# Altova MapForce 2023 Professional Edition



Manuel de l'utilisateur et de référence

### Altova MapForce 2023 Professional Edition Manuel de l'utilisateur et de référence

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

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

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

Published: 2022

© 2016-2022 Altova GmbH

# Table des matières

## 1 Introduction

#### 17

1.1	Nouve	elles fonctions	
	1.1.1	Version 2023	
	1.1.2	Version 2022	
	1.1.3	Version 2021	
	1.1.4	Version 2020	
	1.1.5	Version 2019	
	1.1.6	Version 2018	
1.2	MapFo	orce, c'est quoi?	
	1.2.1	Mappage : Sources et cibles	
	1.2.2	Langages de transformation	
	1.2.3	Scénarios de mappage	
	1.2.4	Intégration avec les produits d'Altova	
1.3	Aperçu de l'interface d'utilisateur		
	1.3.1	Barres	
	1.3.2	Fenêtres	
	1.3.3	Fenêtre de messages	
	1.3.4	Volets	

## 2 **Tutoriels**

2.1	Une sou	urce vers une cible	. 41
	2.1.1	Créer et enregistrer le Design	. 42
	2.1.2	Ajouter composant source	. 43
	2.1.3	Ajouter composant cible	. 45
	2.1.4	Connecter Source et Cible	. 46
	2.1.5	Consulter le résultat de mappage	. 49
2.2	Sources	s multiples vers une cible	. 51
	2.2.1	Préparer les fichiers source	. 52
	2.2.2	Ajouter composant cible	. 53

2.2.3	Vérifier et définir les fichiers d'entrée/de sortie	. 54
2.2.4	Connecter Sources et Cible	. 55
Mappag	es en chaîne	. 58
2.3.1	Préparer le design de mappage	. 58
2.3.2	Configurer la deuxième cible	. 59
2.3.3	Connecter les Cibles	. 61
2.3.4	Filtrer les données	. 61
2.3.5	Prévisualiser et enregistrer la sortie	. 64
Source	s multiples vers cibles multiples	. 65
2.4.1	Préparer le design de mappage	. 67
2.4.2	Configurer l'entrée	. 68
2.4.3	Configurer la Sortie, Partie 1	. 68
2.4.4	Configurer la Sortie, Partie 2	. 71
	2.2.3 2.2.4 Mappag 2.3.1 2.3.2 2.3.3 2.3.4 2.3.5 Sources 2.4.1 2.4.2 2.4.3 2.4.4	<ul> <li>2.2.3 Vérifier et définir les fichiers d'entrée/de sortie</li></ul>

## 3 Notions fondamentales de mappage

3.1	Comp	osants	76
	3.1.1	Ajouter des composants au mappage	78
	3.1.2	Les bases de composant	80
	3.1.3	Chemins de fichier	82
3.2	Conne	exions	87
	3.2.1	Types de connexion	
	3.2.2	Paramètres de connexion	
	3.2.3	Menu contextuel de la connexion	
	3.2.4	Connexions incorrectes	100
	3.2.5	Garder des connexions après avoir supprimé des composants	101
3.3	Procé	dures générales et fonctions	103
	3.3.1	Validation	103
	3.3.2	Généreration de code	105
	3.3.3	Fonctions de Mode Texte	106
	3.3.4	Recherche Mode Texte	109
	3.3.5	Paramètres de mappage	112
3.4	Projet	S	115
	3.4.1	Notions de base du projet	115
	3.4.2	Paramètres de projet	118

4	Com	nposants de structure	120
4.1	XML e	et schéma XML	121
	4.1.1	Paramètres de composant XML	122
	4.1.2	Types dérivés	126
	4.1.3	Valeurs NULL	128
	4.1.4	Commentaires et Instructions de traitement	132
	4.1.5	Sections CDAT A	133
	4.1.6	Caractères génériques - xs:any / xs:anyAttribute	134
	4.1.7	Espaces de noms personnalisés	137
	4.1.8	Gestionnaire de schéma	139
10	Deeee	a de dennées SOI	450

	4.1.8	Gestionnaire de schéma	139
4.2	Bases	de données SQL	156
	4.2.1	Connexion à une source de données	158
	4.2.2	Introduction aux mappages de base de données	248
	4.2.3	Mapper des données dans des bases de données	285
	4.2.4	Joindre des données de base de données	338
	4.2.5	Filtrer et trier les données de base de données (WHERE/ORDER SQL)	355
	4.2.6	Instructions SELECT SQL en tant que tables virtuelles	362
	4.2.7	Mapper des données XML de / vers des champs de base de données	372
	4.2.8	Parcourir et interroger des bases de données	382
	4.2.9	Procédures stockées	396
	4.2.10	Gérer des exceptions de base de données	423
	4.2.11	Traçage de base de données et journalisation d'erreur	435
4.3	Fichiers	s CSV et Texte	446
	4.3.1	Exemple : Mapper des fichiers CSV vers XML	446
	4.3.2	Exemple : Itérer à travers des items	448
	4.3.3	Exemple : Créer des hiérarchies depuis des fichiers texte CSV et de	
		longueur fixe	451
	4.3.4	Configurer les options CSV	455
	4.3.5	FLF à base de données	459
	4.3.6	Configurer les options FLF	463
4.4	Binary	Files	471
	4.4.1	Exemple: Lire des fichiers binaires	473

3.4.3

5	Com	posants de transformation	476
5.1	Entrée	e simple	477
	5.1.1	Ajouter des composants d'entrée simple	478
	5.1.2	Paramètres de composant d'entrée simple	479
	5.1.3	Créer une valeur d'entrée par défaut	481
	5.1.4	Exemple : Utiliser les noms de fichier en tant que paramètres de mappage	482
5.2	Sortie	simple	488
	5.2.1	Ajouter des composants de sortie simples	489
	5.2.2	Exemple : Consulter la sortie de fonction	490
5.3	Les va	ariables	492
	5.3.1	Ajouter des variables	494
	5.3.2	Changer le contexte et l'étendue des variables	498
	5.3.3	Exemple : Compter les lignes de table de base de données	500
	5.3.4	Exemple : Filtre et numéroter les nœuds	501
	5.3.5	Exemple : Grouper et sous-grouper des enregistrements	503
5.4	Joindr	e des données	505
	5.4.1	Ajouter des conditions Join	507
	5.4.2	Joindre trois structures ou plus	510
	5.4.3	Exemple : Joindre des structures XML	511
5.5	Trier le	es données	517
	5.5.1	Trier par clés multiples	519
	5.5.2	Trier par variables	520
5.6	Filtres	et conditions	523
	5.6.1	Exemple : Filtrer des nœuds	525
	5.6.2	Exemple: Retourner une valeur par condition	527
5.7	Value-	Maps	530
	5.7.1	Exemple : Remplacer les jours de la semaine	533
	5.7.2	Exemple : Remplacer des titres de tâche	536
5.8	Regro	uper des données	540
	5.8.1	Exemple : Regrouper les enregistrements par clé	544
5.9	Excep	tions	547
	5.9.1	Exemple : Exception sur condition "Greater Than" (Supérieur à)	548

4.4.2

6	Fond	tions	551
6.1	Notions	s fondamentales de base	
6.2	Gérer I	les bibliothèques de fonction	555
	6.2.1	Bibliothèques locales et globales	557
	6.2.2	Chemins de bibliothèque relatifs	558
6.3	Valeurs	s par défaut et fonctions de nœud	560
	6.3.1	Configuration de la règle	562
	6.3.2	Scénarios Use-Case	565
	6.3.3	Métadonnées de nœud dans des fonctions de nœud	571
6.4	Fonctio	ons définies par l'utilisateur	575
	6.4.1	Notions de bases des FDU	576
	6.4.2	Paramètres UDF	581
	6.4.3	FDU récursives	586
	6.4.4	Implémentation de la consultation	588
6.5	Custor	n Functions	592
	6.5.1	Importer des fonctions XSLT personnalisées	592
	6.5.2	Importer une fonction XQuery 1.0 personnalisée	599
	6.5.3	Importer des bibliothèques Java et .NET personnalisées	604
	6.5.4	Référencer les bibliothèques Java, C# et C++ manuellement	612
6.6	Expres	sions régulières	
6.7	Référe	nce des bibliothèques de fonctions	631
	6.7.1	core   aggregate functions	633
	6.7.2	core   conversion functions	
	6.7.3	core   file path functions	656
	6.7.4	core   generator functions	660
	6.7.5	core   logical functions	
	6.7.6	core   math functions	668
	6.7.7	core   node functions	674
	6.7.8	core   QName functions	679
	6.7.9	core   sequence functions	
	6.7.10	core   string functions	
	6.7.11	db	

5.9.2

6.7.12	lang   datetime functions	
6.7.13	lang   file functions	745
6.7.14	lang   generator functions	747
6.7.15	lang   logical functions	
6.7.16	lang   math functions	
6.7.17	lang   QName functions	
6.7.18	lang   string functions	
6.7.19	xpath2   accessors	
6.7.20	xpath2   anyURI functions	
6.7.21	xpath2   boolean functions	
6.7.22	xpath2   constructors	
6.7.23	xpath2   context functions	
6.7.24	xpath2   durations, date and time functions	
6.7.25	xpath2   node functions	800
6.7.26	xpath2   numeric functions	
6.7.27	xpath2   string functions	
6.7.28	xpath3   external information functions	819
6.7.29	xpath3   formatting functions	
6.7.30	xpath3   math functions	
6.7.31	xpath3   URI functions	
6.7.32	xslt   xpath functions	
6.7.33	xslt   xslt functions	

#### Scénarios de mappage avancé 7

7.1	Mappa	ges en chaîne	
	7.1.1	Exemple : Passage actif	843
	7.1.2	Exemple : Passage inactif	847
7.2	Mapper	r noms de nœud	851
	7.2.1	Obtenir l'accès aux noms de nœud	852
	7.2.2	Accéder aux nœuds de type spécifique	860
	7.2.3	Exemple : Mapper les noms d'élément dans les valeurs d'attribut	864
	7.2.4	Exemple: Regrouper et filtrer les nœuds par nom	867
7.3	Règles	et stratégies de mappage	
	7.3.1	Séquences	873

## 840

	7.3.2	Le contexte de mappage	874
	7.3.3	Contexte de priorité	884
	7.3.4	Composants de cible multiple	889
7.4	Traiter	plusieurs fichiers d'entrée ou de sortie dynamiquement	893
	7.4.1	Mapper plusieurs fichiers d'entrée dans un seul fichier de sortie	896
	7.4.2	Mapper plusieurs fichiers d'entrée dans plusieurs fichiers de sortie	898
	7.4.3	Fournir les noms de fichier en tant que Paramètres de mappage	899
	7.4.4	Consulter plusieurs fichiers de sortie	899
	7.4.5	Exemple : Séparer un fichier XML en plusieurs fichiers	900
	7.4.6	Exemple : Séparer une table de base de données dans plusieurs fichiers	
		XML	901
7.5	Parser	et sérialiser des strings	904
	7.5.1	À propos du composant Parser/Sérialiser	904
	7.5.2	Exemple : sérialiser vers un string (XML vers base de données)	907
7.6	Volets of	de Sortie StyleVision	912
7.7	Génére	r la documentation de mappage	916
	7.7.1	Feuilles de style Power StyleVision prédéfinies	918
	7.7.2	Feuilles de style personnalisées	923

## 8 Débogueur

8.1	Préparation de débogage	930
8.2	À propos du mode Déboguer	931
8.3	Ajouter et supprimer des points d'arrêt	935
8.4	Utiliser la fenêtre Valeurs	937
8.5	Utiliser la fenêtre Contexte	939
8.6	Utiliser la fenêtre Points d'arrêt	941
8.7	Visualiser une sortie partiellement générée	943
8.8	Consulter la valeur actuelle d'un connecteur	944
8.9	Retourner dans le passé récent	945
8.10	Consulter l'historique des valeurs traitées par un connecteur	946
8.11	Configurer le Contexte en une Valeur	947

## 9 Automatiser des mappages et MapForce

948

9.1	Automatisation avec RaptorXML Server	. 949
9.2	Automatisation avec MapForce Server	. 950
9.3	Préparer des mappages pour l'exécution de serveur	. 951
9.4	Compiler des mappages sur des fichiers d'exécution MapForce Server	. 957
9.5	Déployer des mappages sur FlowForce Server	. 960
9.6	Interface de ligne de commande MapForce	. 965

## 10 Ressources globales Altova

10.1	Configuration des Ressources globales, Partie 1		
10.2	Config	uration des Ressources globales, Partie 2	
10.3	Fichier	s XML en tant que Ressources globales	
10.4	Dossie	rs en tant que Ressources globales	
10.5	Bases	de données en tant que Ressources globales	
10.6	Résulta Resso	ats de transformation MapForce et StyleVision comme urces globales	
10.7	Resso	urces globales dans des environnements d'exécution variés	
	10.7.1	Ressources globales dans XSLT, XSLT2, XQuery	
	10.7.2	Ressources globales dans MapForce Server	
	10.7.3	Ressources globales dans FlowForce Server	

## **11 Catalogs in MapForce**

11.1	Comment fonctionnent les catalogues	994
11.2	Structure du catalogue dans MapForce	996
11.3	Personnaliser vos catalogues	998
11.4	Variables d'Environnement	000

#### 12 Plug-in MapForce pour Visual Studio

## **13 Plug-in MapForce pour Eclipse**

13.1	Installer le plug-in MapForce pour Eclipse	1005
13.2	La perspective MapForce	1007
13.3	Accéder aux menus et fonctions communs	1009

# 

## 

13.4	Travaill	er avec des mappages et des projets	. 1013
	13.4.1	Créer un projet MapForce/Eclipse	. 1013
	13.4.2	Créer de nouveaux mappages	. 1015
	13.4.3	Importer des mappages existants dans un projet Eclipse	. 1017
	13.4.4	Configurer un Build et une Génération automatique de code MapForce	. 1020
13.5	Étendre	e le plug-in MapForce pour Eclipse	. 1023

## 14 Commandes de menu

14.1	Fichier		1025
14.2	Édition		1028
14.3	Insérer		1029
14.4	Projet		1032
14.5	Compos	sant	1034
14.6	Connex	ion	1036
14.7	Fonction	n	1037
14.8	Sortie		1038
14.9	Débogu	ier	1040
14.10	Affichag	je	1041
14.11	Outils		1043
	14.11.1	Personnaliser les menus	1044
	14.11.2	Personnaliser les raccourcis de clavier	1045
14.12	Options	; 	1048
	14.12.1	Java	1050
	14.12.2	Base de données	1051
	14.12.3	Proxy de réseau	1056
14.13	Fenêtre		1059
14.14	Aide		1060

Générer du code depuis un mappage...... 1070

Générer du code depuis un projet de mappage...... 1071

## 15 G

15.2.1 15.2.2

or	<b>NOr</b>	atour	do.	CODO	
			ue	LUUE	

1	0	6	5
---	---	---	---

	15.2.3	Construire le projet	1072
	15.2.4	Exécuter l'application	1072
15.3	Généra	ation de code C#	
	15.3.1	Sélectionner la plateforme cible	
	15.3.2	Générer du code depuis un mappage	
	15.3.3	Générer du code depuis un projet de mappage	1077
	15.3.4	Construire les projets .NET Framework	1078
	15.3.5	Construire les projets .NET 5 et .NET Core	1079
15.4	Génére	er du code Java	
	15.4.1	Générer du code depuis un mappage	1082
	15.4.2	Générer du code depuis un projet de mappage	1083
	15.4.3	Construire le projet avec Ant	1084
	15.4.4	Gérer les références JDBC	1085
	15.4.5	Exemple : Générer et exécuter du code Java	1087
15.5	Intégre	r du code généré par MapForce	
	15.5.1	Exemple Java	1096
	15.5.2	Exemple C#	1098
	15.5.3	Exemple C++	1100
	15.5.4	Modifier l'entrée et la sortie par programmation	1101
15.6	Génére	er du code depuis des Schémas XML ou des DTDs	1108
	15.6.1	À propos des Schema Wrapper Libraries (C++)	1111
	15.6.2	À propos des Schema Wrapper Libraries (C#)	1113
	15.6.3	À propos des Schema Wrapper Libraries (Java)	1116
	15.6.4	Intégrer des Bibliothèques Schema Wrapper	1118
	15.6.5	Exemple: Bibliothèques de livres	1121
	15.6.6	Exemple: Purchase Order	1145
15.7	Référe	nce aux classes générées (C++)	1153
	15.7.1	altova::DateTime	1153
	15.7.2	altova::Duration	1156
	15.7.3	altova::DayTimeDuration	1158
	15.7.4	altova::YearMonthDuration	1159
	15.7.5	altova::meta::Attribute	1159
	15.7.6	altova::meta::ComplexType	1160
	15.7.7	altova::meta::Element	1161
	15.7.8	altova::meta::SimpleType	1161

	15.7.9	[YourSchema]::[CDoc]	1163
	15.7.10	[YourSchema]::[ElementType]	1165
	15.7.11	[YourSchema]::MemberAttribute	1166
	15.7.12	[YourSchema]::MemberElement	1167
15.8	Référer	nce aux classes générées (C#)	1168
	15.8.1	Altova.Types.DateTime	1168
	15.8.2	Altova.Types.DateTimeFormat	1171
	15.8.3	Altova.Types.Duration	1172
	15.8.4	Altova.Xml.Meta.Attribute	1175
	15.8.5	Altova.Xml.Meta.ComplexType	1176
	15.8.6	Altova.Xml.Meta.Element	1176
	15.8.7	Altova.Xml.Meta.SimpleType	1177
	15.8.8	[YourSchema].[Doc]	1178
	15.8.9	[YourSchema].[ElementType]	1180
	15.8.10	[YourSchemaType].MemberAttribute	1181
	15.8.11	[YourSchemaType].MemberElement	1182
15.9	Référer	nce aux classes générées (Java)	1184
	15.9.1	com.altova.types.DateTime	1184
	15.9.2	com.altova.types.Duration	1189
	15.9.3	com.altova.xml.meta.Attribute	1192
	15.9.4	com.altova.xml.meta.ComplexType	1193
	15.9.5	com.altova.xml.meta.Element	1193
	15.9.6	com.altova.xml.meta.SimpleType	1194
	15.9.7	com.[YourSchema].[Doc]	1195
	15.9.8	com.[YourSchema].[ElementType]	1196
	15.9.9	com.[YourSchema].[YourSchemaType].MemberAttribute	1198
	15.9.10	com.[YourSchema].[YourSchemaType].MemberElement	1198
15.10	Astuces	s de Génération de Code	1200
15.11	Options	s du Générateur de Code	1201
15.12	Référer	nce SPL	1203
	15.12.1	Structure SPL de base	1203
	15.12.2	Déclarations	1204
	15.12.3	Variables	1206
	15.12.4	Variables prédéfinies	1207
	15.12.5	Créer des fichiers de sortie	1208

15.12.6	Opérateurs	1210
15.12.7	Conditions	
15.12.8	Collections et foreach	
15.12.9	Sous-routines	
15.12.10	Types intégrés	

## 16 L'API de MapForce

16.1	Accéde	er à l'API	
16.2	Le mod		
16.3	Gestior	n d'erreur	
16.4	Exemp	le Projet C#	
16.5	Exemp	le Projet Java	1233
16.6	Exemp	les JScript	1237
	16.6.1	Démarrer Application	
	16.6.2	Accès simple aux documents	
	16.6.3	Générer le code	
	16.6.4	Générer le code (alternatif)	
	16.6.5	Exécuter un mappage	
	16.6.6	Project Tasks	
16.7	Référe	nce d'objet	1251
	16.7.1	Interfaces	
	16.7.2	Énumérations	

17 Act	tiveX	Integ	ration
--------	-------	-------	--------

17.1	Prereq	uisites	1406
17.2	Adding	the ActiveX Controls to the Toolbox	1408
17.3	Integra	tion at Application Level	1410
17.4	Integra	tion at Document Level	1413
17.5	Active>	< Integration Examples	1417
	17.5.1	C#	1417
	17.5.2	HTML	1424
	17.5.3	Java	1431
	17.5.4	VB.NET	1441

17.6	Comma	and Reference	1444
	17.6.1	"File" Menu	1444
	17.6.2	"Edit" Menu	1445
	17.6.3	"Insert" Menu	1446
	17.6.4	"Project" Menu	1446
	17.6.5	"Component" Menu	1447
	17.6.6	"Connection" Menu	1448
	17.6.7	"Function" Menu	1449
	17.6.8	"Output" Menu	1449
	17.6.9	"Debug" Menu	1450
	17.6.10	"View" Menu	1450
	17.6.11	"Tools" Menu	1451
	17.6.12	"Window" Menu	1452
	17.6.13	"Help" Menu	1452
17.7	Object	Reference	1453
	17.7.1	MapForceCommand	1453
	17.7.2	MapForceCommands	1455
	17.7.3	MapForceControl	1456
	17.7.4	MapForceControlDocument	1464
	17.7.5	MapForceControlPlaceHolder	1470
	17.7.6	Enumerations	1472

## 18 Annexes

18.1	Notes c	le prise en charge	1475
	18.1.1	Sources et cibles prises en charge	1475
	18.1.2	Fonctions prises en charge dans le code généré	1476
18.2	Informa	tion des moteurs	1479
	18.2.1	Informations concernant le moteur XSLT et XQuery	1479
	18.2.2	Fonctions XSLT et XPath/XQuery	1485
18.3	Donnée	es techniques	1565
	18.3.1	SE et exigences de mémoire	1565
	18.3.2	Moteurs Altova	1565
	18.3.3	Prise en charge Unicode	1566
	18.3.4	Utilisation Internet	1566

tions de licence	1568
Distribution électronique de logiciel	1568
Activation de logiciel et le license metering	1569
Altova Contrat de licence de l'utilisateur final	1570
1	tions de licence Distribution électronique de logiciel Activation de logiciel et le license metering Altova Contrat de licence de l'utilisateur final

## Index

## 1 Introduction

**MapForce® 2023 Professional Edition** est un outil de mappage de données visuel pour les projets d'intégration des données avancées . MapForce® est une application Windows 32/64-bit qui est exécutée sur Windows 7 SP1 avec mise à jour de la plateforme, Windows 8, Windows 10, Windows 11, et Windows Server 2008 R2 SP1 avec mise à jour de la plateforme ou plus récent. La prise en charge 64-bit est disponible pour les éditions Enterprise et Professional. MapForce s'intègre aussi avec Visual Studio et Eclipse, ainsi qu'avec les produits Microsoft Office. Pour plus d'information, voir <u>Notes de prise en charge</u> <sup>(475)</sup>.



Dernière mise à jour : 07.10.2022

## 1.1 Nouvelles fonctions

Cette section décrit les nouvelles fonctions de chaque release de MapForce. Pour plus de détails, veuillez voir la sous-section respective.

## 1.1.1 Version 2023

#### Version 2023

- Une prise en charge pour les thèmes suivants a été ajoutée : Classique, Clair et Sombre. Pour plus d'information, voir <u>Fenêtre</u><sup>1059</sup>.
- Mises à jour internes et optimisations.
- La prise en charge d'Eclipse a été mise à jour et couvre maintenant les versions suivantes : 2022-09, 2022-06, 2021-03, 2020-12 (éditions Professional et Enterprise). Pour plus de détails, voir <u>Plug-in de MapForce pour Eclipse</u>
- Prise en charge des message pour ODETTE EDI (Enterprise Edition).
- Prise en charge de <u>XII Transformation Registry 5 Specification</u> (*Enterprise Edition*).
- Il est désormais possible de créer des paramètres UDF<sup>682</sup> basés sur base de données et de variables<sup>493</sup> avec une arborescence de tables associées (éditions Professional et Enterprise).
- Il est désormais possible d'envoyer une structure de requête application/x-www-form-urlencoded au service REST (*Enterprise Edition*).
- Prise en charge des Répertoires UN/EDIFACT D.21B et D.22A Directories (Enterprise Edition).
- Prise en charge pour SQLite 3.39.2, MariaDB 10.9.2 et PostgreSQL 14.5 (éditions Professional et Enterprise). Pour en savoir plus sur toutes les bases de données prises en charge, voir <u>Bases de données</u><sup>156</sup>.
- Prise en charge du <u>Gestionnaire de schéma XML</u><sup>(133)</sup> est un outil qui propose un moyen centralisé d'installer et de gérer des schémas XML pour une utilisation sur toutes les applications activées par XBRL d'Altova.
- Prise en charge des délimiteurs EDI mappables (*Enterprise Edition*). La fonction est actuellement prise en charge pour les normes EDI suivantes : EDIFACT, X12 et NCPDP SCRIPT.

## 1.1.2 Version 2022

#### Version 2022 Release 2

- Mises à jour et optimisations internes
- La prise en charge d'Eclipse a été mise à jour et couvre maintenant les versions suivantes : 2021-12, 2021-09; 2021-06; 2021-03 (éditions Professional et Enterprise). Pour des détails, voir <u>Plug-in de MapForce pour Eclipse</u>
- Prise en charge de Visual Studio 2022 dans MapForce Plug-in pour Visual Studio et génération de code (éditions Professional et Enterprise). Pour plus d'informations, voir <u>MapForce Plug-in pour Visual Studio</u> et <u>génération de code</u>.
- Prise en charge de .NET 6.0 dans génération de code (éditions Professional et Enterprise). Pour les détails, voir <u>génération de code</u><sup>1006</sup>.

- De nouvelles versions de base de données sont prises en charge : PostgreSQL 14, SQLite 3.37.2, MariaDB 10.6.5, MySQL 8.0.28, IBM DB2 11.5.7 (*éditions Professional et Enterprise*). Pour en savoir plus sur toutes les bases de données prises en charge, voir <u>Bases de données</u><sup>156</sup>.
- Il est désormais possible de prévisualiser des images dans la fenêtre **Projet**(*éditions Professional et Enterprise*). Pour plus d'informations, voir <u>Notions de base de projet</u><sup>(17)</sup>.
- Il est désormais possible de créer des indicateurs de classement pour des composants XBRL cible (*Enterprise Edition*) conformes à EBA.

### Version 2022

- Mises à jour et optimisations internes
- La prise en charge d'Eclipse a été mise à jour et couvre maintenant les versions suivantes : 2021-09; 2021-06; 2020-03; 2020-12 (éditions Professional et Enterprise). Pour des détails, voir <u>Plug-in de</u> <u>MapForce pour Eclipse</u>
- <u>Copy-all connections</u> prend désormais en charge JSON. Cette fonction est uniquement disponible pour les types JSON compatibles (*Enterprise Edition*).
- Un nouveau volet de sortie StyleVision dénommé *Texte* a été introduit. Si un fichier SPS est annexé à un composant, le nouveau format texte brut peut être vu dans MapForce (*éditions Professional et Enterprise*). Pour plus d'informations, voir <u>Volets de sortie StyleVision</u><sup>912</sup>.
- Prise en charge de JSON Schema dans les variables 42 et paramètres UDE 61 (Enterprise Edition).
- Prise en charge pour bases de données NoSQL : MongoDB et CouchDB (*Enterprise Edition*). Pour en savoir plus sur toutes les bases de données prises en charge, voir <u>Bases de données</u><sup>(156)</sup>.
- Une nouvelle bibliothèque de fonction bson est désormais disponible, qui vous permet de créer et manipuler quelques-uns des types BSON (*Enterprise Edition*).
- Prise en charge des Répertoires UN/EDIFACT D.20B et D.21A.
- Prise en charge de SWIFT 2021.

## 1.1.3 Version 2021

#### Version 2021 Release 3

• Prise en charge pour le nouveau JSON Schema <u>Draft 2019-09</u> et <u>Draft 2020-12</u> (*uniquement Enterprise Edition*).

#### Version 2021 Release 2

- XSLT 3.0 est désormais pris en charge en tant que langage de mappage. Voir <u>Générer le code</u> XSLT<sup>105</sup>. MapForce inclut désormais également de nouvelles fonctions built-in qui sont prises en charge quand le langage de mappage est XSLT 3.0. Pour plus d'informations, voir <u>Référence des</u> <u>bibliothèques de fonctions</u><sup>631</sup>.
- Lorsque vous générez le code C#, vous pouvez choisir .NET Core 3.1 et .NET 5.0 en tant qu'infrastructures cibles depuis les <u>options de génération de code</u><sup>(20)</sup> (ceci s'ajoute à la prise en charge existante pour les projets .NET Framework). Pour les détails, voir <u>Générer le code C#</u><sup>(07)</sup>.
- Mises à jour internes et optimisations.

- Un mappage MapForce peut lire des données BLOB (binary large object) depuis des fichiers binaires et écrire des fichiers binaires sur le disque. Cela permet, par exemple de lire des champs BLOB depuis une base de données et de les enregistrer en tant que fichiers d'image sur le disque, ou de lire des fichiers binaires comme des PDF depuis le disque et de les enregistrer en tant que des champs xs:base64Binary dans le cadre d'un fichier XML. Voir <u>Fichiers binaires</u> <sup>471</sup> pour plus d'informations.
- De nouvelles versions de base de données sont prises en charge : MariaDB 10.4, 10.5
- De nouvelles versions d'Eclipse sont prises en charge : 2019.09, 2019.12, 2020.03, 2020.06
- Lorsque vous réunissez plusieurs tables ou modes de base de données en utilisant des composants « join » SQL dans un mappage, vous pouvez définir le mode « join » soit en tant que LEFT OUTER JOIN soit INNER JOIN, voir <u>Modifier le Mode Join</u><sup>340</sup>.
- Mises à jour internes et optimisations.

## 1.1.4 Version 2020

#### Version 2020 Release 2

- Une nouvelle fenêtre <u>Gérer la fenêtre des bibliothèques</u><sup>(33)</sup> est disponible qui vous permet de consulter et de gérer toutes les bibliothèques de fonction importées au niveau de document et de programme (cela comprend les fonctions définies par MapForce et d'autres types de bibliothèques). Cela vous permet, par exemple, de copier-coller aisément des fonctions définies par l'utilisateur d'un mappage à un autre, voir <u>Copier-coller des UDF entre les mappages</u><sup>(33)</sup>.
- Lorsqu'un fichier de mappage importe des bibliothèques, le chemin des fichiers de bibliothèque importée est relatif au fichier de mappage par défaut, voir <u>Chemins de bibliothèque relatifs</u>. Vous pouvez toujours importer des mappages au niveau de l'application, comme dans les releases précédentes, mais dans ce cas, le chemin de bibliothèque est toujours absolu.
- Si un fichier de mappage importe des bibliothèques XSLT ou XQuery, vous pouvez générer un code XSLT ou XQuery qui référence les fichiers de bibliothèque importés en utilisant un chemin relatif. La nouvelle option est disponible dans le dialogue <u>Paramètres de mappage</u><sup>(112)</sup>.
- <u>L'API MapForce</u><sup>(22)</sup> a été amélioré avec de nouveaux membres qui vous permettent de gérer des bibliothèques importées de manière programmatique (par exemple les ajouter ou les supprimer).
- Le code généré pour des bibliothèques de wrapper de schéma XML propose maintenant un plus grand contrôle sur les espaces de noms d'élément et des préfixes. De nouvelles méthodes sont disponibles pour déclarer ou contourner des espaces de noms pour un élément, ou pour apposer un élément avec un espace de noms préfixé. Voir <u>Exemple : Purchase Order</u> Le Control de la control de
- De nouvelles versions de base de données sont prises en charge : PostgreSQL 12.1 et Informix 14.10.
- Mises à jour et optimisations internes

- Prise en charge pour Visual Studio 2019 dans <u>MapForce Plug-in pour Visual Studio</u><sup>(100)</sup> et <u>génération</u> <u>de code</u><sup>(106)</sup>.
- Prise en charge d'Eclipse 4.9 4.12, voir <u>MapForce Plug-in pour Eclipse.</u>
- Si un package Oracle contient des procédures ou des fonctions publiques stockées, celles-ci sont aussi disponibles dans le mappage, voir <u>Ajouter des procédures stockées dans le mappage</u><sup>339</sup>.
- Vous pouvez configurer un composant de base de données de manière à ce que les noms d'objet de base de données soient traités en tant que relatifs par rapport au schéma par défaut, mais non liés à un schéma particulier. Cela vous permet d'économiser du temps si vous avez besoin de passer à une base de données à l'avenir, voir <u>Changer de base de données et de schémas</u><sup>257</sup>.

- Vous pouvez déployer des Ressources globales créées dans MapForce vers FlowForce Server, voir <u>Déployer des Ressources globales dans FlowForce Server</u><sup>(90)</sup>.
- Lorsque vous souhaitez remplacer des valeurs avec une table de consultation, vous pouvez coller des données tabulaires (paires key-value) depuis des sources externes comme CSV ou Excel dans le mappage. De même, il est plus facile de gérer des cas lorsqu'une valeur n'est pas trouvée dans la table de consultation prédéfinie, le traitement de ce type de valeurs ne nécessite plus l'utilisation de la fonction substitute-missing. Voir <u>Utiliser Value-Maps</u><sup>600</sup>.
- Mises à jour et optimisations internes

## 1.1.5 Version 2019

#### Version 2019 Release 3

- La plus grande partie de l'interface graphique de l'utilisateur est maintenant optimisée pour les écrans à densité de pixels élevée (HiDPI)
- Prise en charge du <u>Traçage de base de données et archivage d'erreurs</u>
   <sup>435</sup>
- Meilleure prise en charge de la gestion de transaction de base de données dans le code de programme généré et l'exécution de serveur. Si une erreur se produit lorsqu'un mappage met à jour une base de données, il est possible de continuer l'exécution de mappage même si certaines opérations ont échoué, voir <u>Gérer les exceptions de base de données</u><sup>423</sup>.
- Un mappage peut encoder ou décoder des données Base64 à l'aide des nouvelles fonctions <u>charset-encode</u><sup>762</sup> et <u>charset-decode</u><sup>760</sup>.
- De nouvelles versions de base de données sont prises en charge : PostgreSQL 11, Db2 pour i 7.2 et 7.3
- Prise en charge d'Eclipse 4.7 4.10, voir <u>Plug-in MapForce pour Eclipse</u><sup>1004</sup>
- Prise en charge des paramètres explicites du chemin du Java Virtual Machine depuis MapForce, voir <u>Paramètres Java</u><sup>(105)</sup>
- Mises à jour et optimisations internes

- De nouvelles versions de base de données sont prises en charge : Firebird 3.0, Informix 12.10, MariaDB 10.2, MySQL 8.0, PostgreSQL 10, SQL Server 2017, SQL Server pour Linux (s'ajoute à la prise en charge pour les versions précédentes)
- Vous pouvez appliquer de manière sélective une valeur par défaut ou une fonction du nœud qui correspond à vos critères personnalisés. Par exemple, vous pouvez appliquer une fonction de nœud uniquement aux nœuds de type spécifique dont le nom correspond à une expression régulière, voir <u>Appliquer des fonctions de nœud et des défauts aux conditions</u>
- Lors de la création de fonctions de nœuds, vous pouvez traiter les données de manière conditionnelles sur la base des métadonnées comme le nom de nœud ou l'annotation de nœud, voir <u>Fournir des métadonnées de nœud aux fonctions de nœud</u><sup>571</sup>.
- Les mappages qui contiennent des fonctions de regroupement sont exécutés bien plus rapidement dans un code généré C#, C++, ou Java
- Mises à jour et optimisations internes

## 1.1.6 Version 2018

#### Version 2018 Release 2

- Prise en charge pour les bases de données suivantes : MariaDB 10.2, Teradata 16
- Il est possible d'ajouter au mappage facilement des fonctions intégrées, des fonctions définies par l'utilisateur et des constantes en double-cliquant sur une zone vide dans le mappage (voir <u>Notions</u> <u>fondamentales des fonctions</u><sup>652</sup>).
- Mises à jour et optimisations internes

- Prise en charge de la génération de code de programmation pour Visual Studio 2013, 2015 et 2017, voir <u>Générateur de Code</u>
- Prise en charge des versions de base de données suivantes : Sybase ASE 16, PostgreSQL 9.6, MySQL 5.7
- Mises à jour et optimisations internes

## 1.2 MapForce, c'est quoi?

#### Site web d'Altova : Source de connées

MapForce est un outil puissant et flexible qui permet un mappage graphique any-to-any de différents formats de données. Voir <u>Mappage : Sources et Cibles</u> <sup>24</sup> pour une liste complète de formats de données disponibles. MapForce vous permet de mapper une source vers une cible, une source vers de multiples cibles, de multiples sources vers une cible, ou de multiples sources vers de multiples cibles. Pour en savoir plus sur des scénarios de mappage, voir <u>Scénarios de mappage</u> <sup>26</sup> et <u>Tutoriels</u> <sup>40</sup>. MapForce fournit également une plage complète de traitement de données et options de filtrage, telles que les <u>fonctions</u> <sup>551</sup>, <u>variables</u> <sup>492</sup>, <u>filtres et</u> <u>conditions</u> <sup>552</sup>, qui vous permet de manipuler vos données.

Afin de pouvoir réaliser un mappage, vous pouvez fournir une structure de données qui décrit la structure de chacun de vos fichiers source et cible. Par exemple, un schéma XML définit la structure d'un document XML. Le mappage (de la source à la cible) est réalisé par le biais d'une interface utilisateur graphique glisserdéplacer. Vous n'êtes pas obligé d'écrire un code de programme pour le mappage. MapForce vous génère le code. Vous pouvez ensuite utiliser ce code pour transformer des documents qui sont dotés de la structure de données source en documents ayant la structure de données cible.

### Modèle abstrait

Le modèle abstrait ci-dessous illustre un des scénarios de base d'une transformation de données dans MapForce. La première étape montre qu'une structure abstraite dénommée **sourceschema.xsd** est mappée avec une autre structure abstraite dénommée **outputSchema.xsd**. Le mappage génère le code de transformation dans le langage de transformation sélectionné. La deuxième étape montre que le contenu du fichier XML source, qui est basé sur **sourceschema.xsd**, est mappé avec le fichier XML cible, qui est basé sur **outputSchema.xsd**. Le mappage du contenu du fichier source vers le fichier cible est réalisé par le code de transformation généré à l'étape précédente.



Toutes les éditions de MapForce sont disponibles en tant qu'applications 32-bit. Les éditions MapForce Professional et Enterprise sont également disponibles en tant qu'applications 64-bit.

#### Conventions

Les fichiers de mappage illustrés ou référencés dans le manuel peuvent être trouvés sous les emplacements suivants :

- C:\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples
- C:\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\Tutorial
- C:

\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\Tutorial\BasicTutoria ls

#### Dans cette section

Cette section est organisée dans les rubriques suivantes :

- <u>Mappage : Sources et cibles</u><sup>24</sup> ;
- Langages de transformation<sup>25</sup>;
- <u>Scénarios de mappage</u><sup>26</sup>;
- Intégration avec les produits d'Altova<sup>28</sup>

## 1.2.1 Mappage : Sources et cibles

Dans MapForce, les termes *source* et *cible* sont des termes essentiels qui se réfèrent à des structures de données desquels ou vers lesquels les données sont mappées, respectivement. Les technologies qui peuvent être utilisées en tant que sources et cibles de mappage sont recensées ci-dessous.

#### MapForce Basic Edition

• XML et Schéma XML.

MapForce Professional Edition

- XML et Schéma XML ;
- Fichiers plats, y compris des valeurs séparées par virgule (CSV) et un format de champ à longueur fixe (FLF);
- Bases de données : toutes les bases de données relationnelles, y compris Microsoft Access et les bases de données SQLite
- Fichiers binaires (contenu BLOB brut).

MapForce Enterprise Edition

- XML et Schéma XML ;
- Fichiers plats, y compris des valeurs séparées par virgule (CSV) et un format de champ à longueur fixe (FLF);
- Les données provenant des fichiers de texte de legacy peuvent être mappées et converties sur d'autres formats avec MapForce FlexText ;
- Bases de données SQL : toutes les bases de données relationnelles, y compris Microsoft Access et les bases de données SQLite ;
- Bases de données NoSQL ;
- Fichiers binaires (contenu BLOB brut) ;
- Famille EDI des formats, y compris UN/EDIFACT, ANSI X12, HL7, IATA PADIS, SAP IDoc, TRADACOMS;

- Fichiers JSON ;
- Microsoft Excel 2007 et les fichiers ultérieurs ;
- Fichiers d'instance XBRL et taxonomies ;
- Protocol Buffers.

