valeur constante `E` et `PI`, sont accédés sans devoir spécifier aucun argument.

Exemples XSLT

Voici quelques exemples montrant comment appeler des méthodes et des champs statiques :

```

<xsl:value-of xmlns:jMath="java:java.lang.Math"
               select="jMath:cos(3.14)" />

<xsl:value-of xmlns:jMath="java:java.lang.Math"
               select="jMath:cos( jMath:PI() )" />

<xsl:value-of xmlns:jMath="java:java.lang.Math"
               select="jMath:E() * jMath:cos(3.14)" />

```

Veuillez noter que les fonctions d'extension ci-dessus prennent la forme `prefix:fname()`. Le préfixe dans les trois cas indiqués est `jMath:`, qui est associé avec l'URI d'espace de nom `java:java.lang.Math`. (L'URI d'espace de nom doit commencer avec `java:`. Dans les exemples ci-dessus, il est étendu pour contenir le nom de classe (`java.lang.Math`.) La partie `fname()` des fonctions extension doit correspondre au nom d'une classe publique (par ex. `java.lang.Math`) suivi du nom d'une méthode statique publique avec son/ses argument/s (comme `cos(3.14)`) ou un champ statique public (comme `PI()`).

Dans les exemples ci-dessus, Le nom de classe a été inclus dans l'espace de nom URI. S'il n'était pas contenu dans l'URI d'espace de nom, il devrait être inclus dans la partie `fname()` de la fonction d'extension. Par exemple :

```

<xsl:value-of xmlns:java="java:"
               select="java:java.lang.Math.cos(3.14)" />

```

Exemple XQuery

Un exemple semblable dans XQuery serait :

```

<cosine xmlns:jMath="java:java.lang.Math">
  {jMath:cos(3.14)}
</cosine>

```

12.2.2.1.5 Java: Méthodes d'instance et Champs d'instance

Une méthode d'instance a un objet Java qui lui est passé en tant que son premier argument de l'appel de méthode. Un tel objet Java est généralement créé en utilisant une fonction d'extension (par exemple un appel de constructeur) ou un paramètre/variable de feuille de style. Un exemple XSLT de ce type serait :

```

<xsl:stylesheet version="1.0" exclude-result-prefixes="date"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="java:java.util.Date"
  xmlns:jlang="java:java.lang">
  <xsl:param name="CurrentDate" select="date:new()" />
  <xsl:template match="/">
    <enrollment institution-id="Altova School"
      date="{date:toString($CurrentDate)}"
      type="{jlang:Object.toString(jlang:Object.getClass( date:new() ))}" />
    </enrollment>
  </xsl:template>
</xsl:stylesheet>

```

Dans l'exemple ci-dessus, la valeur du nœud `enrollment/@type` est créée comme suit :

1. Un objet est créé avec un constructeur pour la classe `java.util.Date` (avec le constructeur `date:new()`).
2. Cet objet Java est passé en tant que l'argument de la méthode `java.lang.Object.getClass`.
3. L'objet obtenu par la méthode `getClass` est passé en tant que l'argument de la méthode `java.lang.Object.toString`.

Le résultat (la valeur de `@type`) aura une chaîne avec la valeur : `java.util.Date`.

Un champ d'instance est théoriquement différent d'une méthode d'instance dans le sens où ce n'est pas un objet Java en soit qui est passé en tant qu'argument au champ d'instance. Au lieu, un paramètre ou une variable est passée en tant que l'argument. Néanmoins, le paramètre/variable peut contenir elle-même la valeur retournée par un objet Java. Par exemple, le paramètre `CurrentDate` prend la valeur retournée par un constructeur pour la classe `java.util.Date`. Cette valeur est ensuite passée en tant qu'un argument à la méthode d'instance `date:toString` afin de fournir la valeur de `/enrollment/@date`.

12.2.2.1.6 Types de données : XPath/XQuery en Java

Lorsqu'une fonction Java est appelée depuis l'intérieur d'une expression XPath/XQuery, le type de données des arguments de la fonction est important pour déterminer laquelle des multiples classes Java possédant le même nom est appelé.

dans Java, les règles suivantes sont suivies :

- S'il y a plus d'une seule méthode Java portant le même nom, mais chacune contenant un nombre différent d'arguments que les autres, alors c'est la méthode Java qui correspond le mieux au nombre d'arguments dans l'appel de la fonction qui est sélectionnée.
- La chaîne XPath/XQuery, le nombre et les types de données booléens (*voir la liste ci-dessous*) sont convertis implicitement en un type de données Java correspondant. Si le type XPath/XQuery fourni peut être converti en plus d'un seul type Java (par exemple, `xs:integer`), alors, le type Java sélectionné est celui qui est déclaré pour la méthode sélectionnée. Par exemple, si la méthode Java appelée est `fx(decimal)` et que le type de données XPath/XQuery fourni est `xs:integer`, alors `xs:integer` sera converti dans le type de données `decimal` de Java.

La table ci-dessous recense les conversions implicites de la chaîne XPath/XQuery, le nombre et les types booléens en types de données Java.

<code>xs:string</code>	<code>java.lang.String</code>
<code>xs:boolean</code>	<code>boolean (primitive)</code> , <code>java.lang.Boolean</code>
<code>xs:integer</code>	<code>int</code> , <code>long</code> , <code>short</code> , <code>byte</code> , <code>float</code> , <code>double</code> , et les classes de wrapper de ces catégories comme <code>java.lang.Integer</code>
<code>xs:float</code>	<code>float (primitive)</code> , <code>java.lang.Float</code> , <code>double (primitive)</code>
<code>xs:double</code>	<code>double (primitive)</code> , <code>java.lang.Double</code>

<code>xs:decimal</code>	<code>float (primitive), java.lang.Float,</code> <code>double(primitive), java.lang.Double</code>
-------------------------	--

Les sous-types des types de données du schéma XML recensés ci-dessus (et qui sont utilisés dans XPath et XQuery) seront aussi convertis dans le/s type/s Java correspondant à ce type d'ancêtre du sous-type.

Dans certains cas, il peut ne pas être possible de sélectionner la méthode Java correcte sur la base du cas suivant.

- L'argument fourni est une valeur `xs:untypedAtomic` de 10 et il est prévu pour la méthode `mymethod(float)`.
- Néanmoins, il existe une autre méthode dans la classe qui prend un argument d'un autre type de données : `mymethod(double)`.
- Puisque les noms de la méthode sont les mêmes et que le type fourni (`xs:untypedAtomic`) peut être converti correctement soit à `float` ou `double`, il est possible que `xs:untypedAtomic` soit converti en `double` au lieu de `float`.
- Par conséquent, la méthode sélectionnée ne sera pas la méthode requise et peut ne pas produire le résultat attendu. Pour contourner ce problème, vous pouvez créer une méthode définie par l'utilisateur avec un autre nom et utiliser cette méthode.

Les types qui ne sont pas couverts dans la liste ci-dessus (par exemple, `xs:date`) ne sera pas converti et générera une erreur. Néanmoins, veuillez noter que dans certains cas, il peut être possible de créer le type Java requis en utilisant un constructeur Java.

12.2.2.1.7 Types de données : Java en XPath/XQuery

Lorsqu'une méthode Java retourne une valeur, le type de donnée de la valeur est de type chaîne, numérique ou booléen, alors il est converti dans le type XPath/XQuery correspondant. Par exemple, les types de données `java.lang.Boolean` et `boolean` de Java sont convertis en `xsd:boolean`.

Des arrays unidimensionnels retournés par des fonctions sont étendus à une séquence. Les arrays multidimensionnels ne seront pas convertis et devraient donc être encapsulés.

Lorsqu'un objet Java encapsulé ou un type de données différent de chaîne, numérique ou booléen est retourné, vous pouvez assurer la conversion dans le type XPath/XQuery requis tout d'abord en utilisant une méthode Java (par ex. `toString`) pour convertir l'objet Java en une chaîne. Dans XPath/XQuery, la chaîne peut être modifiée pour correspondre à la représentation lexicale du type requis puis convertie dans le type requis (par exemple, en utilisant l'expression `cast as`).

12.2.2.2 Fonctions d'extension .NET

Si vous utilisez .NET Framework 2+ sur un ordinateur Windows, vous pouvez utiliser des fonctions d'extension écrites dans n'importe quel langage .NET (par exemple, C#). Une fonction d'extension .NET peut être utilisée dans une expression XPath ou XQuery pour invoquer un constructeur, une propriété ou une méthode (statique ou instance) dans une classe .NET. (Dernier test effectué avec .NET Framework 4.8.1 en mai 2025.)

Une propriété d'une classe .NET est appelée en utilisant la syntaxe `get_PropertyName()`.

Cette section est organisée dans les sous-sections suivantes :

- [.NET: Constructeurs](#) ⁶⁴⁷
- [.NET: Méthodes statiques et champs statiques](#) ⁶⁴⁷
- [.NET: Méthodes d'instance et champs d'instance](#) ⁶⁴⁸
- [Types de données : XPath/XQuery en .NET](#) ⁶⁴⁹
- [Types de données : .NET en XPath/XQuery](#) ⁶⁵⁰

Forme la fonction d'extension

La fonction d'extension de l'expression XPath/XQuery doit avoir la forme `prefix:fname()`.

- La partie `prefix:` est associée avec un URI qui identifie la classe .NET en cours d'adressage.
- La partie `fname()` identifie le constructeur, la propriété ou la méthode (statique ou d'instance) à l'intérieur de la classe .NET, et fournit tout argument, le cas échéant.
- L'URI doit commencer avec `clitype:` (qui identifie la fonction comme étant une fonction d'extension .NET).
- La forme `prefix:fname()` de la fonction d'extension peut être utilisée avec les classes de système et avec les classes dans un assemblage chargé. Néanmoins, si une classe doit être chargée, les paramètres supplémentaires contenant l'information requise devront être fournis.

Paramètres

Pour charger un assemblage, utiliser les paramètres suivants :

<code>asm</code>	Le nom de l'assemblage à charger.
<code>ver</code>	Le numéro de la version (quatre entiers maximum séparés par des points).
<code>sn</code>	Le jeton clé du nom solide de l'assemblage (16 chiffres hex).
<code>from</code>	Un URI qui donne l'emplacement de l'assemblage (DLL) à charger. Si l'URI est relatif, il est relatif au document XSLT ou XQuery. Si ce paramètre est présent, tout autre paramètre est ignoré.
<code>partialname</code>	Le nom partiel de l'assemblage. Il est fourni dans <code>Assembly.LoadWith.PartialName()</code> , qui tentera de charger l'assemblage. Si <code>partialname</code> est présent, tout autre paramètre est ignoré.
<code>loc</code>	La région, par exemple, <code>en-US</code> . Le défaut est <code>neutral</code> .

Si l'assemblage doit être chargé depuis un DLL, utiliser le paramètre `from` et omettre le paramètre `sn`. Si l'assemblage doit être chargé depuis le Global Assembly Cache (GAC), utiliser le paramètre `sn` et omettre le paramètre `from`.

Un signe d'interrogation doit être inséré avant le premier paramètre, et les paramètres doivent être séparés par un point-virgule. Le nom du paramètre donne sa valeur avec un signe égal (*voir exemple ci-dessous*).

Exemples de déclarations d'espace de nom

Un exemple de déclaration d'espace de nom dans XSLT qui identifie le système class `System.Environment`:

```
xmlns:myns="clitype:System.Environment"
```

Un exemple d'une déclaration d'espace de nom dans XSLT qui identifie la classe devant être chargée en tant que `Trade.Forward.Scrip`:

```
xmlns:myns="clitype:Trade.Forward.Scrip?asm=forward;version=10.6.2.1"
```

Un exemple d'une déclaration d'espace de nom dans XQuery qui identifie la classe de système `MyManagedDLL.testClass`. Deux cas peuvent être distingués :

1. Lorsque l'assemblage est chargé depuis le GAC:

```
declare namespace cs="clitype:MyManagedDLL.testClass?asm=MyManagedDLL;
ver=1.2.3.4;loc=neutral;sn=b9f091b72dccfba8";
```
2. Lorsque l'assemblage est chargé depuis le DLL (références partielles ou complète ci-dessous) :

```
declare namespace cs="clitype:MyManagedDLL.testClass?from=file:///C:/Altova
Projects/extFunctions/MyManagedDLL.dll;
```

```
declare namespace cs="clitype:MyManagedDLL.testClass?from=MyManagedDLL.dll;
```

Exemple XSLT

Voici un exemple complet XSLT qui appelle les fonctions dans la classe de système `System.Math`:

```
<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:output method="xml" omit-xml-declaration="yes" />
  <xsl:template match="/">
    <math xmlns:math="clitype:System.Math">
      <sqrt><xsl:value-of select="math:Sqrt(9)"/></sqrt>
      <pi><xsl:value-of select="math:PI()"/></pi>
      <e><xsl:value-of select="math:E()"/></e>
      <pow><xsl:value-of select="math:Pow(math:PI(), math:E())"/></pow>
    </math>
  </xsl:template>
</xsl:stylesheet>
```

La déclaration d'espace de nom de l'élément `math` associe le préfixe `math:` avec l'URI `clitype:System.Math`. Le début `clitype:` de l'URI indique que ce qui suit identifie soit une classe de système soit une classe chargée. Le préfixe `math:` dans les expressions XPath associe les fonctions d'extension avec l'URI (et, par extension, la classe) `System.Math`. Les fonctions d'extension identifient des méthodes dans la classe `System.Math` et fournissent les arguments le cas échéant.

Exemple XQuery

Voici un fragment d'exemple XQuery similaire à l'exemple XSLT ci-dessus :

```
<math xmlns:math="clitype:System.Math">
  {math:Sqrt(9)}
</math>
```

Comme pour l'exemple XSLT ci-dessus, la déclaration d'espace de nom identifie la classe .NET, dans ce cas une classe de système. L'expression XQuery identifie la méthode à appeler et fournit l'argument.

12.2.2.2.1 .NET: Constructeurs

Une fonction d'extension peut être utilisée pour appeler un constructeur .NET. Tous les constructeurs sont appelés avec la pseudo-fonction `new()`. S'il y a plus d'un constructeur pour une classe, le constructeur qui correspond le mieux au nombre d'arguments fourni est sélectionné. Si aucun constructeur n'est jugé conforme aux arguments fournis, alors une erreur 'No constructor found' sera retournée.

Des constructeurs qui retournent des types de données XPath/XQuery

Si le résultat d'un appel de constructeur .NET peut être [converti implicitement dans des types de données XPath/XQuery](#) ⁶⁴⁴, alors la fonction d'extension .NET retournera une séquence qui est un type de données XPath/XQuery.

Des constructeurs qui retournent des objets .NET

Si le résultat d'un appel de constructeur .NET ne peut pas être converti en un type de données XPath/XQuery adéquat, le constructeur crée un objet .NET encapsulé avec un type qui est le nom de la classe retournant cet objet. Par exemple, si un constructeur pour la classe `System.DateTime` est appelé (avec `System.DateTime.new()`), alors un objet ayant le type `System.DateTime` est retourné.

Le format lexical de l'objet retourné peut ne pas être conforme au format lexical d'un type de données XPath. Dans ce cas, la valeur retournée devrait donc être : (i) convertie dans le format lexical du type de données XPath requis ; puis (ii) au type de données XPath requis.