## **1.2.2** Langages de transformation

Dans MapForce, un langage de transformation est utilisé pour générer le code de transformation qui effectue les mappages. Vous pouvez sélectionner/modifier un langage de transformation à tout moment. MapForce permet la visualisation du code de transformation dans la langue sélectionnée. Pour plus d'informations, voir <u>Générer et visualiser le code XSLT</u><sup>105</sup>. Vous pouvez également générer le code par le biais de la commande de menu **Fichier | Générer code dans** et utilisez ce code pour la transformation de tout document de données valide conformément au schéma de composant de source. Dépendant de l'édition MapForce, vous pouvez choisir la langue préférée pour vos transformations de données comme suit :

MapForce Basic Edition	MapForce Professional et Enterprise Edition
<ul> <li>XSLT 1.0</li> <li>XSLT 2.0</li> <li>XSLT 3.0</li> </ul>	<ul> <li>XSLT 1.0</li> <li>XSLT 2.0</li> <li>XSLT 3.0</li> <li>BUILT-IN</li> <li>XQuery</li> <li>Java</li> <li>C#</li> <li>C++</li> </ul>

Pour sélectionner un langage de transformation, suivez les étapes suivantes :

- Dans le menu Sortie, cliquez sur le nom que vous souhaitez utiliser pour la transformation.
- Cliquez sur le nom du langage dans la barre d'outils Sélection du langage (affichée ci-dessous).



Lorsque vous changez de langage de transformation du mappage, certaines fonctions de MapForce ne sont pas prises en charge pour ce langage. Pour plus d'information, voir <u>Notes de prise en charge</u><sup>1475</sup>.

Pendant que vous concevez ou visualisez des mappages, MapForce valide l'intégrité de vos schémas ou transformations. Si une erreur de validation devait apparaître, MapForce l'affiche dans <u>la fenêtre des</u> <u>Messages</u><sup>35</sup>. Ceci est très utile car vous pouvez immédiatement la réviser et corriger ces erreurs.

#### Langages de transformation dans les éditions MapForce Professional et Enterprise

Lorsque vous choisissez un langage de transformation Java, C# ou C++, MapForce génère les projets et les solutions requis de manière à ce que vous puissiez les ouvrir directement dans Visual Studio ou Eclipse. Pour des scénarios d'intégration de données avancés, vous pouvez aussi étendre le programme généré avec votre propre code, en utilisant les bibliothèques Altova et l'API MapForce.

#### <u>BUILT-IN</u>

Quand vous sélectionnez BUILT-IN comme langage de transformation pour votre mappage, MapForce utilise son moteur de transformation natif pour exécuter le mappage de données. MapForce utilise également cette option de manière implicite à chaque fois que vous visualisez la sortie d'un mappage où le langage de transformation est Java, C#, ou C++.

Le moteur BUILT-IN exécute les mappages sans avoir besoin de processeurs externes, ce qui pourrait être un bon choix si l'utilisation de la mémoire vous cause des soucis. Si vous n'avez pas besoin de générer le code de programme dans un langage spécifique, utilisez BUILT-IN comme option par défaut car il prend en charge la plupart des fonctions de MapForce comparé à d'autres langages (voir <u>Notes de prise en charge</u><sup>(1776)</sup>). De plus, si vous sélectionnez BUILT-IN comme langage de transformation, vous serez en mesure d'automatiser le mappage avec MapForce Server. Pour plus d'information, voir <u>Automatiser les tâches de MapForce avec les produits d'Altova</u><sup>(1176)</sup>.

## 1.2.3 Scénarios de mappage

Les scénarios peuvent être différents selon les critères suivants : (i) sources et cibles, et (ii) complexité de mappages. Différentes structures de données peuvent être utilisées en tant que sources et cibles : par exemple, Schéma XML, un fichier XML avec un schéma assigné, des bases de données, etc. Pour en savoir plus sur les formats acceptables des sources et cibles, voir <u>Mappage : Sources et cibles</u><sup>24</sup>.

La complexité des designs de mappage est illustrée dans les scénarios suivants mais pas limitée à ceux-ci :

- Mapper une source vers une cible Pour plus d'informations sur ce type de mappage, voir <u>Tutoriel 1</u><sup>41</sup>.
- Fusionner des sources de données multiples dans une cible. Pour plus d'informations, voir <u>Tutoriel</u>
   2<sup>51</sup>.
- Filtrer les données de telle manière que seul un sous-ensemble des données en question est mappé à un fichier cible. Voir le <u>Tutoriel 3</u><sup>58</sup>.
- Mapper la structure et le contenu de la source au fichier cible. Voir le <u>Tutoriel 4</u><sup>65</sup>.

Peu importe la technologie avec laquelle vous travaillez, MapForce détermine généralement automatiquement la structure de vos données ou suggère la fourniture d'un schéma pour vos données. MapForce peut également générer des schémas depuis un exemple de fichier d'instance. Par exemple, si avez un fichier d'instance XML mais pas de définition de schéma, MapForce peut la générer pour vous. MapForce rend donc les données à l'intérieur des fichiers XML disponibles pour le mappage vers d'autres fichiers ou formats. Pour en savoir plus sur les termes et fonctions de MapForce, voir <u>Tâches de base</u><sup>74</sup> et <u>Aperçu de l'interface d'utilisateur</u><sup>30</sup>.

Pour un accès et une gestion plus faciles, vous pouvez organiser les designs de mappage dans les projets de mappage. Cette fonction est disponible dans les éditions MapForce Professional et Enterprise. En plus de la génération de code pour les mappages individuels au sein du projet, vous pouvez générer le code de programme depuis des projets entiers.

Dans MapForce, vous pouvez entièrement personnaliser non seulement l'apparence de l'environnement de développement, mais aussi les différents paramètres ayant trait à chaque technologie et à chaque type de composant de mappage. Par exemple :

 Lors d'un mappage de ou vers XML, vous pouvez choisir (i) d'inclure une référence de schéma, ou (ii) si la déclaration XML doit être supprimée dans les fichiers XML de sortie. Vous pouvez aussi choisir l'encodage des fichiers générés (par exemple, UTF-8).

- Lorsque vous mappez vers et depuis des bases de données, vous pouvez définir des paramètres tels que la période time-out pour exécuter des instructions de bases de données. Il est également possible de choisir si MapForce doit utiliser des transactions de base de données, ou s'il doit découper le nom de schéma de la base de données des noms de table lorsqu'il génère le code.
- Dans le cas de XBRL, vous pouvez sélectionner les modes de structure que MapForce doit afficher : le mode Bases de lien de présentation et définition, le mode Lien de base de Table, ou le mode Tous les concepts.

Les exemples ci-dessous illustrent les designs de mappage qui utilisent les mêmes types de source et structures de cible (*Exemple 1*) et des différents (*Exemple 2*). Les deux exemples de mappage sont simples dans le sens que seule une source et une cible sont utilisées.

#### Exemple 1 : mappage XML

Avec MapForce, vous concevez toutes les transformations de mappage visuellement. Par exemple, dans le cas d'XML, vous pouvez connecter chaque élément et attribut dans un fichier XML à un élément ou attribut d'un autre fichier XML. Donc, vous donnez l'instruction à MapForce de lire des données depuis l'élément source (ou l'attribut) et de l'écrire dans l'élément cible (ou l'attribut). La transformation d'un fichier XML vers un autre fichier XML est illustrée ci-dessus :



#### Exemple 2 : Mappage de base de données

Lors du travail avec des bases de données dans les éditions MapForce Professional ou Enterprise, vous pouvez voir n'importe quelle colonne de base de données dans la zone de mappage MapForce et mapper des données vers ou depuis celle-ci en établissant des connexions visuelles. Comme c'est le cas avec d'autres produits d'Altova MissionKit, lors de la configuration d'une connexion de base de données provenant de MapForce, vous pouvez choisir le pilote de base de données de manière flexible et le type de connexion (ADO, ADO.NET, ODBC ou JDBC) conformément à votre infrastructure existante et les besoins de mappage de données. De plus, vous pouvez générer visuellement des requêtes SQL, utiliser les procédures stockées, ou requérir une base de données directement (la prise en charge varie selon le type de base de données, d'édition et de pilote). Un exemple de transformation de données depuis une base de données dans un fichier XML est donné ci-dessous :



## **1.2.4** Intégration avec les produits d'Altova

Les transformations peuvent être exécutées au sein de MapForce en utilisant les moteurs built-in XSLT/XQuery. MapForce peut également être utilisé en tandem avec d'autres produits d'Altova (*voir ci-dessous*).

#### <u>XMLSpy</u>

Si <u>XMLSpy</u> est installé sur le même appareil, vous pouvez ouvrir et éditer tout type de fichier pris en charge en ouvrant XMLSpy directement depuis les contextes spécifiques de MapForce. Par exemple, la commande de menu **Composant | Éditer une définition de schéma dans XMLSpy** est disponible quand vous cliquez sur une composante XML.

#### RaptorXML Server

Vous pouvez choisir d'exécuter la XSLT générée directement dans MapForce et de consulter le résultat de transformation des données immédiatement. Si vous nécessitez une performance accrue, vous pouvez traiter le mappage en utilisant <u>RaptorXML Server</u>, un moteur de transformation XML extrêmement rapide.

#### MapForce Serveur (Éditions Enterprise et Professional)

Vous pouvez automatiser les tâches de MapForce avec l'aide de <u>Altova MapForce Server</u>, qui peut être installé sur Windows, Linux et les systèmes macOS. MapForce Server vous permet d'exécuter les transformations spécifiées dans un mappage, non seulement depuis la ligne de commande du système d'exploitation respectif mais aussi à travers les appels d'API (.NET, COM, Java).

#### FlowForce Serveur (Éditions Enterprise et Professional)

Vous pouvez également automatiser les tâches de MapForce avec l'aide de <u>Altova FlowForce Server</u>, qui peut être installé sur Windows, Linux et les systèmes macOS. FlowForce Server vous permet d'exécuter des tâches de MapForce Server conformément à un calendrier.

#### StyleVision (Éditions Enterprise and Professional)

Avec l'aide de <u>StyleVision</u>, vous pouvez concevoir ou réutiliser des feuilles de style StyleVision Power existantes et consulter le résultat des transformations de mappage en tant que documents HTML, RTF, PDF ou Word 2007+.

#### MapForce en tant que plug-in

Les éditions MapForce Professional et Enterprise peuvent être installées en tant que plug-in des environnements de développement intégrés de Visual Studio et Eclipse. Ainsi, vous pouvez concevoir des mappages et obtenir l'accès aux fonctions MapForce sans quitter votre environnement de développement préféré.

Pour plus d'information sur l'automatisation des tâches, voir <u>Automatiser les tâches de MapForce avec les</u> produits d'Altova<sup>948</sup>. Pour en savoir plus sur l'utilisation de MapForce en tant que plug-in, voir <u>Plug-in pour</u> <u>Visual Studio</u><sup>1009</sup> et <u>Plug-in pour Eclipse</u><sup>1009</sup>.

## **1.3** Aperçu de l'interface d'utilisateur

L'interface graphique d'utilisateur de MapForce est organisée comme un environnement de développement intégré. Les principaux composants d'interface sont illustrés ci-dessous. Vous pouvez changer les paramètres d'interface en utilisant la commande de menu **Outils | Personnaliser**. Utilisez les Vue boutons affichés dans le coin du haut à droite de chaque fenêtre pour les afficher, masquer, épingler ou ancrer. Si vous devez rétablir les barres d'outils et les fenêtres à leur état par défaut, utilisez la commande de menu **Outils | Restaurer les barres d'outils et les fenêtres**.

L'image ci-dessous illustre les parties principales de l'interface utilisateur graphique de MapForce.



Pour plus d'information sur les fonctionnalités et fonctions de chaque partie, voir la page respective ci-dessous.

#### Dans cette section

Cette section est organisée comme suit :

- <u>Barres</u><sup>31</sup>;
- Fenêtres 31 :
- Fenêtre de messages<sup>35</sup> :
- Volets 36

## 1.3.1 Barres

Cette rubrique donne un aperçu des barres disponibles.

#### Barre de menu et barre d'outils

La barre de **menu** affiche les items de menu. Chaque barre d'outils affiche un groupe de touches représentant les commandes de MapForce. Vous pouvez repositionner les barres d'outils en glissant leur poignée à l'endroit désiré. La capture d'écran ci-dessous illustre la barre de **menu** et les barres d'outils. L'interface actuelle dépend de votre édition de MapForce et des paramètres que vous choisissez.

Is <u>F</u> ichier	<u>É</u> dition	<u>l</u> nsérer	<u>P</u> rojet	<u>C</u> omposant	Connexion	Fonction	<u>S</u> ortie	Dé <u>b</u> oguer	Aff <u>i</u> chage	O <u>u</u> tils	F <u>e</u> nêtre /	Ai <u>d</u> e
🗋 🖻 🗄 (	9 🕘 🗠	~   <u>∦</u> [	è 🗳 🗙	👬 🕼 📬 💝	• Default	- 🖪 .	<u>i ASLT ASUTe</u>	<u>XSUTa KQ.</u> <u>JAVA</u> <u>C</u> #		▶ ■ २७	H. 💎 💎 (	•
🛃 🚺 EDIQ [	lı 🖳 🚹 🛛		abçî var D	🗛 🖓 🐺 🕵 👘	32 🕐 😫 🛉	0 00 00 📼			100% -	R R E	F 🖅 🛍 👼	4

#### Barre de statut d'application

La barre de statut d'application apparaît en bas de la fenêtre d'application et montre l'information au niveau de l'application. Les info-bulles sont affichées quand vous déplacez la souris au-dessus d'un bouton de barre d'outils. Si vous utilisez la version 64-bit de MapForce, le nom d'application apparaît dans la barre de statut avec le suffixe (x64). Il n'y a pas de suffixe pour la version 32-bit.

## 1.3.2 Fenêtres

Cette rubrique donne une vue d'ensemble des fenêtres disponibles.

#### Fenêtre de bibliothèques

La fenêtre **Bibliothèques** recense les fonctions intégrées MapForce, qui sont organisées en bibliothèques. La liste des fonctions disponibles change sur la base du langage de transformation que vous avez choisi soit depuis le menu **Sortie**, soit depuis la barre d'outils **Sélection de langage**. Pour plus d'information, voir Langages de transformation<sup>25</sup>. Si vous avez créé des fonctions définies par l'utilisateur, ou si vous avez importé des bibliothèques externes, elles apparaîtront aussi dans la fenêtre des **Bibliothèques**.

Bibliothèques 2			
🔁 core		^	
_ aggregate functions			
avg	résultat =		
count	résultat =		
max	résultat =		
max-string	résultat =		
min	résultat =		
min-string	résultat =		
string-join	résultat =		
sum	résultat =		
■= conversion functions			
boolean	résultat =		
format-date	résultat =		
format-dateTime	résultat =		
format-number	résultat =	1	
format-time	résultat =	1	
number	résultat =		
parse-date	résultat =		
parse-dateTime	résultat =		
parse-number	résultat =		
parse-time	résultat =		
string	résultat =	¥	
Chercher les fonctions		•	
Ajouter/Supprimer les bibliothèques			
🖀 Bibliothèques  🛅 Projet			

Pour chercher des fonctions par leur nom ou leur description, saisissez la valeur de recherche dans le champ de saisie situé en bas de la fenêtre **Bibliothèques**. Pour trouver toutes les occurrences d'une fonction (dans le mappage actif actuellement), cliquez avec la touche de droite sur la fonction et choisissez **Trouver tous les appels** depuis le menu contextuel. Vous pouvez aussi consulter le type de données de fonction et la description directement depuis la fenêtre **Bibliothèques**. Pour plus d'informations, voir <u>Fonctions</u><sup>651</sup>.

### Fenêtre Projet (Éditions Enterprise et Professional)

MapForce prend en charge l'Interface de document multiple et permet de regrouper vos mappages dans des projets de mappage. La fenêtre **Projet** montre tous les fichiers et les dossiers qui ont été ajoutés au projet. Les fichiers de projet ont l'extension \*.mfp (MapForce Project). Pour rechercher des mappages au sein des projets, cliquez n'importe où au sein de la fenêtre **Projet** et appuyez sur **CTRL + F**.Pour plus d'information, voir <u>Projets</u>.



#### Fenêtre(s) de mappage

MapForce utilise une Multiple Document Interface (MDI). Chaque fichier de mappage que vous ouvrez dans MapForce a une fenêtre séparée. Cela vous permet de travailler avec plusieurs fenêtres de mappage et d'arranger ou de redimensionner les fenêtres de plusieurs manières dans la fenêtre principale (parent) de MapForce. Vous pouvez aussi arranger toutes les fenêtres ouvertes en utilisant les mises en page Windows standard : Disposer horizontalement, Disposer verticalement, Cascade. Lorsque de multiples mappages sont ouverts dans MapForce, vous pouvez passer rapidement entre les onglets disposés dans la partie inférieure du volet **Mappage** (*voir la capture d'écran ci-dessous*).

•			
Mapping	XSLT2	Output	
New Desig	gn1 🛛 🐻 N	lew Design2	🗞 New Design3

Vous pouvez ouvrir les options de gestion de la Fenêtre en utilisant la commande de menu **Fenêtre | Fenêtres**. Le dialogue **Fenêtres** vous permet de réaliser différentes actions y compris activer, enregistrer, fermer ou minimiser les fenêtres de mappage ouvertes. Pour choisir plusieurs fenêtres dans le dialogue **Fenêtres**, cliquez sur les entrées nécessaires tout en maintenant la touche **Ctrl** appuyée.

#### Gérer la fenêtre Bibliothèques

Cette fenêtre vous permet de consulter et de gérer toutes les fonctions définies par l'utilisateur (FDU) et les bibliothèques personnalisées importées (y compris les fichiers Java .class compilés et les fichiers .NET DLL assembly) qui sont utilisés par les mappages actuellement ouverts.

Par défaut, la fenêtre Gérer des bibliothèques n'est pas visible. Pour l'afficher, choisir une des deux options suivantes :

- Dans le menu View, cliquer sur Gérer Bibliothèques.
- Cliquer Ajouter/Supprimer Bibliothèques en bas de la fenêtre Bibliothèques.



Vous pouvez choisir de consulter des UDF et des bibliothèques uniquement pour le document de mappage (fichier .mfd) qui est activé actuellement, ou pour tous les mappages ouverts. Pour consulter des fonctions et des bibliothèques importées pour tous les documents de mappages ouverts actuellement, cliquer avec la touche de droite dans la fenêtre et sélectionner **Afficher les documents ouverts** depuis le menu contextuel.

Pour afficher le chemin du document de mappage ouvert au lieu du nom, cliquer avec la touche de droite dans la fenêtre et sélectionner **Afficher les chemins de fichier** depuis le menu contextuel.

Pour plus d'informations, voir <u>Gérer les Bibliothèques de fonction</u><sup>555</sup>.

#### Fenêtre de vue d'ensemble

La fenêtre **Aperçu** vous donne une vue d'ensemble du <u>volet de mappage</u><sup>33</sup>. Utilisez-la pour naviguer rapidement vers un emplacement particulier dans la zone de mappage lorsque la taille de mappage est très importante. Pour vous rendre à un emplacement particulier sur le mappage, cliquez et glissez le rectangle rouge.



## 1.3.3 Fenêtre de messages

La fenêtre de **Messages** (*voir la capture d'écran ci-dessous*) affiche des statuts de de validation, des messages, des erreurs et/ou des avertissements lorsque vous prévisualisez ou <u>validez</u><sup>103</sup> un mappage. Cliquez sur le texte souligné dans la fenêtre de **Messages** pour voir un composant ou une structure qui a causé le message d'information, d'avertissement ou d'erreur.

Me	ssages	×
<b>F</b>	▼▼▲□⊡⊠ ۹۶۶×	
2	🖵 🕑 <u>BooksToLibrary.mfd</u> : Mapping validation successful 0 error(s), 1 warning(s)	
	$r = \underline{\mathbb{A}}  ightarrow  ext{publication}$ has no input connection and will be generated only once.	
	You can try to connect $\triangleright$ <u>book</u> with $\triangleright$ <u>publication</u> to resolve.	
<u>∼</u>	🔚 🛄 The output component 🤮 library has no output file name set. A default file name will be used.	
r la		
~	~	
	~	
6	4	-

#### Valider l'icône du statut de validation

Lorsque vous validez un mappage, MapForce vérifie, par exemple, des sortes de composants non pris en charge et des connexions incorrectes ou manquantes. Le résultat de la validation est affiché dans la fenêtre de **Messages** avec une des icônes de statut suivantes :

Icône	Signification
0	Validation a été achevée avec succès.
<ul> <li>Image: Second sec</li></ul>	Validation a été achevée avec des avertissements.
8	Validation a échoué.

La fenêtre **Messages** peut afficher tout type de message supplémentaire suivant : messages d'information, avertissements et erreurs.

Icône	Signification
•	Indique un message d'information. Les messages d'information ne stoppent pas l'exécution de mappage.
	Indique un message d'avertissement. Les avertissements ne stoppent pas l'exécution de mappage. Ils peuvent être générés, par exemple, lorsque vous ne créez pas de connexions à des connecteurs d'entrée obligatoires. Dans ces cas, la sortie sera encore générée pour les composants dont les connexions valides existent.
0	Indique une erreur. Lorsqu'une erreur se produit, l'exécution de mappage échoue et aucune sortie n'est générée. L'aperçu du code XSLT ou XQuery n'est pas possible non plus.

Pour souligner le composant ou la structure qui a causé l'information, l'avertissement ou le message d'erreur, cliquez sur le texte souligné dans la fenêtre **Messages**.

#### Actions liées aux messages

La fenêtre Messages vous permet de prendre les actions suivantes :

Icône	Description
Þ	Filtrer les messages par sévérité : messages d'information, erreurs, avertissements. Choisir <b>Tout cocher</b> pour comprendre tous les niveaux de sécurité (il s'agit du comportement par défaut). Choisir <b>Tout décocher</b> pour supprimer tous les niveaux de sévérité du filtre. Dans ce cas, seule l'exécution générale ou le message de statut de validation est affiché.
	Passer à la ligne suivante.
	Passer à la ligne précédente.
<u>D</u>	Copier la ligne choisie dans le presse-papiers.
Đ	Copier la ligne choisie dans le presse-papiers, y compris toutes les lignes imbriquées en- dessous.
A	Copier les contenus entiers de la fenêtre <b>Messages</b> dans le presse-papiers.
٩	Trouver un texte spécifique dans la fenêtre <b>Messages</b> . En option, pour trouver uniquement des mots, choisir <b>Correspondance mot entier uniquement</b> . Pour trouver du texte tout en préservant la casse minuscule ou majuscule, choisir <b>Respecter la casse</b> .
đ	Trouver un texte spécifique en commençant depuis la ligne sélectionnée jusqu'à la fin.
\$	Trouver un texte spécifique en commençant depuis la ligne sélectionnée actuellement jusqu'au début.
×	Supprimer la fenêtre Messages.

Lorsque vous travaillez avec plusieurs fichiers de mappage simultanément, vous souhaiterez peut-être afficher des messages d'information, d'avertissement ou d'erreur dans des onglets individuels pour chaque mappage. Dans ce cas, cliquez sur les onglets numérotés disponibles à gauche de la fenêtre de **Messages** avant de valider le mappage.

## 1.3.4 Volets

Cette rubrique donne un aperçu des volets disponibles.

#### Volet Mappage

Le volet **Mappage** est la surface de travail dans laquelle vous configurez les <u>mappages</u><sup>103</sup>. Vous pouvez ajouter des composants de mappage (comme des fichiers, des schémas, des constantes, des variables, etc.) dans la zone de mappage à partir du menu **Insérer**. Pour plus d'information, voir <u>Ajouter des composants au</u>
<u>mappage</u><sup>78</sup>. Vous pouvez aussi glisser des fonctions affichées dans la fenêtre **Bibliothèques** dans le volet **Mappage**. Pour plus d'informations, voir <u>Ajouter une fonction au mappage</u><sup>552</sup>.

### Volet XSLT

Le volet **XSLT** affiche le code de transformation XSLT généré depuis votre mappage. Pour passer à ce volet, choisir XSLT, XSLT 2 ou XSLT 3 en tant que <u>langage de transformation</u><sup>25</sup>, puis cliquez sur l'onglet portant le même nom.

Ce volet vous propose une numérotation de ligne et un pliage de code. Pour agrandir ou réduire des portions de code, cliquer sur les icônes "+" et "-" du côté gauche de la fenêtre. Toute portion du code réduit est affichée avec un symbole d'ellipse. Pour consulter le code réduit sans l'agrandir, déplacez le curseur de la souris sur l'ellipse. Une infobulle s'ouvre, qui affiche le code à consulter comme indiqué dans l'image ci-dessous. Veuillez noter que si le texte consulté est trop grand pour être affiché dans l'infobulle, une ellipse supplémentaire apparaît à la fin de l'infobulle.

1	xml ver</th <th>sion="1.0" encoding="UTF-8"?&gt;</th> <th></th> <th>*</th>	sion="1.0" encoding="UTF-8"?>		*	
2	!	->			
11	<pre><xsl:style< pre=""></xsl:style<></pre>	esheet version="1.0" xmlns:xsl="http://www.w3.or	rg/1999/		
	XSL/Trans	form" xmlns:xs="http://www.w3.org/2001/XMLSchema	exclude-		
	result-pr	efixes="xs">			
12	<xsl:< td=""><td>output method="xml" encoding="UTF-8" indent="ye</td><td>s"/&gt;</td><td></td></xsl:<>	output method="xml" encoding="UTF-8" indent="ye	s"/>		
13	🤶 <xsl:< td=""><td>template match="/"&gt;</td><td></td><td></td></xsl:<>	template match="/">			
14	<	<pre>xsl:variable name="var1_initial" select="."/&gt;</pre>			
15	⊝ <	PersonList>			
16		<pre><xsl:attribute <="" name="xsi:noNamespaceSchemaLo" pre=""></xsl:attribute></pre>	cation"		
	namespace	="http://www.w3.org/2001/XMLSchema-instance">fil	le:///C:/		
	Users/alt	ova/Documents/Altova/MapForce2020/MapForceExampl	les/		
	PersonLis	t.xsd			
17	+	<pre><xsl:for-each <="" select="(./Company/Employees/n&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;ode())[./&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;self::*]" td=""><td><pre>&lt;</pre></td></xsl:for-each></pre>	<pre>&lt;</pre>		
31	<	/PersonList>			
32	- <td><xsl:variable name="var2_filter" select="."></xsl:variable></td> <td></td> <td></td>	<xsl:variable name="var2_filter" select="."></xsl:variable>			
33	<td><person></person></td> <td></td> <td></td>	<person></person>			
34		<xsl:attribute name="role"></xsl:attribute>			
		<xsl:value-of select="local-name(.)"></xsl:value-of>			
				×.	
Mapp	ing XSLT	<first></first>			
Dorr	onlict mfdt	<xsl:value-of select="FirstName"></xsl:value-of>	4 5	~	
Pers	onListania		4 P	^	
Overview	v	<last></last>	<b>▼</b> ₽	×	
		<xsl:value-of select="LastName"></xsl:value-of>			
				-	

Pour configurer les paramètres de vues, y compris l'indentation, les marqueurs de fin de ligne, et autres), cliquez avec la touche de droite sur le volet et choisissez **Paramètres mode texte** du menu contextuel. De manière alternative, cliquez sur in (**Paramètres du mode Texte**) dans la barre d'outils.

### Volet XQuery (Éditions Enterprise et Professional)

Le volet **XQuery** affiche le code de transformation XQuery généré depuis votre mappage, lorsque vous cliquez sur la touche **XQuery**. Ce volet est disponible lorsque vous sélectionnez XQuery en tant que langage de

transformation. Ce volet fournit également des fonctions de numérotation de lignes et de pliage de code, qui fonctionne d'une manière semblable au volet XSLT (voir ci-dessus).

### Volet Requête BD (Éditions Enterprise et Professional)

Le volet **Requête BD** vous permet de requêter directement toutes les bases de données principales. Vous pouvez travailler avec plusieurs connexions actives vers des bases de données différentes. Pour plus d'informations, voir <u>Parcourir et interroger des bases de données</u><sup>(332)</sup>.



### Volet Sortie

Le volet **Sortie** affiche le résultat de la transformation de mappage. Si le mappage génère plusieurs fichiers, vous pouvez naviguer de manière séquentielle à travers chaque fichier généré.

1	xml version="1.0" encoding="UTF-8"?	
2	<pre>PersonList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>	
	<pre>xsi:noNamespaceSchemaLocation="file:///C:/Users/altova/Documents/</pre>	
	Altova/MapForce2020/MapForceExamples/PersonList.xsd">	
3	<pre></pre>	
4	<first>Vernon</first>	
5	<last>Callaby</last>	
6	-	
7	<pre></pre>	
8	<first>Frank</first>	
9	<last>Further</last>	
10	-	
11	<pre></pre>	
12	<first>Loby</first>	
13	<last>Matise</last>	
14	-	
15	<pre></pre>	
16	<first>Susi</first>	
17	<last>Sanna</last>	
18	-	
19	<pre>_</pre>	
		Ψ.
Марр	aing XSLT Output	
👩 Pers	ionList.mfd*	⊳ ×

Ce volet fournit également des fonctions de numérotation de lignes et de pliage de code, qui fonctionne d'une manière semblable au volet XSLT (voir ci-dessus).

### Volets de Sortie de StyleVision (Éditions Enterprise and Professional)

Si vous avez installé <u>Altova StyleVision</u>, les volets de sortie de StyleVision deviennent disponibles à côté du volet **Sortie**. Les volets de sortie de StyleVision vous permettent de consulter et d'enregistrer la sortie de mappage dans des formats HTML, RTF, PDF et Word 2007+. Cela est possible grâce aux fichiers StyleVision Power Stylesheet (SPS) conçus dans StyleVision et attribués à un composant de mappage dans MapForce.

# 2 **Tutoriels**

Avec l'aide des tutoriels, vous serez à même de comprendre et utiliser les capacités de transformation de données basiques de MapForce. Les tutoriels vous guideront à travers les bases et ce étape par étape. Au fur et à mesure, les tutoriels deviennent de plus en plus complexes. Pour cette raison, il est recommandé de les suivre par séquence. Il est avantageux de posséder des connaissances de XML et de Schéma XML.

#### Exemples de fichiers

Les fichiers de mappage illustrés ou référencés dans ces tutoriels sont disponibles dans le <u>dossier</u> <u>BasicTutorials</u><sup>24</sup>. Si vous avez des doutes sur les possibles effets qu'aura une modification sur les exemples originaux fournis avec MapForce, pensez à créer des sauvegardes avant de procéder aux modifications.

#### Liste des tutoriels

#### <u>Une source vers une cible</u>

<u>Ce tutoriel</u> montre comment utiliser les mécanismes clés de MapForce pour mapper les nœuds d'un fichier source aux nœuds d'un fichier cible. Ce tutoriel poursuit son explication comment convertir un fichier XML défini par un schéma XML en fichier XML défini par un schéma XML différent.

#### Sources multiples vers une cible

<u>Ce tutoriel</u> montre comment fusionner des données depuis de multiples fichiers XML source à un fichier cible.

#### Mappages enchaînés

Dans ce tutoriel<sup>68</sup>, nous créons un mappage simple comme dans le premier tutoriel, puis nous filtrons les données produites par ce mappage et passons les données filtrées au deuxième fichier cible.

#### Sources multiples vers cibles multiples

<u>Ce tutoriel</u> montre comment lire des données depuis multiples fichiers d'instance XML situés dans le même dossier et de les écrire dans de multiples fichiers XML générés instantanément.

Tutoriels

:

# 2.1 Une source vers une cible

Ce tutoriel décrit comment créer un mappage pour un des scénarios les plus basiques. Notre objectif est de prendre les données depuis un fichier A avec un schéma XML A y assigné et mettre ces données dans un fichier B avec un schéma XML B y assigné. Donc, notre fichier cible aura les mêmes données que dans la source mais avec un schéma différent (structure). Grâce au code de transformation, vous verrez comment la structure du fichier source aura changé. Veuillez noter que le mappage peut uniquement être effectué entre les structures, mais nous ne pourrons pas visualiser le résultat de la transformation dans <u>le volet de sortie</u>. Pour cela, à des fins d'illustration, nous utilisons un fichier XML dénommé Books.xml (voir la liste de code ci-dessous). Les grandes lignes de notre méthode seront comme suit :

- 1. Puisque nous utilisons deux structures de données, nous allons créer deux composants (*Source* et *Cible*) dans notre conception de mappage.
- 2. Puis, nous allons mapper des nœuds en connectant un nœud source à un nœud cible désiré. Ce sont ces connexions qui constituent le mappage et déterminent quel nœud source correspond à quel nœud cible.
- 3. Puisque la transformation d'un document en un autre est réalisée en utilisant un langage de transformation appropriée, tel que XSLT, nous sélectionnons un langage de transformation.
- 4. Nous utilisons les moteurs de transformation built-in de MapForce pour transformer le document XML source (Books.xml) en un document cible requis. Ce document cible sera un document XML valide conformément à l'XSD cible (Library.xsd).
- 5. Finalement, nous allons pouvoir enregistrer le fichier XML.

L'image ci-dessous illustre un modèle abstrait de la transformation de données utilisée dans le présent tutoriel



Le modèle abstrait ci-dessus illustre deux étapes du processus de mappage. Dans la première étape, la structure de Books.xsd est mappée vers la nouvelle structure dénommée Library.xsd. Le mappage est effectué par le biais d'un langage de transformation<sup>25</sup>. Le choix du langage de transformation dépend de votre édition de MapForce. Dans notre cas, XSLT2 a été choisi en tant que langage de transformation. Dans la deuxième étape, le contenu de Books.xml, dont Books.xsd y est assigné, est mappé vers le fichier cible

(BooksOutput.xml) et basé sur un nouveau schéma (Library.xsd). La liste de codes ci-dessous montre les données échantillon de Books.xml qui seront utilisées en tant que source de données.

```
<books>
<book id="1">
<book id="1">
<book id="1">
<book id="1">
<book id="1">
<book id="1">
<book id="2">
<book id="2"</book id="2">
<book id="2"</book id="2">
<book id="2"</book id="2"</book id="2">
<book id="2"</book id=
```

C'est ainsi que nous voulons que nos données aient l'air dans le fichier cible dénommé BooksOutput.xml :

```
<library>
  <last_updated>2015-06-02T16:26:55+02:00</last_updated>
   <publication>
      <id>1</id>
      <author>Mark Twain</author>
      <title>The Adventures of Tom Sawyer</title>
      <genre>Fiction</genre>
      <publish_year>1876</publish_year>
   </publication>
   <publication>
      <id>2</id>
      <author>Franz Kafka</author>
      <title>The Metamorphosis</title>
      <genre>Fiction</genre>
      <publish_year>1912</publish_year>
   </publication>
</library>
```

Certains noms d'élément dans l'XML source et cible ne sont pas identiques. Notre objectif est de remplir les éléments <author>, <title>, <genre> et <publish\_year> du fichier cible depuis les éléments équivalents dans le fichier source (<author>, <title>, <category>, <year>). L'attribut id dans le fichier source doit être mappé vers l'élément <id> dans le fichier. Enfin, nous devons remplir l'élément <last\_updated> du fichier cible avec la date et l'heure à laquelle le fichier a été mis à jour dernièrement.

Pour effectuer la transformation de données requise, suivez les étapes décrites dans les sous-sections cidessous.

### 2.1.1 Créer et enregistrer le Design

Cette rubrique explique comment créer un nouveau design, sélectionner un langage de transformation, valider et enregistrer votre mappage.

### Créer un nouveau design

Afin de pouvoir effectuer une transformation, vous devrez créer un nouveau design de mappage, qui peut être fait d'une des manières suivantes :

- Allez au menu Fichier and cliquez sur Nouveau. Puis sélectionnez Mappage et cliquez sur OK.
- Cliquez sur 🖾 dans la batte d'outils. Puis sélectionnez **Mappage** et cliquez sur **OK**.

#### Sélectionner un langage de transformation

Dépendant de votre édition de MapForce, différents <u>langages de transformation</u><sup>25</sup> sont disponibles. Pour ce tutoriel, nous avons choisi XSLT2. Vous pouvez sélectionner ce langage de transformation d'une des manières suivantes :

- Cliquez sur (XSLT2) dans la barre d'outils.
- Dans le menu Sortie, cliquez sur XSLT 2.0.

#### Valider et enregistrer le mappage

La validation d'un mappage est une étape optionnelle qui vous permet de voir et de corriger des erreurs et des avertissements de mappage potentiels avant d'exécuter le mappage. Vous pouvez valider votre mappage à toute étape. Pour vérifier si le mappage est valide, effectuez une des étapes suivantes :

- Dans le menu Fichier, cliquez sur Valider mappage.
- Cliquez sur 📴 (Valider) dans la barre d'outils.

La fenêtre Messages affiche les résultats de validation comme suit :

Messages	-	<b>ņ</b>	×
New Design1: Mapping validation successful - 0 error(s)	), 0 warning(	s)	*
(C)			

Pour enregistrer le mappage, effectuez une des étapes suivantes :

- Cliquez sur Enregistrer dans le menu Fichier.
- Cliquez sur 🖬 (Enregistrer) dans la barre d'outils.

À votre convenance, le mappage créé dans ce tutoriel est enregistré en tant que Tut1\_schemaToSchema.mfd.

## 2.1.2 Ajouter composant source

À cette étape, nous souhaitons ajouter un fichier XSD file, qui sera la structure du premier composant, et un fichier XML, qui fournira les données pour ce composant. Le fichier source dénommé Books.xsd peut être ajouté au mappage d'une des manières suivantes :

- Cliquez
   (Insérer Fichier/Schéma XML) dans la barre d'outils.
- Dans le menu Insérer, cliquez sur Schéma XML/Fichier.
- Glissez Books.xsd depuis Windows Explorer dans la zone de mappage.

Lorsque vous ajoutez un schéma, MapForce suggère d'ajouter un fichier XML échantillon. Cliquez sur **Naviguer** et recherchez Books.xml qui est placé dans le même dossier. Donc notre fichier source contient un schéma et du contenu. Dans les propriétés de chaque composant source ou cible, nous pouvons spécifier un fichier XSD et un fichier XML. Le fichier XSD définit la structure du document dans ce composant. Le fichier XML fournit les données de ce composant (source ou cible) et doit être valide par le schéma de ce composant. Si un composant est créé depuis un fichier XSD, vous êtes invité à avoir un fichier XML, le fichier XSD qui est référencé depuis un fichier XML sera utilisé pour définir la structure des données du composant. S'il n'existe aucune référence à un XSD, MapForce vous demandera s'il peut générer un fichier XSD pour ce composant.

#### Afficher la structure

Maintenant que le fichier source a été ajouté à la zone de mappage, vous pouvez voir sa structure. Dans MapForce, cette structure est connue en tant qu'un composant de mappage ou simplement comme <u>composant</u><sup>74</sup>. Vous pouvez élargir des éléments dans le composant en cliquant sur l'icône ⊞. De manière alternative, vous pouvez appuyer sur la clé + sur le pavé numérique. La capture d'écran ci-dessous illustre le composant source :



Books dans la barre de titre indique le nom du composant. Le nœud du niveau supérieur représente le nom du fichier d'instance XML : Fichier : Books.xml. Les éléments XML dans la structure sont représentés par l'icône . Les attributs XML sont représentés par l'icône . Les petits triangles, affichés des deux côtés du composant, représentent les entrées de données du côté gauche et les sorties du côté droit. Dans MapForce, ces triangles sont appelés connecteurs d'entrée et connecteurs de sortie, respectivement.

#### Déplacer et redimensionner les composants

Pour déplacer le composant dans le volet de mappage, cliquez sur l'en-tête du composant et glissez la souris sur un nouvel emplacement. Pour redimensionner le composant, faites glisser le coin droit du composant. Vous pouvez aussi double-cliquer sur le coin de manière à ce que MapForce ajuste la taille automatiquement.

# 2.1.3 Ajouter composant cible

La prochaine étape est d'ajouter un composant cible et de définir ses paramètres. Pour ajouter le fichier cible

dénommé Library.xsd au mappage, cliquez sur (Insérer le Schéma XML/Fichier). Cliquez sur Ignorer lorsque vous êtes invité par MapForce à fournir un fichier d'instance. À cette étape, le design de mappage ressemble à l'exemple suivant :



Veuillez noter que lorsque vous ouvrez library.xsd, il est affiché en tant que fichier XML file dans le composant. En fait, MapForce crée uniquement une référence vers le fichier XML dénommé Library.xml, mais ce fichier XML lui-même n'existe pas encore. Donc notre compostant cible a un schéma mais pas de contenu.