Trois choses peuvent être réalisées avec un objet .NET créé par un constructeur :

- Il peut être réalisé dans une variable :

```
<xsl:variable name="currentdate" select="date:new(2008, 4, 29)"
xmlns:date="clitype:System.DateTime" />
```
- Il peut être passé à une fonction d'extension (voir [Méthode d'instance et Champs d'instance](#) ⁶⁴²):

```
<xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
xmlns:date="clitype:System.DateTime" />
```
- Il peut être converti en une chaîne, nombre ou booléenne :
- ```
<xsl:value-of select="xs:integer(date:get_Month(date:new(2008, 4, 29)))"
xmlns:date="clitype:System.DateTime" />
```

### 12.2.2.2.2 .NET: Méthodes statiques et Champs statiques

Une méthode est appelée directement par son nom et en fournissant les arguments pour la méthode. Le nom utilisé dans l'appel doit être exactement conforme à une méthode statique publique dans la classe spécifiée. Si le nom de la méthode et le nombre des arguments qui ont été donnés dans l'appel de la fonction correspondent à plus d'une méthode dans une classe, alors les types des arguments fournis sont évalués pour obtenir la

meilleure correspondance possible. Si une correspondance ne peut pas être trouvée sans ambiguïté, une erreur sera rapportée.

**Note :** Un champ dans une classe .NET est considéré être une méthode sans argument. Une propriété est appelée en utilisant la syntaxe `get_PropertyName()`.

## Exemples

Un exemple XSLT montrant un appel vers une méthode avec un argument (`System.Math.Sin(arg)`):

```
<xsl:value-of select="math:Sin(30)" xmlns:math="clitype:System.Math"/>
```

Un exemple XSLT montrant un appel vers un champ (considéré une méthode sans argument) (`System.Double.MaxValue()`):

```
<xsl:value-of select="double:MaxValue()" xmlns:double="clitype:System.Double"/>
```

Un exemple XSLT montrant un appel vers une propriété (syntaxe est `get_PropertyName()`) (`System.String()`):

```
<xsl:value-of select="string:get_Length('my string') "
xmlns:string="clitype:System.String"/>
```

Un exemple XQuery montrant un appel vers une méthode avec un argument (`System.Math.Sin(arg)`):

```
<sin xmlns:math="clitype:System.Math">
 { math:Sin(30) }
</sin>
```

### 12.2.2.2.3 .NET: Méthodes d'instance et Champs d'instance

Une méthode d'instance a un objet .NET qui lui est passé en tant que son premier argument de l'appel de méthode. Un tel objet .NET est généralement créé en utilisant une fonction d'extension (par exemple un appel de constructeur) ou un paramètre/variable de feuille de style. Un exemple XSLT de ce type serait :

```
<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:output method="xml" omit-xml-declaration="yes"/>
 <xsl:template match="/">
 <xsl:variable name="releasedate"
 select="date:new(2008, 4, 29)"
 xmlns:date="clitype:System.DateTime"/>
 <doc>
 <date>
 <xsl:value-of select="date:ToString(date:new(2008, 4, 29))" />
 </date>
 </doc>
 </template>
</stylesheet>
```



```

 xmlns:date="clitype:System.DateTime"/>
 </date>
 <date>
 <xsl:value-of select="date:ToString($releasedate) "
 xmlns:date="clitype:System.DateTime"/>
 </date>
</doc>
</xsl:template>
</xsl:stylesheet>

```

Dans l'exemple ci-dessus, un constructeur `System.DateTime (new(2008, 4, 29))` est utilisé pour créer un objet .NET de type `System.DateTime`. Cet objet est créé deux fois, une fois en tant que la valeur de la variable `releasedate`, une deuxième fois en tant que le premier et seul argument de la méthode `System.DateTime.ToString()`. La méthode d'instance `System.DateTime.ToString()` est appelée deux fois, les deux fois avec le constructeur `System.DateTime (new(2008, 4, 29))` en tant que son premier et seul argument. Dans une de ces instances, la variable `releasedate` est utilisée pour obtenir l'objet .NET.

## Méthodes d'instance et champs d'instance

La différence entre une méthode d'instance et un champ d'instance est théorique. Dans une méthode d'instance, un objet .NET est passé directement en tant qu'un argument ; dans un champ d'instance, un paramètre ou une variable est passé à la place—bien que le paramètre ou la variable puisse contenir un objet .NET. Par exemple, dans l'exemple ci-dessus, la variable `releasedate` contient un objet .NET, et c'est cette variable qui est passée en tant que l'argument de `ToString()` dans le deuxième constructeur d'élément `date`. C'est pourquoi, l'instance `ToString()` dans le premier élément `date` est une méthode d'instance alors que le deuxième est considéré être un champ d'instance. Le résultat produit dans les deux instances, néanmoins, est le même.

### 12.2.2.2.4 Types de données : XPath/XQuery en .NET

Lorsqu'une fonction d'extension .NET est utilisée dans une expression XPath/XQuery, les types de données des arguments de la fonction sont importants pour déterminer laquelle des méthodes .NET multiples possédant le même nom est appelée.

Dans .NET, les règles suivantes sont suivies :

- S'il y a plus d'une méthode portant le même nom dans une classe, les méthodes disponibles pour la sélection sont réduites à celles qui ont le même nombre d'arguments que l'appel de fonction.
- Les types de données chaîne, nombre et booléen XPath/XQuery (*voir la liste ci-dessous*) sont convertis implicitement en un type de données .NET correspondant. Si le type XPath/XQuery fourni peut être converti en plus d'un type .NET (par exemple, `xs:integer`), puis, ce type .NET est sélectionné ce qui est déclaré pour la méthode sélectionnée. Par exemple, si la méthode .NET appelée est `fx(double)` et que le type de données fourni XPath/XQuery est `xs:integer`, alors `xs:integer` sera converti dans le type de données `double` de .NET.

La table ci-dessous recense les conversions implicites des types chaîne, nombre et booléen XPath/XQuery en types de données .NET.

<code>xs:string</code>	<code>StringValue</code> , <code>string</code>
<code>xs:boolean</code>	<code>BooleanValue</code> , <code>bool</code>
<code>xs:integer</code>	<code>IntegerValue</code> , <code>decimal</code> , <code>long</code> , <code>integer</code> , <code>short</code> , <code>byte</code> , <code>double</code> , <code>float</code>
<code>xs:float</code>	<code>FloatValue</code> , <code>float</code> , <code>double</code>
<code>xs:double</code>	<code>DoubleValue</code> , <code>double</code>
<code>xs:decimal</code>	<code>DecimalValue</code> , <code>decimal</code> , <code>double</code> , <code>float</code>

Les sous-types des types de données de XML Schema recensés ci-dessus (et qui sont utilisés dans XPath et Query) seront aussi convertis dans le/s type/s .NET correspondant au type ancêtre du sous-type.

Dans certains cas, il peut ne pas être possible de sélectionner la méthode .NET correcte sur la base des informations fournies. Veuillez considérer le cas suivant, par exemple.

- L'argument fourni est une valeur `xs:untypedAtomic` de 10 et est prévue pour la méthode `mymethod(float)`.
- Néanmoins, il y a une autre méthode dans la classe qui prend un argument d'un autre type de données : `mymethod(double)`.
- Puisque les noms de la méthode et le type fourni (`xs:untypedAtomic`) ont pu être convertis correctement en `float` ou `double`, il est possible que `xs:untypedAtomic` soit converti en `double` au lieu de `float`.
- Par conséquent, la méthode sélectionnée ne sera pas celle requise et peut ne pas produire le résultat attendu. Pour contourner ce problème, vous pouvez créer une méthode définie par l'utilisateur avec un nom différent et utiliser cette méthode.

Les types qui ne sont pas couverts dans la liste ci-dessus (par exemple `xs:date`) ne seront pas convertis et généreront une erreur.

#### 12.2.2.2.5 Types de données : .NET en XPath/XQuery

Lorsqu'une méthode .NET retourne une valeur et que le type de données de la valeur est de type chaîne, numérique ou booléenne, il est converti dans le type XPath/XQuery correspondant. Par exemple, le type de données `decimal` de .NET est converti en `xsd:decimal`.