### Paramètres de composant

Désormais, nous allons devoir renommer le composant cible Books\_output.xml. Ceci nous permettra d'éviter toute confusion dans les prochains tutoriels, étant donné que nous allons utiliser un fichier séparé dénommé Library.xml, qui a son propre contenu et est basé sur le même schéma Library.xsd. Afin de renommer le fichier cible, double-cliquez sur l'en-tête du composant cible. Ceci ouvre la <u>boîte de dialogue des Paramètres de composant</u> (*voir la capture d'écran ci-dessous*), dans laquelle nous allons devoir modifier le nom du fichier cible comme suit :

😢 Component Sett	tings		×
Component name:	BooksOutput		
Library.xsd		Browse	Edit
Input XML File			
		Browse	Edit
Output XML File			
BooksOutput.xml		Browse	Edit

Le design de mappage a désormais l'air de ceci :

욚	Books		🛱 BooksOutput	
Ģĺ	File: Books.xml	File/String	🖃 🗋 File: BooksOutput.xml File	<u>/s</u> p
¢ <sup>i</sup> ∈	books	¢	þ .⊟ <b>&lt;&gt;</b> library	Þ
	<b>⊟</b> <>book	¢	Iast_updated	Þ
Þ	id	Þ		Þ
Þ	Author	¢	⊳ <mark>≺&gt;</mark> id	Þ
Þ	title	¢	> author	Þ
Þ	Category	¢	title	Þ
Þ	<>>year	¢	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	Þ
			publish_year	Þ

Note : le nom de l'en-tête fait uniquement référence au nom de composant et non pas au nom de schéma sur lequel ce fichier est basé. Pour voir le nom de schéma, ouvrez la boîte de dialogue Paramètres de composant.

### 2.1.4 Connecter Source et Cible

À cette étape, nous allons mapper les données dans le fichier source au fichier cible. L'objectif est de mapper deux types d'entrée aux nœuds cible : (i) nœuds source et (ii) données dont les deux sont mappés de manière simultanée. La plupart des données sont issus de Books.xml. Dans notre exemple, nous fournirons également des informations sur la date et l'heure actuelles utilisant la fonction XPath <u>current-dateTime</u><sup>781</sup>.

#### Connexions automatiques

Désormais, nous allons créer une connexion de mappage entre l'élément <book> dans le composant source et l'élément <publication> dans le composant cible. À cette fin, cliquez sur le connecteur de sortie (le petit triangle) situé à droite de l'élément <book> et le glisser dans le connecteur d'entrée de l'élément

#### Tutoriels

publication> dans la cible. Lorsque vous effectuez cette étape, MapForce peut connecter automatiquement tous les éléments enfants de <book> dans le fichier source vers les éléments portant les mêmes noms dans le fichier cible. Dans notre exemple, quatre connexions ont été créées simultanément (*voir capture d'écran cidessous*). Cette fonction est appelée *Auto-connexion des enfants correspondants* et peut être désactivée et personnalisée, le cas échéant.



Vous pouvez activer ou désactiver **Auto-connexion des enfants correspondants** d'une des manières suivantes :

- Cliquez sur 🖽 (Basculer auto-connexion des enfants) dans la barre de menu.
- Dans le menu Connexion, cliquez sur Auto-connexion des enfants correspondants.

### Connecter des items obligatoires

Veuillez noter que certains des connecteurs d'entrée dans le composant cible ont été marqués en orange par MapForce, ce qui indique que ces items sont obligatoires. Ces items sont obligatoires car ils ont été définis de cette manière dans le schéma de fichier. Afin de garantir la validité du fichier XML cible, il est nécessaire de fournir les valeurs pour les items obligatoires comme suit :

- Connectez l'élément <category> dans la source avec l'élément <genre> dans le composant cible.
- Connectez l'élément <year> dans la source avec l'élément Lignes dans le composant cible.

#### Ajouter la date et l'heure actuelles

Enfin, vous devez fournir une valeur pour l'élément <last\_updated>. Si vous déplacez la souris sur son connecteur d'entrée, vous constaterez que l'élément est de type xs:dateTime(voir capture d'écran ci-dessous).

Veuillez noter que, pour voir s'afficher les astuces, la touche de la barre d'outils (Afficher les astuces) doit

être activée. En cliquant sur (Afficher types de données) dans la barre d'outils, vous pouvez aussi rendre le type de données visible à tout moment.



Vous pouvez recevoir la date et l'heure actuelles par le biais d'une fonction XSLT2 current-dateTime. Pour trouver cette fonction, tapez dans la zone de texte située dans le bas de la <u>fenêtre des bilbiothèques</u> <sup>(31)</sup> (*voir la capture d'écran ci-dessous*). En alternative, double-cliquez sur une zone vide à l'intérieur du volet de **mappage** et commencez à saisir current-date.



Pour ajouter la fonction au mappage, glissez la fonction dans le volet de **mappage** et connectez sa sortie à l'entrée de l'élément <last\_updated> (voir la capture d'écran ci-dessous)..



Vous pouvez désormais valider et enregistrer votre mappage, tel qu'affiché dans <u>Créer et enregistrer le</u> design<sup>42</sup>.

# 2.1.5 Consulter le résultat de mappage

MapForce utilise ses moteurs built-in pour générer la sortie et permet de prévisualiser le résultat du mappage directement dans le volet **Sortie** (*voir la capture d'écran ci-dessous*).

1	xml version="1.0" encoding="UTF-8"?	
2	<pre><li>library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi</li></pre>	i
3	<lpre><last_updated>2020-11-18T16:05:48+01:00</last_updated></lpre>	
4	<pre>     <pre></pre></pre>	
5	<id>1</id>	
6	<author>Mark Twain</author>	
7	<title>The Adventures of Tom Sawyer</title>	
8	<pre><genre>Fiction</genre></pre>	
9	<publish_year>1876</publish_year>	
10	<pre>- </pre>	
11	<pre>     <pre></pre></pre>	
12	<id>2</id>	
13	<author>Franz Kafka</author>	
14	<title>The Metamorphosis</title>	
15	<pre><genre>Fiction</genre></pre>	
16	<publish_year>1912</publish_year>	-
17	· · · · · · · · · · · · · · · · · · ·	
Map	ing XSLT2 DB Query   Output	
Boo	ksToLibrary.mfd d D	х

Par défaut, les fichiers affichés dans le volet **Sortie** ne sont pas enregistrés sur disque. À la place, MapForce crée des fichiers temporaires. Pour enregistrer la sortie, ouvrez la volet de **Sortie** et sélectionnez la commande

# de menu Sortie | Enregistrer le fichier de Sortie ou cliquez sur 違 (Enregistrer la sortie générée) dans la barre d'outils.

Pour configurer MapForce afin qu'il écrive la sortie directement sur les fichiers finaux au lieu des fichiers temporaires, rendez-vous sur **Outils | Options | Général**, et cochez la case **Écrire directement sur les fichiers de sortie finaux**. Veuillez noter qu'il n'est pas recommandé d'activer cette option pendant que vous suivez ce tutoriel, car vous risquez d'écraser les fichiers de tutoriel originaux par erreur.

Vous pouvez aussi consulter le code XSLT généré qui effectue la transformation. Pour consulter le code, cliquez sur la touche **XSLT2** située dans la partie inférieure du volet de **mappage**. Afin de générer le code XSLT2 et de l'enregistrer dans un fichier, sélectionnez l'item de menu **Fichier | Générer code dans | XSLT 2.0**. Vous serez invité à choisir un dossier dans lequel le code généré sera enregistré. Une fois que la génération de code est achevée, le dossier de destination contiendra les deux fichiers suivants :

- 1. Un fichier de transformation XSLT, nommé après le schéma cible. Le fichier de transformation a le format suivant : MappingMapTo<TargetFileName>.xslt.
- 2. Un fichier DoTransform.bat, qui vous permet d'exécuter la transformation XSLT avec <u>Altova</u> <u>RaptorXML Server</u> depuis la ligne de commande. Pour exécuter la commande, vous allez devoir installer RaptorXML.

# 2.2 Sources multiples vers une cible

Dans ce tutoriel, vous apprendrez à fusionner les données depuis un nouveau fichier appelé Library.xml avec les données de Books.xml. Le résultat sera un fichier cible dénommé MergedLibrary.xml, qui contiendra les données des deux fichiers source. Le fichier cible sera basé sur le schéma Library.xsd. Veuillez noter que les fichiers source ont différents schémas. Si les fichiers source avaient le même schéma, vous pourriez aussi fusionner leurs données en utilisant une approche différente (voir <u>Sources multiples à Cibles multiples</u>). L'image ci-dessous représente un modèle abstrait de la transformation de données décrite dans le présent tutoriel.



La liste de code ci-dessous montre un extrait de **Books.xm1**, le fichier qui sera utilisé en tant que première source de données.

```
<books>
  <book id="1">
    <author>Mark Twain</author>
    <title>The Adventures of Tom Sawyer</title>
    <category>Fiction</category>
    <year>1876</year>
  </book>
</books>
```

La liste de code ci-dessous montre un extrait de **Library.xml**, le fichier qui sera utilisé en tant que deuxième source de données :

```
<library>
</publication>
<id>5</id>
<author>Alexandre Dumas</author>
<title>The Three Musketeers</title>
<Genre>Rock</Genre>
<publish_year>1844</publish_year>
</publication>
</ Library>
```

C'est ainsi que nous voulons que nos données fusionnées aient l'air dans le fichier cible dénommé merged\_library.xml :

```
<library>
   </publication>
      <id>1</id>
      <author>Mark Twain</author>
      <title>The Adventures of Tom Sawyer</title>
      <Genre>Rock</Genre>
      <publish_year>1876</publish_year>
  </publication>
   </publication>
      <id>5</id>
      <author>Alexandre Dumas</author>
      <title>The Three Musketeers</title>
      <Genre>Rock</Genre>
      <publish_year>1844</publish_year>
   </publication>
</ Library>
```

Pour effectuer la transformation, suivez les étapes décrites dans les sous-sections ci-dessous.

### 2.2.1 Préparer les fichiers source

Le point de départ de ce tutoriel sont deux fichiers source, dont chacun est doté d'un schéma (Books.xsd et Library.xsd) et de données (Books.xml et Library.xml). Pour préparer les fichiers source pour le mappage, suivez les étapes suivantes :

- 1. Ouvrez Books.xsd.
- 2 Lorsque MapForce propose d'ajouter un fichier XML échantillon, cliquez sur **Naviguer** et ouvrez Books.xml.
- 3. Ouvrez Library.xsd.
- 4. Lorsque MapForce propose d'ajouter un fichier XML échantillon, cliquez sur **Naviguer** et ouvrez Library.ml.
- 5. À titre de convenance, placez les fichiers source l'un au-dessus de l'autre ((voir la capture d'écran cidessous).

À cette étape, le design de mappage ressemble à ceci :



Puisque les nouvelles références de mappage de plusieurs fichiers du même dossier, assurez-vous d'enregistrer ce nouveau mappage dans le dossier **BasicTutorials**.

# 2.2.2 Ajouter composant cible

Dans la prochaine étape, nous allons ajouter un fichier cible dénommé Library.xsd. Pour ajouter un fichier

cible, cliquez sur (Insérer le Schéma XML/Fichier). Cliquez sur Ignorer lorsque vous êtes invité par MapForce à fournir un fichier d'instance. Puis, cliquez sur l'en-tête du nouveau composant et glissez-le à droite des deux composants source. Vous pouvez toujours bouger les composants de mappage dans toutes les directions. Néanmoins, le fait de placer un composant source à la gauche d'un composant cible vous permettra une lecture et une compréhension plus claires de votre mappage. Il s'agit là aussi de la convention utilisée pour tous les mappages illustrés dans cette documentation, ainsi que dans les fichiers de mappage échantillon qui accompagnent votre installation MapForce.



À cette étape, le fichier cible n'est doté que d'une structure mais ne contient pas de données. Les données seront fusionnées à un moment ultérieur des deux fichiers source et mappées au fichier cible.

### 2.2.3 Vérifier et définir les fichiers d'entrée/de sortie

À cette étape, le mappage est doté de deux composants source (Books et Library) et d'un composant cible (Library). Désormais, nous avons deux composants du même nom - Library. Pour éviter toute confusion, nous devons modifier les paramètres dans la <u>boîte de dialogue des Paramètres de composant</u><sup>[22]</sup>. Doublecliquez sur l'en-tête de chaque composant. Puis, vérifiez et changez le nom et les fichiers d'entrée/de sortie de chaque composant, tel qu'affiché ci-dessous.

😫 Component Sett	tings		×
Component name:	Books		
Schema File			
Books.xsd		Browse	Edit
Input XML File			
Books.xml		Browse	Edit
Output XML File			
		Browse	Edit

😫 Component Set	tings		×
Component name:	Library		
Schema File			
Library.xsd		Browse	Edit
Input XML File			
Library.xml		Browse	Edit
Output XML File			
		Browse	Edit

😤 Component Settings		×	
Component name:	MergedLibrary		
Schema File			
Library.xsd		Browse	Edit
Input XML File		1	
		Browse	Edit
Output XML File			
MergedLibrary.xml		Browse	Edit

Comme indiqué ci-dessus, le premier composant source lit les données depuis Books.xml. Le deuxième composant source lit les données provenant de Library.xml. Enfin, le composant cible sort les données dans un fichier appelé MergedLibrary.xml.

### 2.2.4 Connecter Sources et Cible

Désormais, nous allons mapper les données depuis deux fichiers source (Books.xml et Library.xml) avec le fichier cible (MergedLibrary.xml). Pour réaliser cet objectif, suivons les étapes suivantes :.

- Connectez l'élément <book> dans le premier composant source avec l'élément <publication> dans MergedLibrary.xml. Lorsque vous effectuez cette étape, MapForce peut connecter automatiquement tous les éléments enfants de <book> dans le fichier source vers les éléments portant les mêmes noms dans le fichier cible. Dans notre exemple, quatre connexions ont été créées simultanément. Pour en savoir plus sur la connexion automatique des éléments enfant, voir le premier tutoriel
- Dès que vous vous connectez à l'élément <book> avec l'élément <publication>, vous remarquerez que certains connecteurs d'entrée dans le composant cible ont été marqués en orange. Ceci indique

que ces éléments sont obligatoires. Pour assurer la validité du fichier XML cible, connectez l'élément <category> avec l'élément <genre> et l'élément <year> avec l'élément <publish\_year>.

- Pour fournir une valeur pour l'élément <last\_updated>, trouvez la fonction dénommée currentdateTime dans la bibliothèque XSLT2. Faites glisser la fonction vers la zone de mappage et connectez result avec l'élément <last\_updated> dans Merged\_library.xml.
- Afin d'instruire MapForce d'écrire des données provenant de la deuxième source sur la cible, cliquez sur le connecteur de sortie de l'élément <publication> dans Library.xml et faites-le glisser dans le connecteur d'entrée de l'élément <publication> dans MergedLibrary.xml. Étant donné que le connecteur d'entrée cible a déjà une connexion établie, le message suivant apparaît :

MapForce	×
The input connector you are trying to connect to has already a connection assigned. Only one connection can be defined per input connector. What action do you want?	
Replace Connection Duplicate Input Cancel	

Dans ce tutoriel, notre objectif est de mapper des données depuis deux sources vers une cible. Cliquez donc sur **Doubler l'entrée**. Ce faisant, vous configurer un composant cible de telle façon qu'il acceptera les données également de la nouvelle source. À présent, le mappage ressemble à ceci :



La capture d'écran démontre que l'élément publication dans le composant cible a été doublé. Le nouveau nœud publication(2) acceptera les données provenant du composant source Library. Veuillez noter que même si le nom de ce nœud apparaît en tant que publication(2) dans le mappage, son nom dans le fichier XML cible sera publication, qui est l'objectif que nous souhaitons atteindre.

Cliquez sur la touche **Sortie** située en bas du volet de **mappage** pour consulter le résultat de mappage. Vous remarquerez que les données provenant de **Books.xml** et Library.xml ont désormais été fusionnées dans le nouveau fichier MergedLibrary.xml. Pour enregistrer la sortie, ouvrez le volet **Sortie** et sélectionnez la commande de menu **Sortie | Enregistrer le fichier de Sortie** ou cliquez sur **G** (Enregistrer la sortie générée) dans la barre d'outils.

À votre convenance, le design de mappage est enregistré dans ce tutoriel en tant que **Tut2\_MultipleToOne.mfd**. Ceci est utile car ce mappage sera utilisé en tant que point de départ dans <u>le</u> <u>prochain tutoriel</u><sup>58</sup>.

# 2.3 Mappages en chaîne

L'objectif de ce tutoriel est de filtrer les données dans MergedLibrary.xml, qui a été créée dans le <u>tutoriel</u> <u>précédent</u><sup>51</sup>, de telle façon qu'un seul sous-ensemble des données en question est affiché dans la sortie. De manière spécifique, nous allons uniquement avoir besoin des books publiés après 1900. L'image ci-dessous illustre un modèle abstrait de la transformation de données décrite dans le présent tutoriel.



Dans le diagramme ci-dessus, les données sont tout d'abord fusionnées depuis deux fichiers source (Books.xml et Library.xml) dans un seul fichier cible appelé MergedLibrary.xml. Ensuite, les données sont transformées avec une fonction de filtre et sont passées au composant suivant appelé FilteredLibrary.xml. Veuillez noter que FilteredLibrary.xml est basé sur le schéma Library.xsd. Le composant intermédiaire agit aussi bien en tant que cible et source des données. Dans MapForce, cette technique est appelée <u>mappages en chaîne</u><sup>[841]</sup>.

Pour effectuer la mappage, suivez les étapes décrites dans les sous-sections ci-dessous.

# 2.3.1 Préparer le design de mappage

Le point de départ de ce tutoriel **Tut2\_MultipleToOne.mfd** (*voir la capture d'écran ci-dessous*). Ce mappage a été conçu dans <u>le tutoriel précédent</u><sup>51</sup>.



Puisque le nouveau mappage référence plusieurs dossiers, assurez-vous d'enregistrer ce nouveau mappage dans le dossier **BasicTutorials**.

# 2.3.2 Configurer la deuxième cible

Désormais, nous devons ajouter et configurer le deuxième fichier cible (FilteredLibrary.xml), qui contiendra uniquement un sous-ensemble d'éléments <book> de MergedLibrary.xml.

### Ajouter le deuxième composant cible

Pour ajouter le deuxième composant cible, cliquez sur le bouton de la barre d'outils (Insérer Fichier/Schéma XML) et ouvrez Library.xsd. Cliquez sur Ignorer lorsque vous êtes invité à fournir consulter un fichier d'instance échantillon. Notre deuxième composant cible n'a qu'une structure mais pas de contenu. À une étape ultérieure, nous allons mapper les données filtrées avec le fichier cible. Le design de mappage a désormais l'air de ceci :



### Configurer le deuxième composant cible

Comme affiché ci-dessus, le mappage a désormais deux composants source (Books et Library) et deux composants cible (MergedLibrary et Library). Pour éviter toute confusion, nous modifierons le nom du composant nouvellement ajouté FilteredLibrary. Pour ce faire, double-cliquez sur l'en-tête du composant le plus à droite et éditer les paramètres de composant

😢 Paramètres de composant			×
Nom de composant: FiteredLibrary			
Fichier de schéma			
Library.xsd	Parcourir	Éditer	
Fichier XML d'entrée	Parcourir	Édition	
Echier VM de certie	Turcount	LUIDOIT	
FilteredLibrary.xml	Parcourir	Éditer	

# 2.3.3 Connecter les Cibles



# 2.3.4 Filtrer les données

À cette étape, nous allons filtrer les données de MergedLibrary de telle façon que seuls les livres publiés après 1900 seront passés au composant FilteredLibrary. Nous allons utiliser le composant Filtre à cette fin.

### Ajouter un filtre

Pour ajouter un filtre, cliquez avec le bouton droit entre MergedLibrary et FilteredLibrary et sélectionnez Insérer filtre : Nœuds/Lignes depuis le menu contextuel.



Le composant de filtre a désormais été ajouté au mappage (voir la capture d'écran ci-dessous).



Tel qu'affiché ci-dessus, le connecteur d'entrée bool est marqué en surbrillance orange, ce qui signifie qu'une entrée est obligatoire. Si vous passez sur un connecteur, vous pouvez voir si une entré<u>e</u> de type xs:boolean

est requise (*voir la capture d'écran ci-dessous*). Pour voir les informations, cliquez sur (Afficher infos) dans la barre d'outils.



### Que les livres après 1900

Le composant de filtre requiert une condition qui retourne soit true ou false. Lorsque la condition booléenne retourne true, les données de la séquence publication actuelle seront copiées sur la cible. Lorsque la condition retourne false, les données ne seront pas copiées. Dans ce tutoriel, la condition requise est de filtrer tous les livres qui ont été publiés après 1900. Pour créer la condition, procédez comme suit :

- 1. Cliquez sur **Constante** dans le menu **Insérer**. Ajouter une constante avec la valeur *1900*. Choisissez **Nombre** en tant que type.
- 2. Dans la fenêtre Libraries, situez la fonction greater et glissez-la sur le volet de mappage.
- 3. Effectuez les connexions de mappage de et vers la fonction greater, tel qu'indiqué ci-dessous. En faisant ceci, vous donnez l'instruction à MapForce de copier l'élément de source actuel <publication> vers l'élément <publication> dans le composant cible quand publish\_year est supérieur à 1900.



# 2.3.5 Prévisualiser et enregistrer la sortie

Vous êtes désormais prêt à consulter et enregistrer la sortie des deux composants cible. Lorsque des composants à cible multiple existent dans le même mappage, vous pouvez choisir lequel consulter en cliquant sur la touche **Preview** se trouve dans un état appuyé, elle indique que ce composant spécifique est actuellement activé pour une consultation et elle génèrera la sortie dans le volet **Sortie**. Il n'est possible d'activer le preview que d'un seul composant à la fois.

C'est pourquoi, lorsque vous souhaitez consulter et enregistrer la sortie du composant intermédiaire MergedLibrary, procédez comme suit :

- 1. Cliquez sur 🚾 dans le composant MergedLibrary.
- 2. Cliquez sur la touche **Sortie** située en bas du volet de **mappage**.
- 3. Dans le menu **Sortie**, cliquez sur **Enregistrer le fichier de Sortie** si vous souhaitez enregistrer la sortie dans un fichier.

Lorsque vous souhaitez consulter et enregistrer la sortie du composant FilteredLibrary :

- 1. Cliquez sur 🖻 dans le composant MergedLibrary.
- 2. Cliquez sur 👁 dans le composant FilteredLibrary.
- 3. Cliquez sur la touche **Sortie** située en bas du volet de **mappage**.
- 4. Dans le menu **Sortie**, cliquez sur **Enregistrer le fichier de Sortie** si vous souhaitez enregistrer la sortie dans un fichier.

Vous avez désormais terminer de concevoir le mappage qui a deux composants cible. Pour plus d'informations concernant le travail avec des composants pass-through, voir <u>Mappages en chaîne</u><sup>[34]</sup>. À votre convenance, le design de mappage est enregistré dans ce tutoriel en tant que <u>Tut3\_ChainedMapping.mfd</u>.

# 2.4 Sources multiples vers cibles multiples

Ce tutoriel vous montre comment mapper les données depuis des fichiers source multiples vers des fichiers cible multiples. Pour illustrer cette technique, nous allons créer un mappage avec les objectifs suivants :

- 1. Lire des données depuis des fichiers XML multiples dans le même répertoire.
- 2. Mapper le schéma de chaque fichier dans un nouveau schéma.
- 3. Pour chaque fichier source XML, générer un nouveau fichier cible XML avec un nouveau schéma.
- 4. Supprimer la déclaration XML et d'espace de noms des fichiers générés.

L'image ci-dessous illustre un modèle abstrait de la transformation de données utilisée dans le présent tutoriel :



### Point de départ

Nous utiliserons trois fichiers source XML en tant qu'exemples. Ils portent la dénomination BookTitle1.xml, BookTitle2.xml et BookTitle3.xml. Chacun des trois fichiers est basé sur Books.xsd et entrepose un seul livre (*voir ci-dessous*).

#### BookTitle1.xml

#### BookTitle2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
```

#### BookTitle3.xml

### Étapes supplémentaires

Nous allons procéder comme suit :

- 1. D'abord, nous devons mapper le schéma source (Books.xsd) au schéma cible (Library.xsd).
- Une fois la transformation effectuée, le mappage générera trois fichiers conformément à ce nouveau schéma (voir la liste de codes ci-dessous).
- 3. Nous allons également configurer le mappage de telle manière que les noms des fichiers générés portent les dénominations Publication1.xml, Publication2.xml et Publication3.xml. Veuillez noter que la déclaration XML et l'espace de noms doivent être supprimés.

#### Publication1.xml