Lorsqu'un objet .NET ou un type de données différent de chaîne, numérique ou booléenne est retourné, vous pouvez assurer la conversion dans le type XPath/XQuery requis tout d'abord en utilisant une méthode .NET (par ex. `System.DateTime.ToString()`) pour convertir l'objet .NET en une chaîne. Dans XPath/XQuery, la chaîne peut être modifiée pour correspondre à la représentation lexicale du type requis puis convertie dans le type requis (par exemple, en utilisant l'expression `cast as`).

### 12.2.2.3 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>` est 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 style 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>
 Total Retail Price =
 $<xsl:value-of select="user:AddMargin(50)"/>

 Total Wholesale Price =
 $<xsl:value-of select="50"/>

 </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`.

```

<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.

## 12.3 Types de données dans les schémas XML générés à partir d'une BD

Lorsqu'un schéma XML est généré à partir d'une base de données (BD), les types de données spécifiques à cette BD sont convertis en types de données XML Schema. Les correspondances entre les types de données BD et les types de données XML Schema pour les BD couramment utilisées sont indiquées dans cette annexe. Sélectionnez dans la liste ci-dessous.

- [ADO](#) <sup>654</sup>
- [MS Access](#) <sup>655</sup>
- [Serveur MS SQL](#) <sup>656</sup>
- [MySQL](#) <sup>657</sup>
- [ODBC](#) <sup>657</sup>
- [Oracle](#) <sup>658</sup>
- [Sybase](#) <sup>659</sup>

### 12.3.1 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

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

## 12.3.2 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

### 12.3.3 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



### 12.3.4 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

### 12.3.5 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

### 12.3.6 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

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.

## 12.3.7 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

<code>smallint</code>	<code>xs:short</code>
<code>tinyint</code>	<code>xs:byte</code>

## 12.4 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](#) <sup>661</sup>
- [Moteurs Altova](#) <sup>661</sup>
- [Prise en charge Unicode](#) <sup>662</sup>
- [Utilisation Internet](#) <sup>662</sup>

### 12.4.1 SE et exigences de mémoire

#### Système d'exploitation

Les applications logicielles d'Altova sont disponibles pour les plateformes suivantes :

- Windows 10, Windows 11
- Windows Server 2016 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.

### 12.4.2 Moteurs Altova

#### Valideur XML

Lors de l'ouverture d'un document XML, l'application utilise son valideur 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 valideur 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 valideur 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 valideur 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.

### 12.4.3 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.

### 12.4.4 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

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.

## 12.5 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](#).

### 12.5.1 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 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



jours et devra aussi acheter une licence à l'issue de la période d'évaluation afin de pouvoir continuer à utiliser le produit.

## 12.5.2 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 licencié 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 de 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](https://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.

## 12.5.3 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/fr/privacy>

## 12.5.4 Packaging License Files with StyleVision Installer

Si vous souhaitez effectuer une installation silencieuse de StyleVision, il est recommandé de modifier la base de données MSI afin d'y inclure votre ou vos fichiers de licence. De cette manière, le programme d'installation installera non seulement le produit, mais également la licence. Pour plus d'informations sur la procédure à suivre, télécharger ce [fichier ZIP](#) depuis le site Web d'Altova et ouvrir le document PDF qu'il contient.

# Index

■

.docx (uniquement Enterprise Edition), 14, 30

## A

**À propos de StyleVision, 547**

**Abréviations,**

utilisées dans le manuel de l'utilisateur, 24

**Activer le logiciel, 543**

**Affichage Fractionner, 30**

**Affichages,**

de l'aperçu de la GUI, 32

**affichages des documents,**

dans GUI, 27

**Afficher balise, 28**

**Aide en ligne, 535, 543**

**Aide PDF, 535, 543**

**Aides à la saisie dans le design,**

activer et désactiver l'affichage, 485

description de, 32

Masquer, 32

Masquer automatiquement, 32

**Aides à la saisie dans le mode Design,**

activer et désactiver l'affichage, 485

**Ajouter un schéma, 452**

**alertes de modification du fichier,**

dans SPS modulaire, 224

**Alignement vertical du contenu des cellules de table,**

dans SPSs, 521

**Aligner contenu des cellules de table,**

dans SPSs, 521

**AltovaXML,**

et FOP, 414

**ancrage,**

Fenêtres de l'Assistant à la saisie du design, 32

**Annexes, 548**

**Aperçu de sortie, 486**

**Aperçu design SPS, 98**

**Aperçu du design,**

fenêtre de la barre latérale, 35

**Aperçu du mécanisme du catalogue, 103**

**Aperçus de feuille de style XSLT,**

dans Aperçus de sortie, 30

**Aperçus de sortie,**

description de, 30

**application ASP.NET, 401**

**application web ASPX, 401**

**Apposer,**

colonne à la table dans SPS, 517

ligne à la table dans SPS, 517

**Arborescence de design,**

et SPS modulaire, 224

fenêtre de la barre latérale, 40

voir aussi assistants à la saisie du design, 32

**Assigner des formats prédéfinis,**

dans le tutoriel Démarrage rapide, 72

**Assistance pour StyleVision, 547**

**Assistance technique pour StyleVision, 547**

**Assistant à la saisie Propriétés,**

Groupe d'événement, 391

**Attributs de classe,**

dans le tutoriel Démarrage rapide, 72

**Authentic Browser, 18**

**Authentic Desktop, 18**

**Authentic View,**

dans les produits Altova, 18

## B

**Balise consécutive, 28**

**Balises,**

réduire et agrandir, 482

**Balises dans Aperçu de design, 28**

**Barre de menu,**

déplacer, 26

**Barre de statut, 484**

**Barre d'outils, 443**

activer et désactiver l'affichage, 484

activer/désactiver l'affichage, 443

ajouter/supprimer les icônes dans, 443

Barre d'outils de la table, 446

Barre d'outils standard, 450

déplacer, 26

Formater barre d'outils, 445

Insérer la barre d'outils Éléments de design, 447

**Barre d'outils, 443**

- personnaliser, 484
- positionner dans GUI, 443
- réinitialiser, 443

**Barre latérale Chercher & Remplacer, 52****Barre latérale des messages, 52****Bas de page,**

- ajouter dans la table, 517
- dans tables, 135

**Base de données,**

- boutons de barre d'outils à éditer, 450

**Base de données (éditions Enterprise et Professional),**

- voir sous BD, 13

**Base year,**

- dans formatage d'entrée, 334

**Blocs de texte définis par l'utilisateur, 122, 508****Blocs de texte XML définis par l'utilisateur, 123****Blueprints pour la mise en page, 171****Bordures,**

- des tables SPS, 519

**Boutons, 164****Boutons de barre d'outils,**

- ajouter et supprimer, 444

**Boutons radio, 164****branches de condition Otherwise, 266**

## C

**Calcul automatique,**

- et échappement de sortie, 331

**Calculs automatiques, 261**

- commande d'insertion dans le design, 490
- comment utiliser, 261
- Créer, éditer, formater, 261
- dans le tutoriel Démarrage rapide, 78
- déplacer, 261
- et conditions, 270
- exemples, 275
- fichiers d'exemples, 263
- Fonctions Java et :NET dans (Enterprise edition uniquement), 261
- formater les résultats de date, 385
- mettre à jour un nœud avec la valeur de, 490
- symbole dans le mode Design, 421

**Cases à cocher, 160****Cases de mise en page, 175****Catalogues dans RaptorXML, 104****Catalogues et variables d'environnement, 107****Cellule (de table),**

- fractionner horizontalement, 519
- fractionner verticalement, 519

**Cellules,**

- fusionner dans les tables SPS, 518

**Centre de téléchargement des composants,**

- sur le site web d'Altova, 547

**Chaînes de formatage,**

- définissant le formatage des entrées, 523

**Chaînes de formatage prédéfinies,**

- pour le formatage des entrées, 523

**Champ de saisie multiligne, 159****Champs d'entrée, 159****Champs numériques,**

- formatage de, 334

**Chercher,**

- utiliser des expressions régulières, 476

**Chercher dans Aperçu de design, 52****Clé d'évaluation,**

- pour votre logiciel Altova, 543

**Codes clé,**

- pour votre logiciel Altova, 543

**Codes-barres, 166****Colonne,**

- apposer à la table SPS, 517
- insérer dans la table SPS, 517
- supprimer de la table dans SPS, 518

**Colonnes (de tables),**

- dissimuler dans sortie HTML, 140

**Commande Annuler, 476****Commande Aperçu avant impression, 474****Commande Change-To, 181****Commande Coller, 476****Commande Copier, 476****Commande Couper, 476****Commande d'encodage, 472****Commande Éditer modèle correspondant, 119****Commande Enregistrer design, 463****Commande Enregistrer les fichiers générés, 469****Commande Fermer (SPS), 458****Commande Imprimer, 474****Commande Ouvrir (SPS), 458****Commande Quitter, 475****Commande Rechercher, 476****Commande Remplacer (éditions Enterprise et Professional), 476**

**Commande Rétablir, 476**  
**Commande Sélectionner tout, 476**  
**Commande Supprimer, 476**  
**Commande Trouver suivant, 476**  
**Commander le logiciel Altova, 543**  
**Commandes,**  
    personnaliser, 530  
**Commandes de formulaire,**  
    commandes de menu pour l'insertion, 489  
**Commenter le contenu, 503, 514**  
**Compatibilité Internet Explorer, 100**  
**Condition,**  
    commande d'insertion dans le design, 501  
**Conditions,**  
    configuration, 266  
    dans le tutoriel Démarrage rapide, 82  
    éditer, 269  
    et calculs automatiques, 270  
    inclure dans, 512  
**Configurer StyleVision, 20**  
**Conteneur de layout, 506**  
**Conteneurs de layout et éléments, 447**  
**Conteneurs de mise en page, 171**  
**Contenu de la page HTML,**  
    Copier-coller dans le design, 114  
**Contenu de la table Excel,**  
    Copier-coller dans le design, 114  
**Contenu du document MS Word,**  
    Copier-coller dans le design, 114  
**Contenu du document Word,**  
    Copier-coller dans le design, 114  
**Contenu dynamique,**  
    dans le tutoriel Démarrage rapide, 60  
**Contenu statique,**  
    dans le tutoriel Démarrage rapide, 67  
**Contenus,**  
    commande d'insertion dans le design, 488  
**Contrat de licence de l'utilisateur final, 664, 666**  
**Conversion d'HTML en XML, 394**  
**CoreCatalog.xml, 104**  
**Créer un nouveau document SPS,**  
    dans le tutoriel Démarrage rapide, 55  
**CustomCatalog.xml, 104**

## D

**dans le dialogue XPath,**  
    débuguer les expressions dans, 431  
    description de, 426, 438  
    tester les expressions dans, 427  
**Date,**  
    formatage de, 334  
**Dates,**  
    comment utiliser dans SPS, 385  
    exemples de manipulation des données avec XPath 2.0, 385  
    formatage de, 385  
**de contenu de document XML,**  
    symbole dans le mode Design, 421  
**Décimaux,**  
    formatage de, 334  
**Désactivé (commande), 503, 514**  
**désactiver-l'échappement de sortie, 331**  
**Design MobileTogether,**  
    exporter vers, 469  
**Designs basés sur des formulaires, 171, 452**  
**Dialogue Éditer paramètres, 482**  
**Dialogue Personnaliser,**  
    pour personnaliser StyleVision, 484  
**Dictionnaire utilisateur par défaut,**  
    pour les vérifications orthographiques SPS, 526  
**Dictionnaires,**  
    pour les vérifications orthographiques SPS, 526  
**Dispositifs de saisie de données, 158**  
    commandes de menu pour l'insertion, 489  
    symbole dans le mode Design, 421  
**Dissimuler balisage, 28**  
**Distribution,**  
    des produits logiciels Altova, 664  
    des produits logiciels d'Altova, 664  
**Document de sortie temporaire, 20**  
**Documentation,**  
    Aperçu de, 24  
**Documents,**  
    Ouverture et fermeture, 27  
**Documents de sortie multiples, 251**  
    emplacement des fichiers, 256  
    et aperçus de sortie, 256  
    mettre en lien entre, 253  
**données XML,**

**données XML,**

- fusionner de multiples sources, 217
- insérer dans le design SPS, 110

**DPI, 472****DTD,**

- comme source SPS, 188
- déclarer des entités non parsées, 363

**DTD et catalogues, 103**

## E

**Échappement de sortie, 331****Éditer le dialogue de l'expression XPath,**

- voir la dialogue XPath, 426

**Éditeur JavaScript, 389, 390**

- dans Aperçu de design, 28

**Élément du document,**

- définition de, 21

**Éléments,**

- définies par l'utilisateur, 122

**Éléments de design, 447****Éléments de document (voir éléments Racine), 187****Éléments définis par l'utilisateur, 122, 515****Éléments racine, 37****Éléments Racine (aka éléments de document),**

- et sources de schéma, 187
- Sélectionner pour schéma, 187

**Encodage,**

- pour les fichiers de sortie, 535

**Encodage de sortie, 472****Encodage des documents de sortie, 472****En-têtes,**

- ajouter dans la table, 517
- dans tables, 135

**Entier,**

- formatage de, 334

**Entités,**

- non parsé, 363
- utiliser comme détenteurs URI, 363

**Espace réservé de contenus,**

- dans le tutoriel Démarrage rapide, 60
- Insérer nœud comme contenus, 110

**Espaces de noms,**

- ajouter à la SPS, 37, 97, 102, 187
- Aperçu de, 40
- dans la SPS, 37

**Exigences de mémoire, 661****Expressions régulières,**

- chercher et remplacer en utilisant, 476

**Expressions XPath,**

- en cours de création dans le dialogue Éditer l'expression XPath, 426, 438
- et styles, 354

**Extensions Altova,**

- fonctions graphiques (voir fonctions graphique), 559

## F

**faire défiler les boutons,**

- dans la fenêtre principale, 27

**FAQ sur StyleVision, 547****Fenêtre de layout, 506****fenêtre de sources de schéma,**

- voir aussi assistants à la saisie du design, 32

**Fenêtres de l'Assistant à la saisie du design,**

- ancrage, 32
- flottant, 32

**Feuilles de style,**

- voir aussi sous feuilles de style CSS, 344
- voir aussi sous feuilles de style XSLT, 344

**Feuilles de style CSS,**

- externe appliqué à, 344
- feuilles de style externes, 344
- importer préséance de média, 344
- voir aussi Styles, 344

**Fichier de travail XML, 37, 97**

- aperçu d'impression, 474
- définition de, 21
- et aperçus de sortie, 30
- imprimer, 474

**Fichiers,**

- Ouvrir récemment utilisés, 475

**Fichiers batch,**

- et tâches prévues, 419