```
library>
<publication>
<id>1</id>
<author>Mark Twain</author>
<title>The Adventures of Tom Sawyer</title>
<genre>Fiction</genre>
<publish_year>1876</publish_year>
</publication>
</library>
Publication2.xml
<library>
```

Publications.xml
<li>library&gt;</li>
<publication></publication>
<id>3</id>
<author>Herman Melville</author>
<title>Moby Dick</title>
<genre>Fiction</genre>
<publish_year>1851</publish_year>

Pour effectuer la transformation de données requise, suivez les étapes décrites dans les sous-sections cidessous.

# 2.4.1 Préparer le design de mappage

Le point de départ de ce tutoriel est semblable au mappage BooksToLibrary.mfd du premier tutoriel<sup>41</sup>, la différence étant qu'aucun fichier XML n'est fourni à cette étape. Notre objectif à cette étape est de mapper une structure abstraite (Books.xsd) avec une autre structure abstraite (Library.xsd). Afin de reproduire le scénario illustré dans la capture d'écran ci-dessous, suivez les étapes suivantes :

- 1. Ouvrez Books.xsd.
- 2. Lorsque MapForce propose d'ajouter un fichier XML échantillon, cliquez sur Skip.
- 3. Ouvrez Library.xsd.
- 4. Lorsque MapForce propose d'ajouter un fichier XML échantillon, cliquez sur Skip.
- 5. Connectez les nœuds respectifs, tel qu'affiché dans la capture d'écran ci-dessous.
- 6. Localisez la fonction current-dateTime dans la fenêtre Bibliothèques.
- 7. Faites glisser la fonction vers la zone de mappage et connectez result avec l'élément last\_updated.



# 2.4.2 Configurer l'entrée

Maintenant, nous voulons trois fichiers XML, chacun contenant un livre, basés sur le même schéma appelé Books.xsd. Ces trois fichiers XML, appelés BookTitle1.xml, BookTitle2.xml, et BookTitle3.xml seront utilisés en tant que fichiers d'entrée. Avant de modifier les paramètres de composant, enregistrez votre mappage en tant que fichier .mfd dans le dossier BasicTutorials. Pour donner l'instruction à MapForce de traiter les fichiers d'instance XML multiples, double-cliquez sur l'en-tête du composant source. Saisissez BookTitle\*.xml en tant que fichier d'entrée dans la boîte de dialogue Paramètres de composant L'astérisque (\*) contenu dans le nom du fichier donne l'instruction à MapForce d'utiliser tous les fichiers avec le préfixe BookTitle en tant qu'entrées de mappage. Étant donné que le chemin est relatif, MapForce recherchera tous les fichiers BookTitle dans le même répertoire que le fichier de mappage. Vous pouvez aussi saisir une chemin absolu, le cas échéant.

😢 Paramètres de composant					
Nom de composant: Books					
Fichier de schéma					
books.xsd	Parcourir	Éditer			
Fichier XML d'entrée					
MapForceExamples\Tutorial\BasicTutorials\Books.xml	Parcourir	Édition			
Fichier XML de sortie	Parcourir	Éditer			

# 2.4.3 Configurer la Sortie, Partie 1

Dans cette étape, nous allons créer le nom de fichier de chaque fichier de sortie. Pour ce faire, nous utiliserons la fonction concat, qui concatène (joint) toutes les valeurs qui lui sont fournies en tant qu'arguments. Lorsque ces valeurs sont réunies, elles créeront un nom de fichier de sortie (par ex., Publication1.xml). Pour générer les nom de fichier en utilisant la fonction concat, suivez les étapes suivantes :

### Ajouter la fonction de concaténation

Recherchez la fonction concat dans la fenêtre **Bibliothèques** et glissez-la dans la zone de mappage (*voir la capture d'écran ci-dessous*). Par défaut, cette fonction a deux paramètres quand elle est ajoutée au mappage. Dans notre exemple, nous avons besoin de trois paramètres. Cliquez sur • (Ajouter paramètre) à l'intérieur du composant de la fonction et ajoutez-y un troisième paramètre. Veuillez noter qu'en cliquant sur (Supprimer paramètre), ceci supprime un paramètre.

f <sup>/</sup> <sub>8</sub> conca	it
[Svalue1 ⊠	
[Svalue2 ⊠	result[:
[Svalue3 ⊠	

#### Insérer une constante

Pour ajouter une constante, cliquez sur **Constante** dans le menu **Insérer**. Lorsque vous êtes invité à fournir une valeur, saisissez publication et laissez l'option **String** inchangée. La constante publication fournit la valeur de string constante publication. Connectez la constante avec la value1 de la fonction concat, telle qu'affichée dans la capture d'écran ci-dessous :



### Fournir l'id

Connectez l'attribut id du composant source avec value2 de la fonction concat. L'attribut id du fichier XML source fournit une valeur d'identifiant unique pour chaque fichier. Cela sert à empêcher que tous les fichiers soient générés avec le même nom. La connexion devient rouge lorsque vous cliquez sur celle-ci.



#### Extraire l'extension de fichier

**Recherchez la fonction** get-fileext dans la fenêtre **Bibliothèques** et glissez-la dans la zone de mappage. Créez une connexion depuis le nœud supérieur du composant source (Fichier : Books.xml) avec le paramètre filepath de cette fonction. Puis, connectez le paramètre extension de la fonction get-fileext value3 à la fonction concat. Ce faisant, vous extrayez uniquement la partie de l'extension (dans ce cas, .xml) depuis le nom de fichier source et vous la passez au nom de fichier de sortie.



# 2.4.4 Configurer la Sortie, Partie 2

Vous pouvez désormais donner l'instruction à MapForce de créer les noms du fichier lors de l'exécution du mappage. Pour ce faire, nous utiliserons des noms de fichier dynamiques (*voir les sous-sections ci-dessous*).

### Noms de fichier dynamiques

À cette étape, nous donnons l'instruction à MapForce de générer les fichiers d'instance de manière dynamique, ce qui signifie que chaque fichier de sortie recevra son nom basé sur les arguments fournis à la fonction **concat**. Pour ce faire, cliquez sur la touche <u>File</u> ou <u>File/String</u> du composant cible et sélectionnez **Utiliser les noms de fichier dynamiques fournis par les mappages**.

dibrary	. 10		
		U <u>t</u> iliser les noms de fichier depuis les paramètres de composant	
Ast_updated     Ast_updated     Ast_updated     Ast_updated     Ast_updated     Ast_updated     Ast_updated     Ast_updated     Ast_updated	~	Uti <u>l</u> iser les noms de fichier dynamiques fournis par les mappages <u>P</u> arser chaînes vers XML S <u>é</u> rialiser XML vers chaînes	
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓		Supprimer les co <u>n</u> nexions Créer composant varia <u>b</u> le	*

### Connecter la fonction avec le nœud dynamique

Nous allons donc connecter le résultat de la fonction concat avec le Fichier : Nœud <dynamic> du composant cible.



### Vérifier les paramètres de composant

Si vous double-cliquez sur l'en-tête du composant cible cette fois-ci, vous remarquerez que les zones de texte **Fichier XML d'entrée** et **Fichier XML de sortie** sont désactivées, et leur valeur affiche *<Noms de fichier fournis par le mappage>(voir la capture d'écran ci-dessous)*. Ceci est une indication que vous avez fourni les noms de fichier d'instance de manière dynamique du mappage. Pour cette raison, il n'est plus pertinent de les définir dans les paramètres de composant.
😢 Paramètres de composant			$\times$
Nom de composant: Library			
Fichier de schema MapForceExamples\Tutorial\BasicTutorials\Librarly.xsd	Parcourir	Éditer	
Fichier XML d'entrée <noms de="" fichier="" fournis="" le="" mappage="" par=""></noms>	Parcourir	Édition	
Fichier XML de sortie <noms de="" fichier="" fournis="" le="" mappage="" par=""></noms>	Parcourir	Éditer	

Vous pouvez à présent exécuter le mappage et voir le résultat, ainsi que le nom des fichiers générés. Ce mappage génère de multiples fichiers de sortie. Vous pouvez naviguer à travers les fichiers de sortie en utilisant les touches gauche et droite dans le coin supérieur gauche du volet **Sortie** ou en choisissant un fichier depuis la liste déroulante (*voir la capture d'écran ci-dessous*).



À votre convenance, le design de mappage est enregistré dans ce tutoriel en tant que **Tut4\_MultipleToMultiple.mfd**.

# 3 Notions fondamentales de mappage

Un design de mappage MapForce (ou simplement un "mappage") est la représentation visuelle de la manière dont les données sont transformées d'un format à un autre. Un mappage consiste en des *composants* que vous ajoutez à la zone de mappage afin de créer vos transformations de données. Un mappage valide consiste en un ou plusieurs *composants de source* connectés à un ou plusieurs *composants cibles*. Vous pouvez exécuter un mappage et consulter son résultat directement dans MapForce. Vous pouvez générer du code et l'exécuter extérieurement. Vous pouvez aussi compiler un mappage dans un fichier d'exécution MapForce et automatiser l'exécution de mappage en utilisant, par exemple <u>MapForce Server</u> ou <u>FlowForce Server</u>. MapForce enregistre les mappages en tant que fichiers portant l'extension .mfd.

La capture d'écran ci-dessous illustre la structure basique d'un mappage de :



#### Nouveau mappage

Pour créer un nouveau mappage, cliquez sur (Nouveau) dans la barre d'outils. En alternative, cliquez sur Nouveau dans le menu Fichier. Puis sélectionnez Mappage et cliquez sur OK. La prochaine étape consistera à ajouter des composants<sup>78</sup> au mappage et à créer les connexions<sup>87</sup>.

# Parties principales d'un mappage

Les sous-sections ci-dessous décrivent les parties principales d'un design de mappage. <u>Composant</u>

Dans MapForce, le terme *composant* est ce qui représente visuellement la structure de vos données, ou indique la manière dont les données doivent être transformées. Les composants sont les pièces centrales de tout mappage et sont représentés en tant que boîtes rectangulaires. Les composants peuvent être divisés en deux grands groupes :

- Les composants source et cible ;
- Les composants de <u>structure</u><sup>120</sup> et de <u>transformation</u><sup>476</sup>.

Veuillez noter que ces deux groupes ne sont pas mutuellement exclusifs. Le premier groupe reflète les relations entre les composants ; par ex., un composant peut être la source pour un composant et la cible pour

un autre composant. MapForce lit les données depuis un composant source et écrit ces données dans un composant cible. Lorsque vous exécutez un mappage, le composant cible instruit MapForce soit à générer un fichier (ou plusieurs fichiers), soit à sortir le résultat en tant que valeur de string pour un traitement ultérieur dans un programme externe. Les types de composants du premier groupe sont décrits ci-dessous :

- Une source est située à gauche du composant cible. MapForce lit les données depuis la source.
- Une cible est située à droite de la source. MapForce écrit les données dans le composant cible.
- Un composant *pass-through* est un sous-type de composants source et cible. Un composant pass-through agit en tant que source et cible. Pour plus d'information, voir les <u>Mappages en chaîne</u><sup>[64]</sup>. Veuillez noter que seuls les composants de structure peuvent être pass-through.

Le deuxième groupe affiche si un composant est doté d'une structure de données ou s'il est utilisé pour transformer les données mappées d'un autre composant.

Pour en savoir plus sur les composants et actions liées au composant, voir Components 76.

#### **Connecteur**

Un connecteur est un petit triangle affiché sur le côté gauche ou droite d'un composant. Les connecteurs affichés à gauche d'un composant fournissent des points d'entrée de données à ce composant. Les connecteurs affichés à droite d'un composant fournissent des points de sortie de données *de ce composant*.

#### <u>Connexion</u>

Une connexion est une ligne que vous pouvez tirer entre deux connecteurs. En tirant des connexions, vous instruisez MapForce de transformer des données d'une manière spécifique (par exemple, lire des données depuis un document XML et de l'écrire dans un autre document XML).

## Dans cette section

Cette section décrit les tâches et concepts les plus communs de MapForce. La section est organisée en sous-sections comme suit :

- <u>Composants</u><sup>76</sup>
- <u>Connexions</u><sup>87</sup>
- Procédures générales et fonctions<sup>103</sup>
- Projets<sup>115</sup>

# 3.1 Composants

Les composants sont des éléments centraux de tout design de mappage dans MapForce. Visuellement, les composants sont représentés comme boîtes rectangulaires dans la zone de mappage. Cette rubrique donne un aperçu des composants structurels et de transformation (*voir exemple ci-dessous*). La distinction est basée sur le fait si un composant a une structure de données ou s'il est utilisé pour transformer des données. Voir la description de ces deux types dans les sous-sections ci-dessous. Voir aussi Notions fondamentales de mappage

#### Exemple de composants

L'exemple de mappage ci-dessous illustre deux composants de source de données (Books et Library), un composant cible de données (MergedLibrary), et un composant de transformation (la fonction current-dateTime).



# Composants de structure

Les composants structurels représentent une structure abstraite de vos données (par ex., un fichier XML). La liste des composants structurels qui peuvent être utilisés dans les sources et cibles de données se trouve sous <u>Composants structurels</u><sup>(20)</sup>. Les composants structurels peuvent lire les données depuis les fichiers ou d'autres sources, écrire des données dans les fichiers ou d'autres sources, ou encore stocker des données à des étapes intermédiaires dans le processus de mappage (par ex., afin de prévisualiser des données). La table ci-dessous donne un aperçu des composants de structure et de leurs boutons de barre d'outils respectifs.

Icône	Description
떒	Composant XML

Icône	Description
	Composant de texte (éditions Professional et Enterprise)
•	Composant de base de données ( <i>éditions</i> <i>Professional et Enterprise</i> pour les bases de données SQL ; <i>Enterprise Edition</i> pour les bases de données NoSQL)
JSON	Composant JSON (Enterprise Edition)
XI	Composant Microsoft Excel (Enterprise Edition)
	Composant WSDL (Enterprise Edition)
EDI	Composant EDI (Enterprise Edition)
XBRL	Composant XBRL (Enterprise Edition)
101 010	Protocol Buffers (Enterprise Edition)

# Composants de transformation

Les composants de transformation vous aide à <u>transformer les données</u><sup>(51)</sup>, <u>stocker un résultat de mappage</u> <u>intermédiaire</u><sup>(32)</sup> pour un traitement ultérieur, <u>remplacer une valeur par une autre valeur</u><sup>(50)</sup>, <u>trier</u><sup>(517)</sup>, <u>grouper</u><sup>(540)</sup>, <u>joindre</u><sup>(505)</sup>, et <u>filtrer</u><sup>(523)</sup> vos données. Vous pouvez également ajouter une <u>exception</u><sup>(547)</sup>, qui interrompt le processus de mappage et affiche une erreur quand une condition définie par un filtre apparaît. La table cidessous donne un aperçu des composants de transformation et de leurs boutons de barre d'outils respectifs.

Icône	Description
	Entrée simple
♪	Sortie simple
Þ	Composant de filtre
az A‡	Composant de tri
<u>f/</u>	Fonction intégrée
۵	Fonction définie par l'utilisateur
(A:Z)	Composant SQL/NoSQL-WHERE/ORDER (éditions Professional et Enterprise)
1	Composant Value- Map
VAB	Variable
<b>"</b>	Fonction de service Web (Enterprise Edition)
۲	Exception (éditions Professional et Enterprise)

Icône	Description
: * *	Constante
te J	Condition If-Else
N	Composant Join (éditions Professional et Enterprise)

## Dans cette section

Cette section donne un aperçu des composants et est organisée comme suit :

- <u>Ajouter des composants</u>
   <sup>78</sup>
- Les bases de composant<sup>80</sup>
- Chemins de fichier
   <sup>82</sup>

# **3.1.1** Ajouter des composants au mappage

Cette rubrique explique comment ajouter des composants au mappage. Pour ajouter un composant, vous allez d'abord devoir <u>créer un nouveau design de mappage</u><sup>74</sup>, ensuite procédez comme suit :

- Dans le menu Insérer, choisissez un type de composant (par ex., XML Schéma/Fichier).
- Glissez un fichier depuis Windows File Explorer dans la zone de mappage. Veuillez noter que cette opération est uniquement possible pour les composants basés sur fichiers compatibles.
- Cliquez sur le bouton pertinent dans la barre d'outils Insérer Composant (voir la capture d'écran cidessous)



Chaque type de composant a un but et une fonction spécifiques. Pour obtenir un aperçu des composants, voir <u>Composants</u><sup>76</sup>. Si vous souhaitez en savoir plus sur les structures de données qui peuvent être utilisées comme sources et cibles de, voir les <u>composants de structure</u><sup>120</sup>. Pour des informations concernant les composants intégrés de MapForce utilisés pour stocker des données temporairement ou les transformer, voir <u>Composants de transformation</u><sup>476</sup>.

Pour une liste complète de composants, voir la référence d'icône de composant ci-dessous.

## Ajouter des composants depuis l'URL

En plus d'ajouter des fichiers locaux en tant que composants de mappage, vous pouvez ajouter des fichiers depuis une URL. Veuillez noter que cette opération est prise en charge lorsque vous ajoutez un fichier en tant que <u>composant de source</u>. Les protocoles pris en charge sont HTTP, HTTPS et FTP. Pour ajouter un composant depuis un URL, suivez les étapes suivantes :

- 1. Dans le menu **Insérer**, sélectionnez le type de composant que vous souhaitez ajouter (par ex., **XML Schéma/Fichier**).
- 2. Dans la boîte de dialogue **Ouvrir** (*voir la capture d'écran ci-dessous*), cliquez sur **Basculer vers** I'URL.

 Saisir l'URL du fichier dans le champ de saisie URL du fichier, et cliquez sur Ouvrir. Assurez-vous que le type de fichier dans le champ de saisie URL du fichier est le même que le type de fichier que vous avez spécifié à l'étape 1.

Open	<b>—</b>
File URL: http:// /Documents/BookLi	ist.xsd 🗸 🗸
Open as: Auto	File load Use cache/proxy
User:	<ul> <li>Remember password between application starts</li> </ul>
Available files Server URL: http://	■ Browse
Cache Profiles	Ē
BookList.xml	
Datal Check Out	
Empl Check In	+
Undo Check Out	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

La liste ci-dessous décrit les options disponibles dans le dialogue **Ouvrir**.

- Souvenez-vous du mot de passe : Si le serveur nécessite une authentification de mot de passe, vous serez invité à saisir le nom d'utilisateur et le mot de passe. Si vous souhaitez que le nom d'utilisateur et le mot de passe soit retenu la prochaine fois, démarrez MapForce, saisissez votre nom utilisateur et mot de passe dans la boîte de dialogue et sélectionnez la case à cocher Se rappeler du mot de passe entre les lancements d'application.
- Ouvrir comme : Définit la grammaire pour le parseur. L'option par défaut et recommandée est Auto.
- Utiliser cache/proxy : Si le fichier que vous chargez ne changera probablement pas, sélectionnez l'option Utiliser cache/proxy pour mettre en cache les données et accélérer le chargement du fichier. Sinon, sélectionnez Recharger si vous voulez recharger le fichier à chaque fois que vous ouvrez le mappage.
- Serveur URL : Pour des serveur qui prennent en charge Web Distributed Authoring et Versioning (WebDAV), vous pouvez chercher des fichiers une fois après avoir saisi l'URL de serveur dans le champ de saisie URL Serveur et cliquez sur Parcourir. Bien que l'aperçu montre tous les types de fichier, assurez-vous de choisir le même type de fichier à l'étape 1 ci-dessus. Autrement, des erreurs surviendront.

 Check in/out : Si le serveur est un Microsoft SharePoint Server, sélectionnez la case à cocher II s'agit d'un serveur Microsoft® SharePoint®. Cela affichera l'état check-in ou check-out du fichier dans la zone d'aperçu. Si vous souhaitez vous assurer que personne ne puisse éditer le fichier sur le serveur pendant que vous l'utilisez, cliquez avec la touche de droite sur le fichier et sélectionnez Check Out (voir la capture d'écran ci-dessous). Pour faire le check-in de tout fichier préalablement en « checkedout », cliquez avec la touche de droite sur le fichier et sélectionnez Check In.

# 3.1.2 Les bases de composant

Cette rubrique explique comment définir, chercher et manipuler les <u>composants de structure</u><sup>76</sup>. Pour plus d'information, veuillez voir les sous-sections ci-dessous.

# Changer les paramètres de composant

Après avoir ajouté un composant dans la zone de mappage, vous pouvez configurer son paramètres applicables depuis le dialogue **Paramètres de composant**. Vous pouvez ouvrir le dialogue **Paramètres de composant** d'une des manières suivantes :

- Double-cliquez sur l'en-tête de composant.
- Sélectionner le composant et cliquez sur les Propriétés dans le menu Composant.
- Cliquez avec la touche de droite sur l'en-tête du composant, cliquez sur Propriétés.

Veuillez noter que les options disponibles dépendent du type de composant. La liste des paramètres pour chaque type de composant est donnée ci-dessous :

- Paramètres de composant XML<sup>122</sup>
- Paramètres de composant de base de données 281
- Paramètres de composant CSV<sup>455</sup>
- Paramètres de composant du Champ à longueur fixe<sup>463</sup>

Pour chaque composant basé sur fichier (par ex., un fichier XML) la touche <u>File</u> (*Édition de base*) ou <u>File/String</u> (*éditions Professional et Enterprise*) apparaît à côté du nœud racine. Ce bouton spécifie des options avancées pour traiter ou générer de multiples fichiers en un seul mappage. Pour plus d'informations, voir <u>Traiter plusieurs</u> <u>fichiers d'entrée ou de sortie</u><sup>633</sup>. Ce bouton vous permet également de définir les options pour les <u>strings de</u> <u>parsage et les données de sérialisation vers les strings</u><sup>604</sup>.

# Chercher un composant

Pour rechercher un nœud spécifique dans un composant, suivez les étapes suivantes :

- 1. Cliquez sur le composant que vous souhaitez parcourir et appuyer sur les touches CTRL+F.
- 2. Saisissez un terme de recherche et cliquez sur **Trouver suivant/précédent/tout** (*voir la capture d'écran ci-dessous*).



Utiliser les options **Avancé** pour définir quel(s) item(s) (nœuds) doivent être recherchés. Vous pouvez aussi restreindre les options de recherche basées sur les connexions spécifiques.

## Aligner des composants

Lorsque vous déplacez des composants dans la zone de mappage, MapForce affiche des repères (lignes pointillées) qui vous aident à aligner des composants (*voir la capture d'écran ci-dessous*).



Pour activer cette option, suivez les étapes ci-dessous :

- 1. Allez au menu Outils et cliquez sur Options.
- 2. Dans le groupe Édition, cochez la case à cocher Aligner les composants sur glissage de souris.

# Dupliquer l'entrée

Parfois, vous devez configurer un composant pour qu'il puisse accepter les données provenant de plusieurs sources. Si vous voulez que le schéma cible accepte les données de plus d'un schéma source, vous pouvez

dupliquer tout nœud d'entrée dans votre composant cible. La duplication d'entrées n'a de sens que pour un composant cible : Les nœuds dupliqués ne peuvent accepter que des données, mais il n'est pas possible de mapper des données depuis des nœuds dupliqués. Vous pouvez dupliquer autant de nœuds que vous souhaitez.

Il existe deux manières de dupliquer l'entrée : (i) en sélectionnant **Ajouter double entrée Avant/ Après** depuis le menu contextuel et (ii) en connectant un nœud source avec un nœud cible, qui est déjà connecté à un nœud différent. La première option est décrite ci-dessous. Des informations sur la deuxième option peuvent être trouvées dans <u>le deuxième tutoriel</u><sup>51</sup>.

#### Ajouter double entrée Avant/Après

Pour dupliquer un nœud d'entrée particulier, cliquez dessus avec la touche de droite et sélectionnez **Ajouter double entrée Avant/ Après** depuis le menu contextuel (*voir la capture d'écran ci-dessous*). Dans l'image cidessous, le nœud author est en train d'être dupliqué de manière à ce que les données puissent être mappées vers le nœud dupliqué d'une autre source de nœud.



**Note :** la duplication d'attributs XML n'est pas permise, puisqu'elle rendra l'instance XML invalide.

# 3.1.3 Chemins de fichier

Un fichier de design de mappage (\*.mfd) peut contenir des références à plusieurs schémas et fichiers d'instance. MapForce utilise les fichiers de schéma afin de déterminer la structure des données à mapper. Dans les éditions MapForce Professional et Enterprise, les mappages peuvent aussi contenir des références aux fichiers StyleVision Power Stylesheets (\*.sps), utilisés pour formater des données pour les sorties comme PDF, HTML et Word. Les mappages peuvent aussi contenir des références à des bases de données basées sur des fichiers comme Microsoft Access ou SQLite.

Les références à des fichiers sont créées par MapForce lorsque vous ajoutez un composant au mappage. Néanmoins, vous pouvez toujours définir ou changer les références de chemin manuellement, le cas échéant.

Cette section fournit des instructions sur comment définir ou changer les chemins vers différents types de fichier référencés par un mappage. La section est organisée en rubriques suivantes :

- <u>Chemins relatifs et absolus</u>
   <sup>83</sup>
- Les chemins dans les environnements d'exécution<sup>85</sup>

# 3.1.3.1 Chemins relatifs et absolus

Cette rubrique explique comment utiliser les chemins absolus et relatifs des fichiers référencés par votre composant. Un chemin absolu affiche tout l'emplacement d'un fichier, en commençant par un répertoire racine (*voir Exemple :. Paramètres de composant XML*). Un chemin relatif affiche l'emplacement du fichier qui est relatif au répertoire de travail actuel : par ex., Books.xml.

Dans la boîte de dialogue des **Paramètres de composant** (*voir l'exemple ci-dessous*), vous pouvez spécifier les chemins absolus ou relatifs pour de nombreux fichiers référencés par le composant. La liste des fonctions d'expression est indiquée ci-dessous :

- Fichiers d'entrée à partir desquels MapForce lit des données ;
- Fichiers de sortie vers lesquels MapForce écrit des données ;
- Fichiers de schéma qui s'appliquent aux composants avec un schéma ;
- Fichiers de structure utilisés par des paramètres d'entrée ou de sortie des fonctions définies par l'utilisateur et des variables ;
- Fichiers StyleVision Power Stylesheet (\*.sps) utilisés pour formater des données pour les sorties comme les PDF, HTML et Word.
- Les fichiers de base de données dans le cas de composants de base de données (éditions *Professional et Enterprise*).

#### Copier-coller et chemins relatifs

Lorsque vous copiez un composant depuis un mappage et que vous le collez dans un autre mappage, MapForce vérifie si les chemins relatifs de fichiers de schéma peuvent être résolus par rapport au dossier du mappage de destination. Si le chemin ne peut pas être résolu, vous serez invité à rendre les chemins relatifs en absolu.

#### Chemins cassés

Lorsque vous ajoutez ou modifiez une référence de fichier dans un mappage, et que le chemin ne peut pas être résolu, MapForce affiche un message d'avertissement. Les références de chemin cassé peuvent arriver dans les cas suivants :

- Vous utilisez des chemins relatifs, puis déplacez le fichier de mappage vers un nouveau répertoire sans déplacer les fichiers de schéma et d'instance.
- Vous utilisez des chemins absolus vers des fichiers dans le même répertoire que le fichier de mappage, puis déplacez le répertoire vers un autre emplacement.

Lorsqu'un de ces scénarios arrivent, MapForce marque le composant en rouge. La solution dans ce cas est de double-cliquer sur l'en-tête de composant et de mettre à jour toutes les références de chemin cassés dans le dialogue **Paramètres de composant**. Voir aussi <u>Changer les paramètres de composant</u><sup>(80)</sup>.

# Exemple : Composant XML

L'exemple ci-dessous affiche comment les chemins de fichier sont utilisés dans un composant XML. Si vous souhaitez enregistrer tous les fichiers relatifs au mappage dans le fichier de mappage (.mfd), vérifiez la boîte *Enregistrer tous les chemins de fichier relatifs au fichier MFD* dans le bas de la boîte de dialogue **Paramètres** 

**de composant**. Il s'agit d'une option recommandée et par défaut qui affecte tous les fichiers référencés par le composant (*affiché dans le cadre rouge dans l'image ci-dessous*). Si vous n'avez pas encore enregistré votre mappage, vous verrez des chemins absolus au schéma et/ou des fichiers d'instance dans la boîte de dialogue **Paramètres de composant**. Pour voir les chemins relatifs dans la boîte de dialogue **Paramètres de composant**, suivez les étapes suivantes :

- 1. <u>Créer un nouveau mappage</u><sup>74</sup> et <u>ajouter un composant de structure</u><sup>78</sup> : par ex., un fichier XML avec un schéma XML y assigné.
- 2. Double-cliquez sur l'en-tête du composant pour ouvrir une boîte de dialogue **Paramètres de composant**.
- 3. Vérifier la boîte *Enregistrer tous les chemins de fichier relatifs au fichier MFD* en bas de la boîte de dialogue **Paramètres de composant**.
- 4. Enregistrer votre mappage.
- 5. Vous pouvez ouvrir maintenant une nouvelle fois les **Paramètres de composant** afin de vérifier les chemins relatifs dans les champs de texte pertinents.

Note : les chemins qui référencent un pilote non-local ou utilisent une URL ne seront par rendus relatifs.

😢 Paramètres de composant		×
Nom de composant: Books		
Fichier de schéma Jsers\ <username>\Documents\Altova\MapForce2022\MapForceExamples\Tutorial\BasicTutorials\Books.xsd Fichier XML d'entrée</username>	Parcourir Éditr	er
Jsers\ <username>\pocuments\Altova\MapForce2022\MapForceExamples\Tutorial\BasicTutorials\books.xml</username>	Parcourir Éditio	on
Fichier XML de sortie	Parcourir Édit	er

Quand la case à cocher *Enregistrer tous les chemins de fichier relatifs au fichier MFD* est sélectionnée, MapForce assurera le suivi des fichiers référencés par le composant même si vous enregistrez le mappage dans un nouveau dossier. Si tous les fichiers se trouvent dans le même répertoire que le mappage, les références de chemin ne seront pas brisées lorsque vous déplacez le répertoire complet vers un nouvel emplacement sur le disque.

Le paramètre Enregistrer tous les chemins de fichier relatifs au fichier MFD s'applique aux fichiers suivants :

- Fichiers de structure utilisés par des paramètres complexes d'entrée ou de sortie des fonctions définies par l'utilisateur et des variables de type complexe ;
- Fichiers plats d'entrée ou de sortie (éditions Professional et Enterprise) ;
- Fichiers de schéma référencés par les composants de base de données qui prennent en charge des champs XML (éditions Professional et Enterprise);
- Fichiers de base de données (éditions Professional et Enterprise);
- XBRL d'entrée ou de sortie, FlexText, EDI, Excel 2007+, fichiers JSON (*uniquement Enterprise Edition*).

# Exemple : Composant de base de données (éditions Professional et Enterprise)

Quand vous ajoutez un fichier de base de données tel que Microsoft Access ou SQLite au mappage, vous pouvez saisir un chemin relatif à la place d'un chemin absolu dans la boîte de dialogue **Sélectionner une** 

base de données (voir la capture d'écran ci-dessous). Avant de saisir des chemins de fichier relatifs, assurezvous d'enregistrer d'abord le fichier de mappage .mfd. Si vous voulez changer le chemin d'un composant de données qui est déjà dans le mappage, cliquez sur **Changer** dans la boîte de dialogue **Paramètres de composant**. Pour plus d'information sur la connexion à une source de base de données, voir <u>Démarrer</u> l'assistant de connexion de la base de données

😢 Choisir une bas	e de données	$\times$
Arristant	Connecter à SQLite	
Connexions existantes	Cliquer sur Parcourir et choisir une base de données SQLite. Cliquer ensuite sur 'Se connecter' pour poursuivre.	
(ADD)	Nanonull.sqlite Parcourir	

**Note**: lorsque vous générez un code de programme, compilez des fichiers d'exécution MapForce Server (.mfx), ou déployez le mappage sur <u>FlowForce Server</u>, un chemin relatif sera converti dans un chemin absolu si vous sélectionnez la case à cocher *Rendre les chemins absolus dans le code généré* dans les paramètres de mappage<sup>(112)</sup>. Pour en savoir plus, voir <u>À propos des chemins dans le code généré</u><sup>(85)</sup>.

# 3.1.3.2 Les chemins dans des environnement d'exécution différents

Si vous générez du code depuis les mappages, compilez les mappages vers les fichiers d'exécution de <u>MapForce Server</u> (.mfx), ou déployez les mappages vers <u>FlowForce Server</u>, les fichiers générés sont exécutés dans l'environnement cible de votre choix : par exemple, <u>RaptorXML Server</u>, <u>MapForce Server</u>, ou une application C#. Pour que le mappage puisse être exécuté avec succès, tout chemin relatif doit être explicite dans l'environnement dans lequel le mappage est exécuté. Les chemins de base pour chaque langage cible sont indiqués ci-dessous :

Langage cible	Chemin de base	
XSLT, XSLT2, XSLT3	Chemin du fichier XSLT.	
XQuery*	Chemin du fichier XQuery.	
C++, C#, Java*	Répertoire de travailde l'application générée.	
Built-in* (pour les aperçus du mappage dans MapForce)	Chemin du fichier de mappage (.mfd).	
Built-in* (pour l'exécution du mappage avec MapForce Server)	Répertoire de travail actuel.	

Langage cible	Chemin de base	
Built-in* (pour l'exécution du mappage avec MapForce Server sous le contrôle de FlowForce Server)	Répertoire de travail de la tâche ou répertoire de travail de FlowForce Server.	

\* Langages disponibles dans MapForce Professional et Enterprise editions

#### Chemin relatif vers chemin absolu

Lorsque vous générez un code de programme, compilez des fichiers d'exécution MapForce Server (.mfx), ou déployez le mappage sur <u>FlowForce Server</u>, un chemin relatif sera converti dans un chemin absolu si vous sélectionnez la case à cocher **Rendre les chemins absolus dans le code généré** dans les <u>paramètres de mappage</u><sup>112</sup>.

Lorsque vous générez un code et que la case à cocher a été sélectionnée, MapForce résout tout chemin relatif basés sur le répertoire du fichier de mappage .mfd, et les rend absolus dans le code généré. Ce paramètre a une incidence sur les fichiers suivants :

- Fichiers d'instance d'entrée et de sortie pour des composants basés sur le fichier ;
- Fichiers Access et base de données SQLite utilisés comme composants de mappage (éditions *Professional et Enterprise*).

## Chemins de bibliothèque dans du code généré

Les fichiers de mappage peuvent contenir en option des références de chemin à des bibliothèques de types variés. Par exemple, vous pouvez importer des fonctions définies par l'utilisateur depuis un autre fichier de mappage, depuis XSLT personnalisé, XQuery\*, C#\* ou des bibliothèques Java\*, ou depuis des fichiers .mff\* (fonction de MapForce). Pour plus d'informations, voir <u>Gérer les Bibliothèques de fonction</u><sup>655</sup>.

#### \* Fonctions disponibles dans les éditions MapForce Professional et Enterprise

L'option **Rendre les chemins absolus dans le code généré** ne s'applique qu'aux composants de mappage, et elle ne touche pas les chemins menant aux bibliothèques externes. Pour toutes les bibliothèques différentes de XSLT et XQuery, le chemin de bibliothèque sera converti en un chemin absolu dans le code généré. Par exemple, si votre fichier de mappage contient des références de bibliothèque comme des fichiers .NET.dll ou Java .class, et que vous souhaitez exécuter le code généré dans un autre environnement, les bibliothèques référencées doivent exister dans le même chemin dans l'environnement cible.

Si vous prévoyez de générer un fichier XSLT ou XQuery depuis un mappage, vous pouvez rendre le chemin de bibliothèque relatif au fichier XSLT ou XQuery généré, comme suit :

- 1. Ouvrez les paramètres de mappage<sup>112</sup>.
- 2. Sélectionnez la case à cocher **Bibliothèques de référence avec des chemins relatifs aux fichiers XSLT/XQuery générés.** Assurez-vous que la bibliothèque XSLT ou XQuery existe dans ce chemin.

# 3.2 Connexions

Une connexion est une ligne qui connecte une source vers une cible. Les connexions représentent visuellement comment les données sont mappées d'un nœud vers un autre. Les sous-sections ci-dessous décrivent différentes actions liées à la connexion que vous pouvez effectuer.

## Créer, copier, supprimer une connexion

Pour créer une connexion entre deux lignes, appuyez et restez sur le <u>connecteur de sortie</u><sup>75</sup> d'un nœud source et glissez-le vers un nœud de destination. Un connecteur d'entrée accepte *une seule* connexion. Si vous tentez d'ajouter une deuxième connexion à la même entrée, MapForce vous demandera de remplacer la connexion par une nouvelle ou de <u>dupliquer l'item d'entrée</u><sup>81</sup>. Un connecteur de sortie peut avoir plusieurs connexions, chacune vers une entrée différente.

Pour copier une connexion vers un item différent, appuyez et tenez appuyée la section grasse à la fin de la connexion (*voir la capture d'écran dans Déplacer une connexion*) et glissez-la vers la destination sélectionnée tout en tenant la clé **Ctrl** appuyée.

Pour supprimer un connexion, cliquez sur la connexion et appuyez sur la touche **Supprimer**. De manière alternative, cliquez avec la touche de droite sur la connexion et sélectionnez **Supprimer** dans le menu contextuel.

#### Entrées obligatoires

Pour vous aider dans le processus de mappage, MapForce marque les entrées obligatoires en orange dans les composants cibles. L'exemple ci-dessous vous montre que dès que vous vous connectez à l'élément book du composant Books de l'élément publication du composant BookOutput, les connecteurs des nœuds obligatoires du composant BooksOutput seront mis en surbrillance. Si vous ne connectez pas d'entrées obligatoires, les nœuds respectifs ne seront pas mappés vers le cible et le mappage sera invalide.



#### Connexions parent manquantes

Lorsque vous créez des connexions entre les nœuds source et cible manuellement, MapForce analyse automatiquement les résultats possibles de mappage. Si vous vous connectez aux deux nœuds enfant sans les connecter à leurs nœuds parent, vous verrez un message de notification qui suggère la connexion du parent du nœud source au parent du nœud cible. Ce message de notification vous aide à éviter des situations dans lesquelles un seul nœud enfant apparaît dans le volet **Sortie**.

Si vous souhaitez déactiver de telles notifications, suivez les étapes suivantes :

- 1. Allez au menu Outils et cliquez sur Options.
- 2. Ouvre le groupe Messages.
- 3. Effacez la case à cocher **quand vous créez une connexion, suggérez la connexion d'items** ancêtres.

# Déplacer une connexion

Pour déplacer une connexion vers un nœud différent, appuyez et tenez appuyée la section épaisse à la fin de la connexion (*voir la capture d'écran ci-dessous*) et glissez-la vers la destination sélectionnée.



# Voir les info-bulles de connexion

Les info-bulles de connexion vous permettent de voir les noms des (i) nœuds auxquels les données sont mappées ou (ii) les nœuds depuis lesquels les données sont mappées. Pour voir les conseils, appuyez sur le bouton de la barre d'outils (Afficher conseils). Pour voir les noms des nœuds auxquels les données sont mappées, pointez le curseur sur la section épaisse d'une connexion près du connecteur de sortie (*voir la capture d'écran ci-dessous*). Pour voir le nom du nœud duquel les données sont mappées, pointez le curseur sur la section près d'un connecteur d'entrée. Si de multiples connexions ont été définies depuis la même sortie, un maximum de dix noms d'item seront affichés dans l'info-bulle.

Dans l'exemple ci-dessous, le nœud cible duquel les données de l'élément book sont mappées est appelé publication.



# Changer les paramètres de connexion

Pour changer les paramètres de connexion, suivez une des étapes suivantes :

- Sélectionner une connexion. Puis, allez au menu Connexion et cliquez sur Propriétés.
- Double-cliquer sur la connexion.

• Cliquez avec la touche de droite sur la connexion, puis cliquez sur Propriétés.

Pour plus d'informations, voir <u>Paramètres de connexion</u><sup>97</sup>.

## Mettre en surbrillance les connexions de manière sélective

MapForce vous permet de marquer les connexions de manière sélective dans un mappage. Cette fonction peut être utile quand votre mappage a de nombreux composants avec de multiples connexions. Marquer des connexions rendra la vérification plus facile, c'est-à-dire vérifier si les nœuds du composant sélectionné sont mappés correctement. Veuillez noter que le terme *connecteur* utilisé pour les touches de la barre d'outils se réfère à une connexion, c'est-à-dire une ligne connectant des nœuds de composant. Voir les options disponibles ci-dessous.

	Afficher des connecteurs de composant sélectionnés (seulement des connexions directes)
R.	Afficher des connecteurs de la source vers le cible (connexions directes et indirectes)

#### Uniquement des connexions directes

Quand le bouton **Direct-connections** n'est *pas* appuyé, vous pouvez voir toutes les connexions en noir. Quand le bouton **Direct-connections** est appuyé, seules les connexions liées au composant sélectionné actuellement sont en noir. Les autres connexions sont en gris clair.

#### Connexions directes et indirectes

Le bouton **Connexions source-à-cible** devient disponible uniquement quand le bouton **Direct-connections** est appuyé. Quand le bouton **Connexions source-à-cible** est appuyé, vous pouvez tracer des connexions du composant sélectionné actuellement, y compris ses connexions directes et les connexions de ses composants connectés jusqu'aux fichiers source et cible.

Pour comprendre comment ces deux options fonctionnent, voir l'exemple ci-dessous.

#### <u>Exemple</u>

La capture d'écran ci-dessous illustre la partie du mappage ChainedPersonList.mfd, qui est disponible dans le dossier MapForceExamples. Dans le mappage ci-dessous, nous avons appuyé sur le bouton Directconnections, cliqué sur l'en-tête du composant concat, mais nous n'avons pas encore appuyé sur le bouton Connexions source-à-cible. Pour cette raison, nous voyons que seules les connexions directes de la fonction concat à la "-" constante, au substring, à la position, et aux composants Contacts sont noirs. Les autres connexions dans le mappage sont en gris clair.



La prochaine étape est d'appuyer sur le bouton **Connexions source-à-cible**. La capture d'écran ci-dessous reflète les changements :



Avec le bouton **Connexions source-à-cible** appuyé, d'autres connexions sont devenus noires : (i) les connexions de la fonction du **substring** à la constante 1 et du composant PersonList, et (ii) la connexion de la fonction **position** au composant PersonList. Toutefois, les connexions entre le composant PersonList et son composant précédent restent gris clair. Donc, quand vous appuyez sur le bouton **Connexions source-à-cible** et cliquez sur le composant, vous pouvez retracer les connexions directes du composant. Si le composant sélectionné est connecté à quelques <u>composants de transformation</u> (par ex., les fonctions, constantes, filtres, composants de tri, composants SQL-NoSQL-WHERE/ORDER, conditions if-else, value-maps), vous pourrez également voir leurs connexions jusqu'aux <u>composants de structure</u> (tels que PersonList ci-dessus), variables, composants join, ou fonctions de service Web auxquels ces composants de transformation sont connectés.

# Dans cette section

Cette section donne un aperçu des connexions et est organisée comme suit :

- <u>Types de connexion</u><sup>90</sup>
- Paramètres de connexion<sup>97</sup>
- <u>Connexion Menu contextuel</u>
   <sup>98</sup>
- <u>Connexions incorrectes</u><sup>100</sup>
- Garder des connexions après avoir supprimé des composants<sup>101</sup>

# 3.2.1 Types de connexion

Les types de connexion suivants sont disponibles dans MapForce :

- <u>Connexions orientées vers la cible</u><sup>91</sup> (Standard) ;
- <u>Connexions orientées vers la source</u><sup>(9)</sup> (Contenu mixte) ;
- <u>Connexions d'enfants correspondants</u><sup>(93)</sup>
- <u>Connexions copy-all</u><sup>95</sup> (Copier les items d'enfant).

#### Connexions orientées vers la cible par rapport aux connexions orientées vers la source

Les connexions orientées vers la cible et les connexions orientées vers la source sont mutuellement exclusives. Le choix entre ces deux options dépend de l'ordre dans lequel les nœuds ont besoin d'être mappés. Dans les connexions orientées vers la cible, l'ordre des nœuds dans la sortie est déterminé par le schéma *cible*. Ce type de connexion est adapté pour la plupart des scénarios de mappage et est le type de connexion par défaut utilisé dans MapForce. Une connexion orientée vers la cible est affichée en tant que ligne pleine (*voir la capture d'écran ci-dessous*).

> ()Name	ľ	2	ķ	
l> ⊡ () Office	1	Standard		-⊟ <b>()</b> Office
🖒 🚽 🗘 Name	1			<b>{}</b> Name  >
🖒 🕀 🌔 Desc	l:	}	¢	-⊟ <b>()</b> Desc þ

Les connexions orientées vers la cible ne sont éventuellement pas adaptées quand vous voulez mapper des nœuds XML avec du contenu mixte (nœuds enfant et texte). Dans ce cas, <u>une connexion orientée vers la</u> <u>source</u> est recommandée : L'ordre des nœuds dans la sortie est déterminée par le schéma *source*.

## Enfants correspondants et connexions copy-all

Les connexions d'enfants correspondants et connexions copy-all appartiennent à une sous-section de connexions orientées vers la cible et la cible. Les enfants correspondants et connexions copy-all mappent des données entre les nœuds avec des nœuds enfant qui sont semblables dans les composants source et cible. Les connexions copy-all sont semblables aux connexions d'enfants correspondants, mais ne sont dotées que d'une connexion épaisse à la place de multiples connexions, qui empêche la zone de mappage d'être visuellement encombrée.

La section fournit des informations sur chaque type de connexion et sur les scénarios quand ces types de connexion sont utiles.

# 3.2.1.1 Connexions orientées vers la source

Une connexion orientée vers la source vous permet de mapper le contenu mixte (nœuds texte et enfant) dans le même ordre que dans le fichier XML *source*. Une connexion de contenu mixte est affichée en tant que ligne pointillée au niveau du nœud parent (*voir le Mappage l'élément <para>*). Cette rubrique explique comment mapper du contenu mixte. Elle affiche également l'effet d'utiliser les connexions standard (orientées vers la cible) avec du contenu mixte.

- **Note :** les connexions orientées vers la source peuvent également être utilisées dans les champs de base de données avec du contenu mixte (*éditions Professional et Enterprise*).
- Note : en vue d'accepter du contenu mixte, les composants cible doivent avoir des nœuds de contenu mixte.

#### Mappage de contenu mixte

Cette rubrique explique comment mapper le contenu mixte en utilisant une connexion axée sur la source Vous aurez besoin des fichiers suivants : Tut-OrgChart.xml, Tut-OrgChart.mfd et Tut-OrgChart.xsd., qui sont disponibles dans le <u>dossier du Tutoriel</u><sup>24</sup>.

#### Instance XML source

Un snippet du Tut-OrgChart.xml est affiché ci-dessous. Dans cet exemple, nous nous concentrerons sur l'élément du contenu mixte <para> avec ses nœuds enfant <bold> et <italic>. L'élément <para> contient aussi une instruction de traitement (<?sort alpha-ascending?>) ainsi qu'un commentaire (<!--Company details... -->), qui peuvent aussi être mappés, comme indiqué ci-dessous. Veuillez noter la séquence du texte et des nœuds gras/italique dans le fichier d'instance XML.

8	e	<desc></desc>
9	0	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
	nanoe	electronic technologies for <italic>multi-core processors.</italic> February 1999 saw the
	unvei	ling of the first prototype <bold>Nano-grid.</bold> The company hopes to expand its
	opera	tions <italic>offshore</italic> to drive down operational costs.
10		<pre><?sort alpha-ascending?></pre>
11		Company details: location and general company information
12		
13	e	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
14		
15	-	

#### <u> Mapper l'élément <para></u>

L'image ci-dessous illustre une partie de Tut-Orgchart.mfd. Dans l'exemple ci-dessous, la ligne pointillée montre que l'élément <para> contient du contenu mixte. Pour créer des connexions de contenu mixte, suivez les étapes suivantes :

- 1. Sélectionnez la commande de menu **Connexion | Auto-connexion des enfants correspondants**, qui se connectera aux <u>nœuds enfant correspondant</u> automatiquement. De manière alternative, vous pouvez mapper manuellement le nœud <para> avec ses nœuds enfant.
- Connectez l'élément <para> dans le composant source avec l'élément <para> dans le composant cible. Un message vous demandera si vous souhaitez définir la connexion en tant que « orientée vers la source ».
- 3. Cliquez **Oui** pour créer une connexion à contenu mixte.
- 4. Cliquez sur le volet Sortie pour voir le résultat du mappage. Cliquez sur le bouton (Wrap) dans la barre d'outils du volet Sortie pour afficher la liste de code entière (par ex., n'allant pas au-delà de la barre de défilement) dans le volet Sortie. Le contenu mixte du nœud <para> a été mappé dans le même ordre qu'il apparaît dans le fichier source XML.



#### Traiter les instructions et les commentaires

Si votre mappage a des instructions de traitement et/ou commentaires et que vous voulez les mapper, suivez les étapes ci-dessous :

- 1. Cliquez de la touche de droite sur la connexion du contenu mixte (ligne pointillée) et sélectionnez **Propriétés**.
- 2. Sous Source-Drive (contenu mixte), sélectionnez les cases à cocher Mapper les instructions de traitement et/ou les Commentaires de mappage.

#### Connexions orientées vers la cible avec du contenu mixte

Choisir des connexions orientées vers la cible pour du contenu mixte peut avoir des conséquences indésirables. Pour voir comment des connexions orientées vers la cible affectent l'ordre des nœuds à contenu mixte, suivez la instructions suivantes :

- 1. OUVYEZ Tut-ExpReport.mfd dans le dossier du Tutoriel.
- Cliquez sur le bouton de barre d'outils (<u>Auto-connexion des enfants correspondants</u>). Supprimez la case à cocher Créer des connexions copy-all dans les <u>pour les connexions d'enfants</u> <u>correspondants</u>. Ceci empêchera MapForce de créer les <u>connexions copy-all</u>
   Ceci empêchera MapForce de créer les <u>connexions copy-all</u>
- Créer une connexion entre le nœud para dans la source et le nœud para dans la cible. Un message vous demandera si vous souhaitez définir les connexions en tant que « orientées vers la source ». Cliquez sur Non. Ceci crée une connexion orientée vers la cible.
- 4. Cliquez sur le volet **Sortie** pour voir le résultat du mappage (*la capture d'écran ci-dessous*).



La capture d'écran ci-dessus monte que le contenu de l'élément text() dans la source a été mappé vers la cible. Toutefois, l'ordre des nœuds enfant (gras et italique) dans la sortie correspond à l'ordre de ces nœuds dans le schéma XML cible. Ceci signifie que les éléments gras et italiques ne sont pas intégrés dans le texte, mais sont mappés séparément.

# 3.2.1.2 Connecter des enfants correspondants

Les connexions d'enfants correspondants connectent automatiquement tous les nœuds enfant qui sont dotés des mêmes noms dans les fichiers source et cible. Pour activer cette option, procédez comme suit :

- Cliquez sur le 🗮 bouton de barre d'outils (Auto-connexion des enfants correspondants).
- Allez au menu Connexion et cliquez sur Auto-connexion des enfants correspondants.

## Paramètres pour les connexions d'enfants correspondants

Pour changer les pour les connexions d'enfants correspondants, clic droit sur une connexion et sélectionnez l'option **Connecter des enfants correspondants** depuis le menu contextuel (*voir la capture d'écran cidessous*). De manière alternative, allez au menu **Connexion** et cliquez sur **Paramètres pour connecter des enfants correspondants**.



La liste ci-dessous décrit les options disponibles dans la boîte de dialogue **Paramètres pour connecter des** 

enfants correspondants. Les paramètres dans la boîte de dialogue ne s'appliquent que si le barre d'outils (Toggle auto-connexion des enfants) est actif. <u>Paramètres généraux</u>

- *Ignorer la casse :* Les enfants correspondants seront connectés indépendamment de la casse des noms de nœuds d'enfant.
- *Ignorer les espaces de noms :* Les enfants correspondants seront connectés indépendamment des espaces de noms des nœuds d'enfant.
- Récursif : Cette option crée de nouvelles connexions entre les nœuds correspondants s'ils ont les mêmes noms. Et cela n'a aucune importance à quel niveau les nœuds sont imbriqués dans l'arborescence.
- Mixer les attributs et les éléments : Cette option permet la création de connexions entre les attributs et les éléments qui portent les mêmes noms. Par exemple, une connexion est créée si deux nœuds Name existent, même si l'un est un élément et l'autre un attribut.
- Créer des connexions copier-tout : Ce paramètre est actif par défaut. Il crée (si possible) <u>une</u> <u>connexion copy-all</u><sup>(93)</sup>, qui mappe les données entre des nœuds enfant étant similaires ou identiques.

#### Connexions existantes

- Ignorer les connexions de sortie existantes : Cette option crée des connexions supplémentaires pour tous les nœuds correspondants, même s'ils ont déjà des connexions sortantes.
- Retenir : Cette option retient les connexions existantes.
- *Écraser :* Cette option écrase les connexions existantes.
- Supprimer tout existant : Cette option supprime toutes les connexions existantes avant d'en créer de nouvelles.

#### Supprimer des connexions en tant que groupe

Si vous souhaitez supprimer des connexions en tant que groupe, suivez les instructions ci-dessous :

1. Clic droit sur un nom de nœud dans le composant.

2. Sélectionnez **Supprimer des connexions | Supprimer toutes <...> connexions** depuis le menu contextuel (*voir la capture d'écran ci-dessous*).

ļ	Supprimer les co <u>n</u> nexions		Supprimer toutes les connexions directes
Ì	Créer composant varia <u>b</u> le		Supprimer toutes les connexions enfant entrantes
	Fonctions nœud et défauts		Supprimer toutes les connexions enfant entrantes et sortantes

- Supprimer toutes les connexions directes : Cette option supprime toutes les connexions qui sont directement mappées vers ou depuis des nœuds sélectionnés.
- Supprimer Toutes les connexions enfant entrantes : Cette option est active uniquement si vous avez cliqué avec la touche droite sur un nœud parent dans un composant cible. Cette option supprime toutes les connexions enfant entrantes du nœud parent sélectionné.
- Supprimer Toutes les connexions enfant sortantes : Cette option est active uniquement si vous avez cliqué avec la touche droite sur un nœud parent dans un composant source. Cette option supprime toutes les connexions enfant du nœud parent sélectionné.

# 3.2.1.3 Connexions copier tout

Les connexions copier-tout mappent les données entre des nœuds avec des items enfant qui sont similaires ou identiques. Les connexions « Copier tout » ne sont possibles que pour des formats identiques (par ex., JSON à JSON ou XML à XML). Ce principe s'applique également à tous les composants texte : fichiers plats, fichiers FlexText et EDI. Puisqu'il s'agit de fichiers texte pour ces formats, vous pouvez les combiner et créer une connexion copy-all entre les fichiers EDI et FlexText, par exemple. Voir Mappage : Sources et cibles et cibles pour en savoir plus sur les formats qui peuvent être utilisés en tant que sources de données et cibles de données.

Le principal avantage des connexions copier-tout est qu'elles simplifient visuellement l'espace de travail du mappage : Une connexion représentée par une ligne épaisse est créée à la place de multiples connexions (*voir l'exemple dans Créer toutes les connexions copier-tout manuellement*). Les sous-sections ci-dessous expliquent comment créer des connexions copy-all automatiquement et manuellement.

# Créer des connexions copy-all automatiquement

Pour créer une connexion copy-all automatiquement, suivez les étapes suivantes :

- 1. Allez au menu **Connexion**.
- 2. Cliquez sur Paramètres pour connecter des enfants correspondants.
- 3. Vérifiez la case Créer des connexions copier-tout et cliquez sur OK.
- 4. Appuyez sur la touche de la barre d'outils **Toggle auto-connexion d'enfants**. De manière alternative, allez au menu **Connexion** et cliquez sur **Auto-connexion des enfants correspondants**.

Si des types et/ou des noms de nœuds enfant dans la source et la cible ne sont pas les mêmes, une connexion copy-all ne sera pas créée automatiquement, et vous devrez en créer une automatiquement.

## Créer des connexions copier-tout manuellement

Pour créer une connexion copy-all manuellement, suivez les étapes suivantes :

- 1. Ajouter un fichier source : Cliquez sur **XML Schema/File** dans le menu **Insert** et recherchez Books.xml situés dans le <u>dossier BasicTutorials</u><sup>24</sup>.
- Ajouter un fichier cible : Cliquez sur XML Schema/File dans le menu Insérer et recherchez Library.xsd situé dans le même dossier que Books.xml. Cliquez sur Ignorer lorsque vous êtes invité par MapForce à ajouter un exemple de fichier XML.
- 3. Mappez le nœud <book> du composant books au nœud <publication> du composant library. Comme la structure des éléments de <book> et <publication> ne coïncide pas entièrement, la connexion copy-all n'est pas créée. À la place, la fonction Auto-connexion des enfants correspondants connecte automatiquement tous les nœuds enfant avec le même nom, ce qui est expliqué dans le <u>Tutoriel 1</u><sup>46</sup>.
- 4. Pour modifier la connexion automatique à une connexion copy-all, cliquez avec la touche droite sur la connexion entre <book> et <publication> et sélectionnez Copy-All (copiez les éléments enfant) depuis le menu contextuel.
- 5. Une fenêtre pop-up proposera de remplacer les connexions existantes par une connexion copy-all. Cliquez sur **OK**. Maintenant la source et la cible ont une connexion copy-all (*voir la capture d'écran*).



Dans le mappage ci-dessus, seuls deux nœuds enfant sont identiques dans les deux structures : <author> et <title>. Pour cette raison, une connexion copy-all existe entre ces nœuds. Les nœuds enfant qui ne sont pas les mêmes ne peuvent pas être connectés. La capture d'écran montre que l'id n'est pas incluse dans la connexion copy-all car son type n'est pas le même dans la source et dans la cible : l'id est un attribut dans la source et un élément dans la cible. Si vous tentez de créer une connexion entre les éléments qui ne sont pas les mêmes, par ex., <category> et <genre>, MapForce vous invitera à la remplacer ou à dupliquer l'entrée (voir la capture d'écran ci-dessous).

La <u>duplication de l'entrée</u> <sup>(B1)</sup> n'a de sens que si vous souhaitez que la cible accepte des données depuis plus d'une entrée, ce qui n'est pas nécessaire ici. Si vous choisissez de remplacer la connexion "Copier-tout", un message vous invite à nouveau soit à résoudre, soit à supprimer la connexion "Copier-tout". Cliquez sur **Résoudre la connexion copier-tout** si vous souhaitez remplacer la connexion copier-tout par des <u>connexions orientées vers la cible</u> individuelles. Si vous préférez supprimer entièrement la connexion copy-all, cliquez sur **Supprimer connexions enfant**.

#### Important :

Lors de la création de connexions copier-tout entre un schéma et un paramètre d'une <u>fonction définie par</u> <u>l'utilisateur</u><sup>575</sup>, les deux composants doivent être basés sur le même schéma. Néanmoins, il n'est pas nécessaire qu'ils possèdent les mêmes éléments racine.

# 3.2.2 Paramètres de connexion

La boîte de dialogue **Paramètres de connexion** définit les paramètres d'une connexion. Pour ouvrir cette boîte de dialogue, double-cliquez sur la connexion. De manière alternative, cliquez avec la touche de droite sur une connexion et sélectionnez **Propriétés** depuis le menu contextuel. Les paramètres sont divisés en deux parties : les types de connexion et les paramètres d'annotation. Pour plus d'information, veuillez voir les soussections ci-dessous.

Paramètres de connexion X		
Type de connexion Orienté vers la cible (Sta Copier-Tout (Copier les i Orienté vers la source (c Instructions de trait Commentaires de M	indard) tems enfants) contenu mixte) tement de mappage appage	
Paramètres d'annotation Description : mixed Commencer l'emplacement O Connexion de source Milieu O Connexion cible	t re	
Alignement O Horizontal O Vertical O Incliné	Position Au-dessus de la ligne En-dessous de la ligne	
	OK Annuler	

#### Types de connexion

Vous pouvez choisir un des types de connexion décris ci-dessous:

- Les connexions <u>orientées vers la cible (Standard)</u><sup>(91)</sup> sont adaptées pour la plupart des scénarios de mappage.
- Les connexions <u>Copy-all (Copier tous les items enfant)</u><sup>(95)</sup>: Si des composants source et cible ont des nœuds identiques ou similaires avec des nœuds enfant correspondants, une connexion copy-all sera automatiquement créée entre les nœuds correspondants.
- Les connexions <u>orientées vers la source (contenu mixte)</u><sup>(91)</sup> mappent le contenu mixte (nœuds texte et enfant) dans le même ordre que dans le fichier XML source. Si vous sélectionnez

**Mapper les instructions de traitement** et/ou **Mapper les commentaires**, vous pourrez inclure ces groupes de données dans le fichier sortie (*voir la capture d'écran ci-dessous*).).

6	
7	<
	italic>multi-core processors.February 1999 saw the unveiling of the first prototype <b< th=""></b<>
	hopes to expand its operations <italic>offshore</italic> to drive down operational costs.
8 🔵	sort alpha-ascending?
9 🔵	Company details: location and general company information
10	
11	ara>White papers and further information will be made available in the near future.

#### Paramètres d'annotation

La section des **paramètres d'annotation** section vous permet d'étiqueter la connexion. Cette option est disponible pour tous les types de connexion. Pour annoter une connexion, suivez les instructions cidessous :

- 1. Cliquez avec la touche de droite sur la connexion et sélectionnez **Propriétés** depuis le menu contextuel. En alternative, double-cliquez sur la connexion.
- 2. Saisissez le nom de la connexion sélectionnée actuellement dans le champ **Description**. Cela active toutes les options dans la section **Paramètres d'annotation**.