**Fichiers CSS,**

- gérer dans la barre latérale de l'Aperçu du design, 35

**Fichiers de catalogue, 103****Fichiers de sortie,**

- générer, 102

**Fichiers générés, 102****Fichiers PXF, 404**

- créer, 404

**Fichiers PXF, 404**

- éditer, 407
- en cours de déploiement, 408
- enregistrer sous, 468

**Fichiers récemment utilisés, 475****Fichiers source pour SPS, 97****filtre XPath,**

- sur modèles globaux, 234

**Filtres,**

- pour consulter les modèles de manière sélective, 449

**Filtres de design,**

- activer et désactiver, 486

**Filtres de modèle, 449****Filtres pour modèles de design,**

- activer et désactiver, 486

**Filtres sur nœud-modèles, 242****Filtres XPath sur nœud-modèles, 242****Flottant,**

- Fenêtres de l'Assistant à la saisie du design, 32

**Fonction de XSLT 'unparsed-entity-uri', 363****Fonctions,**

- dans XPath, défini par l'utilisateur, 369
- StyleVision, 14

**Fonctions de produit,**

- liste de, 14

**Fonctions d'extension .NET,**

- constructeurs, 647
- conversions de type de données .NET en XPath/XQuery, 650
- conversions de type de données XPath/XQuery en .NET, 649
- dans expressions XPath, 426, 438
- méthodes d'instance, champs d'instance, 648
- méthodes statiques, champs statiques, 647
- pour XSLT et XQuery, 644
- prise en charge de, dans le Mode Authentic, 426, 438

**Fonctions d'extension dans les scripts MSXSL, 651****Fonctions d'extension Java,**

- constructeurs, 641
- conversions de type de données XPath/XQuery en Java, 643
- conversions de type de données, Java en XPath/XQuery, 644
- dans expressions XPath, 426, 438
- fichiers de classe définis par l'utilisateur, 637
- fichiers JAR définis par l'utilisateur, 640
- méthodes d'instance, champs d'instance, 642
- méthodes statique, champs statiques, 641
- pour XSLT et XQuery, 635

- prise en charge de, dans le Mode Authentic, 426, 438

**Fonctions d'extension pour XSLT et XQuery,**

- Extensions Altova, 559
- Fonctions d'extension Java, 635, 644
- voir sous Fonctions d'extension .NET, 644
- voir sous les fonctions d'extension Java, 635

**Fonctions Java et .NET (Enterprise edition uniquement),**

- dans Calculs automatiques, 261

**Fonctions XPath,**

- dans le dialogue XPath, 426, 438
- définies par l'utilisateur, 369

**Fonctions XPath définies par l'utilisateur, 369****Formatage,**

- alignement de texte, 445
- aperçu des procédures, 328
- boutons de barre d'outils pour, 445
- de champs numériques, 334
- des formats prédéfinis HTML, 445
- nœuds sur insertion, 112
- pour tables, 135
- propriétés de texte, 445
- recense, 445
- voir aussi Présentation, 328

**Formatage de valeur, 334****Formatage des entrées,**

- définissant les chaînes de formatage pour, 523
- des dates, 385

**Formater les numéros,**

- dans numérotation automatique, 316

**Formats prédéfinis,**

- commande d'insertion dans le design, 491
- Sur insertion d'un nœud, 112
- symbole dans le mode Design, 421

**Fractionner cellule de table,**

- horizontalement, 519
- verticalement, 519

**Fragment de design,**

- insérer, 506

**Fragments de design, 245****Fusionner cellules,**

- dans les tables SPS, 518

## G

**Gestion des packs, 206****Gestionnaire de schéma,**

**Gestionnaire de schéma,**

- Aperçu CLI, 210
- aperçu de, 196
- Commande Aide CLI, 211
- Commande de mise à jour CLI, 215
- Commande de mise à niveau CLI, 216
- Commande désinstallation CLI, 214
- Commande Info CLI, 211
- Commande Initialiser CLI, 212
- Commande Installer CLI, 212
- Commande Liste CLI, 213
- Commande Réinitialiser CLI, 214
- comment exécuter, 199
- corriger un schéma, 204
- désinstaller un schéma, 205
- installer un schéma, 204
- mettre à niveau un schéma, 204
- recenser les schémas par statut dans, 202
- réinitialiser, 205
- statut de schémas dans, 202

**Gestionnaires d'événement,**

- attribuer les fonctions à, 391

**Graphiques,**

- aperçu de l'utilisation dans SPS, 152
- voir aussi sous Images, 152

**GUI,**

- affichages des documents dans, 27
- description de, 26
- documents multiples dans, 27
- Fenêtre principale de, 27

## H

**Hyperlien,**

- commande d'insertion dans le design, 500

**Hyperliens,**

- inclure dans, 512

## I

**IE 9,**

- Voir sous la compatibilité Internet Explorer, 100

**Image,**

- commande d'insertion dans le design, 492

**Images,**

- accéder pour le rendu de la sortie, 152
- dans le tutoriel Démarrage rapide, 67
- et unparsedentityuri, 152
- fichiers d'exemples, 157
- localiser via des entités non parsées, 363
- spécifier URI pour, 152
- symbole dans le mode Design, 421
- types pris en charge, 154

**Images intégrées, 472****Importation de modèles XSLT,**

- dans SPS, 249

**importation HTML, 394**

- Créer un nouvel SPS, 394
- de listes HTML, 398
- de Tables HTML, 398
- design SPS, 396
- Générer les fichiers depuis SPS, 400
- structure de schéma, 396

**importation XSLT, 365****Information de Copyright, 664****Information juridique, 664****Informations générales, 661****Informations techniques, 661****Inscription pour votre logiciel Altova, 543****Insérer,**

- colonne dans la table SPS, 517
- ligne dans la table SPS, 517

**Insérer des éléments de design via la barre d'outils, 447****Installation silencieuse,**

- modifier fichier MSI, 666
- packager les fichiers de licence avec l'installateur, 666

**Intégration d'images, 472****Interface,**

- voir GUI, 26

**Interface utilisateur,**

- voir GUI, 26

**Internet usage,**

- dans les produits Altova, 662

**Items définis par l'utilisateur, 122, 508****items TOC,**

- en cours de création, 307
- formater, 307



# J

## JavaScript,

Voir sous Scripts, 389

# L

## Licence, 666

information à propos de, 664

packager les fichiers de licence avec l'installateur, 666

## Licence de produit de logiciel, 666

## Licences,

pour votre logiciel Altova, 543

## License metering,

des produits Altova, 665

## Lien hypertexte, 165, 321

création et modification, 323

Et des entités non parsées, 323

localiser via des hyperliens, 363

mettre en lien vers des ressources externes, 323

mettre en lien vers les signets, 323

retirer et supprimer, 323

symbole dans le mode Design, 421

## Liens,

voir sous hyperliens, 165, 321

## Ligne,

apposer à la table SPS, 517

dans conteneurs de layout, 506

insérer dans la table SPS, 517

supprimer de la table dans SPS, 518

## Ligne de commande, 410

et paramètres, 285

et tâches prévues, 419

## Ligne horizontale,

commande d'insertion dans le design, 495

dans le tutoriel Démarrage rapide, 67

## Lignes (de tables),

Agrandir/réduire dans sortie HTML, 140

## Liste des propriétés, 522

## Listes, 147

dans le tutoriel Démarrage rapide, 82

Importé du document HTML, 398

inclure dans, 511

Listes (statiques et dynamiques), 496

Listes de choix, 161

Listes dynamiques, 147, 149, 496

Listes statiques, 147, 496, 511

Logiciel complémentaire,

à télécharger, 547

# M

Manuel de l'utilisateur, 535, 543

## Masquer,

fonction d'assistants à la saisie du design, 32

## Masquer automatiquement,

fonction d'assistants à la saisie du design, 32

## Menu,

personnaliser, 530

## Menu Affichage, 484

## Menu Aide, 543

## Menu Éditer, 476

## Menu Fenêtre, 542

## Menu Fichier, 452

commande Quitter, 475

Fichier | Aperçu avant impression, 474

Fichier | Encodage, 472

Fichier | Enregistrer design, 463

Fichier | Enregistrer les fichiers générés, 469

Fichier | Enregistrer sous, 468

Fichier | Fermer, 458