- 2. Utilisez les groupes restants pour définir les paramètres de **démarrage emplacement**, **alignement** et **position** de l'étiquette.
- 3. Appuyez sur le bouton de la barre d'outils (Afficher les annotations). Si le bouton n'est pas encore visible dans la barre d'outils, activez le bouton dans la barre d'outils Afficher les options.

🕀 🛄 footnote	Mortagae seruising	🕀 🛄 footnote
Mortgage Servicing Rights	Mortgage servicing	🖃 🚯 Mortgage Servicing
= id	Þ I	— <b>— =</b> id
= unitRef		<b></b> unitRef

Note : si la touche de la barre d'outils Afficher annotations est inactive, vous pouvez toujours consulter l'annotation si vous placez le curseur de la souris sur la connexion. L'annotation apparaîtra comme info-bulle si la touche de la barre d'outils (Afficher les conseils) est active dans la barre d'outils Afficher les options.

# 3.2.3 Menu contextuel de la connexion

Cette rubrique décrit les commandes disponibles dans le menu contextuel de la connexion. Lorsque vous cliquez avec la touche de droite sur une connexion, les commandes contextuelles suivantes sont disponibles :

	Connecter les enfants correspondants
×	Supprimer Entf
	Aller à la source : Date
	Aller à la cible : Date
~	Orienté vers la cible (Standard)
	Copier-Tout (Copier les items enfants)
	Ori <u>e</u> nté vers la source (contenu mixte)
Āţz	I <u>n</u> sérer tri : nœuds/lignes
Ŷ	Insérer <u>f</u> iltre : nœuds/lignes
R.Z	Ins <u>é</u> rer SQL-WHERE/ORDER
	Insé <u>r</u> er Value-Map
	<u>P</u> ropriétés

Pour plus d'information, veuillez voir les sous-sections ci-dessous. Paramètres généraux

- Connecter les enfants correspondants : Ouvre la boîte de dialogue Connecter les enfants • correspondants<sup>(33)</sup>. Cette commande est activée lorsque la connexion est susceptible d'avoir des enfants correspondants.
- Supprimer : Supprime la connexion sélectionnée.
- Aller à la source : <item name> Met en surbrillance le <u>connecteur de sortie</u><sup>75</sup> de la connexion sélectionnée.
- Aller à la cible : <item name> Met en surbrillance le connecteur d'entrée<sup>75</sup> de la connexion sélectionnée.

#### Types de connexion

Voir les détails sur les types de connexion dans <u>Types de connexion</u><sup>90</sup> et <u>Paramètres de connexion</u><sup>97</sup>.

#### Commandes d'insertion

- *Insérer Tri : Nœuds/Lignes :* Ajoute un composant de tri<sup>617</sup> entre un nœud source et un nœud cible. *Insérer Filtre : Nœuds/Lignes :* Ajoute un composant de filtre<sup>623</sup> entre un nœud source et un nœud cible.
- Insérer SQL/NoSQL-WHERE/ORDER: Ajoute un composant SQL/NoSQL-WHERE/ORDER entre un • nœud source et un nœud cible (éditions Professional et Enterprise). Pour tout détail, voir Filtrer et trier des données de base de données<sup>355</sup>.
- Insérer Value-Map : Ajoute une value-map<sup>530</sup> entre le nœud source et le nœud cible. •

#### Propriétés

Ouvre la boîte de dialogue des <u>Paramètres de connexion</u><sup>97</sup>.

# 3.2.4 Connexions incorrectes

Il existe des situations dans lesquelles vous allez éventuellement vouloir changer un schéma d'une source ou d'une cible. Les changements à un schéma peuvent affectés la validité de leur mappage et le résultat dans plusieurs connexions incorrectes. La rubrique explique comment corriger de telles connexions après avoir changé le fichier de schéma. Suivez les instructions dans l'exemple ci-dessus pour comprendre comment gérer les connexions incorrectes.

1. Ouvrez Tut-ExpReport.mfd disponible sous le <u>dossier du Tutoriel</u><sup>24</sup>. La partie de ce mappage est affiché ci-dessous.



- 2. Ouvrez ExpReport-Target.xsd dans l'éditeur (par ex., <u>Altova XMLSpy</u>) et modifiez l'élément racine Company dans le schéma cible en Company-EU. Il n'est pas nécessaire de fermer MapForce.
- Une fois avoir édité l'élément racine du schéma cible, l'invite des fichiers Changés apparaît dans MapForce. Cliquez sur la touche Recharger. Puisque l'élément racine a été supprimé, le composant affiche plusieurs nœuds incorrects.
- 4. Cliquez sur Sélectionner un nouvel élément racine dans la partie supérieure du composant (voir la capture d'écran ci-dessous). Vous pouvez aussi changer l'élément racine en cliquant avec la touche de droite sur l'en-tête du composant et en sélectionnant Changer l'élément racine depuis le menu contextuel.

🙄 Sélectionner élément racine 🛛 🗙			
Nom	Espace de noms URI	r	
CashAdvance	http://my-company.com/namespace		
Company-EU	http://my-company.com/namespace		
Date	http://my-company.com/namespace		

- 5. Sélectionnez Company-EU en tant que nouvel élément racine et cliquer sur **OK**. L'élément racine company-EU est maintenant visible dans la partie supérieure du composant.
- 6. Maintenant, vous devez déplacer la connexion du nœud incorrect Company au nouvel élément racine. Appuyez et tenez appuyée la section épaisse (*voir la flèche rouge ci-dessous*) de la connexion de Company. Puis glissez la connexion vers l'élément racine Company-EU.



Une boîte de dialogue de notification vous demandera si vous souhaitez déplacer tous les nœuds enfant connectés correspondants. Vous pouvez choisir entre uniquement vouloir déplacer la connexion sélectionnée ou la connexion sélectionnée avec ses nœuds enfant qui correspondent aux nœuds enfant du nouvel élément racine. Dans notre exemple, nous avons choisi l'option **Inclure des connexions décroissantes**. Dès que vous cliquez sur le bouton, tous les nœuds incorrects disparaîtront du composant.

Note : si le nœud vers lequel vous mappez porte le même nom que le nœud source, mais a un espace de noms différent, le dialogue de notification contiendra une touche supplémentaire Inclure les espaces de noms décroissants et mapper l'espace de noms. Cliquez sur cette touche pour déplacer les connexions enfant du même espace de noms en tant que nœud parent de source vers les mêmes nœuds enfant sous le nœud d'espace de noms différent.

## Solution alternative

Une solution alternative au problème discuté ci-dessus pourrait supprimer les nœuds incorrects dont vous n'avez éventuellement plus besoin dans votre mappage. Par exemple, si vous supprimez la connexion entre la fonction concat et le nœud Nom, le Nom disparaîtra du composant ExpReport-Target.

## Les connexions incorrectes dans les bases de données (éditions Professional et Enterprise)

Si votre base de données a des connexions incorrectes, vous allez devoir <u>changer les paramètres de</u> <u>composant</u> <sup>(20)</sup> Cliquer sur le bouton **Changer** dans la boîte de dialogue **Paramètres de composant** vous permet de sélectionner une base de données différente ou de changer les tables dans votre composant de base de données. Toutes les connexions valides/correctes et les données de base de données pertinentes seront retenues si vous sélectionnez une base de données de la même structure.

# 3.2.5 Garder des connexions après avoir supprimé des composants

MapForce vous permet de garder les connexions même après avoir supprimé quelques <u>composants de</u> <u>transformations</u> : par ex., les variables, composants de tri et de filtre, value-maps, entrées simples, composants SQL/NoSQL-WHERE/ORDER. Les connexions peuvent être simples ou multiples. Garder les connexions peut être particulièrement utile avec de multiples connexions enfant car vous ne devrez pas restaurer chaque connexion simple enfant manuellement après avoir supprimé le composant de transformation. Pour activer cette option, allez à **Outils | Options | Édition** et sélectionnez **Suppression de composant intelligent (garder les connexions utiles)**. Par défaut, cette option est désactivée, ce qui signifie que supprimer un composant de transformation supprimera également ses connexions directes.

## Exemple

L'exemple de fichier appelé Tut3-ChainedMapping est utilisé pour illustrer une suppression de composant intelligent. L'exemple de fichier est disponible dans le dossier <u>BasicTutorials</u><sup>24</sup>.

#### Avant la suppression

La capture d'écran ci-dessous montre que les <u>connexions copy-all</u> existent entre le composant MergedLibrary et le filtre publication, et entre le filtre publication et le composant FilteredLibrary. Maintenant, nous voulons supprimer le filtre publication mais retenir les connexions copy-all. Pour ce faire, sélectionnez la case à cocher **Suppression de composant intelligent** dans la boîte de dialogue des **Options** (*voir la capture d'écran ci-dessus*).



#### Après la suppression

Après avoir supprimé la fonction publication, la connexion copy-all a été créée directement entre le nœud publication dans MergedLibrary et le nœud publication dans FilteredLibrary (voir la capture d'écran ci-dessous).



**Note :** si un composant de filtre est connecté aux deux sorties on-true et on-false, les connexions des deux sorties seront retenues.

# 3.3 **Procédures générales et fonctions**

Outre la création de mappages, vous pouvez également valider votre mappage et la sortie, générer le code, utiliser les fonctions du mode Texte et définir les paramètres de mappage. Cette section est organisée dans les rubriques suivantes :

- Validation<sup>103</sup>
- <u>Génération de code</u><sup>105</sup>
- Fonctions de Mode Texte
- <u>Recherche Mode Texte</u><sup>109</sup>
- Paramètres de mappage<sup>112</sup>

# 3.3.1 Validation

Cette rubrique explique comment valider des mappages. Cette rubrique vous montre également comment prévisualiser, enregistrer et valider votre sortie.

# Valider les mappages

MapForce valide les mappages automatiquement quand vous cliquez sur le volet **Sortie**. Vous pouvez aussi valider votre mappage manuellement, ce qui peut vous aider à identifier et corriger des erreurs potentielles avant d'exécuter le mappage. Pour valider un mappage manuellement, cliquez sur le volet **Mappage**, puis procédez à l'une des étapes suivantes :

- Cliquez sur Valider mappage dans le menu Fichier.
- Cliquez sur 📴 (Valider) dans la barre d'outils.

Lorsque vous validez un mappage, MapForce vérifie, par exemple, des types de composants non pris en charge et des connexions incorrectes ou manquantes. Pour en savoir plus sur les statuts de validation, voir <u>Fenêtre de messages</u><sup>35</sup>. La fenêtre **Messages** vous permet également de prendre des <u>actions liées aux</u> <u>messages</u><sup>36</sup>. Pour afficher le résultat de chaque validation dans un onglet individuel, cliquez dans l'onglet numéroté disponible à gauche de la fenêtre **Messages**. Cela peut être utile, par exemple, si vous travaillez avec de multiples fichiers simultanément.

#### Validation des composants de transformation

La validation des <u>composants de transformation</u> fonctionne comme suit :

- Si un **connecteur d'entrée** obligatoire est non-connecté, un message d'erreur est généré et la transformation est interrompue.
- Si un **connecteur de sortie** n'est pas connecté, un avertissement est généré et le processus de transformation se poursuit. Le composant, qui a causé l'avertissement, et ses données sont ignorés et ne sont pas mappés vers la cible.

# Prévisualiser et valider la sortie

MapForce vous permet de prévisualiser la sortie sans avoir à exécuter et à compiler le code généré avec un processeur ou compilateur externe. En général, il est recommandé de consulter la sortie de transformation

dans le cadre de MapForce avant de traiter le code généré extérieurement. Lorsque vous consultez les résultats de mappage, MapForce exécute le mappage et affiche la sortie dans le <u>volet de sortie</u><sup>33</sup>.

Une fois que les données sont disponibles dans le volet **Sortie**, vous pouvez les valider et les enregistrer si nécessaire. Pour plus d'information, voir <u>Validation</u><sup>103</sup>. Vous pouvez aussi utiliser la commande **Trouver** (**Ctrl** + **F**) pour situer rapidement un motif de texte particulier dans le fichier de sortie. Pour plus d'informations, voir <u>Recherche Mode Texte</u><sup>109</sup>. Tout avertissement, erreur ou messages d'information lié à l'exécution de mappage est affiché dans la <u>fenêtre Messages</u><sup>35</sup>.

Si vous sélectionnez C++, C# ou Java (*éditions Professional et Enterprise*) en tant que <u>langage de</u> <u>transformation</u>, MapForce exécute le mappage utilisant son moteur de transformation built-in et affiche le résultat dans le volet **Sortie**.

Pour enregistrer la sortie de transformation, cliquez sur le volet **Sortie**, puis procédez à l'une des étapes suivantes :

- Cliquez sur Enregistrer fichier de sortie dans le menu Sortie.
- Cliquez sur 🛄 (Enregistrer sortie générée) dans la barre d'outils.

#### Charger les options

Lorsque vous consultez des fichiers de sortie volumineux, MapForce limite la quantité de données affichées dans le volet **Sortie**. Dans ce cas, le bouton **Charger plus** apparaît dans la partie inférieure du volet (*voir la capture d'écran ci-dessous*). Cliquer sur le bouton **Charger plus** ajoute le prochain morceau de données. Vous pouvez configurer les paramètres d'apercu depuis l'onglet **Général** du dialogue **Options**. Pour plus d'informations, voir <u>Options MapForce</u>

Result file size: 324.3 MB. 3% of the result are displayed. Load more Load all	Load all
--	----------

#### Valider la Sortie

Dès que la sortie devient disponible dans le volet **Sortie**, vous pouvez la valider par rapport au schéma y associé. Veuillez noter que la touche **Valider Sortie** et sa commande de menu correspondante (Sortie | **Valider fichier de sortie**) sont uniquement activées si le fichier de sortie prend en charge la validation par rapport à un schéma. Le résultat de la validation est affiché dans la fenêtre **Messages**, par exemple : Pour valider la sortie, suivez une des étapes suivantes :

- Ouvrez le volet Sortie et cliquez sur 🖾 (Valider Sortie) dans la barre d'outils.
- Ouvrez le volet Sortie et cliquez sur Valider Fichier de sortie dans le menu Sortie.

La capture d'écran ci-dessous illustre une validation qui n'a pas réussi. La fenêtre **Messages** contient des informations détaillées sur les erreurs. Par exemple, si vous cliquez sur le lien <Name>, MapForce marquera cet élément dans le volet **Sortie**.

🔁 😣 C:\Documents and Settings\pWy Documents\Altova\MapForce2011\MapForceExamples\Tutorial\Tut-F
Element <u><name></name></u> 💴 is not allowed at this location under element <u><company-person></company-person></u> 💴 .
□□ □ Reason: The following elements are expected at this location (see below)
<pre></pre> CompanyLogo> <pre>&gt;&gt;</pre>
Error location: <u>Company-Person</u> / <u>Name</u>
📅 <u>cvc-model-group</u> : Element <u><name></name></u> 🏧 unexpected by type '(anonymous)' 🏧 of element <u><c(< u=""></c(<></u>
🛄 cvc-ett.5.2.1: The element <u><company-person></company-person></u> 🏧 is not valid with respect to the actual type

# 3.3.2 Généreration de code

Vous pouvez générer du code depuis un mappage dépendant du langage que vous avez sélectionné en tant que <u>langage de transformation</u><sup>25</sup> de données. Vous pouvez générer du code dans les langages suivants :

- XSLT 1.0/XSLT 2.0/XSLT 3.0
- XQuery (éditions Professional et Enterprise)
- Java (éditions Professional et Enterprise)
- C# éditions Professional et Enterprise)
- C++ éditions Professional et Enterprise)

#### <u>Exemple</u>

Si vous voulez générer le code XSLT, suivez les instructions ci-dessous :

- 1. Sélectionnez l'item de menu Fichier | Générer code dans | XSLT .
- Choisir le dossier dans lequel vous souhaitez enregistrer le fichier XSLT et cliquez sur OK. MapForce génère le code et affiche le résultat de l'opération dans la <u>fenêtre Messages</u><sup>35</sup>.

Pour consulter le code généré XSLT, cliquez sur le volet **XSLT** en bas de la fenêtre **Mappage**. Les mêmes étapes s'appliquent à tout autre langage recensé ci-dessus.

Une fois que la génération de code est achevée, le dossier de destination contiendra les deux fichiers suivants :

- Un fichier de transformation XSLT, nommé après le schéma cible. Le fichier de transformation a le format suivant : <a href="mailto:suivant:/mapping>MapTo<TargetFileName>.xslt">mapping>MapTo<TargetFileName>.xslt</a>. <a href="mailto:suivant">Mapping>MapTo<TargetFileName>.xslt</a>. <a href="mailto:suivant">Mapping> est la valeur du champ Nom d'application dans les paramètres de mappage</a> <a href="mailto:suivant">suivant : <a href="mailto:suivant:mapping>MapTo<TargetFileName>.xslt">mapping> est la valeur du champ Nom d'application dans les paramètres de mappage</a> <a href="mailto:suivant">suivant : <a href="mailto:suivant:mapping>MapTo<TargetFileName>.xslt">mapping> est la valeur du champ Nom du composant de mappage</a> <a href="mailto:suivant:mapping>">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant de mappage</a> <a href="mailto:suivant:mapping">suivant:mapping> est le nom du composant</a> <a href="ma
- Un fichier DoTransform.bat, qui vous permet d'exécuter la transformation XSLT avec <u>Altova</u> <u>RaptorXML Server</u> depuis la ligne de commande. Pour exécuter la commande, vous allez devoir installer RaptorXML.

# **3.3.3** Fonctions de Mode Texte

Les <u>volet de sortie</u><sup>(38)</sup>, <u>volet XQuery</u><sup>(37)</sup>, et <u>volet XSLT</u><sup>(37)</sup> sont dotés de nombreuses aides visuelles afin de rendre l'affichage du texte plus facile : par ex., les marges, le marquage de texte, les guides de retrait, les marqueurs de fin de ligne et les marqueurs d'espace blanc. Vous pouvez personnaliser ces fonctions dans la boîte de dialogue **Paramètres du mode Texte** (*voir la capture d'écran ci-dessous*). Les paramètres dans le dialogue s'appliquent à l'application entière.

😢 Paramètres Mode text	e		×	
Marges ✓ Marge n° de lignes ✓ Marge signet ✓ Marge de repli	Onglets Taille onglet : 4 Insérer onglets Insérer espaces	Aide visuelle ✓ Guides retrait Marqueur de fin de ligne Marqueurs espace blanc	OK Annuler Appliquer	
<ul> <li>Activer marquage automatique</li> <li>Faire correspondre le mot sélectionné</li> <li>Faire correspondre toute sélection depuis: 1 </li> <li>caractères.</li> <li>Respecter la casse</li> </ul>				
Haut Bas Gauche Droite Ctrl + Haut Ctrl + Bas Ctrl + Gauche Ctrl + Droite	Une ligne vers le haut Une ligne vers le bas Une colonne à gauche Une colonne à droite Faire défiler une ligne Faire défiler une ligne Un mot à gauche Un mot à droite	e vers le haut vers le bas	~	

Pour ouvrir le dialogue Paramètres Mode Texte, suivez une des étapes suivantes :

- Sélectionnez Sortie | Paramètres Mode Texte.
- Cliquez sur 🖾 (**Paramètres Mode Texte**) dans la barre d'outils.
- Cliquez avec la touche de droite sur la zone vide du volet **Sortie** et sélectionnez **Paramètres Mode texte** depuis le menu contextuel.

Certaines des aides à la navigation peuvent être activées depuis la barre d'outils **Mode Texte**, le menu d'application ou les raccourcis de clavier. Pour plus d'information sur les raccourcis, voir la section **Key Map** du dialogue des **Paramètres Mode Texte** affiché ci-dessus.

Voir la liste des paramètres disponibles ci-dessous.

Marges

#### Marge à lignes numérotées

La numérotation des lignes est affichée dans la marge de numérotation, qui peut être activée ou désactivée dans le dialogue **Paramètres Mode Texte**. Lorsqu'une portion de texte est réduite, la portion de numérotation du texte réduit est également dissimulée.

#### Marge à signet

Les lignes dans le document peuvent être marquées par un signet pour une référence et un accès rapides. Si la case à cocher **marge à signet** dans le dialogue des **Paramètres Mode Texte** est sélectionnée, les signets sont affichés dans la marge à signet (*voir la capture d'écran ci-dessous*). Si la case à cocher **marge à signet** n'est pas sélectionnée, les lignes marquées d'un signet de couleur cyan.

6		<lastname>Little</lastname>
7	0	Address>
8		<street>Long Way</street>
9		<city>Los-Angeles</city>
10		<zip>34424</zip>
11		<state>CA</state>
12		
13		
14	0	<lineltems></lineltems>
15	6	<lineltem> </lineltem>
24	6	<lineltem></lineltem>
33		

Vous pouvez éditer et parcourir les signets utilisant les commandes affichées dans la table ci-dessous. Les commandes sont disponibles dans le menu **Sortie** et également à travers le menu contextuel lorsque vous cliquez sur la touche droite du volet **Sortie**, **XSLT** ou **XQuery**.

	Insérer/Supprimer Signet (Ctrl + F2)
<b>i</b>	Aller au Signet suivant (F2)
	Aller au Signet précédent (Shift + F2)
<b>L</b>	Supprimer tous les Signets (Ctrl + Shift + F2)

#### Marge pliable

Le pliage de source se réfère à la capacité d'élargir et de réduire les nœuds. Cette fonction est affichée dans la marge pliable de source. La marge peut être activée et désactivée dans le dialogue **Paramètres Mode Texte**. Pour agrandir ou réduire des portions de texte, cliquer sur les nœuds + et - du côté gauche de la fenêtre. Toute portion du code réduit est affichée avec un symbole d'ellipse(*voir la capture d'écran ci-dessous*). Pour consulter le code réduit sans l'agrandir, déplacez le curseur de la souris sur l'ellipse. Une infobulle s'ouvre qui affiche le code à consulter, tel qu'indiqué dans la capture d'écran ci-dessous. Veuillez noter que si le texte consulté est trop grand pour être affiché dans l'infobulle, une ellipse supplémentaire apparaît à la fin de l'infobulle.



#### ■ Activer le marquage automatique

Le paramètre **Activer le marquage automatique** vous permet de voir toutes les correspondances du morceau de texte sélectionné. La sélection est marquée en bleu clair, et les correspondances sont marquées en brun clair. La sélection et ses correspondances sont marquées en tant que carrés gris dans la barre de défilement. La position actuelle du curseur est affichée en tant que marqueur bleu dans la barre de défilement. Une sélection peut être définie comme étant un mot entier ou un nombre fixe de caractères. Vous pouvez aussi spécifier si la casse doit être prise en compte ou pas.

Pour sélection de caractère, vous pouvez spécifier le nombre minimum de caractères qui doivent correspondre, en commençant depuis le premier caractère dans la sélection. Par exemple, vous pouvez choisir de faire correspondre deux caractères ou plus. En ce qui concerne les recherches de mots, les éléments suivants sont considérés être des mots séparés : noms d'élément (sans crochets), les crochets de balises d'éléments, les noms d'attributs et les valeurs d'attribut sans guillemets.

#### Aide visuelle

#### Guides de retrait

Les guides de retrait sont des lignes verticales qui indiquent l'étendue d'un retrait d'une ligne. Ils peuvent être activés et désactivés dans le dialogue **Paramètres Mode Texte**. Les options **Insérer onglets** et **Insérer espaces** prennent effet quand vous utilisez l'option **Sortie | Texte XML Pretty-Print**.

#### Marqueurs de fin de ligne et marqueurs d'espace blanc

Les marqueurs de fin de ligne et les marqueurs d'espace blanc (*voir la capture d'écran ci-dessous*) peuvent être basculés dans le dialogue **Paramètres Mode Texte**. Les flèches représentent les tabulations. L'abréviation *CR* est synonyme de retour de chariot. Les pointillés représentent les espaces de caractères.

1	xml version="1.0" encoding="UTF-8"? CR
2	<pre>cbooks xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
	<pre>xsi:noNamespaceSchemaLocation="books.xsd"&gt;CR</pre>
3	⊖ → <book id="1" ·="">CR</book>
4	→ → <author>Mark Twain</author>
5	→ → <title>The Adventures of Tom Sawyer</title> CR
6	→→ <category>Fiction</category> C
7	$\rightarrow \rightarrow \langle year \rangle 1876 \langle year \rangle \mathbb{R}$
8	
9	-

#### Autres paramètres du Mode Texte
### Coloration syntaxique

La coloration syntaxique est une autre aide visuelle qui rendent la lecture des listes de codes plus conviviale. La coloration syntaxique dépend de la valeur sémantique du texte. Par exemple, dans des documents XML, selon que le nœud XML est un élément, un attribut, un contenu, une section CDATA, un commentaire ou une instruction de traitement, le nom de nœud (et dans certains cas, le contenu du nœud) sera coloré d'une autre teinte.

#### Zoom avant et arrière

Vous pouvez zoomer en avant et en arrière (en faisant défiler la roulette de la souris) tout en maintenant la touche **Ctrl** appuyée. En alternative, appuyez sur les touches - ou + tout en maintenant la touche **Ctrl** appuyée.

### Pretty-print

La commande **Pretty-Print XML Text** reformate le document XML actif dans le **Mode Texte** pour donner un affichage structuré du document. Par défaut, chaque nœud enfant est séparé de son parent par quatre espaces. Cela peut être personnalisé depuis le dialogue **Paramètres Mode Texte**. Pour imprimer automatiquement un document XML, sélectionnez la commande de menu **Sortie | Pretty-Print XML** 

Text ou cliquez sur 🔄 (Pretty-print) dans la barre d'outils.

#### Retour automatique à la ligne

Le retour à la ligne aide à afficher une liste de codes à l'intérieur des bords de l'espace de travail. Si le paramètre de retour à la ligne n'est pas activé, certaines parties de texte peuvent ne pas être entièrement visibles dans l'espace de travail. Pour activer le retour automatique à la ligne dans le document actif,

sélectionnez la commande de menu **Sortie | Word Wrap** ou cliquez sur (Word Wrap) dans la barre d'outils.

# 3.3.4 Recherche Mode Texte

Le texte dans le volet **Sortie**, le volet **Xquery**, et le volet **XSLT** peut être recherché avec un plage d'options et d'aides visuelles extensives.

Vous pouvez rechercher un terme dans tout le document ou à l'intérieur d'une sélection de texte. Pour démarrer une recherche, appuyez sur **Ctrl+F** ou sélectionnez la commande de menu **Édition | Recherche**. Vous pouvez saisir un string ou utiliser la zone de liste déroulante pour sélectionner un string d'un des derniers 10 strings. Quand vous saisissez ou sélectionnez un string, toutes les correspondances sont mises en surbrillance, et les positions des correspondances sont indiquées par des marqueurs orange dans la barre de défilement (*voir la capture d'écran ci-dessous*). La position de la correspondance sélectionnée actuellement (en surbrillance grise) dépend du dernier emplacement du curseur.

Vous pouvez voir le nombre total de correspondances et la position de l'index de la correspondance actuelle

sélectionnée. Utilisez les boutons (**Précédent**) et (**Suivant**) pour basculer entre les correspondances.



### Options de Recherche

Vous pouvez spécifier des critères de recherche à l'aide des boutons situés sous le champ de recherche. La liste des options disponibles est donnée dans la table ci-dessous.

Option	lcôn e	Description
Respecter la casse	Aa	Réalise une recherche sensible à la casse : par ex., <i>Address</i> n'est pas la même chose que <i>address</i> .
Correspondance du mot entier	Abc	Seuls les mots identiques correspondront.
Utiliser une expression régulière	•*	Une fois activé, le terme de recherche sera lu en tant qu'expression régulière. Voir les <i>expressions régulières</i> ci-dessous.
Trouver l'ancre	I	La position de l'ancre dépend du dernier emplacement du curseur. Cliquez sur les boutons <b>Précédent</b> et <b>Suivant</b> ne modifie pas la position de l'ancre.
Trouver dans la sélection	<b>bh</b>	Une sélection est un morceau de texte marqué. Pour trouver un terme au sein d'une sélection, marquez un morceau de texte, appuyez sur <b>Ctrl+F</b> , veillez à ce que le bouton <b>Chercher dans la sélection</b> est appuyé et saisissez le terme dans le champ de recherche.

### Expressions régulières

Vous pouvez utiliser des expressions régulières (regex) pour trouver un string de texte. Pour ce faire, activez l'option **Utiliser les expressions régulières** (*voir la table ci-dessus*). Ensuite, saisissez une expression

régulière dans le champ de recherche Cliquer sur 😻 (**Générateur d'expression régulière**) vous donne une liste d'exemples d'expression régulière (*voir ci-dessous*). La capture d'écran ci-dessous montre une expression régulière qui aide à trouver des adresses e-mail.



#### Métacaractères d'expression régulière

La table ci-dessous affiche des métacaractères que vous pouvez utiliser, et remplacer du texte. Tous les métacaractères à l'exception des deux derniers correspondent aux éléments de menu dans le **Générateur d'expression régulière** (*voir ci-dessus*).

ltem de menu	Métacaractères	Description
Tout caractère		Correspond à n'importe quel caractère. Il s'agit d'un espace réservé pour un seul caractère.
Gammes de caractère	[]	Correspond à tout caractère se trouvant dans cet ensemble. Par exemple, [abc] correspond à tous les caractères a, b ou c. Vous pouvez aussi utiliser des plages : par ex., [a-z] pour tout caractère en minuscule.
Caractères absent de la plage de caractères	[^]	Correspond à tout caractère ne se trouvant pas dans cet ensemble. Par exemple, [^A-za-z] correspond à tout caractère sauf à un caractère alphabétique.
Début du mot	/<	Correspond au début d'un mot.
Fin du mot	/>	Correspond à la fin d'un mot.
Début de ligne	^	Correspond au début d'une ligne sauf si utilisé à l'intérieur d'un ensemble( <i>voir ci-dessus</i> ).
Fin de ligne	\$	Correspond à la fin d'une ligne. Par exemple, A+\$ correspond à un ou

ltem de menu	Métacaractères	Description
		plusieurs
Expression balisée	(abc)	Les parenthèses marquent le début et la fin d'une expression balisée. Les expressions balisées peuvent être utiles lorsque vous devez baliser (« vous souvenir ») d'une région comparée afin de pouvoir vous y référer ultérieurement. Il est possible de baliser jusqu'à neuf sous-expressions, puis de les rétro-référencer ultérieurement. Par exemple, (the) \1 correspond au string the the. Cette expression peut être expliquée comme suit : Faites correspondre le string the et souvenez-vous en comme région balisée ; l'expression doit être suivie d'un espace et d'une rétro-référence à la région balisée correspondante précédente.
0 ou plus de correspondances	*	Correspond à zéro ou plusieurs matches de l'expression précédente. Par exemple, sa*m correspond à Sm, Sam, Saam, Saaam etc.
1 ou plus de correspondances	+	Correspond à une ou plusieurs occurrences de l'expression précédente. Par exemple, sa+m correspond à Sam, Saam, Saaam etc.
	\n	Où n est 1 à 9, n se réfère à la première région balisée de neuf régions ( <i>voir ci-dessus</i> ).
	\x	Vous permet d'utiliser un caractère x, qui aurait généralement une signification particulière. Par exemple, \[ sera interprété comme [ et non pas comme le début d'un jeu de caractères.

### Chercher caractères spéciaux

Si l'option **Utiliser des expressions régulières** est activée, vous pouvez rechercher n'importe quel des caractères spéciaux suivants au sein du texte :

- \t (Onglet)
- \r (Retour de chariot)
- \n (Nouvelle ligne)
- \\ (Backslash)

Par exemple, pour trouver un caractère de tabulation, appuyez sur **Ctrl + F**, choisissez l'option **Utiliser les** expressions régulières, puis saisissez \t dans le dialogue **Recherche**.

# 3.3.5 Paramètres de mappage

La boîte de dialogue **Paramètres de mappage** (*voir la capture d'écran ci-dessous*) vous permet de définir les paramètres spécifiques au document. Pour ouvrir cette boîte de dialogue, allez au menu **Fichier** et cliquez sur **Paramètres de mappage**. En alternative, cliquez avec la touche de droite dans la zone vide du volet de mappage et sélectionnez **Paramètres de mappage** depuis le menu contextuel.

😢 Paramètres de mappag	e	$\times$
Génération de code		
Nom d'application:	Mapping	
Nom pack de base Java :	com.mapforce	
Rendre les chemins abs	olus dans le code généré	
Bibliothèques de référer	nce avec des chemins relatifs aux fichiers XSLT / XQuery générés	
Assurer la convention d système de fichier local	e chemin Windows pour la sortie de chemin de fichier pour les fichiers d'un	
Paramètres de fichier de so	rtie	
Terminaisons de ligne:	Défaut plate-forme V	
(prises en charge dans l'ex	écution Built-in et la génération de code C#, Java et C++)	
Version de Schema XML		
● v1.1 if <xs:schema vc:<br="">v1.0 sinon</xs:schema>	ninVersion="1.1" >	
O Toujours v1.1		
O Toujours v1.0		
	OK Annuler	

Les paramètres disponibles sont décrits dans les sous-pages ci-dessous.

- Génération de code
  - Nom d'application : Définit le préfixe du fichier XSLT généré ou le nom d'application Java, C# ou C++ (éditions Professional et Enterprise).
  - Nom de package de base Java (éditions Professional et Enterprise) : Cette option s'applique lorsque Java est sélectionné en tant que langage de transformation. L'option définit le nom de package de base pour la sortie Java.
  - Rendre les chemins absolus dans le code généré : Cette case à cocher touche tous les chemins dans les composants de mappage, sauf les chemins vers les fichiers de bibliothèque externes (comme les bibliothèques XSLT). La case à cocher définit si les chemins de fichier doivent être relatifs ou absolus dans le code de programme généré, dans les fichiers d'exécution de <u>MapForce Server</u> (.mfx) et dans les fonctions de mappage déployées dans <u>FlowForce Server</u>. Pour plus d'informations, voir <u>Chemins dans des environnements d'exécution</u><sup>85</sup>.
  - Bibliothèques de référence avec des chemins relatifs aux fichiers XSLT/XQuery générés : Cette case à cocher s'applique quand le langage de mappage est XQuery (éditions Professional et Enterprise), soit XSLT. Cette option est utile si votre mappage référence une bibliothèque XSLT ou XQuery et si vous prévoyez générer des fichiers XSLT ou XQuery depuis le mappage. Sélectionnez cette case à cocher si vous souhaitez que les chemins de bibliothèque soient relatifs au répertoire du code XSLT ou XQuery généré. Si la case à cocher n'est pas sélectionnée,

#### **114** Notions fondamentales de mappage

les chemins de bibliothèque seront absolus dans le code généré. Voir aussi chemins de <u>Bibliothèque dans le code généré</u><sup>86</sup>.

 Assurer la convention de chemin Windows pour le chemin de fichier : Cette case à cocher s'applique si le langage de mappage est XQuery (éditions MapForce Professional et Enterprise), XSLT 2.0 ou XSLT 3.0. Cette case à cocher garantit que les conventions de chemin Windows sont suivies. Lors de la sortie XSLT 2.0, XSLT 3.0 ou XQuery, le nom de fichier traité actuellement est extrait en interne et en avec l'aide de la fonction document-uri, qui retourne un chemin sous le format file://URI pour des fichiers locaux. Lorsque cette case à cocher est sélectionnée, un chemin de spécification file://URI est converti automatiquement dans un chemin de fichier Windows complet (e.g., C:\...) pour simplifier le traitement supplémentaire.

### Paramètres de fichier de sortie (éditions Professional et Enterprise)

La zone de liste **fin de ligne** vous permet de spécifier les fins de ligne des fichiers de sortie. La *Plateforme par défaut* signifie l'option par défaut pour les systèmes d'exploitation cible : par exemple, Windows (CR+LF), macOS (LF) ou Linux (LF). Vous pouvez aussi choisir une fin de ligne spécifique manuellement. Les paramètres que vous choisissez ici sont importants lorsque vous compilez un mappage vers un fichier d'exécution de <u>MapForce Server</u> (.mfx) ou lorsque vous déployez un mappage sur <u>FlowForce Server</u> exécuté sur un système d'exploitation différent.

#### Version de Schéma XML

Cette option permet de définir la version de schéma XML utilisée dans le fichier de mappage. Veuillez noter que pas toutes les versions 1.1. spécifiques sont actuellement prises en charge. Si la déclaration xs:schema vc:minVersion="1.1" est présente, la version 1.1 sera utilisée ; sinon, la version 1.0 sera utilisée.

Si le document XSD n'a pas d'attribut vc:minVersion ou si la valeur de l'attribut vc:minVersion est autre que 1.0 ou 1.1, alors XSD 1.0 sera le mode par défaut. Ne confondez pas l'attribut vc:minVersion avec l'attribut xsd:version. Le premier attribut a le numéro de version XSD, alors que le deuxième contient le numéro de version du document . Changer ce paramètre dans un mappage existant entraîne un rechargement de tous les schémas de la version de schéma XML sélectionnée, et peut aussi modifier sa validité.

### Paramètres d'opération de service Web (édition Enterprise)

Les champs **définitions WDSL**, **Service**, **Point de terminaison** et **Opération** sont automatiquement remplis si le document de mappage fait partie de la mise en place de service Web.

# 3.4 Projets

Outre le fait de créer des mappages individuels, vous pouvez aussi créer des projets de mappage qui contiennent plusieurs mappages. Des mappages qui sont ajoutés à un projet sont facilement accessibles depuis la fenêtre **Projet** (*voir la capture d'écran ci-dessous*).

Projet X
MapForceExamples
🕀 🛅 Mapping Folder
-⊞ 🛅 XML Schemas
- 🕀 🛅 JSON
- 🕀 💼 Databases
🕀 🛅 Text Files
🕀 🛅 FlexText
🕀 🛅 Protocol Buffers
- 🕀 💼 Web service calls
🕀 🛅 OOXML Excel 2007+
Terret Constraints and the second se
🗏 Bibliothèques

L'avantage principal des projets est que vous pouvez définir des paramètres de génération de code commun (comme le langage cible et le répertoire de sortie) pour tous les mappages, y compris ceux contenus dans ce projet en particulier. Vous pouvez aussi créer des dossiers dans des projets et spécifiez des paramètres de génération de code personnalisés pour chaque dossier individuel dans un projet. Pour plus d'informations concernant le code de programme généré par MapForce (dans C++, C#, et Java), voir <u>Générateur de code</u><sup>(06)</sup>. Les fichiers de projet de MapForce sont enregistrés avec l'extension .mfp.

Dans l'édition MapForce Enterprise, vous pouvez aussi créer des projets de Service Web. De tels projets vous permettent de générer du code de programme Java ou C# qui met en place les services Web SOAP, sur la base de fichiers WSDL (Web Services Description Language) existants.

# 3.4.1 Notions de base du projet

Les sous-sections ci-dessous vous aideront à démarrer avec le projet. Les procédures associées aux projets peuvent être largement divisées en (i) créant un projet, (ii) organisant un projet, et en (iii) réalisant différentes actions. Voir les informations concernant ces procédures dans les sous-sections ci-dessous.

### Nouveau projet

Pour créer un nouveau projet, procédez comme suit :

- 1. Cliquez sur le bouton dans la barre d'outils. En alternative, allez au menu **Fichier** and cliquez sur **Nouveau**.
- 2. Sélectionnez Fichier de projet, puis cliquez sur OK.

3. Saisissez le nom du projet dans le dialogue **Enregistrer le projet sous** et cliquez sur **Enregistrer**. Le nouveau projet est maintenant affiché dans la fenêtre de **Projet**.

Pour fermer un projet, allez au menu **Projet** et cliquez sur **Fermer Projet**.

### Organisation du projet

### Ajouter un mappage à un projet

Si vous voulez ajouter un mappage actuellement actif au projet, suivre une des étapes suivantes :

- Allez au menu Projet, cliquez sur Ajouter Fichier actif au projet.
- Cliquez avec la touche de droite sur le projet pertinent dans la fenêtre Projet et sélectionnez Ajouter Fichier actif au projet depuis le menu contextuel.

Pour ajouter des fichiers de mappage existant au projet, suivez une des étapes suivantes :

- Allez au menu Projet, cliquez sur Ajouter Fichier au projet.
- Cliquez avec la touche de droite sur le projet pertinent dans la fenêtre **Projet** et sélectionnez **Ajouter Fichiers au projet**.
- Astuce : si vous souhaitez ajouter de multiples fichiers, tenez appuyée la clé Ctrl tout en sélectionnant les fichiers dans la boîte de dialogue Ouvrir.

### Supprimer un fichier depuis un projet

Pour supprimer un fichier ou un dossier d'un projet, suivez une des étapes suivantes :

- Sélectionnez le fichier que vous souhaitez supprimer dans la fenêtre **Projet**. Cliquez avec la touche de droite sur le fichier et sélectionnez **Supprimer** depuis le menu contextuel.
- Sélectionnez le fichier pertinent dans la fenêtre **Projet** et appuyez sur **Supprimer**.

Les fichiers de projet de MapForce ont l'extension . mfp. Vous pouvez ouvrir des projets de MapForce de la même façon que les mappages : Allez au menu **Fichier** et cliquez sur **Ouvrir**. Par défaut, quand vous exécutez MapForce pour la première fois, vous verrez le projet MapForceExamples.mfp dans la fenêtre **Projet**.

### Actions liées au projet

### Rechercher un projet

Pour rechercher un projet pour des fichiers, suivez les instructions ci-dessous :

- 1. Dans la fenêtre **Projet**, cliquez sur le projet ou le dossier que vous recherchez.
- Appuyez sur Ctrl + F. La boîte de dialogue Recherche vous permet de définir les options de recherche. Par exemple, si vous voulez inclure les noms de dossier dans la recherche, sélectionnez l'option Chercher dans les noms de dossier (voir la capture d'écran ci-dessous).



### Générer le code pour votre projet

Dans les projets, vous pouvez générer le code pour des (i) mappages individuels, (ii) un dossier spécifique ou (iii) tout le projet.

Pour générer un code pour un mappage ou un dossier dans votre projet, cliquez avec la touche de droite sur le mappage pertinent ou dossier et sélectionnez **Générer du code** ou **Générer du code dans**. Si vous sélectionnez **Générer du code**, le code sera généré dans le langage spécifié dans les <u>paramètres de</u> projet<sup>(113)</sup>. Vous pouvez aussi choisir de générer le code dans une des langues disponibles dans votre édition de MapForce. Pour plus d'information, voir <u>Gényération de code</u><sup>(105)</sup>.

Pour générer du code dans tout le projet, allez au menu **Projet** et sélectionnez **Générer le Code pour tout le projet**. En alternative, cliquez avec la touche de droite sur le nom du projet dans la fenêtre **Projet** et sélectionnez **Générer le code**. Le code sera généré dans le langage spécifié dans les <u>paramètres de projet</u><sup>119</sup>. En ce qui concerne tout le projet, vous pouvez aussi sélectionner un langage pour générer le code dans le menu **Projet** ou dans le menu contextuel du projet. Le choix des langages dépend de votre édition de MapForce. Pour plus d'information, voir <u>Génération de code</u><sup>105</sup>.

#### Prévisualiser les images

La fenêtre de **Projet** vous permet de prévisualiser des images dans les formats suivants :.png, .jpeg, .gif, .bmp, .tiff et .ico (*voir la capture d'écran ci-dessous*). En double-cliquant sur le fichier de l'image, vous l'ouvrirez dans une application externe, qui dépend de l'association du fichier dans Windows.



Pour tout projet, vous pouvez spécifier les paramètres de génération de code qui toucheront tous les mappages dans un projet. Pour ouvrir le dialogue **Paramètres de projet** (*voir la capture d'écran ci-dessous*), suivez une des étapes suivantes :

- Cliquer avec la touche de droite sur le nom du projet dans la fenêtre Projet et sélectionnez les Propriétés à partir du menu contextuel
- Allez au menu Projet et cliquez sur Propriétés.

Paramètres de projet		$\times$
Nom de projet :	MapForceExamples	
Répertoire de projet :	C:\Users \Documents\Altova\MapForce2022\Maj	
Paramètres de sortie		
Nom de sortie :	MapForceExamples	
Répertoire de sortie :	C:\Users\xxxxx\Documents\Altova\MapForce2022\Map Parcourir	
Langue :	Java 🗸	
Paramètres Java		
Nom de pack de base :	com.mapforce	
	OK Annuler	

Les paramètres disponibles sont recensés ci-dessous. Veuillez noter que le nom du projet et le répertoire du projet ne peuvent pas être changés une fois le projet créé.

- Nom de sortie : La valeur saisie dans le champ de texte détermine les noms du projet/de la solution généré(e) et autres objets du code généré.
- Répertoire de sortie : Définit le dossier Windows où le code généré provenant de tous les mappages dans ce projet sera enregistré. Par défaut, la sortie est enregistrée dans le répertoire MapForceExamples\output\.
- Langage : Définit le langage de génération de code pour tous les fichiers de mappage dans ce projet.
- *Nom de package de base :* Ce paramètre s'applique si Java est sélectionné en tant que langage de transformation. Le paramètre définit le nom du paquet de base dans le projet Java généré.

# 3.4.3 Dossiers de projet

MapForce vous permet d'organiser les mappages à l'intérieur d'un projet dans les dossiers. Vous pouvez créer autant de dossiers dont vous avez besoin et y ajouter des mappages. De tels dossiers sont virtuels et existent uniquement à l'intérieur d'un projet MapForce : Ces dossiers ne correspondent pas aux dossiers de votre système d'exploitation. L'un des avantages de créer des dossiers dans un projet est que vous pouvez définir des paramètres de génération de code communs pour tous les fichiers de mappage dans ce fichier particulier. Pour créer un dossier à l'intérieur d'un projet de MapForce, procédez comme suit :

- 1. Allez au menu **Projet**, et cliquez sur **Créer dossier**. En alternative, cliquez avec la touche de droite sur le projet dans la fenêtre **Projet** et sélectionnez **Créer dossier**.
- 2. Dans le dialogue **Propriétés**, (*voir la capture d'écran ci-dessous*)saisissez les paramètres de génération de code requis et cliquez sur **OK**.

Propriétés						Х
Nom: Schémas XML Paramètres de génération de code Outiliser les paramètres de projet par défaut						
Utiliser les par	ramètres su	ivants :				
Répertoire d	e sortie :				Parcourir	
Langue :		XSLT 1.0		$\sim$		
				OK	Annuler	

La liste ci-dessous décrit les paramètres que vous pouvez définir dans la boîte de dialogue des Propriétés.

- Nom : Il s'agit du nom du dossier dans votre projet.
- Utiliser les paramètres de projet par défaut : Cette option signifie que les paramètres de génération de code dans le dossier actuel sont les mêmes que pour le projet entier. C'est pourquoi, lorsque vous générez du code depuis votre projet, MapForce utilisera les paramètres de génération de code définis dans les paramètres de projet<sup>(113)</sup>. Si votre dossier nécessite des paramètres de génération de code personnalisés, choisissez Utiliser les paramètres suivants et spécifiez le répertoire de sortie du code et le langage, le cas échéant.
- *Répertoire de sortie :* Il s'agit du dossier où le code généré provenant de tous les mappages dans ce projet sera enregistré.
- Langage : Cette option définit le langage de de génération de code pour tous les fichiers de mappage dans ce dossier.

#### **Composants de structure** 4

Cette section fournit des informations sur différents formats de données que vous pouvez utiliser comme sources et cibles de données :

- XML et Schéma XML<sup>121</sup> Bases de données<sup>156</sup> ٠
- •
- Fichiers CSV et Texte Fichiers binaires 471 ٠
- •

# 4.1 XML et schéma XML

### Site web d'Altova : Site web d'Altova :

Cette section fournit des informations sur les aspects spécifiques aux composants XML. Pour des informations sur les scénarios de transformation des données de base, voir <u>Tutoriels</u><sup>40</sup>.

### Générer un fichier de schéma

### Générer un Schéma XML

Lorsque vous ajoutez un fichier XML sans référence de schéma, MapForce suggèrera générer un schéma XML pour vous. Vous serez ensuite invité à sélectionner le répertoire dans lequel le schéma généré devrait être enregistré. Pour ajouter un fichier ou schéma XML, utilisez la commande de menu **Insérer | Schéma/Fichier** 

XML ou le bouton de la barre d'outils

Lorsque MapForce génère un schéma depuis un fichier XML, les types de données pour des éléments/attributs doivent être inférés depuis le document d'instance XML et ne sont pas exactement ce à quoi vous vous attendez. Nous vous recommandons de contrôler si le schéma généré est une représentation précise des données d'instance.

Si les éléments ou les attributs dans plus d'un espace de noms sont présents, MapForce génère un Schéma XML séparé pour chaque espace de noms distinct ; c'est pourquoi plusieurs fichiers peuvent être créés sur le disque.

### DTD comme structure du document

À partir de MapForce 2006 SP2, les DTD namespace-aware sont pris en charge pour les composants source et cible. Pour rendre les mappages possibles, les URI d'espace de nom sont extraits depuis les déclarations d'attribut de DTD xmlns. Toutefois, certains DTD contiennent des déclarations d'attribut xmlns\* sans URI d'espace de nom (par ex., des DTD utilisés par StyleVision). Pour rendre ces DTD utilisables dans MapForce, définissez l'attribut xmlns avec l'URI d'espace de noms comme indiqué ci-dessous :

### Dans cette section

La section est organisée en rubriques suivantes :

- Paramètres de composant XML<sup>122</sup>
- <u>Types dérivés</u><sup>126</sup>
- Valeurs NULL<sup>128</sup>
- <u>Commentaires et Instructions de traitement</u><sup>132</sup>
- <u>Sections CDATA</u><sup>133</sup>
- <u>Caractères génériques : xs:any/xs:anyAttribute</u><sup>134</sup>
- Espaces de noms personnalisés<sup>137</sup>
- <u>Gestionnaire de schéma XML</u><sup>139</sup>

# 4.1.1 Paramètres de composant XML

Après avoir ajouté un composant XML dans la zone de mappage, vous pouvez configurer son paramètres applicables depuis le dialogue **Paramètres de composant** (*voir la capture d'écran ci-dessous*). Vous pouvez ouvrir le dialogue **Paramètres de composant** d'une des manières suivantes :

- En double-cliquant sur l'en-tête de composant.
- En cliquant avec la touche de droite sur l'en-tête du composant et sélectionnez Propriétés.
- En sélectionnant le composant dans le mappage et cliquez sur les **Propriétés** dans le menu **Composant**.

😢 Paramètres de composant	$\times$			
Nom de composant: Books				
Books.xsd Parcourir Éditer				
Fichier XML d'entrée				
Books.xml Parcourir Édition				
Fichier XML de sortie				
Parcourir Éditer				
Préfixe pour l'espace de noms de cible:				
Ajouter référence schéma/DTD (laisser le champ vide pour un accès absolu du schéma):				
Écrire la déclaration XML				
Convertir les valeurs en types cible (désactiver pour conserver le formatage des valeurs numériques ou de date au risque d'écrire une sortie invalide)				
Sortie pretty print				
Créer signature num (exécution Built-in seulement) Paramètres de signature				
En cas d'échec de la création : O Arrêter le traitement				
Continuer sans signature				
Encodage de sortie				
Nom d'encodage: Unicode UTF-8 $\checkmark$				
Ordre des octets: Little Endian V Inclure indicateur				
Fichier de Feuille de style StyleVision Power				
Parcourir Créer				
Activer les optimisations de traitement d'entrée sur la base des min/maxOccurs				
Enregistrer tous les chemins de fichier relatifs au fichier MFD				
OK Annuler				

Les paramètres disponibles sont décrits dans les sous-pages ci-dessous.

## Paramètres généraux

Nom de composant

Le nom de composant est généré automatiquement lorsque vous créez un composant. Néanmoins, vous pouvez changer le nom à tout moment. Le nom de composant peut contenir des espaces et les caractères de point final. Le nom de composant ne doit pas contenir barres obliques, des barres obliques inversées, des points-virgule, des guillemets doubles, des espaces de début et de fin. Si vous souhaitez changer le nom du composant, veuillez noter :

- Si vous souhaitez déployer le mappage vers FlowForce Server, le nom du composant doit être unique.
- Il est recommandé de n'utiliser que des caractères qui peuvent être saisis dans la ligne de commande. Les caractères de types nationaux peuvent présenter des encodages différents dans Windows et dans la ligne de commande.
- E Fichier de schéma

Spécifie le nom ou le chemin du fichier de schéma XML utilisé par MapForce pour valider et mapper des données. Pour changer de fichier de schéma, cliquez sur **Parcourir** et choisissez un nouveau fichier. Pour éditer le fichier dans <u>Altova XMLSpy</u>, cliquez sur **Éditer**.

Fichier d'entrée XML

Spécifie le fichier d'instance XML à partir duquel MapForce lira des données. Ce champ est important pour un composant de source et est rempli lorsque vous avez tout d'abord créé le composant et l'attribuez à un fichier d'instance XML. Dans le composant source, le nom de fichier d'instance est également utilisé pour détecter l'élément racine XML et le schéma référencé, et pour valider par rapport au schéma sélectionné. Pour changer de fichier de schéma, cliquez sur **Parcourir** et choisissez un nouveau fichier. Pour éditer le fichier dans <u>Altova XMLSpy</u>, cliquez sur **Éditer**.

Fichier de sortie XML

Spécifie le fichier d'instance XML à partir duquel MapForce écrira des données. Ce champ est significatif pour un composant cible. Pour changer de fichier de schéma, cliquez sur **Parcourir** et choisissez un nouveau fichier. Pour éditer le fichier dans <u>Altova XMLSpy</u>, cliquez sur **Éditer**.

■ Préfixe pour l'espace de noms cible

Vous permet de saisir un préfixe pour l'espace de nom cible. Avant d'assigner le préfixe, assurez-vous que l'espace de nom cible est défini dans le schéma cible.

■ Ajouter une référence schéma/DTD

Ajoute le chemin du fichier de schéma XML référencé vers l'élément racine de la sortie XML. Le chemin de schéma saisi dans ce champ est écrit dans un/des fichier(s) d'instance cible généré(s) dans l'attribut xsi:schemaLocation ou dans la déclaration DOCTYPE si un DTD est utilisé.

*MapForce Professional et Enterprise Edition* : Si vous générez du code dans XQuery ou C++, ajouter la référence DTD n'est pas pris en charge.

La saisie d'un chemin dans ce champ vous permet de définir où le fichier de schéma référencé par le fichier d'instance XML devra se trouver. Cela permet de garantir que l'instance de sortie peut être validée à la destination de mappage lorsque le mappage est exécuté. Vous pouvez saisir une adresse http://dans ce champ, ainsi qu'un chemin relatif ou absolu.

Désactiver cette option vous permet de déconnecter l'instance XML depuis le schéma XML référencé ou le DTD. Ceci peut être utile, par exemple, si vous voulez envoyer la sortie XML à quelqu'un qui n'a pas accès au schéma XML sous-jacent.

E Écrire la déclaration XML

Par défaut, l'option est activée, ce qui signifie que la déclaration XML est écrite dans la sortie. La table cidessous montre comment cette fonction est prise en charge dans les langages cibles de MapForce et les moteurs d'exécution.

Langage cible/ Moteur d'exécution	Lorsque la sortie est un fichier	Lorsque la sortie est un string
Built-In éditions Professional et Enterprise)	Oui	Oui
MapForce Server Professional et Enterprise editions)	Oui	Oui
XSLT, XQuery	Oui	Non
Générateur de code (C++, C#, Java) (Professional et Enterprise editions)	Oui	Oui

■ Convertir les valeurs en types de cible

Cette option vous permet de définir() les types de schéma XML cible doivent être utilisés du mappage, ou(ii) si les données mappées vers le composant cible doivent être traitées en tant que valeurs string. Par défaut, ce paramètre est activé. La désactivation de cette option vous permet de retenir le formatage précis des valeurs. Par exemple, cela peut être utile pour satisfaire une facette de pattern dans un schéma exigeant un nombre spécifique de chiffres décimaux dans une valeur numérique. Vous pouvez utiliser des fonctions de mappage pour formater le nombre en tant que string dans le format requis, puis mapper ce string dans la cible.

Veuillez noter que la désactivation de cette option désactivera la détection de valeurs invalides, par exemple écrire des lettres dans des champs numériques.

■ Sortie Pretty print

Reformate le document XML de sortie pour lui donner un aspect structuré. Chaque nœud enfant est décalé de son parent par un seul caractère de tabulation.

Gréer signature numérique (Enterprise Edition)

Vous permet d'ajouter une signature numérique dans le fichier d'instance de sortie XML. L'ajout d'une signature numérique est possible si vous sélectionnez Built-in en tant que langage de a transformation .

### Encodage de sortie

Vous permet de spécifier les paramètres suivants du fichier d'instance de sortie :

- Nom d'encodage
- Tri d'octets

• Si le caractère byte order mark (BOM) doit être inclus

Par défaut, tout nouveau composant à l'encodage défini dans l'option *Encodage par défaut pour les nouveaux composants*. Vous pouvez accéder à cette option depuis **Outils | Options** (section *Généralités*).

Si le mappage génère XSLT 1.0/2.0, l'activation de la case à cocher *Byte Order Mark* n'a pas de conséquence, étant donné que ces langages ne prennent pas en charge Byte Order Marks.

### Fichier de Feuille de style StyleVision Power

Cette option vous permet de sélectionner ou de créer un fichier de feuille de style Altova StyleVision. Ces fichiers vous permettent de sortir des données depuis le fichier d'instance XML à une variété de formats appropriés pour le reporting, comme HTML, RTF et autres. Voir aussi <u>Utiliser des chemins relatifs sur un</u> <u>Composant</u><sup>(33)</sup>.

### Autres paramètres

Activer les optimisations de traitement d'entrée basées sur min/maxOccurs

Cette option vous permet une gestion spéciale pour les séquences connues pour contenir exactement un item, comme les attributs ou les éléments enfant requis avec minOccurs et maxOccurs="1". Dans ce cas, le premier item de la séquence est extrait, puis l'item est traité directement en tant que valeur atomique (et pas en tant que séquence).

Si les données d'entrée **ne sont pas valides** par rapport au schéma, une séquence vide peut survenir dans un mappage, ce qui interrompt le mappage et affiche un message d'erreur. Pour permettre le traitement d'une **entrée invalide**, désactivez cette case à cocher.

Enregistrer tous les chemins de fichier relatifs au fichier MFD

Lorsque cette option est activée, MapForce enregistre les chemins de fichier affichés dans le dialogue **Paramètres de composant** relatif à l'emplacement du fichier de MapForce Design (.mfd). Voir aussi Utiliser des chemins relatifs et absolus<sup>83</sup>.

# 4.1.2 Types dérivés

Cette rubrique explique comment utiliser les types dérivés dans les mappages. Les types dérivés sont définis dans la <u>Spécification de schéma W3C XML (la section 2.5.2)</u>. Pour un bref aperçu des types dérivés et primitifs, voir <u>la documentation Microsoft</u>. Pour utiliser les types dérivés dans un mappage, vous devez spécifier l'attribut xsi:type dans votre fichier XML (par ex., <Address xsi:type="UK-Address">).

### Scénarios possibles

Cette sous-section décrit un scénario possible d'utilisation d'un type dérivé. Par exemple, nous avons une entreprise avec deux filiales : une au Royaume-Uni et l'autre aux États-Unis d'Amérique. Maintenant, nous voulons avoir deux listes (UKCustomers et USCustomers), dont chacune inclura des informations sur l'adresse de la filiale respective et tous les clients associés à cette filiale.

#### Composants de structure

### Définition des types dérivés

Les captures d'écran ci-dessous illustrent la définition des types dérivés appelés US-Address et UK-Address (<u>XMLSpy</u> mode Schéma). Les éléments UK-Address et UK-Address ont le même type de base appelé AddressType qui inclut les éléments Name, Street et City. Dans l'élément US-Address, le type de base a été élargi pour inclure Zip et State, tandis que l'élément UK-Address inclut le type de base et l'élément Postcode. À des fins d'illustration, nous mapperons uniquement l'élément UK-Address vers le fichier cible.



### Type dérivé dans un mappage

Les instructions ci-dessous montrent comment mapper les données depuis le type dérivé. Notre objectif est de mapper l'information sur les bureaux au UK vers l'élément UKCustomers. Les exemples de fichiers sont disponibles dans le dossier Tutoriel.

- 1. Dans le menu **Insérer**, cliquez sur **Schéma XML/Fichier** et ouvrez DerivedTypeSource.xml. Ce fichier XML est basé sur DerivedTypeSource.xsd.
- 2. Insérez le fichier cible appelé DerivedTypeTarget.xsd. Notez que le schéma cible ne doit pas inclure l'attribut xsi:type.
- 3. Cliquez sur la touche <u>TYPE</u> à côté de l'élément Address dans le composant source. Cette touche indique que les types dérivés existent pour cet élément dans le schéma.
- 4. La boîte de dialogue **Types dérivés** (*voir la capture d'écran ci-dessous*) vous permet de sélectionner tous types dérivés disponibles pour cet élément spécifique. Dans notre modèle de mappage, nous voulons que seule UK-Address soit mappée.



- 5. Dès que vous sélectionnez la case à cocher à côté du type dérivé UK-Address, un nouvel élément appelé Address xsi:type="UK-Address" apparaît dans le composant.
- 6. Maintenant, connectez les nœuds tel qu'affichés dans le mappage ci-dessous.



### <u>Sortie</u>

Cliquez sur le volet Sortie vous montrera le résultat suivant :

Le modèle de mappage est enregistré sous Tutorial\DerivedType.mfd. Vous pouvez également ajouter un autre fichier XML source qui comprend des informations sur les clients au UK et mapper ces données au nœud Customers dans le composant cible. De cette manière, l'élément UKCustomers inclura l'information sur l'adresse UK et tous les clients associés à cette filiale.

# 4.1.3 Valeurs NULL

Cette section décrit comment MapForce gère les valeurs NULL dans les composants source et cible. Pour pouvoir utiliser l'attribut xsi:nil="true" dans votre fichier XML, vous devez spécifier l'attribut nillable="true" pour les élément(s) dans votre fichier de schéma. Pour en savoir plus sur les attributs nillable et xsi:nil, voir la spécification W3C. Notez que l'attribut xsi:nil n'est pas visible dans une arborescence de composant dans le volet **Mappage**.

Les sous-sections ci-dessous illustrent quelques scénarios possibles de valeurs NULL de mappage.

### Valeurs NULL dans les composants XML

Cette sous-section décrit quelques uns des scénarios possibles de mappage des éléments avec un attribut xsi:nil="true".

<u>Uniquement l'élément source a xsi:nil="true"/Les deux éléments source et cible ont xsi:nil="true"</u> Ce scénario a les conditions suivantes :

- La connexion est <u>target-driven<sup>91</sup></u>.
- L'élément source a un attribut xsi:nil="true". L'élément cible correspondant n'a pas cet attribut.
- En alternative, les deux éléments source et cible peuvent avoir les attributs xsi:nil="true".
- Les attributs nillable="true" doivent être définis dans les schémas source et cible.
- Les éléments source et cible sont de type simple.

Dans ce cas, l'élément cible aura l'attribut xsi:nil="true" dans le fichier de sortie, tel qu'affiché dans le modèle de fichier de sortie ci-dessous (*en surbrillance jaune*).

```
<book id="7">
<author>Edgar Allan Poe</author>
<title>The Murders in the Rue Morgue</title>
<category xsi:nil="true"/>
<year>1841</year>
<OrderID id="213"/>
</books>
```

**Note :** si l'attribut nillable="true" n'est *pas* défini dans le schéma cible, l'élément cible correspondant sera vide dans la sortie.

<u>Seul l'élément cible a xsi:nil="true"</u> Ce scénario a les conditions suivantes :

- La connexion est <u>target-driven</u><sup>91</sup>.
- L'élément source n'a pas d'attribut xsi:nil="true".
- L'élément cible correspondant a un attribut xsi:nil="true".
- L'élément source et cible peut être de type simple ou complexe.

Dans ce cas, l'élément source écrasera l'élément cible contenant l'attribut xsi:nil="true". L'exemple cidessous montre un modèle de fichier de sortie. L'élément <genre> inclut l'attribut xsi:nil="true" dans l'élément cible. Toutefois, cet élément a été écrasé lors de l'exécution de mappage. Pour cette raison, l'élément <genre> (*en surbrillance jaune*) a Fiction dans la sortie.

```
</publication>
<id>l</id>
<author>Mark Twain</author>
<title>The Adventures of Tom Sawyer</title>
<Genre>Rock</Genre>
<year>1876</year>
<OrderID id="124"/>
</publication>
```

<u>L'élément source Complex-type/les éléments complex-type ont xsi:nil="true"</u> Ce scénario a les conditions suivantes :

- La connexion est <u>target-driven</u><sup>91</sup>.
- L'élément source est de type complexe. Dans notre exemple, l'élément source a un attribut id="213" et un attribut xsi:nil="true". L'élément cible correspondant est de type complexe et a un attribut id="124", mais n'a pas d'attribut xsi:nil="true".
- En alternative, les éléments source et cible, dont les deux sont de type complexe, peuvent avoir des attributs xsi:nil="true".

Dans ce cas, l'élément source écrasera l'élément cible (*en surbrillance jaune ci-dessous*. Toutefois, l'attribut xsi:nil="true" ne sera pas écrit dans le fichier de sortie automatiquement. Pour voir l'attribut xsi:nil="true" dans l'élément cible dans le fichier de sortie, utilisez la connexion <u>copy-all</u><sup>95</sup>.

```
<book id="7">
    <author>Edgar Allan Poe</author>
    <title>The Murders in the Rue Morgue</title>
    <year>1841</year>
    <orderID id="213"/>
</books>
```

### Fonctions utiles

Les fonctions suivantes pourraient vous aider à vérifier, remplacer et assigner les valeurs NULL :

- <u>is-xsi-nil</u><sup>675</sup> : Aide à vérifier explicitement si un élément source a un attribut xsi:nil définit à true.
- <u>substitute-missing</u><sup>200</sup> : Substitue une valeur NULL dans l'élément source avec quelque chose de spécifique.
- <u>set-xsi-nil</u><sup>677</sup>: Assigne l'attribut xsi:nil="true » à un élément cible. Ceci fonctionne pour les éléments cible de types simple et complexe.
- <u>substitute-missing-with-xsi-nil</u><sup>678</sup>: S'il y a un contenu, il sera écrit à l'élément cible ; s'il y a des valeurs qui manquent, utilisez cette fonction résultera en élément cible avec un attribut xsi:nil="true" dans la sortie.\_\_
- Connecter une fonction <u>exists</u><sup>663</sup> à un élément source avec une valeur NULL retourne true, même si l'élément n'a pas de contenu.

Veuillez noter que les fonctions qui génèrent xsi:nil ne peuvent pas être passées par les fonctions ou composants qui opèrent uniquement sur des valeurs (telles que la fonction **if-else**).

### Valeurs NULL dans les composants de base de données

Cette sous-section affiche comment les valeurs NULL sont traitées dans les composants de base de données.

### Mapper les champs de base de données NULL dans les éléments NULL

Les éléments cibles qui obtiennent les valeurs NULL depuis les champs de base de données ne sont pas créés dans la sortie automatiquement. Pour voir de tels éléments dans la sortie, vous devez (i) ajouter les attributs nillable="true" aux éléments cibles pertinents dans le fichier de schéma et (ii) utilisez la fonction <u>substitute-missing-with-xsi-nil</u><sup>678</sup> dans le mappage. L'exemple ci-dessous montre comment gérer les valeurs NULL dans les mappages avec un composant de base de données source.

### Table d'application dans le volet DB Query.

L'exemple de mappage est situé dans le chemin suivant : Tutorial\DBNullToXML.mfd. Pour notre exemple, nous avons choisi uniquement une table (Application) depuis la base de données Accounts (voir ci-dessous).

đ	1 🖻				
	AppID 🔹	AppName •	Description •	Category •	URL •
1	1	Altova MapForce	Best data mapping tool!	IDE	https://www.altova.com/mapforce
2	2	Notepad	[NULL]	[NULL]	[NULL]

Pour voir la table Application, suivez les étapes suivantes :

- Ouvrez le volet DB Query.
- Sélectionnez la base de données Accounts pour voir sa structure dans le Navigateur de la base de données.
- Cliquez avec la touche de droite dans la table Application et cliquez sur Afficher dans l'Éditeur SQL | SELECT.
- Cliquez sur le bouton (Exécuter Requête). La table Application apparaîtra dans l'onglet Résultats.

Pour en savoir plus sur les requêtes de bases de données, voir <u>Parcourir et interroger des bases de</u> <u>données</u><sup>332</sup>.

### <u>Mappage</u>

La table Application ci-dessus montre que le deuxième enregistrement a des valeurs NULL dans les champs Description, Catégorie et URL. À des fins d'illustration, nous mapperons presque la totalité des colonnes directement vers les éléments cible correspondants. Pour la colonne URL, nous utiliserons la fonction **substitute-missing-with-xsi-nil** pour que les valeurs NULL dans l'élément cible ait un attribut xsi:nil="true" (voir le mappage ci-dessous).



### <u>Sortie</u>

Le fichier de sortie ci-dessous affiche que le premier enregistrement de la table a été entièrement écrit à la sortie, tandis que le deuxième enregistrement a été écrit uniquement partiellement. Les valeurs de base de données NULL sont absentes de la sortie, à l'exception de l'élément uRL. Depuis que l'élément uRL est doté de l'attribut nillable="true" dans le fichier de schéma et nous avons utilisé la fonction substitute-missingwith-xsi-nil, l'élément uRL est désormais doté de l'attribut xsi:nil="true" dans la sortie (*en surbrillance jaune*).