Fichier | Imprimer, 474

Fichier | Nouveau, 452

Fichier | Ouvrir, 458

## Menu Inclure dans, 509

## Menu Insérer, 488

Insérer | Calcul automatique, 490

Insérer | Condition, 501

Insérer | Contenus, 488

Insérer | Désactivé, 503

Insérer | Fragment de design, 506

Insérer | Hyperlien, 500

Insérer | Image, 492

Insérer | Ligne horizontale, 495

Insérer | Paragraphe, 491

Insérer | Paragraphe spécial, 491

Insérer | Reste des contenus, 489

Insérer | Signet, 498

Puces et numérotation, 496

**Menu Outils, 525****menu Propriétés, 522**

Puces et numérotation, 522

**Menu Table, 516****Métadonnées de sortie du document HTML, 360****Mettre à jour les nœuds (éditions Enterprise et Professional),**

Avec un résultat de Calcul automatique, 261

**Mettre en place un nouveau document SPS,**

dans le tutoriel Démarrage rapide, 55

**Microsoft Office 2007 (uniquement Enterprise Edition), 14, 30****Mise en page,**

d'aperçus dans la GUI, 32

**Mode Design,**

affichage de balise, 28

description de, 28

et Éditeur JavaScript, 28

symboles dans la conception SPS, 421

**Modèle,**

Changer la correspondance de nœud pour, 181

contenir dans, 509

insérer, 503

**Modèle de variable, 241**

contenir dans, 510

insérer, 505

**Modèle défini par l'utilisateur,**

contenir dans, 510

insérer, 504

**Modèle du nouveau document, 251**

insérer, 252

**Modèle fichier XML (éditions Enterprise et Professional), 97**

définition de, 21

**Modèle global complexe, 234****Modèle global simple, 234****Modèle local, 234****Modèle principal, 234**

définition de, 21

**Modèle TOC,**

création et modification, 303

formater, 307

références de niveau dans, 305

reflevels dans, 305

structurer, 305

**Modèles,**

activer et désactiver l'affichage, 486

arborescence de, 40

contenant des lignes de table et des colonnes dans, 133

supprimer autour des lignes et colonnes de table, 133

**Modèles basés sur type, 234****Modèles conditionnels, 501**

symbole dans le mode Design, 421

voir sous : Conditions, 266

**Modèles d'élément,**

définies par l'utilisateur, 122

**Modèles définis par l'utilisateur, 119****Modèles globaux, 234**

dans le tutoriel Démarrage rapide, 90

Effet sur reste-des-contenus, 113

**Modèles nommés, 234****Modèles pour nœuds,**

voir Nœud-modèles, 242

**Modèles XSLT, 40**

gérer dans la barre latérale de l'Aperçu du design, 35

Importer dans SPS, 249

**Modules,**

gérer dans la barre latérale de l'Aperçu du design, 35

**Modules de mise en page,**

étapes de création, 171

**Moteurs Altova,**

dans les produits Altova, 661

**msxsl:script, 651**

## N

**Niveau TM,**

contenir dans, 514

**Niveaux TOC, 296, 298****Nœud,**

Changé ce qui est créé comme, 181

**Nœud contextuel,**

dans le dialogue XPath, 426, 438

**Nœud du document,**

définition de, 21

**Nœud-modèles,**

Défini par l'utilisateur, 119

Et changer en modèles enfant, 242

et filtres XPath, 242

et modèles globaux, 242

opérations sur, 242

**nœuds de document XML,**

symbole dans le mode Design, 421

**Nombre,**

**Nombre,**

formatage de, 334

**Nouveauté de XSLT, 365****Nouveaux modèles de document,**

et structure de design, 253

URL de, 253

**Nouvelle commande, 452****Numérotation automatique, 316****Numéroter les nœuds automatiquement, 316**

## O

**Office Open XML (uniquement Enterprise Edition), 14, 30****OOXML (uniquement Enterprise Edition), 14, 30****Opérateurs XPath,**

dans le dialogue XPath, 426, 438

**options d'arborescence du schéma, 535****Options de prise en charge, 24****Options du projet, 535****Options du vérificateur d'orthographe,**

pour les SPS, 526

**Ouvrir,**

fichiers récemment utilisés, 475

## P

**Paragraphe,**

commande d'insertion dans le design, 491

contenir dans, 511

**Paragraphe spécial,**

commande d'insertion dans le design, 491

contenir dans, 511

**Paramètres, 285**

Aperçu des paramètres définis par l'utilisateur, 40

création et modification, 482

dans SPS, 285

description générale, 285

et Authentic View, 285

et ligne de commande, 285

et sources de schéma, 289

fragment pour le design, 287

gérer dans la barre latérale de l'Aperçu du design, 35

situer des nœuds dans de multiples documents avec, 289

**Paramètres BD,**

création et modification, 482

**Paramètres de réseau, 206****paramètres du proxy, 535****Période d'évaluation,**

des produits logiciels Altova, 664

des produits logiciels d'Altova, 664

**Personnalisation de catalogues, 106****Personnaliser les dictionnaires,**

pour les vérifications orthographiques SPS, 526

**Personnaliser StyleVision, 530****Pixels,**

et longueurs des supports d'impression, 472

et résolution d'écran, 472

**Plateformes,**

pour les produits Altova, 661

**Précédence,**

de styles, 44

**Présentation,**

aperçu des procédures, 328

voir aussi Formats, Formater, 328

**Prise en charge Unicode,**

dans les produits Altova, 662

**Problèmes avec l'affichage, 20****Procédure générale d'utilisation, 96****Procédures d'édition du contenu, 109****Procédures d'édition supplémentaires, 362****processeur FO (Enterprise Edition),**

configuration, 20

**Processeur XQuery,**

dans les produits Altova, 661

**Processeurs,**

à télécharger, 547

**Processeurs XSLT,**

dans les produits Altova, 661

**Propriétés,**

définir, 48

des tables SPS, 446, 519

et groupes de propriété, 48

fenêtre de la barre latérale, 48

voir aussi assistants à la saisie du design, 32

**Propriétés des documents de sortie, 258****propriétés document HTML, 360****Propriétés du document, 258****Proxy de réseau, 535****Puces et numérotation, 147, 149, 496, 522**

inclure dans, 511

# R

## Raccourcis,

personnaliser pour le clavier, 530

## Raccourcis clavier,

personnaliser pour les commandes, 530

## RaptorXML, 410

et FOP, 414

## Référence de l'utilisateur, 420

## Référence TOC, 306

## Références croisées, 319

## Références d'entité,

et échappement de sortie, 331

## Références de caractère,

et échappement de sortie, 331

## Références de texte, 319

## Référentiel de style,

et feuilles de style CSS externes, 344

et styles globaux, 348

fenêtre de la barre latérale, 44

voir aussi assistants à la saisie du design, 32

## Regroupement, 271

exemple group-by (Persons.sps), 274

exemple group-by (Scores.sps), 275

## Remplacer,

utiliser des expressions régulières, 476

## Remplacer dans Aperçu de design, 52

## Réseau, 206

## Restaurer barres d'outils et fenêtres, 535

## Reste-des-contenus, 113

commande d'insertion dans le design, 489

dans le tutoriel Démarrage rapide, 90

et modèles globaux, 234

## RootCatalog.xml, 104

# S

## Schéma principal, 234

## Schéma principal (uniquement Enterprise Edition), 37

## Schémas,

comme source SPS, 188

consulter par le biais des catalogues, 106

définies par l'utilisateur, 193

## Schémas défini par l'utilisateur, 193

## Schémas et catalogues, 103

## Schémas XML et DTD,

comme source SPS, 188

## Scripts,

Aperçu de, 40

dans l'arborescence de Design, 389

définir fonctions JavaScript, 390

et fonctions JavaScript, 389

Les fonctions JavaScript comme gestionnaires d'événement, 391

Utiliser dans un SPS, 389

## SE,

pour les produits Altova, 661

## Sections CDATA, 110

## Signet, 165, 321

commande d'insertion dans le design, 498

création et modification, 321

supprimer, 321

## Signet TdM,

contenir dans, 514

## Signets,

inclure dans, 512

## Signets (ancres),

symbole dans le mode Design, 421