```
<Application>
<AppID>1</AppID>
<AppID>1</AppID>
<AppName>Altova MapForce</AppName>
<Description>Best data mapping tool!</Description>
<Category>IDE</Category>
<URL>https://www.altova.com/mapforce</URL>
</Application>
<Application>
<AppID>2</AppID>
<AppName>Notepad</AppName>
<URL xsi:nil="true"/>
</Application>
</Pre>
```

### Mapper les éléments NULL aux champs de base de données NULL

Lorsque vous mappez un élément NULL XML à la colonne de la base de données, MapForce écrit la valeur NULL à la colonne de base de données correspondante. Vous pouvez aussi utiliser la fonction <u>set-null</u><sup>(24)</sup> si vous voulez définir un champ de base de données à NULL. Pour en savoir plus sur les fonctions liées à la base de données, voir <u>Voir la bibliothèque BD</u><sup>(23)</sup>.

## 4.1.4 **Commentaires et Instructions de traitement**

Les commentaires et instructions de traitement sont définies dans les <u>Spécifications W3C</u>. Cette rubrique explique comment insérer des commentaires et instructions de traitement dans les composants XML cible. Notez que les commentaires et nœuds de l'instruction de traitement n'ont que des connexions d'entrée. Les commentaires et instructions de traitement ne peuvent pas être définis pour les nœuds qui font partie d'une <u>connexion copy-all</u><sup>45</sup>.

### Insérer un commentaire/instruction de traitement

Pour insérer une instruction de traitement ou un commentaire, suivez les étapes ci-dessous :

 Double-cliquez sur un élément dans le composant et sélectionnez Ajouter Commentaire/Instruction de traitement Avant/Après. Lorsque vous insérez une instruction de traitement, vous allez aussi avoir besoin de saisir son nom. Dans cet exemple ci-dessous, une instruction de traitement appelée xmlstylesheet a été insérée après l'élément state.



- 3. Pour fournir la valeur d'un commentaire ou une instruction de traitement, vous pouvez utiliser une constante, par exemple (*voir la capture d'écran ci-dessus*).
- **Note:** les instructions de traitement multiples peuvent être ajoutés avant ou après tout élément dans le composant de cible.
- **Note :** seul un commentaire peut être ajouté avant et après un nœud de cible unique. Pour créer des commentaires multiples, utilisez <u>la fonction d'entrée dupliquée</u><sup>[8]</sup>.

#### Supprimer un commentaire/instruction de traitement

Pour supprimer un commentaire/instruction de traitement, double-cliquez sur le nœud respectif, sélectionnez **Commentaire/Instruction de traitement**, puis sélectionnez **Supprimer Commentaire/Instruction de traitement** depuis le menu contextuel.

# 4.1.5 Sections CDATA

Les sections CDATA sont utilisées pour échapper des blocs de texte contenant des caractères qui devraient normalement être interprétés en tant que balise. Pour plus d'informations sur les sections de CDATA, voir la <u>Spécification W3C</u>. Les nœuds cible recevant des données comme sections CDATA peuvent être comme suit : Données XML, données XML intégrées dans les champs de base de données et éléments enfant XML de dimensions typées dans une cible XBRL. Les sections CDATA peuvent aussi être définies sur des nœuds dupliqués et des nœuds xsi:type.

Pour créer une section CDATA, double-cliquez sur le nœud cible pertinent et sélectionnez **Écrire contenu comme section CDATA**. Une invitation apparaît vous avertissant que les données d'entrée ne doivent pas contenir le délimiteur de fermeture de la section CDATA <u>]]></u>. L'icône <u>[c..</u> apparaît en-dessous de l'onglet élément, qui indique que ce nœud cible est maintenant défini comme section CDATA.

### Exemple

Cet exemple ci-dessous montre un scénario dans lequel la section CDATA pourrait être utile. L'exemple de mappage appelé MapForceExamples\HTMLinCDATA.mfd (voir la capture d'écran ci-dessous) a les aspects suivants :

- L'élément Subsection a du contenu mixte. Pour plus d'information sur les nœuds de contenu mixte, voir <u>Connexions orientées vers la source</u><sup>91</sup>.
- Avec l'aide de la fonction concat, le contenu de l'élément Trademark aura les balises <b></b>.
- Le contenu de l'élément Keyword aura des balises <i></i>.
- Les données avec les nouvelles balises sont transmises aux nœuds text() dupliqués dans le même ordre que dans le document source.
- La sortie du nœud MixedContent est ensuite transmise dans le nœud Description dans le composant cible ShortInfo. Le nœud Description a été défini comme section CDATA.



### <u>Sortie</u>

Cliquez dans le volet **Sortie** pour consulter la section CDATA dans le nœud Description (*voir la capture d'écran ci-dessous*).

7	⇔ <info></info>
8	<title>MapForce</title>
9	<description><![CDATA[Altova <b>MapForce</b> 2014 Enterprise Edition is the premier <i>XML</i></th></tr><tr><th></th><th>/ <i>database</i> / <i>flat file</i> / <i>EDI</i> data mapping tool that auto-generates mapping code in</th></tr><tr><th></th><th><i>XSLT</i> 1.0/2.0, <i>XQuery</i> , <i>Java</i> , <i>C++</i> and <i>C#</i> . It is the definitive tool for</th></tr><tr><th></th><th>data integration and information leverage.]]></description>
10	-

# 4.1.6 Caractères génériques - xs:any / xs:anyAttribute

Cette rubrique explique comment gérer les caractères génériques dans les mappages. Les caractères génériques xs:any (et xs:anyAttribute) vous permettent d'utiliser tout élément/attribut depuis les schémas. Pour plus d'informations sur les caractères génériques, voir la <u>Spécification W3C</u>.

#### Caractères génériques dans la définition de schéma

La capture d'écran ci-dessous affiche qu'un élément xs:any a été défini comme élément enfant de l'élément Person (mode Schéma dans <u>Altova XMLSpy</u>).



Caractères génériques dans MapForce.

Lorsqu'un caractère générique est défini pour un élément et/ou attribut, ce nœud aura un bouton 🗮 (**Changer Sélection**) à côté de celui-lui (*voir la capture d'écran ci-dessous*).



### Sélection de caractère générique

Désormais, notre objectif est d'ajouter un autre élément comme nœud séparé. Cliquez sur le bouton bouver voir la liste des éléments que vous pouvez ajouter à l'arborescence. Notez que seuls les éléments et attributs globalement déclarés dans votre schéma peuvent être vus dans la boîte de dialogue **sélections de caractère générique**(*voir la capture d'écran ci-dessous*).

🙁 Séle	ections de caractère générique	$\times$	
Sélectio	nner des éléments globaux à afficher dans des nœuds séparés :		
	Address	~	
	Altova		
	Department		
	Office		
	Person		
		w.	
Pour uti différen	liser les éléments ou les attributs depuis un schéma t, cliquer sur Importer un schéma différent		
Afin d'utiliser les éléments avec un nom et un contenu arbitraire, veuillez plutôt passer à "Afficher les éléments enfants avec un nom dynamique" sur l'élément parent.			
	OK Annuler		

Pour notre exemple, nous avons choisi Département. Notez que les éléments de caractère générique et les attributs sont insérés après le nœud avec le bouton 🗐. Maintenant, nos composants ont l'air comme suit :

🕀 🜔 Person
= xs:anyAttribute
PrimaryKey
() ForeignKey
() EMail
🕻 xs:any 🔚 Annie
Department (xs:any)
First
() Last
PhoneExt
() Title

Vous pouvez maintenant mapper de/vers ces nœuds comme d'habitude. Dans un composant, les éléments de caractère générique et les attributs sont marqués avec (xs:any) et (xs:anyAttribute), respectivement (voir la capture d'écran ci-dessus).

### Supprimer les caractères génériques

Pour supprimer un nœud de caractère générique, cliquez sur le bouton 🗐 et la case à cocher correspondante dans la boîte de dialogue **sélections de caractère générique**.

### Éléments/attributs d'un schéma différent

La boîte de dialogue **sélections de caractère générique** (*voir ci-dessus*) vous permet d'utiliser les éléments/attributs d'un schéma différent. Cliquer sur le bouton **Importer un schéma différent** vous donnera

les options suivantes : (i) importer un fichier de schéma et (ii) générer un schéma wrapper (voir la description ci-dessous).

#### Importer un schéma

L'option **Importer schéma** importe le schéma externe dans le schéma actuel attribué au composant. Veuillez noter que cette option écrase le schéma existant du composant sur le disque. Si le schéma actuel est un schéma à distance qui a été ouvert depuis une URL (voir <u>Ajouter des composants depuis une URL</u><sup>78</sup>) et non depuis le disque, il ne peut pas être modifié. Dans ce cas, utilisez l'option **Générer schéma wrapper**.

#### Générer un schéma wrapper

L'option **Générer schéma wrapper** crée un nouveau fichier de schéma appelé un *schéma wrapper*. L'avantage de l'utilisation de cette option est que le schéma existant du composant n'est pas modifié. Au lieu, un nouveau schéma sera créé qui comprendra aussi bien le schéma existant que le schéma à importer. Si vous choisissez cette option, vous serez invité à choisir où le schéma wrapper devra être enregistré. Par défaut, le schéma de wrapper a le format suivant : somefile-wrapper.xsd.

Une fois avoir enregistré le schéma wrapper, celui-ci sera attribué automatiquement par défaut au composant. MapForce vous demandera également si vous voulez ajuster l'emplacement du schéma pour que vous puissiez référencer le schéma principal précédent. Cliquez sur **Oui** pour passer au schéma précédent ; sinon, cliquez sur **Non** pour garder l'attribution du schéma wrapper récemment créé au composant.

### Caractères génériques et noms de nœud dynamiques

Néanmoins, il peut se produire des situations dans lesquelles les éléments et/ou les attributs dans une instance sont trop nombreux pour être déclarés dans le schéma. Considérez le fichier exemple suivant :

Pour de telles situations, il est recommandé d'utiliser les noms dynamiques de nœuds à la place des caractères génériques. Pour plus d'information, voir <u>Nom du nœud stratégies de mappage</u><sup>851</sup>.

# 4.1.7 Espaces de noms personnalisés

Lorsqu'un mappage produit une sortie XML, MapForce dérive automatiquement l'espace de nom (ou définit des espaces de noms) de chaque élément et attribut depuis le schéma cible. Il s'agit du comportement par défaut approprié pour la plupart des scénarios de la génération de sortie XML. Toutefois, il existe des cas pour lesquels vous voulez déclarer manuellement l'espace de nom d'un élément directement depuis le mappage.

Déclarer des espaces de noms personnalisés est uniquement pertinent pour les composants XML cible, et s'applique uniquement aux éléments. La commande **Add Namespace** n'est pas disponible pour les attributs, nœuds du caractère générique et pour les nœuds qui reçoivent les données de la connexion copy-all.

Pour comprendre comment les espaces de noms personnalisés fonctionnent, suivez les instructions dans la sous-section ci-dessous.

### Déclarer l'espace de nom manuellement

Pour cet exemple, vous allez avoir besoin du mappage suivant : BasicTutorials\Tut1-SchemaToSchema.mfd.

#### Ajouter un espace de nom

Ouvrez le mappage, cliquez avec la touche de droite sur le nœud bibliothèque dans le composant BooksOutput et sélectionnez Add Namespace depuis le menu contextuel. Désormais, deux nœuds sont disponibles dans l'élément bibliothèque : namespace et prefix (voir la capture d'écran ci-dessous).



### Fournir les valeurs d'espace de noms

La prochaine étape est de fournir les valeurs aux nœuds namespace et prefix. À cette fin, nous utiliserons les deux constantes avec les valeurs de string suivantes : altova.library et lib (voir la capture d'écran cidessous).



**Note :** les deux connecteurs d'entrée namespace et prefix doivent être mappés, même si vous leur fournissez des valeurs vides.

### <u>Sortie</u>

Dans la sortie générée, un attribut xmlns:<prefix>="<namespace>" est ajouté à l'élément, où <prefix> et <namespace> sont des valeurs qui proviennent du mappage. La sortie générée ressemblera à l'exemple suivant(*notez la partie en surbrillance*) :

```
<?xml version="1.0" encoding="UTF-8"?>
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:lib="altova.library" xsi:noNamespaceSchemaLocation="Library.xsd">
...
```

Vous pouvez aussi déclarer plusieurs espaces de noms pour le même élément, si nécessaire. À cette fin, cliquez une nouvelle fois sur la touche de droite dans le nœud et sélectionnez **Ajouter espace de noms** depuis le menu contextuel. Une nouvelle paire de nœuds d'espace de noms et de préfixe deviennent disponibles, auxquels vous pouvez connecter les valeurs du nouveau préfixe et de l'espace de noms.

### Déclarer un espace de nom par défaut

Si vous voulez déclarer un espace de noms par défaut, mappez une valeur de string vide à prefix. La sortie ressemblera alors à l'exemple suivant (*notez la partie en surbrillance*):

```
<?xml version="1.0" encoding="UTF-8"?>
<library xmlns="altova.library" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="Library.xsd">
...
```

Si vous souhaitez créer des préfixes pour les noms d'attribut, par exemple <number

prod:id="prod557">557</number>, vous pouvez le faire soit en utilisant l'accès dynamique aux attributs d'un nœud (voir <u>Mapper des noms de nœud</u><sup>(651</sup>), soit en éditant le schéma de manière à ce qu'il ait un attribut prod:id pour <number>.

### Supprimer l'espace de noms

Pour supprimer une déclaration d'espace de noms précédemment ajoutée, cliquer avec la touche de droite dans le nœud ns:namespace et choisissez **Supprimer espace de noms** du menu contextuel.

## 4.1.8 Gestionnaire de schéma

XML Gestionnaire de schéma est un outil qui propose un moyen centralisé d'installer et de gérer des schémas XML (DTD pour XML et Schémas XML) pour une utilisation sur toutes les applications XBRL Altova, y compris MapForce

- Sur Windows, Gestionnaire de schéma a une interface utilisateur graphique (voir la capture d'écran cidessous) et est aussi disponible dans la ligne de commande. (Les applications desktop d'Altova sont disponibles sur Windows uniquement ; voir la liste ci-dessous.)
- Sur Linux et Mac Gestionnaire de schéma, l'outil est disponible uniquement dans la ligne de commande. (Les applications serveur d'Altova sont disponibles sur Windows, Linux et macOS ; *voir la liste ci-dessous*.)

🌆 XML Schema Manager	×
<ul> <li>XML Schema Manager</li> <li>CBCR - Country-by-Country Reporting</li> <li>CML - Chemical Markup Language</li> <li>DAISY (DTD) - Document Type Definition files for the Digital Accessible Information System</li> <li>DITA - OASIS Darwin Information Typing Architecture</li> <li>DITA (DTD) - OASIS Darwin Information Typing Architecture</li> <li>DOCBOOK (DTD) - Docbook Markup Language</li> <li>✓ EPUB - Electronic Publication</li> <li>HL7v3 NE - Health Level 7 V3, Normative Edition</li> <li>HR-XML - Human Resources Open Standards</li> <li>J2EE (DTD) - Java 2 Platform Enterprise Edition DTDs</li> </ul>	×
<ul> <li>MCAXML - National Coffee Association XML</li> <li>NEWSML (DTD) - News Markup Language</li> <li>NITF - News Industry Text Format</li> <li>OOXML - Office Open ECMA-376 XML Schema files</li> </ul>	
<ul> <li>P3P - Platform for Privacy Preferences Project</li> <li>RIXML - Research Information Exchange Markup Language</li> <li>SMIL (DTD) - Synchronized Multimedia Integration Language</li> </ul>	
<ul> <li>SVG (DTD) - Scalable Vector Graphics</li> <li>TEILITE - Text Encoding Initiative Lite</li> <li>TLD (DTD) - Java Server Pages Tag Library</li> </ul>	
Select the packages you want to install and then click "Apply".	^
< <p><u>Patch Selection</u> <u>Deselect All</u> <u>Reset Selection</u> <u>Apply</u></p>	> <u>C</u> lose

### Application d'Altova qui fonctionnent avec Schema Manager

Applications desktop (Windows uniquement)	Applications de serveur (Windows, Linux, macOS)
XMLSpy (toutes éditions)	RaptorXML Server, RaptorXML+XBRL Server

MapForce ( toutes éditions )	StyleVision Server
StyleVision ( toutes éditions )	
Authentic Desktop Enterprise Edition	

### Installation et désinstallation de Gestionnaire de schéma

Gestionnaire de schéma est installé automatiquement quand vous installez d'abord une nouvelle version de l'Altova Mission Kit Enterprise Edition ou toute application XML-schema-aware d'Altova (*voir la table ci-dessus*).

De même, il est supprimé automatiquement lorsque vous désinstallez la dernière application XML-schemaaware 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 <u>catalogue XML</u> pour mapper les références de schéma aux fichiers locaux. Dans le cas de larges schémas XML, le traitement sera plus rapide que si les schémas étaient à un emplacement à distance.
- Tous les schémas majeurs sont disponibles via Gestionnaire de schéma et sont régulièrement mis à jour pour les dernières versions. Ceci vous fournit une ressource unique pour gérer tous vos schémas et les mettre à disposition de toutes les applications XML-schema-aware d'Altova.
- Les changements réalisés dans Gestionnaire de schéma prennent effet pour tous les produits d'Altova sur cet appareil.

### Comment cela fonctionne ?

Altova stocke tous les schémas XML utilisés dans les produits Altova en ligne. Ce référentiel est mis à jour lorsque de nouvelles versions de schémas sont publiées. Gestionnaire de schéma affiche des informations sur les derniers schémas disponibles lorsqu'ils sont appelés dans son formulaire GUI de même que sur CLI. Vous pouvez ensuite installer, mettre à jour ou désinstaller les schémas via Gestionnaire de schéma.

Gestionnaire de schéma installe également les schémas d'une autre manière. Sur le site web d'Altova (<u>https://www.altova.com/schema-manager</u>), vous pouvez sélectionner un schéma et ses Schémas dépendants que vous souhaitez installer. Le site web préparera un fichier de type <code>.altova\_xmlschemas</code> 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 <u>installer</u> <sup>(61)</sup>, 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 :

Windows	C:\ProgramData\Altova\pkgs\.cache
Linux	/var/opt/Altova/pkgs\.cache
macOS	/var/Altova/pkgs

Ce répertoire cache est mis à jour régulièrement avec le dernier statut des schémas dans l'emplacement de stockage en ligne d'Altova. Ces mises à jour sont réalisées aux moments suivants :

- À chaque fois que vous lancez Gestionnaire de schéma.
- Lorsque vous exécutez MapForce pour la première fois dans un jour donné du calendrier.
- Si MapForce 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 <u>mise à jour</u><sup>(154)</sup> 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 <u>reset</u><sup>153</sup>, et (ii) exécutez la commande <u>initialize</u><sup>150</sup>. (En alternative, exécutez la commande <u>reset</u> avec l'option --i.)

# 4.1.8.1 Exécuter Schema Manager

### Interface utilisateur graphique

Vous pouvez accéder à la GUI de Gestionnaire de schéma des manières suivantes :

- Durant l'installation de MapForce: Vers la fin de la procédure d'installation, sélectionnez la case à cocher Invok e 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 MapForce: 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 <u>Altova website</u>: 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*).

🌆 XML Schema Manager	×
<ul> <li>EPUB - Electronic Publication</li> <li>XML Schema files for the Electronic Publication file format. https://www.w3.org/community/epub3/</li> </ul>	*
☑ 🛃 2.0	
HL7v3 NE - Health Level 7 V3, Normative Edition XML schema files for Health Level 7 V3 Normative Editions. https://www.bl7.org/implement/standards/	
<ul> <li>HR-XML - Human Resources Open Standards XML schema files for the Human Resources Open Standards. https://schemas.liquid-technologies.com/HR-XML/2007-04-15/</li> <li>2007</li> </ul>	
Image: NCAXML - National Coffee Association XML	
NEWSML (DTD) - News Markup Language	
NITF - News Industry Text Format XML schema files for the News Industry Text Format. https://iptc.org/std/NITF/	
3.4	
OOXML - Office Open ECMA-376 XML Schema files	
P3P - Platform for Privacy Preferences Project	Ŧ
The following packages will be installed: HR-XML 2007 - Human Resources Open Standards NITF 3.4 - News Industry Text Format	
Patch Selection Deselect All Reset Selection Apply Cance	

### Interface de ligne de commande

Vous pouvez exécuter Gestionnaire de schéma depuis une interface de ligne de commande en sélectionnant son fichier exécutable, xmlschemamanager.exe.
Le fichier xmlschemamanager.exe est situé dans le dossier suivant :

- Sur Windows : C:\ProgramData\Altova\SharedBetweenVersions
- Sur Linux ou macOS (application serveur uniquement) : %INSTALLDIR%/bin, où %INSTALLDIR% est le répertoire d'installation du programme.

Vous pouvez alors utiliser toute commande dans la section de référence de la commande CLI<sup>149</sup>.

Pour afficher l'aide pour la commande, exécutez l'étape suivante :

- Sur Windows : xmlschemamanager.exe --help
- Sur Linux ou macOS (application serveur uniquement) : sudo ./xmlschemamanager --help

# 4.1.8.2 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.

□ 🗹 CBCR - Country-by-Country Reporting	
XML Schema files for the Country-by-Country Reporting.	
https://www.oecd.org/ctp/country-by-country-reporting-xml-schema-user-guide-for-tax-administration	ns.htm
2.0	
🗉 🗌 CML - Chemical Markup Language	
🗉 🗌 DAISY (DTD) - Document Type Definition files for the Digital Accessible Information System	
DITA - OASIS Darwin Information Typing Architecture	
🗄 🗌 DITA (DTD) - OASIS Darwin Information Typing Architecture	
🗉 🗌 DOCBOOK (DTD) - Docbook Markup Language	
EPUB - Electronic Publication	
XML Schema files for the Electronic Publication file format.	
https://www.w3.org/community/epub3/	
2.0	
HL7v3 NE - Health Level 7 V3, Normative Edition	
XML schema files for Health Level 7 V3 Normative Editions.	
https://www.hl7.org/implement/standards/	
2010	
2008	

• 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.

#### <u>Points à noter</u>

- 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 <u>list</u><sup>152</sup> est utilisée avec différentes options pour pour recenser les différentes catégories de schémas :

xmlschemamanager.exe list	Recense tous les schémas installés et disponibles ; ceux pouvant être mis à niveau sont également indiqués.
xmlschemamanager.exe list -i	Recense les schémas installés uniquement ; ceux pouvant être mis à niveau sont également indiqués
xmlschemamanager.exe list -u	Recense les schémas pouvant être mis à niveau

Note : Sur Linux et macOS, use sudo ./ xmlschemamanager list

# 4.1.8.3 Retoucher ou Installer un schéma

## Retoucher un schéma installé

Occasionnellement, des schémas XML peuvent recevoir des patchs (mises à niveau ou révisions) de leurs émetteurs. Lorsque Gestionnaire de schéma détecte que des patchs sont disponibles, ceux-ci sont indiqués dans les listes de schéma de Gestionnaire de schéma et vous pouvez installer les patchs rapidement.

### <u>Dans la GUI</u>

Les patchs sont indiqués par l'icône k. (Voir aussi la rubrique précédente sur les <u>catégories de statut</u>).) Si les patchs sont disponibles, le bouton **Patch Selection** sera activé. Cliquez dessus pour sélectionner et préparer tous les patchs pour installation. Dans la GUI, l'icône de chaque schéma sera patchée de a **a a**, et le volet des Messages en bas du dialogue recense les patchs qui doivent être appliqués. Lorsque vous êtes prêt pour installer des patchs sélectionnés, cliquez sur **Appliquer**. Tous les correctifs seront appliqués ensemble. Notez que si vous décochez un schéma marqué pour une correction, vous désinstallerez de fait ce schéma.

#### <u>Sur le CLI</u>

Pour appliquer un patch dans l'interface de ligne de commande :

- Exécuter la commande <u>list -u</u><sup>152</sup>. Cela liste tout schéma lorsque des mises à niveau sont disponibles.
- 2. Exécutez la commande <u>upgrade</u><sup>155</sup> pour installer les patchs.

## Installer un schéma disponible

Vous pouvez installer des schémas en utilisant soit la GUI Gestionnaire de schéma ou en envoyant à Gestionnaire de schéma les instructions d'install via la ligne de commande.

Note : Si le schéma actuel référence d'autres schémas, les schémas référencées sont aussi installés.

#### <u>Dans la GUI</u>

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 <u>site web d'Altova</u> et générer un fichier téléchargeable <u>.altova\_xmlschemas</u>. 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**.

### <u>Sur le CLI</u>

Pour installer des schémas via la ligne de commande, exécutez la commande <u>install</u><sup>(51)</sup> :

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 <a href="mailto:aname>-<version">name>-<version</a>. (Les identifiants de schémas sont affichés quand vous exécutez la commande <a href="mailto:list">list</a>.) Vous pouvez saisir autant de schémas que vous le souhaitez. Pour plus de détails, voir la description de la commande <a href="mailto:list">list</a>.

Note : sur Linux ou macOS, utilisez la commande sudo ./xmlschemamanager.

#### Installer un schéma requis

Lorsque vous exécutez une commande activée par XML dans MapForce, et que MapForce 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**.

# 4.1.8.4 Désinstaller un schéma, Réinitialiser

## Désinstaller un schéma

Vous pouvez désinstaller des schémas en utilisant soit la GUI Gestionnaire de schéma ou en envoyant à Gestionnaire de schéma les instructions d'installation via la ligne de commande.

**Note :** si la Police que vous voulez désinstaller référence d'autres Schémas, alors les Schémas référencées sont également désinstallées.

#### <u>Dans la GUI</u>

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.

#### <u>Sur le CLI</u>

Pour désinstaller des schémas via la ligne de commande, exécutez la commande désinstaller<sup>163</sup> :

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 <u>list</u><sup>(52)</sup>.) Vous pouvez saisir autant de schémas que vous le souhaitez. Pour plus de détails, voir la description de la commande <u>désinstaller</u><sup>(53)</sup>.

**Note :** sur Linux ou macOS, utilisez la commande sudo ./xmlschemamanager.

## Réinitialiser Gestionnaire de schéma

Vous pouvez réinitialiser Gestionnaire de schéma. Ceci supprime toutes les schémas installés et le répertoire de mise sous cache.

- Dans la GUI, cliquez sur Reset Selection.
- Dans la CLI, exécutez la commande reset<sup>153</sup>.

Une fois avoir exécuté cette commande, vous devrez exécuter la commande <u>initialize</u><sup>(50)</sup>, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécutez la commande <u>reset</u><sup>(153)</sup> avec l'option -i.

Notez que <u>reset</u>  $-i^{(153)}$  restaure l'installation originale du produit, il est recommandé d'exécuter la commande <u>update</u> après avoir réalisé la réinitialisation. En alternative, exécutez la commande <u>reset</u> avec les options -i and -u.

# 4.1.8.5 Interface de ligne de commande (CLI)

Pour appeler Gestionnaire de schéma dans la ligne de commande, vous devez connaître le chemin de l'exécutable. Par défaut, l'exécutable Gestionnaire de schéma est installé dans le chemin suivant :

C:\ProgramData\Altova\SharedBetweenVersions\XMLSchemaManager.exe

**Note :** sur les systèmes Linux et macOS, une fois que vous avez changé le répertoire à celui contenant l'exécutable, vous pouvez appeler l'exécutable avec **sudo** ./xmlschemamanager. Le préfixe ./ indique que l'exécutable est le répertoire actuel. Le préfixe **sudo** indique que la commande que la commande doit être exécutée avec des privilèges root.

## Syntaxe de ligne de commande

La syntaxe générale pour utiliser la ligne de commande est la suivante :

<exec> -h | --help | --version | <command> [options] [arguments]

Dans l'extrait ci-dessus, la barre verticale i 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.

### 4.1.8.5.1 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.

Veuillez noter les points suivants :

- Vous pouvez invoquer de l'aide en saisissant la commande suivie par -h ou --help, par exemple :
   <exec> list-h
- Si vous tapez -h or --help directement après la commande exécutable et avant une commande, vous recevrez une aide générale (pas d'aide pour la commande), par exemple : <exec> -h list

### Exemple

La commande suivante affiche une aide concernant la commande list :

xmlschemamanager help list

## 4.1.8.5.2 info

Cette commande affiche des informations détaillées pour chacun des schémas fournis en tant qu'argument de schéma. Cette information inclut le titre, la version, description, l'éditeur et tout schéma soumis et tout schéma référencé, et informe si le schéma a été installé ou non.

### Syntaxe

<exec> info [options] Schema+

- L'argument schéma est le nom d'un schéma ou une partie du nom de schéma. (Pour afficher une ID de pack de schéma et des informations détaillées sur son statut d'installation, vous devriez utiliser la commande <u>list</u><sup>(152)</sup>.)
- 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

## 4.1.8.5.3 initialize

Cette commande initialise l'environnement Gestionnaire de schéma. Elle crée un répertoire de cache où les informations concernant tous les schémas sont stockées localement. L'initialisation est réalisée automatiquement la première fois qu'une application schema-cognizant d'Altova est installée. Vous n'aurez pas besoin d'exécuter cette commande dans des circonstances normales, mais vous devrez l'exécuter généralement après la commande reset.

### Syntaxe