## Signets TOC, 296

assistant pour, 300

créer, 300

et niveaux, 300

## Site web d'Altova, 547

## Sortie HTML, 102

et prise en charge d'image, 154

## sortie PDF (Enterprise Edition), 102

et prise en charge d'image, 154

## sortie RTF (Éditions Enterprise et Professional), 102

## sortie RTF (Enterprise Edition),

et prise en charge d'image, 154

## sortie Texte (Éditions Enterprise et Professional), 102

## Sorties multiples de document, 507

## Sources de schéma, 97, 452

Aperçu de, 40

changer sources, 289

et éléments racine (éléments de document), 187

fenêtre de la barre latérale, 37

gérer dans la barre latérale de l'Aperçu du design, 35

multiples dans SPS (Enterprise edition), 187

multiples sources et XPath, 289

Sélectionner pour SPS, 187

**Sources de schéma, 97, 452**

Sources multiples et situer nœuds, 289

**SPS,**

description générale de, 19  
et Authentic View (éditions Enterprise et Professional), 19  
et feuilles de style XSLT, 19  
et StyleVision, 19  
fermer, 458  
ouvrir, 458  
recharger, 458

**SPS modulaires,**

activation et désactivation, 224  
ajouter le module SPS, 224  
aperçu, 220  
créer, 224  
effet sur ordre de préséance, 224  
et déclarations d'espace de nom, 221  
et fichiers de travail XML, 221  
et fichiers modèles XML, 221  
et les alertes de modification du fichier, 224  
et objets modulaires, 221  
et scripts, 221  
et sources de schéma, 221, 224  
et styles CSS, 221, 224  
exemple de projet, 228  
fonctionne avec, 224  
le module SPS à ajouter, 224

**Structure du design, 185****Structure du design SPS, 185****Structure du fichier SPS, 185****Styles,**

assigner des feuilles de style CSS à SPS, 344  
combiner plusieurs, 357  
de données XML, 354  
définir, 46  
définir des styles locaux, 350  
définir les styles globaux dans SPS, 348  
et groupes de propriété, 46  
fenêtre de la barre latérale, 46  
média pour feuilles de style externes assignées, 344  
ordre de cascade, 343  
précédence de, 44  
préséance de styles, 348  
règles CSS combinées, 357  
terminologie de, 343  
travailler avec dans StyleVision, 343  
via des expressions XPath, 354  
voir aussi assistants à la saisie du design, 32

**styles Composite, 357****Styles CSS,**

dans le tutoriel Démarrage rapide, 72  
dans SPS modulaire, 224  
voir aussi Styles, 46

**Styles des documents de sortie, 258****Styles du document, 258****Styles Globaux,**

voir sous Styles, 348

**Styles locaux,**

voir sous Styles, 350

**StyleVision,**

fonctions de produit, 14  
manuel de l'utilisateur, 13

**StyleVision Power Stylesheet,**

voir sous SPS, 13

**StyleVisionBatch, 20, 410****Supprimer,**

colonne de la table dans SPS, 518  
ligne de la table dans SPS, 518  
table dans SPS, 516

**Symboles dans le mode Design,**

d'appareils d'entrée de données, 421  
d'hyperliens, 421  
d'images, 421  
de calculs automatiques, 421  
de contenu du document XML, 421  
de formats prédéfinis, 421  
de modèles conditionnels, 421  
de nœuds de document XML, 421  
de signets (ancres), 421

# T

**Table,**

afficher/masquer bordures dans StyleVision, 519  
ajouter des en-têtes et des pieds de page, 517  
alignement vertical du contenu des cellules, 521  
apposer colonne à, 517  
apposer ligne à, 517  
contenu des cellules, 516  
éditer propriétés de, 519  
en-têtes et pied de page, 516  
insérer colonne dans, 517  
insérer ligne dans, 517  
insérer une table statique, 516

**Table,**

- naviguer, 516
- supprimer colonne de, 518
- supprimer dans SPS, 516
- supprimer ligne de, 518

**Table de contenus,**

- voir sous TOC, 293

**Table dynamique,**

- boutons de barre d'outils pour éditer, 446

**Table statique,**

- boutons de barre d'outils pour éditer, 446
- insérer, 516
- insérer dans SPS, 446

**Tables,**

- Agrandir/réduite les lignes, 140
- aperçu, 125
- créer, 495
- créer des tables dynamiques, 128
- créer des tables statiques, 127
- en-têtes et bas de page dans PDF, 135
- Fermer bouton pour masquer les colonnes, 140
- formater, 135
- fusionner cellules dans, 518
- importé du document HTML, 398
- masquer les colonnes vides, 140
- styles pour des lignes alternées, 354
- traitement conditionnel dans, 133

**Tables (SPS),**

- boutons de barre d'outils à éditer, 446
- éditer les propriétés, 446

**Tables CALS/HTML, 141, 520****Tables dans Aperçu de design,**

- contenant dans et supprimer les modèles, 133
- représentation de, 133

**Tables dynamiques, 125**

- différence des lignes ajoutées/insérées, 128
- en-têtes et bas de page dans, 128
- et modèles globaux, 128
- tables dynamiques intégrées, 128
- voir aussi les tables SPS, 128
- voir aussi Tables, 135

**Tables HTML, 141, 520****Tables SPS,**

- voir aussi Tables dynamiques, 125
- Voir aussi tables statiques, 125

**Tables statiques, 125**

- voir aussi les tables SPS, 127
- voir aussi Tables, 135

**tables XML (Éditions Enterprise et Professional), 125****Tâche planifiée,**

- créant une commande StyleVisionBatch comme, 419
- fichiers batch StyleVisionBatch dans, 419

**TDM,**

- Aperçu de l'utilisation, 293
- commande de menu, 507
- exemple, hiérarchique et séquentiel, 312
- exemple, simple, 308
- Items marqués pour l'inclusion, 296

**Terminologie,**

- utilisée dans StyleVision, 21

**Texte vertical,**

- dans cellules de table, 135
- dans les cases de mise en page, 175

**TOCrefs,**

- voir sous références TOC, 306

**Traitement automatisé, 410****Transformations FO, 414****Transformations XSLT, 414****Trier, 279**

- des groupes et à l'intérieur des groupes, 271, 274, 275
- fichiers d'exemples, 281
- Mécansime de tri, 279
- Trier la clé, 279

**Trier la clé, 279****Tutoriel Démarrage rapide,**

- attributs de classe, 72
- Calculs automatiques, 78
- conditions, 82
- configurer un nouveau document SPS, 55
- contenu dynamique, 60
- Contenu statique, 67
- créer nouveau document <% SPS>, 55
- Espace réservé de contenus, 60
- fichiers requis, 54
- formats prédéfinis, 72
- générer feuilles de style XSLT, 95
- images, 67
- introduction, 54
- Lignes horizontales, 67
- listes de choix, 82
- modèles globaux, 90
- recense, 82
- rest-of-contents, 90
- Styles CSS, 72
- test Authentic View (éditions Enterprise et Professional), 95
- texte statique, 67

**Types en tant qu'unités de traitement,**

modèles globaux, 234

**Types globaux,**

dans modèles, 234

## U

**Un texte statique,**

dans le tutoriel Démarrage rapide, 67

et échappement de sortie, 331

**URI,**

détenir dans entités non parsées, 363

**Utilisation, 96****Utilité de ligne de commande, 20**

## V

**Variables, 285, 290****Variables d'environnement, 107****Variables d'Environnement utilisées dans les catalogues, 104****Vérificateur orthographique,**

dans StyleVision, 525

**version XPath dans SPS, 99****Version XSLT,**

paramètre pour SPS, 450

**version XSLT dans SPS, 99**

## W

**Windows,**

prise en charge pour les produits Altova, 661

**Word 2007 (uniquement Enterprise Edition), 14, 30****Word 2007+ sortie (Enterprise edition), 102****WordML (uniquement Enterprise Edition), 14, 30**

## X

**XML,**

insérer dans le design, 123

**XMLSpy, 18****XPath,**

situer des nœuds de multiples documents, 289

**XPath 1.0,**

et dates, 385

**XPath 2.0,**

et dates, 385

**XSLT,**

Insérer le fragment de code dans le design, 123

**XSLT à SPS, 365****XSLTelements,**

Insérer comme code dans le design, 122

## Z

**Zone de liste,**

dans le tutoriel Démarrage rapide, 82