```
<exec> initialize | init [options]
```

### <u>Options</u>

La commande initialize accepte les options suivantes :

silent,s	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux.
verbose,v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux.
help,h	Afficher l'aide pour la commande.

# Exemple

La commande suivante initialise Gestionnaire de schéma:

xmlschemamanager initialize

# 4.1.8.5.4 install

Cette commande installe un ou plusieurs schémas.

## Syntaxe

<exec> install [options] Schema+

Pour installer de multiples schémas, ajoutez l'argument schéma de nombreuses fois.

L'argument schéma est l'un des suivants :

- Un identifiant de schéma (avoir un format de <name>-<version>, par exemple : cbcr-2.0). Pour trouver les identifiants de schémas que vous voulez, exécutez la commande list <sup>12</sup>. 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 <u>Introduction à SchemaManager : Comment cela fonctionne-t-il ?</u><sup>(39)</sup>.

#### <u>Options</u>

La commande install accepte les options suivantes :

silent,s	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux.
verbose,v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux.
help,h	Afficher l'aide pour la commande.

## Exemple

La commande suivante installe le schéma CBCR 2.0 (Country-By-Country Reporting) et le dernier DocBook DTD:

xmlschemamanager install cbcr-2.0 docbook

## 4.1.8.5.5 list

Cette commande recense les schémas sous la gestion de Gestionnaire de schéma. La liste affiche comme suit

- Tous les schémas disponibles
- · Les schémas contenant le string dans leur nom soumis comme argument de schéma
- Seuls les schémas installés
- Seuls les schémas qui peuvent être mis à niveau

## Syntaxe

<exec> list | ls [options] Schema?

Si aucun argument de schéma n'est soumis, alors toutes les schémas disponibles sont recensés. Autrement, les schémas sont recensés par des options soumises (*voir l'exemple ci-dessous*). Notez que vous pouvez soumettre l'argument de schéma de nombreuses fois.

#### <u>Options</u>

La commande list accepte les options suivantes :

installed,i	Recenser uniquement la liste des schémas installés. Le réglage par défaut est faux.
upgradeable,u	Recenser uniquement les schémas lorsque des mises à niveau (patchs) sont disponibles. Le réglage par défaut est faux.
help,h	Afficher l'aide pour la commande.

## Exemples

- Pour exécuter tous les schémas disponibles, exécutez : xmlschemamanager list
- Pour recenser les schémas installés, exécutez : xmlschemamanager list -i
- Pour recenser tous les schémas qui contiennent soit "doc", soit "nitf" dans leur nom, exécutez xmlschemamanager list doc:

## 4.1.8.5.6 reset

Cette commande supprime tous les schémas installés et le répertoire de mise sous cache. Vous réinitialiserez complètement votre environnement de schéma. Une fois avoir exécuté cette commande, vous devrez exécuter la commande <u>initialize</u><sup>(50)</sup>, pour pouvoir recréer le répertoire de mise sous cache. En alternative, exécuter la commande <u>reset</u> avec l'option -i. Pusique <u>reset</u> -i restaure l'installation originale du produit, nous vous recommandons que vous exécutez la commande <u>update</u><sup>(64)</sup> après avoir réalisé la réinitialisation et l'initialisation. En alternative, exécutez la commande <u>reset</u> avec les options -i and -u

## Syntaxe

<exec> reset [options]

#### **Options**

La commande reset accepte les options suivantes :

init,i	Initialiser Gestionnaire de schéma après le reset. Le réglage par défaut est faux.
update,u	Mettre à jour la liste de schémas disponibles dans le cache. Le réglage par défaut est faux.
silent,s	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux.
verbose,v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux.
help,h	Afficher l'aide pour la commande.

# Exemples

- Pour réinitialiser Gestionnaire de schéma, exécuter : xmlschemamanager reset
- Pour réinitialiser Gestionnaire de schéma et l'initialiser, exécutez : xmlschemamanager reset -i
- Pour réinitialiser Gestionnaire de schéma, initialiser-le et mettez à jour sa liste de schéma, exécutez : xmlschemamanager reset -i-u

## 4.1.8.5.7 uninstall

Cette commande désinstalle un ou plusieurs schémas. Par défaut, tout schéma référencé par la taxonomie actuelle sera également désinstallé. Pour désinstaller uniquement le schéma actuel et garder les schémas référencés, définir l'option --k.

## Syntaxe

<exec> uninstall [options] Schema+

Pour désinstaller de multiples schémas, ajoutez l'argument schéma de nombreuses fois.

#### 154 Composants de structure

L'argument schéma est l'un des suivants :

- Un identifiant de schéma (avoir un format de <name>-<version>, par exemple : cbcr-2.0). Pour trouver les identifiants de schéma qui sont installés, exécutez la commande list -i
   Vous pouvez aussi utiliser un nom de schéma abrégé s'il est unique, par exemple docbook. Si vous utilisez un nom abrégé, alors tous les schémas qui contiennent une abréviation dans leur nom seront désinstallés.
- Le chemin vers un fichier .altova\_xmlschemas téléchargé depuis le site web d'Altova. Pour information sur ces fichiers, voir <u>Introduction à SchemaManager : Comment cela fonctionne-t-il ?</u>

#### **Options**

La commande désinstaller accepte les options suivantes :

keep-references,k	Définir cette option pour garder les schémas référencés. Le réglage par défaut est faux.
silent,s	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux.
verbose,v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux.
help,h	Afficher l'aide pour la commande.

## Exemple

La commande suivante désinstalle les schémas CBCR 2.0 et EPUB 2.0 et leurs dépendances : xmlschemamanager uninstall cbcr-2.0 epub-2.0

La commande suivante désinstalle la taxonomie eba-2.10 mais pas les schémas qu'elle référence : xmlschemamanager uninstall --k cbcr-2.0

## 4.1.8.5.8 update

Cette commande requête la liste des schémas disponibles depuis l'emplacement de stockage en ligne et met à jour le répertoire de mise sous cache local. Vous devriez exécuter cette commande sauf si vous avez réalisé un reset<sup>153</sup> et initialize<sup>150</sup>.

## Syntaxe

<exec> update [options]

#### <u>Options</u>

La commande mise à jour accepte les options suivantes :

silent,s	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux.
verbose,v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux.

--help, --h

Afficher l'aide pour la commande.

# Exemple

La commande suivante met à jour le cache local avec la liste des derniers schémas :

xmlschemamanager update

## 4.1.8.5.9 upgrade

Cette commande met à niveau toutes les schémas installés qui peuvent être mis à niveau à la dernière version *patchée* disponible. Vous pouvez identifier des schémas à mettre à niveau en exécutant la commande <u>list</u> – <u>u</u><sup>(12)</sup>.

**Note :** La commande mettre à niveau supprime une Police dépréciée si aucune version plus récente n'est disponible.

# Syntaxe

<exec> upgrade [options]

#### **Options**

La commande mise à niveau accepte les options suivantes :

silent,s	Afficher uniquement des messages d'erreur. Le réglage par défaut est faux.
verbose,v	Afficher des informations détaillées lors de l'exécution. Le réglage par défaut est faux.
help,h	Afficher l'aide pour la commande.

# 4.2 Bases de données SQL

# Site web d'Altova : & Mappage de base de données

MapForce 2023 offre une prise en charge solide pour le mappage de bases de données en formats XML, fichiers plats et autres formats de base de données. Avec l'édition MapForce Enterprise, vous pouvez également mapper des bases de données en formats EDI, Excel 2007+, JSON, XBRL et des services Web.

La table ci-dessous énumère toutes les bases de données prises en charge. Bien qu'Altova œuvre pour prendre en charge d'autres bases de données, une connexion et un traitement des données réussis ont uniquement été testés avec les bases de données énumérées ci-dessous. Si votre application Altova est une version 64-bit, assurez-vous que vous avez accès aux pilotes de base de données 64-bit nécessaires pour la base de données spécifique à laquelle vous tentez de vous connecter.

Base de données	Notes
Firebird 2.x, 3.x	
IBM DB2 DB2 8.x, 9.x, 10.x, 11.x	
IBM Db2 i6.x, 7.4	Les fichiers logiques sont pris en charge et présentés en tant qu'affichage.
IBM Informix 11.70 et ultérieur	Informix supporte les connexions via ADO, JDBC et ODBC. L'implémentation ne supporte pas les types de données d'objet importants dans un des langages de génération de code. MapForce générera un message d'erreur (pendant la génération de code) si un de ces type de données est utilisé.
MariaDB 10 et ultérieur	
Microsoft Access 2003 et ultérieur	Au moment de la rédaction de ce manuel (début septembre 2019), il n'a y pas de Microsoft Access Runtime disponible pour Access 2019. Vous pouvez vous connecter à une base de données Access 2019 depuis les produits Altova uniquement si Microsoft Access 2016 Runtime est installé et uniquement si la base de données n'utilise pas le type de données « Large Number ».
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005 et ultérieur Microsoft SQL Server on Linux	
MySQL 5 et ultérieur	
Oracle 9i et ultérieur	
PostgreSQL 8 et ultérieur	Les connexions PostgreSQL sont prises en charge aussi bien en tant que connexions natives qu'en tant que connexions sur base pilote par le biais d'interfaces (pilotes) comme ODBC ou JDBC. Les connexions natives ne nécessitent pas de pilotes.

Base de données	Notes
Progress OpenEdge 11.6	
SQLite 3.x	Les connexions SQLite sont prises en charge en tant que connexions directes natives du fichier de base de données SQLite. Aucun pilote séparé n'est nécessaire.
Sybase ASE 15, 16	
Teradata 16	Les connexions sont prises en charge avec ADO.NET, JDBC et ODBC.
	Lorsqu'un mappage insère des données dans une table de base de données, les champs d'identité générés sur base de données ne sont pas pris en charge.

# Mappages de base de données dans des environnements d'exécution variés

Lorsque vous générez un code de programme depuis un mappage ou lorsque vous compilez un mappage vers les fichiers d'exécution MapForce Server, ou lorsque vous déployez un mappage vers FlowForce Server, les détails de connexion de la base de données enregistrés avec les fichiers générés sont adaptés aux pilotes applicables ou pris en charge pour l'environnement cible choisi, comme indiqué dans la table ci-dessous. Par exemple, si le langage de transformation de mappage est défini sur Java, les connexions ADO sont converties en JDBC lorsque le code Java est généré depuis le mappage.

Lorsque le mappage est exécuté dans un environnement autre que MapForce, vous devrez vous assurer que les détails de connexion de la base de données sont pertinents pour l'appareil sur lequel le mappage sera exécuté (par exemple le pilote de base de données est installé, le chemin de base de données est correct, le serveur de base de données est accessible, etc.).

Certains types de connexion de base de données ne sont pas pris en charge dans certains environnements cible, comme indiqué dans la table ci-dessous.

Type de connexion /En vironnement d'exécution	C#	C++	Java	MapForce Server sur Windows	MapForce Server sur Linux/Mac
ADO	Pont ADO	Tel quel	Converti en JDBC	Tel quel	Converti en JDBC
ADO.NET	Tel quel	Défini par l'utilisateur	Converti en JDBC	Tel quel	Converti en JDBC
JDBC	Défini par l'utilisateur	Défini par l'utilisateur	Tel quel	Tel quel	Tel quel
ODBC	Pont ODBC	Pont ODBC	Converti en JDBC	Tel quel	Converti en JDBC

Type de connexion /En vironnement d'exécution	C#	C++	Java	MapForce Server sur Windows	MapForce Server sur Linux/Mac
Native PostgreSQL	Non pris en charge	Non pris en charge	Non pris en charge	Tel quel	Tel quel
Native SQLite	Non pris en charge	Non pris en charge	Non pris en charge	Tel quel	Tel quel

Légende de la table :

- "Tel quel" signifie que le type de connexion de base de données (par exemple, JDBC, ODBC) reste tel que défini dans MapForce.
- "Converti en JDBC" signifie que la connexion de base de données sera convertie en une URL de connexion de base de données semblable à JDBC.
- "Pont ADO" ou "Pont ODBC" signifie que le string de connexion reste tel que défini dans MapForce, mais que le code généré utilisera une classe appropriée qui agit comme un pont ADO (ou un pont ODBC respectivement), par exemple, System.Data.OleDb.OleDbConnection ou System.Data.Odbc.OdbcConnection.
- "Défini par l'utilisateur" signifie que, afin que la connexion puisse fonctionner dans le code généré, vous devrez saisir manuellement les détails de connexion dans le dialogue <u>Paramètres de composant de base de données</u><sup>(231)</sup>. Selon les cas, ces détails de connexion devront être saisis sous **Paramètres spécifiques à BD ADO/OLE** ou sous **Paramètres spécifiques à JDBC**.

Voir aussi :

- Préparer des mappages pour l'exécution de serveur<sup>951</sup>
- Compiler des Mappages sur les fichiers d'exécution MapForce Server<sup>957</sup>
- <u>Déployer des mappages sur FlowForce Server</u><sup>960</sup>
- <u>Générateur de Code</u><sup>1065</sup>

# 4.2.1 Connexion à une source de données

Dans les cas les plus simples, une base de données peut être un fichier local tel qu'un fichier de base de données Microsoft Access ou SQLite. Dans un scénario plus avancé, une base de données peut résider sur un serveur à distance ou un serveur de base données de réseau qui n'utilise pas nécessairement le même système d'exploitation que celui de l'application qui s'y connecte et qui consomme les données. Par exemple, alors que MapForce fonctionne sur un système d'exploitation Windows, la base de données à partir de laquelle vous souhaitez accéder aux données (par exemple MySQL) pourrait, elle, fonctionner sur une machine Linux.

Pour interagir avec des types de bases de données aussi bien locaux qu'à distance, MapForce repose sur les interfaces de connexion de données et les pilotes de base de données d'ores et déjà disponibles sur votre système d'exploitation ou publiés régulièrement par les distributeurs principaux de bases de données. Dans un environnement technologique de bases de données en constante évolution, cette approche permet une plus grande flexibilité et interopérabilité multiplateforme.

Le diagramme suivant illustre, de manière simplifiée, les options de connectivité des données disponibles entre MapForce (illustré en tant qu'application de client générique) et un stockage de données (qui peut être un serveur de base de données ou un fichier de base de données.



\* Les connexions natives directes sont prises en charge pour les bases de données SQLite, PostgreSQL, Couch DB et MongoDB. Pour se connecter à ce type de bases de données, il n'est pas nécessaire d'installer un pilote supplémentaire sur votre système.

Tel qu'indiqué dans le diagramme ci-dessus, MapForce peut accéder à tout type de base de données principale par le biais des technologies d'accès aux données suivantes :

- ADO (Microsoft® ActiveX® Data Objects), qui, à son tour, utilise un fournisseur OLE DB sous-jacent (Object Linking and Embedding, Database)
- ADO.NET (un ensemble de bibliothèques disponibles dans le Microsoft .NET Framework qui permet une interaction avec des données)
- JDBC (Java Database Connectivity)
- ODBC (Open Database Connectivity)
- **Note :** certains fournisseurs ADO.NET ne sont pas pris en charge ou ne le sont que de manière limitée. See <u>Notes de prise en charge d'ADO.NET</u><sup>178</sup>.

## À propos des technologies d'accès aux données

L'interface de connexion aux données que vous choisissez dépend largement de votre infrastructure logicielle existante. En règle générale, vous choisirez la technologie d'accès aux données et le pilote de base de données qui s'intègre le plus étroitement avec le système de base de données auquel vous souhaitez vous connecter. Par exemple, pour se connecter à une base de données Microsoft Access 2013, vous construiriez en principe une chaîne de connexion ADO qui utilise un fournisseur natif comme le **Microsoft Office Access Database Engine OLE DB Provider**. En revanche, pour vous connecter à Oracle, vous choisirez peut-être de télécharger et d'installer les dernières interfaces JDBC, ODBC ou ADO .NET depuis le site Internet Oracle.

Alors que les pilotes pour les produits Windows (comme Microsoft Access ou SQL Server) sont éventuellement déjà disponibles sur votre système d'exploitation Windows, ils peuvent ne pas être disponibles pour d'autres types de base de données. Les distributeurs des principales bases de données publient régulièrement des logiciels de client de base de données et des pilotes accessibles au public qui fournissent un accès multiplateforme aux bases de données respectives par le biais de combinaisons de ADO, ADO.NET, ODBC ou JDBC. De plus, plusieurs pilotes tiers peuvent être disponibles pour une ou plusieurs des technologies mentionnées ci-dessus. Dans la plupart des cas, il existe plusieurs manières de se connecter à la base de données requise depuis votre système d'exploitation, et, par conséquent, depuis MapForce. Les fonctions disponibles, paramètres de performance et les problèmes connus varieront généralement selon la technologie d'accès des données ou pilotes utilisés.

# 4.2.1.1 Lancer l'Assistant de la connexion

MapForce fournit un assistant de connexion à la base de données qui vous guide dans les étapes requises afin de configurer une connexion vers une source de données. Avant de suivre les étapes de l'assistant, veuillez noter que pour certains types de bases de données il est nécessaire d'installer et de configurer séparément plusieurs prérequis de base de données, tels qu'un pilote de base de données ou un logiciel client de base de données. Ceux-ci sont généralement fournis par les distributeurs de base de données respectifs, et comprennent la documentation taillée sur votre version Windows spécifique. Pour une liste de pilotes regroupés par base de données, voir <u>Aperçu des pilotes de base de données</u>.

Pour démarrer l'assistant de connexion à la base de données (*voir la capture d'écran ci-dessous*), suivez les étapes suivantes :

• Dans le menu Insérer, cliquez sur Base de données.

L'assistant de connexion de connexion à la base de données (voir la capture d'écran ci-dessous) est lancé. À gauche de la fenêtre, vous pouvez sélectionner la manière la plus appropriée de connexion à la base de données parmi les possibilités suivantes :

- L'assistant de connexion qui vous invite à choisir votre type de base de données, puis vous guide à travers les étapes pour se connecter à une base de données de ce type
- Sélectionnez une connexion existante
- Sélectionnez une technologie d'accès aux données : ADO, ADO.NET, ODBC ou JDBC
- Utilisez une ressource globale Altova dans laquelle la connexion à la base de données est stockée
- Une connexion PostgreSQL native

Dans le volet de l'assistant de connexion (*voir la capture d'écran ci-dessous*), les bases de données peuvent être stockées de manière alphabétique par le nom du type de base de données ou par l'utilisation récente. Sélectionnez l'option que vous souhaitez dans la liste de choix *Trier par*. Après avoir sélectionné le type de base de données auquel vous voulez vous connecter, cliquez sur **Suivant**.

🤭 Se connecter à u	ne source de données			>
	Assistant connexion			
Assistant connexion	Veuillez choisir une base de données	source puis cliquer su	r Suivant.	
	Si le fournisseur de base de données listé ci-dessous, veuillez créer une c	avec lequel vous soul onnexion utilisant soit	haitez travailler n'es ADO soit directeme	st pas nt ODBC.
Connexions		Trier par:	nom	~
existantes				
	Firebird (ODBC/JDBC)			
"IIIUU	IBM DB2 (ODBC/JDBC)			
Connexions ADO	IBM Informix (ODBC/JDBC)			
	MariaDB (ODBC)			
E	Microsoft Access (ADO)			
ODBC	MICrosoft SQL Server (ADO)			
Connexions	MySQL (ODBC)			
ODBC	Oracle (ODBC/JDBC)			
5	PostgreSQL			
	Progress (ODBC)			_
Connexions	SQLITE			
IDRC	Sybase (ODBC)			
	Teradata (ODBC/JDBC)			
- CO				
ADO.NET				
Connexions				
ADO.NET				
60				
POSTGRE				
Connexions				
PostgreSQL				
- CSU				
Fichiers CSV/TSV				
Fichiels Cov/15v				
	Sauter l'étape de configuration p	our l'assistant		
			Suivant >	Fermer

L'assistant vous guidera à travers les prochaines étapes conformément au type de base de données, à la technologie de connexion (ADO, ADO.NET, ODBC, JDBC) et au pilote qui sera utilisé. Pour consulter des exemples applicables aux différents types de base de données, voir <u>Exemples de connexion à des bases de données</u> (191).

En alternative à l'utilisation de l'assistant, vous pouvez utiliser une des technologies d'accès à la base de données suivantes :

<u>Configurer une connexion ADO</u><sup>165</sup>

- <u>Configurer une connexion ADO.NET</u><sup>171</sup>
- Configurer une connexion ODBC<sup>178</sup>
- <u>Configurer une connexion JDBC<sup>182</sup></u>

# 4.2.1.2 Aperçu des pilotes de base de données

Le tableau suivant regroupe les pilotes de base de données les plus communs que vous pouvez utiliser pour vous connecter à une base de données particulière par le biais d'une technologie d'accès aux données particulières. Veuillez noter que cette liste n'a pas la prétention d'être exhaustive ou prescriptive ; vous pouvez utiliser d'autres solutions natives ou tierces outre les pilotes indiqués ci-dessous.

Bien qu'un certain nombre de pilotes de base de données sont peut-être déjà disponibles sur votre système d'exploitation Windows, il se peut que vous nécessitiez le téléchargement d'un pilote alternatif. En ce qui concerne certaines bases de données, le dernier pilote fourni par le distributeur de bases de données est susceptible de fonctionner mieux que le pilote fourni avec le système d'exploitation.

Les distributeurs de bases de données peuvent fournir des pilotes soit en tant que pack séparément téléchargeables soit fournis avec le logiciel de client de la base de données. Dans ce cas, le logiciel de client de base de données inclut normalement tous les pilotes de base de données requis ou vous proposent une option au cours de l'installation pour sélectionner les pilotes et les composants que vous souhaitez installer. En règle générale, le logiciel de client de base de données consiste en des services d'administration et de configuration utilisés pour simplifier l'administration de base de données et la connectivité, et comporte une documentation expliquant comment installer et configurer le client de base de données et tous ses composants.

La configuration correcte du client de base de données est une étape cruciale pour la mise en place d'une connexion réussie à la base de données. Avant d'installer et d'utiliser votre logiciel de client de base de données, nous vous recommandons vivement de lire attentivement les instructions d'installation et de configuration du client de base de données, ceux-ci peuvent varier pour chaque version de base de données et pour chaque version Windows.

Afin de comprendre les possibilités et les limites de chaque technologie d'accès de données, pour chaque type de base de données, veuillez vous référer à la documentation de ce produit de base de données spécifique et testez également la connexion en comparaison avec votre environnement spécifique. Pour éviter les problèmes de connectivité les plus courants, veuillez consulter les notes et recommandations générales suivantes :

- Certains fournisseurs ADO.NET ne sont pas pris en charge ou ont une prise en charge limitée. Voir Notes de prise en charge ADO.NET<sup>(73)</sup>.
- Lors de l'installation d'un pilote de base de données, nous vous recommandons d'en utiliser un qui ait la même plate-forme que l'application Altova (32-bit ou 64-bit). Par exemple, si vous utilisez une application Altova 32-bit sur un système d'exploitation 64-bit, installez le pilote 32-bit et configurez votre connexion de base de données à l'aide du pilote 32-bit, voir aussi <u>Consulter les pilotes ODBC</u> <u>disponibles</u><sup>181</sup>.
- Lors de la configuration d'une source de données ODBC, il est recommandé de créer le non de source de données (DSN) en tant que System DSN au lieu de l'User DSN. Pour plus d'informations, voir <u>Configurer une connexion ODBC</u><sup>[178]</sup>.
- Si la base de données cible est MySQL ou MariaDB through ODBC, l'option Retourner des lignes correspondantes au lieu des lignes affectées doit être activée dans l'onglet Cursor/Results de MySQL ODBC Connector. En alternative, si vous saisissez le string de connexion manuellement par le

biais de l'assistant Connexion de base de données, ajouter Option=2 au string de connexion, par exemple : Dsn=mydsn;Option=2;

- Lors de la configuration d'une source de données JDBC, veuillez vous assurer que le JRE (Java Runtime Environment) ou le Java Development Kit (JDK) soit installé et que la variable d'environnement CLASSPATH du système d'exploitation soit configurée. Pour plus d'informations, voir <u>Configurer une</u> <u>connexion JDBC Connection</u><sup>162</sup>.
- En ce qui concerne les instructions et les détails de prise en charge de tout pilote ou logiciel de client de base de données que vous installez à partir d'un fournisseur de base de données, contrôlez la documentation fournie avec le paquet d'installation.

Base de données	Interface	Drivers
Firebird	ADO.NET	Fournisseur de données Firebird ADO.NET ( <u>https://www.firebirdsql.org/en/additional-downloads/</u> )
	JDBC	Pilote Firebird JDBC ( <u>https://www.firebirdsql.org/en/jdbc-driver/</u> )
	ODBC	Pilote Firebird ODBC ( <u>https://www.firebirdsql.org/en/odbc-driver/</u> )
IBM DB2	ADO	Fournisseur IBM OLE DB pour DB2
	ADO.NET	Fournisseur IBM Data Server pour .NET
	JDBC	Pilote IBM Data Server pour JDBC et SQLJ
	ODBC	Pilote IBM DB2 ODBC
IBM DB2 for i	ADO	<ul> <li>Fournisseur IBM DB2 pour i5/OS IBMDA400 OLE DB</li> <li>Fournisseur IBM DB2 pour i5/OS IBMDARLA OLE DB</li> <li>Fournisseur IBM DB2 pour i5/OS IBMDASQL OLE DB</li> </ul>
	ADO.NET	Fournisseur .NET Framework Data pour IBM i
	JDBC	Pilote IBM Toolbox pour Java JDBC
	ODBC	Pilote iSeries Access ODBC
IBM Informix	ADO	Fournisseur IBM Informix OLE DB
	JDBC	Pilote IBM Informix JDBC
	ODBC	Pilote IBM Informix ODBC
Microsoft Access	ADO	<ul> <li>Fournisseur Microsoft Jet OLE DB</li> <li>Fournisseur Microsoft Access Database Engine OLE DB</li> </ul>
	ADO.NET	Fournisseur de données .NET Framework pour OLE DB
	ODBC	Pilote Microsoft Access
MariaDB	ADO.NET	En défaut d'un connecteur .NET dédié pour MariaDB, utiliser <b>Connector/NET</b> pour MySQL ( <u>https://dev.mysql.com/downloads/connector/net/</u> ).

Base de données	Interface	Drivers
	JDBC	Connecteur MariaDB /J (https://downloads.mariadb.org/)
	ODBC	Connecteur MariaDB /ODBC (https://downloads.mariadb.org/)
Microsoft SQL Server	ADO	<ul> <li><u>Microsoft OLE DB Driver for SQL Server</u> (MSOLEDBSQL)</li> <li>Fournisseur Microsoft OLE DB pour SQL Server (SQLOLEDB)</li> <li>SQL Server Native Client (SQLNCLI)</li> </ul>
	ADO.NET	<ul> <li>.NET Framework Fournisseur de données pour SQL Server</li> <li>.NET Framework Fournisseur de données pour OLE DB</li> </ul>
	JDBC	<ul> <li>Pilote Microsoft JDBC pour SQL Server (<u>https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server</u>)</li> </ul>
	ODBC	<ul> <li>Pilote ODBC pour Microsoft SQL Server         <ul> <li>(<u>https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server</u>)</li> </ul> </li> </ul>
MySQL	ADO.NET	Connecteur/NET ( <u>https://dev.mysql.com/downloads/connector/net/</u> )
	JDBC	Connecteur/J ( <u>https://dev.mysql.com/downloads/connector/j</u>
	ODBC	Connecteur/ODBC ( <u>https://dev.mysql.com/downloads/connector/odbc</u> )
Oracle	ADO	<ul><li>Fournisseur Oracle pour OLE DB</li><li>Fournisseur Microsoft OLE DB pour Oracle</li></ul>
	ADO.NET	Fournisseur de données Oracle pour .NET ( <u>http://www.oracle.com/technetwork/topics/dotnet/index-085163.html</u> )
	JDBC	<ul> <li>JDBC Thin Driver</li> <li>Pilote JDBC Oracle Call Interface (OCI)</li> <li>Ces pilotes sont généralement installés pendant l'installation de votre client de base de données Oracle. Connectez-vous par le biais du pilote OCI Driver (par le Thin Driver) si vous utilisez le composant Oracle XML DB.</li> </ul>
	ODBC	<ul> <li>Microsoft ODBC pour Oracle</li> <li>Le pilote Oracle ODBC (généralement installé pendant l'installation de votre client de base de données Oracle)</li> </ul>
PostgreSQL	JDBC	Pilote PostgreSQL JDBC ( <u>https://jdbc.postgresql.org/download.html</u> )
	ODBC	psqlODBC ( <u>https://odbc.postgresql.org/</u> )
	Native Connection	Disponible. Il n'est pas nécessaire d'installer des pilotes si vous utilisez une connexion native.

Base de données	Interface	Drivers
Progress	JDBC	Connecteur JDBC ( <u>https://www.progress.com/jdbc/openedge</u> )
OpenEdge	ODBC	Connecteur ODBC ( <u>https://www.progress.com/odbc/openedge</u> )
SQLite	Native Connection	Disponible. Il n'est pas nécessaire d'installer des pilotes si vous utilisez une connexion native.
Sybase	ADO	Fournisseur Sybase ASE OLE DB
	JDBC	jConnect™ pour JDBC
	ODBC	Pilote Sybase ASE ODBC
Teradata	ADO.NET	Fournisseur de données .NET pour Teradata ( <u>https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata</u> )
	JDBC	Pilote JDBC Teradata (https://downloads.teradata.com/download/connectivity/jdbc-driver)
	ODBC	Pilote ODBC Teradata pour Windows ( <u>https://downloads.teradata.com/download/connectivity/odbc-</u> <u>driver/windows</u> )

# 4.2.1.3 Configurer une connexion ADO

Microsoft ActiveX Data Objects (ADO) est une technologie d'accès aux données qui vous permet de vous connecter à une variété de sources de données par le biais de OLE DB. OLE DB est une interface alternative à ODBC ou JDBC; elle fournit un accès uniforme aux données dans un environnement COM (Component Object Model). ADO est un précurseur du plus nouvel <u>ADO.NET</u><sup>(71)</sup> et l'une des manières possibles de se connecter aux bases de données natives de Microsoft comme Microsoft Access ou SQL Server, bien que vous puissiez aussi l'utiliser pour d'autres sources de données.

Ce qui est important, c'est que vous pouvez choisir entre de multiples fournisseurs ADO, et certains doivent être téléchargés et installés sur votre poste de travail avant de pouvoir les utiliser. Par exemple, pour se connecter au serveur SQL, les fournisseurs ADO suivants sont disponibles :

- Microsoft OLE DB *Driver* pour SQL Server (MSOLEDBSQL)
- Microsoft OLE DB *Provider* pour SQL Server (SQLOLEDB)
- SQL Server Native Client (SQLNCLI)

De la liste des fournisseurs ci-dessus, nous recommandons le MSOLEDBSQL; vous pouvez le télécharger depuis <u>https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15</u>. Veuillez noter qu'il doit correspondre avec la plate-forme de MapForce (32-bit ou 64-bit). Les fournisseurs SQLOLEDB et SQLNCLI sont considérés être dépréciés et ne sont donc pas recommandés.

Microsoft OLE DB Provider for SQL Server (SQLOLEDB) est connu pour avoir des soucis avec la

liaison de paramètre de requêtes complexes comme Common Table Expressions (CTE) et des déclarations imbriquées SELECT.

#### Pour configurer une connexion ADO :

- 1. Lancer l'assistant de connexion de base de données<sup>(60)</sup>.
- 2. Cliquez sur ADO Connections.

fois la
-

3. Cliquez sur Générer.

Provider	Connection	Advanced	Al			
Select t	he data you w	ant to conne	ct to:			
OLE	DB Provider(	s)				^
Micr	osoft Jet 4.0 C	DLE DB Provi	der			
Micr	osoft Office 12	2.0 Access D	atabase	Engine OLE	E DB Pro	
Micr	osoft Office 15	5.0 Access D	atabase	Engine OLE	E DB Pro	
Micr	osoft OLE DB	Provider for	Analysis	Services 11	.0	
Micr	osoft OLE DB	Provider for	ODBC D	rivers		
Micr	osoft OLE DB	Provider for (	Oracle			
Micr	osoft OLE DB	Provider for	Search			
Micr	osoft OLE DB	Provider for	SQL Sen	ver		
Micr	osoft OLE DB	Simple Provi	der			
MSE	)ataShape					
OLE	DB Provider f	for Microsoft	Directory	Services		
SQL	Server Native	e Client 10.0		_		
VSE	E Versioning E	Enlistment Ma	anager P	roxy Data S	ource	~
×						
						_
				<u>N</u>	<u>l</u> ext >>	

4. Sélectionnez le fournisseur de données par le biais duquel vous souhaitez vous connecter. La table cidessous montre quelques scénarios communs.

Pour se connecter à cette base de données	Utiliser ce fournisseur
Microsoft Access	<ul> <li>Fournisseur Microsoft Office Access Database Engine OLE DB (recommandé)</li> <li>Microsoft Jet OLE DB Provider</li> <li>Si le Microsoft Office Access Database Engine OLE DB Provider n'est pas disponible dans la liste, assurez-vous d'avoir installé soit Microsoft Access ou Microsoft Access Database Engine Redistributable (<u>https://www.microsoft.com/en-</u> us/download/details.aspx?id=54920) sur votre ordinateur.</li> </ul>
Serveur SQL	<ul> <li>Microsoft OLE DB Driver pour SQL Server (MSOLEDBSQL) - c'est le fournisseur OLE DB recommandé. Si vous voulez que ce fournisseur apparaisse dans la liste, vous devez le télécharger depuis <u>https://docs.microsoft.com/en-</u> us/sql/connect/oledb/download-oledb-driver-for-sql- server?view=sql-server-ver15 et l'installer.</li> <li>Microsoft OLE DB Provider pour SQL Server (OLEDBSQL)</li> </ul>

I

Pour se connecter à cette base de données	Utiliser ce fournisseur
	SQL Server Native Client (SQLNCLI)
Autre base de données	Sélectionnez le fournisseur applicable à votre base de données.
	Si un fournisseur OLE DB de votre base de données n'est pas disponible, veuillez installer le pilote requis depuis le distributeur de base de données (voir <u>Aperçu des pilotes de base de</u> <u>données</u> (102) ). En alternative, vous pouvez établir une connexion ADO.NET, ODBC ou JDBC.
	Si le système d'exploitation comporte un pilote ODBC pour la base de données requise, vous pouvez également utiliser le <b>Fournisseur Microsoft OLE DB pour les Pilotes ODBC</b> ou préférablement opter pour une <u>connexion ODBC</u> <sup>178</sup> .

5. Après avoir sélectionné le fournisseur de votre choix, cliquez sur Suivant et achever l'assistant.

Les étapes suivantes de l'assistant sont spécifiques au fournisseur que vous avez choisi. En ce qui concerne SQL Server, vous devrez fournir ou sélectionner le nom d'hôte du serveur de base de données, la méthode d'authentification, le nom de la base de données ainsi que le nom d'utilisateur et le mot de passe de la base de données. Pour un exemple, consultez <u>Se connecter à Microsoft SQL Server (ADO)</u><sup>215</sup>. En ce qui concerne Microsoft Access, vous serez invité à parcourir ou fournir le chemin du fichier de base de données. Pour un exemple, consultez <u>Se connecter à Microsoft Access (ADO)</u><sup>215</sup>.

La liste complète des propriétés d'initialisation (paramètres de connexion) est disponible dans l'onglet **AII** de la boîte de dialogue de connexion - ces propriétés varient dépendant du fournisseur choisi et doivent éventuellement être explicitement définies de telle manière que la connexion soit possible. Les prochaines sections fournissent des orientations sur la configuration des propriétés d'initialisation de base pour les bases de données Microsoft Access et SQL Server :

- <u>Configurer les propriétés de liaison de données SQL Server</u>
- <u>Configurer les propriétés de liaison de données Microsoft Access</u>

## 4.2.1.3.1 Se connecter à une base de données Microsoft Access existante

Cette approche est convenable si vous souhaitez vous connecter à une base de données Microsoft Access qui n'est pas protégée par un mot de passe. Si la base de données est protégée par un mot de passe, configurez le mot de passe de la base de données tel qu'indiqué dans <u>Se connecter à Microsoft Access (ADO)</u><sup>212</sup>.

#### Pour se connecter à une base de données Microsoft Access existante :

- 1. Exécuter l'assistant de connexion de base de données (voir <u>Lancer l'Assistant de la connexion de la</u> <u>base de données</u><sup>(60)</sup>).
- 2. Choisir Microsoft Access (ADO), puis cliquez sur Suivant.
- 3. Cherchez le fichier de base de données ou saisissez le chemin y menant (soit relatif, soit absolu).
- 4. Cliquez sur **Se connecter**.

# 4.2.1.3.2 Configurer les propriétés de liaison de données SQL Server

Lorsque vous vous connectez à une base de données Microsoft SQL Server par le biais d'<u>ADO</u><sup>(66)</sup>, veuillez vous assurer que les liaisons de données suivantes sont configurées dans l'onglet **Toutes** de la fenêtre de dialogue Propriétés des liaisons de données.

Nom	Valeur	1
Application Name	Taua	
Connect Timeout	nue	
Current Language		
Data Source	MY-WORKSTATION	
DataTypeCompatibility	0	
Extended Properties		
Failover Partner		
Failover Partner SPN		
General Timeout	0	
Initial Catalog		
Initial File Name		
Integrated Security		<u> </u>

Fenêtre de dialogue Propriétés des liaisons de données

Propriété	Notes
Sécurité intégrée	Si vous avez sélectionné le fournisseur de données <b>SQL Server</b> <b>Native Client</b> dans l'onglet <b>Fournisseur</b> , configurez cette propriété sur un caractère d'espace.
Persister les infos de sécurité	Configurer cette propriété sur <b>True</b> .

# 4.2.1.3.3 Configurer les propriétés de liaison de données Microsoft Access

Lorsque vous vous connectez à une base de données Microsoft <u>ADO</u><sup>(165)</sup>, veuillez vous assurer que les propriétés suivantes sont configurées correctement dans l'onglet **Toutes** de la fenêtre de dialogue Propriétés des liaisons de données.

umisseur	Connexion	Propriét	és avancées	Toutes	
/oici les val lonnées. Po liquez sur l	eurs des prop our modifier ur e bouton Mod	riétés d'ir ne valeur lifier la va	nitialisation pou , sélectionnez aleur.	ir ce type ( une propri	de iété, puis
Nom			Valeur		-
Data S	ource				
Extend	ed Properties				
Jet OL	EDB:Bypass (	Choic	False		-
Jet OL	EDB:Bypass (	Jserln	False		=
Jet OL	EDB:Compact	t With	False		
Jet OL	EDB:Create S	ystem	False		
Jet OL	EDB:Databas	e Loc	1		
Jet OL	EDB:Databas	e Pas			
Jet OL	EDB:Don't Co	py Lo	False		
Jet OL	EDB:Encrypt	Datab	False		
Jet OL	EDB:Engine 1	Гуре	0		
Jet OL	EDB:Global B	ulk Tr	1		
Jet OL	EDB:Global P	artial	2		-
Modifier	a valeur				
		OK	Accude		Aido

Fenêtre de dialogue Propriétés des liaisons de données

Propriété	Notes
Source de données	Cette propriété stocke le chemin vers le fichier de base de données Microsoft Access. Afin d'éviter des problèmes de connectivité, il est recommandé d'utiliser le format de chemin UNC (Universal Naming Convention), par exemple : \\anyserver\share\$\filepath
Jet OLEDB : Base de données du système	Cette propriété stocke le chemin vers le fichier d'information du groupe de travail. Vous devrez éventuellement régler la valeur de cette propriété explicitement avant de pouvoir vous connecter à une base de données Microsoft Access.

Propriété	Notes
	Si vous ne pouvez pas vous connecter en raison d'une erreur de "fichier d'information de groupe de travail", localisez le fichier d'information de groupe de travail ( <b>System.MDW</b> ) concernant votre profil d'utilisateur et configurez la valeur de propriété sur le chemin du fichier <b>System.MDW</b> .
	Description de la propriété Jet OLEDB:System database
	Valeur de la propriété C:\Users\john.doe\AppData\Roaming\Microsoft \Access Rétablir la valeur OK Annuler
Jet OLEDB: Mot de passe de la Base de données	Si la base de données est protégée par un mot de passe, configurez la valeur de cette propriété sur le mot de passe de la base de données.
	Description de la propriété
	Jet OLEDB:Database Password Valeur de la propriété
	Rétablir la valeur OK Annuler

# 4.2.1.4 Configurer une connexion ADO.NET

ADO.NET est un ensemble de bibliothèques Microsoft .NET Framework conçues pour interagir avec des données, y compris des données provenant de bases de données. Afin de se connecter à une base de données depuis MapForce par le biais de ADO.NET, il est nécessaire d'avoir Microsoft .NET Framework 4 ou plus élevé. Comme affiché ci-dessous, vous vous connectez à une base de données par le biais d'ADO.NET en sélectionnant un fournisseur .NET et en fournissant un string de connexion.

Un fournisseur de données .NET est une collection de classes qui permet de se connecter à une source de données de type particulier (par exemple, un Serveur SQL Server, ou une base de données Oracle), d'exécuter des commandes, et d'en extraire des données. En d'autres termes, avec ADO.NET, une application comme MapForce interagit avec une base de données par le biais d'un fournisseur de données. Chaque fournisseur de données est optimisé pour travailler avec le type spécifique de sources de données pour lequel il a été conçu. Il existe deux types de fournisseurs .NET :

1. offert par défaut avec Microsoft .NET Framework.

 offert par des revendeurs des principales bases de données, en tant qu'une extension du .NET Framework. Ce genre de fournisseurs ADO.NET doit être installé séparément et peut être généralement téléchargé depuis le site Internet du revendeur de base de données respectif.

**Note :** certains fournisseurs ADO.NET ne sont pas pris en charge ou ne le sont que de manière limitée. Voir <u>Notes de prise en charge d'ADO.NET</u><sup>178</sup>.

#### Pour configurer une connexion ADO.NET :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur ADO.NET Connections.
- 3. Sélectionnez un fournisseur de données .NET depuis la liste.

La liste de fournisseurs disponibles par défaut avec .NET Framework apparaît dans la liste des « Fournisseurs ». Les fournisseurs de données .NET spécifiques au revendeur apparaîtront uniquement dans la liste s'ils sont déjà installés sur votre système. Pour les rendre disponibles, les fournisseurs spécifiques au revendeur doivent être installés dans le GAC (Global Assembly Cache), en exécutant le fichier .msi ou .exe fourni par le revendeur de base de données.

4. Saisir un string de connexion à une base de données. Un string de connexion définit les informations de connexion à la base de données sous la forme de paramètres de connexion de paires de clés/valeurs contenues entre des points-virgules. Par exemple, une string de connexion comme Data Source=DBSQLSERV; Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass permet de se connecter à la base de données du serveur SQL ProductsDB sur le serveur DBSQLSERV, avec le nom d'utilisateur dbuser et le mot de passe dbpass. Vous pouvez créer une string de connexion en saisissant les paires de clé/valeur directement dans la fenêtre de dialogue « Connection String ». Une autre option est de la créer avec Visual Studio (voir <u>Créer une string de connexion dans Visual Studio</u> 173).

La syntaxe de la string de connexion dépend du fournisseur sélectionné depuis la liste « Fournisseur ». Par exemple voir <u>Strings de connexion ADO.NET d'échantillon</u><sup>176</sup>.

Connexions	ADO.NET	
Sélectionner un fournis cliquer sur 'Se connec	sseur depuis la liste et saisir une chaîne de connexion valide, puis ter' pour continuer.	
Fournisseur:	.Net Framework Data Provider for SqlServer	•
Chaîne de connexion:	Data Source=DBSQL14;Initial Catalog=AdventureWorks2014;User ID=dbuser;Password=dbpass	
	-	
	<u>S</u> e connecter <u>F</u> ermer	

5. Cliquez sur Se connecter.

## 4.2.1.4.1 Créer une chaîne de connexion dans Visual Studio

Afin de vous connecter à une source de données en utilisant ADO.NET, il est nécessaire de disposer d'une chaîne de connexion de base de données valide. Les instructions suivantes vous montrent comment créer une chaîne de connexion depuis Visual Studio.

#### Pour créer une chaîne de connexion dans Visual Studio :

- 1. Dans le menu **Tools**, cliquer sur **Connect to Database**.
- 2. Choisir une source de données depuis la liste (dans cet exemple, Microsoft SQL Server). Le fournisseur de données est rempli automatiquement sur la base de votre choix.

Choose Data Source	? ×
Data <u>s</u> ource: Microsoft Access Database File Microsoft ODBC Data Source <u>Microsoft SQL Server</u> Microsoft SQL Server Database File Oracle Database <other> Data provider: .NET Framework Data Provider for SQL 5 ~</other>	Description Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.
Always <u>u</u> se this selection	Continue Cancel

3. Cliquer sur Continue.

Modify Connection	?	×
Enter information to connect to the selected data source or o choose a different data source and/or provider.	lick "Chang	e" to
Data <u>s</u> ource:		
Microsoft SQL Server (SqlClient)	<u>C</u> hange	
S <u>e</u> rver name:		
DBSQLSERV ~	<u>R</u> efrest	h
Log on to the server		
◯ Use <u>W</u> indows Authentication		
Use SQL Server Authentication		
User name: dbuser		
Password:		
Save my password		_
Connect to a database		
Select or enter a <u>d</u> atabase name:		_
ProductsDB		~
○ Attac <u>h</u> a database file:		_
	<u>B</u> rowse	
Logical name:		
	Ad <u>v</u> anced	I
Test Connection OK	Cance	I

- 4. Saisir le nom de l'hôte du serveur ainsi que le nom d'utilisateur et le mot de passe de la base de données. Dans cet exemple, nous nous connectons à la base de données ProductsDB sur le serveur DBSQLSERV, à l'aide de l'authentification de serveur SQL.
- 5. Cliquer sur **OK**.

Si la connexion à la base de données est établie avec succès, elle apparaît dans la fenêtre du Server Explorer. Vous pouvez afficher la fenêtre de Server Explorer à l'aide de la commande de menu **View | Server Explorer**. Pour obtenir la string de connexion à la base de données, cliquer avec la touche de droite sur la fenêtre Server Explorer et sélectionner **Properties**. La string de connexion est maintenant affichée dans la fenêtre Properties de Visual Studio. Veuillez noter que, avant de coller la string dans la fenêtre « Connection String » de MapForce, vous devrez remplacer les astérisques (\*) par le mot de passe.

# 4.2.1.4.2 Chaînes de connexion échantillons ADO.NET

Pour configurer une connexion ADO.NET, veuillez sélectionner un fournisseur ADO.NET depuis le dialogue de connexion à la base de données et saisir un string de connexion (voir aussi <u>Configurer une connexion</u> <u>ADO.NET</u><sup>177</sup>). Les strings de connexion échantillon ADO.NET pour les bases de données variées sont recensés ci-dessous, sous le fournisseur .NET pertinent.

## Fournisseur de données .NET pour Teradata

#### Ce fournisseur peut être téléchargé depuis le site web Teradata

(<u>https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata</u>). Un string de connexion d'échantillon ressemble à l'exemple suivant :

Data Source=ServerAddress;User Id=USer;Password=password;

# Fournisseur de données .NET Framework pour IBM i

Ce fournisseur est installé dans le *IBM i Access Client Solutions - Windows Application Package*. Un string de connexion d'échantillon ressemble à l'exemple suivant :

DataSource=ServerAddress;UserID=user;Password=password;DataCompression=True;

Pour plus d'informations, consulter le fichier d'aide ".Référence technique du fournisseur .NET" contenu dans le pack d'installation ci-dessus.

## Fournisseur de données .NET Framework pour MySQL

Ce fournisseur peut être téléchargé depuis le site Internet MySQL (<u>https://dev.mysql.com/downloads/connector/net/</u>). Un string de connexion d'échantillon ressemble à l'exemple suivant :

Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;

Voir aussi : <u>https://dev.mysql.com/doc/connector-net/en/connector-net-programming-connecting-connection-</u> <u>string.html</u>

## Fournisseur de données .NET Framework pour Serveur SQL

Un string de connexion d'échantillon ressemble à l'exemple suivant :

Data Source=DBSQLSERV; Initial Catalog=ProductsDB; User ID=dbuser; Password=dbpass

Voir aussi : https://msdn.microsoft.com/en-us/library/ms254500(v=vs.110).aspx

# Fournisseur de données 10.1.2 IBM DB2 pour .NET Framework 4.0

Database=PRODUCTS;UID=user;Password=password;Server=localhost:50000;

Note : Ce fournisseur est généralement installé avec le paquet IBM DB2 Data Server Client. Si le fournisseur est absent de la liste des fournisseurs ADO.NET après l'installation du paquet IBM DB2 Data Server Client, veuillez vous référer à la note technique suivante : <u>https://www-01.ibm.com/support/docview.wss?uid=swg21429586</u>.

Voir aussi :

•

https://www.ibm.com/support/knowledgecenter/en/SSEPGG\_10.1.0/com.ibm.swg.im.dbclient.adonet.ref.doc/d oc/DB2ConnectionClassConnectionStringProperty.html

## Fournisseur de données Oracle pour .NET (ODP.NET)

Le pack d'installation qui comprend le fournisseur ODP.NET peut être téléchargé depuis le site Internet Oracle (voir <u>http://www.oracle.com/technetwork/topics/dotnet/downloads/index.html</u>). Un string de connexion d'échantillon ressemble à l'exemple suivant :

Data Source=DSORCL;User Id=USer;Password=password;

DSORCL est le nom de la source de données qui pointe vers un nom de service Oracle défini dans le fichier **tnsnames.ora**, comme décrit dans la section <u>Connexion à Oracle (ODBC)</u><sup>[227]</sup>.

Pour vous connecter sans configurer un nom de service dans le fichier tnsnames.ora, utiliser un string comme

```
Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=port)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));User
Id=user;Password=password;
```

Voir aussi : <u>https://docs.oracle.com/cd/B28359\_01/win.111/b28375/featConnecting.htm</u>

# 4.2.1.4.3 Notes de prise en charge ADO.NET

La table suivante recense des pilotes de base de données ADO.NET connus qui ne sont pas pris en charge actuellement ou présentent une prise en charge limitée dans MapForce.

Base de données	Pilote	Notes de prise en charge
Toutes les bases de données	.Net Framework Data Provider for ODBC	Prise en charge limitée. Des problèmes connus existent avec des connexions Microsoft Access. Nous recommandons d'utiliser les connexions directes ODBC à la place.
	.Net Framework Data Provider for OleDb	Prise en charge limitée. Des problèmes connus existent avec des connexions Microsoft Access. Nous recommandons d'utiliser les connexions directes ADO à la place.
Firebird	Firebird ADO.NET Data Provider	Prise en charge limitée. Nous recommandons d'utiliser ODBC ou JDBC à la place.
Informix	IBM Informix Data Provider for .NET Framework 4.0	N'est pas pris en charge. Utiliser <b>DB2 Data</b> <b>Server Provider</b> à la place.
IBM DB2 for i (iSeries)	.Net Framework Data Provider for i5/OS	N'est pas pris en charge. Utiliser <b>.Net</b> <b>Framework Data Provider for IBM i</b> à la place, installé en tant que partie de <i>IBM i</i> <i>Access Client Solutions - Windows</i> <i>Application Package</i> .
Oracle	.Net Framework Data Provider for Oracle	Prise en charge limitée. Bien que ce pilote soit fourni avec le .NET Framework, Microsoft n'encourage pas son utilisation car il est obsolète.
PostgreSQL	-	Aucun pilote ADO.NET pour ce fournisseur n'est pris en charge. Utiliser une connexion native à la place.
Sybase	-	Aucun pilote ADO.NET pour ce fournisseur n'est pris en charge.

# 4.2.1.5 Configurer une connexion ODBC

ODBC (Open Database Connectivity) est une technologie d'accès de données largement utilisée qui vous permet de vous connecter à une base de données depuis MapForce. Elle peut être utilisée soit en tant que principal moyen pour se connecter à la base de données, ou comme alternative aux connexions pilotées native, OLE DB ou JDBC.

Pour se connecter à une base de données par le biais d'ODBC, vous devrez tout d'abord créer un nom de source de données (DSN) ODBC sur le système d'exploitation. Cette étape n'est pas nécessaire si le DSN a déjà été créé, par exemple par un autre utilisateur du système d'exploitation. Le DSN représente une manière uniforme de décrire la connexion de base de données à toute application client prise en charge ODBC du système d'exploitation, y compris MapForce. Les variables peuvent être de types suivants :

- DSN Système
- DSN Utilisateur
- DSN Fichier

Une source de données *System* est accessible par tous les utilisateurs avec des privilèges sur le système d'exploitation. Une source de données *Utilisateur* est disponible pour l'utilisateur l'ayant créée. Enfin, si vous créez un *DSN Fichier*, la source de données sera créée en tant que fichier avec une extension .dsn que vous pouvez partager avec d'autres utilisateurs, à condition qu'ils aient installé les pilotes utilisés par la source de données.

Tout DSN déjà disponible sur votre machine est regroupé par la boîte de dialogue de connexion de base de données lorsque vous cliquez sur **connexions ODBC** dans la boîte de dialogue de connexion ODBC.

ODSN fichier         Nom de la source de données       Pilote         Excel Files       Microsoft         MS Access Database       Microsoft         OfficeSales_DB2       Microsoft         Dracle_User_DSN       Oracle da         est       Microsoft         /isio Database Samples       Microsoft	Excel Driver (*.xls, *.xlsx, *.xlsm Access Driver (*.mdb, *.accdb) Access Driver (*.mdb)
Nom de la source de données         Pilote           Excel Files         Microsoft           MS Access Database         Microsoft           DifficeSales_DB2         Microsoft           Dracle_User_DSN         Oracle da           vest         Microsoft           Visio Database Samples         Microsoft	Excel Driver (*.xls, *.xlsx, *.xlsm Access Driver (*.mdb, *.accdb) Access Driver (*.mdb)
Excel Files         Microsoft           MS Access Database         Microsoft           DfficeSales_DB2         Microsoft           Dracle_User_DSN         Oracle da           est         Microsoft           /isio Database Samples         Microsoft	Excel Driver (*.xls, *.xlsx, *.xlsm Access Driver (*.mdb, *.accdb) Access Driver (*.mdb)
MS Access Database Microsoft OfficeSales_DB2 Microsoft Oracle_User_DSN Oracle da test Microsoft Visio Database Samples Microsoft	Access Driver (*.mdb, *.accdb) Access Driver (*.mdb)
OfficeSales_DB2 Microsoft Oracle_User_DSN Oracle da test Microsoft Visio Database Samples Microsoft	Access Driver (*.mdb)
Oracle_User_DSN Oracle da Microsoft Visio Database Samples Microsoft	
test Microsoft Visio Database Samples Microsoft	is OraClient11g_home1
Visio Database Samples Microsoft	Access Driver (*.mdb)
	Access Driver (*.mdb, *.accdb)
₽ \$ ⊘ \$	Access Driver (*.mdb) Access Driver (*.mdb, *.accd

Boîte de dialogue de connexions ODBC

Si un DSN vers la base de données requise n'est pas disponible, l'assistant de connexion de base de données de MapForce vous guidera pendant la création ; néanmoins, vous pouvez aussi le créer directement sur votre système d'exploitation Windows. Dans tous les cas, avant de commencer, assurez-vous que le pilote ODBC

applicable pour votre base de données se trouve dans la liste des pilotes ODBC disponibles pour le système d'exploitation (voir <u>Consulter les pilotes ODBC disponibles</u><sup>181</sup>).

#### Pour se connecter en utilisant un nouveau DSN :

- 1. Lancer l'assistant de connexion de base de données <sup>160</sup>.
- 2. Dans le dialogue de connexion à la base de données, cliquez sur **Connexions ODBC**.
- 3. Sélectionnez un type de source de données (DSN Utilisateur, DSN Système, DSN Fichier).

Pour créer un DSN Système, vous avez besoin de droits d'administration sur le système d'exploitation et MapForce doit être exécuté en tant qu'administrateur.

- 4. Cliquez sur Ajouter 🏂 .
- Sélectionnez un pilote, puis cliquez sur DSN Utilisateur ou DSN Système (dépendant du type de DSN que vous voulez créer). Si le pilote applicable pour votre base de données n'est pas dans la liste, téléchargez-le de la base de données et installez-le (voir <u>Apercu des pilotes de base de données</u><sup>(12)</sup>).
- 6. Dans la boîte de dialogue qui apparaît, remplissez toute information de connexion spécifique au pilote pour compléter la configuration.

Pour que la connexion soit établie avec succès, vous devrez fournir le nom d'hôte (ou l'adresse IP) du serveur de base de données, ainsi que le nom d'utilisateur et le mot de passe de la base de données. Il peut y avoir d'autres paramètres de connexion optionnels - ces paramètres varient entre les fournisseurs de bases de données. Pour lire des informations plus détaillées concernant les paramètres spécifiques à chaque méthode de connexion, veuillez consulter la documentation du fournisseur du pilote. Pour se connecter en utilisant un DSN existant : Ceci vous permet de réutiliser les détails liés à la connexion de base de données à chaque fois que vous voulez vous connecter à la base de données. Veuillez noter que les DSN Utilisateur sont ajoutés à la liste des DSN Système.

#### Pour se connecter en utilisant un DSN existant :

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquez sur **ODBC Connections**.
- Sélectionnez le type de la source de données existante (DSN Utilisateur, DSN Système, DSN Fichier).
- 4. Cliquez sur l'enregistrement DSN existant, puis cliquez sur Connexion.

#### Pour construire une chaîne de connexion basée sur un fichier .dsn existant :

- 1. Lancer l'assistant de connexion de base de données<sup>60</sup>.
- 2. Cliquez sur **ODBC Connections**.
- 3. Sélectionnez Générer un string de connexion, puis cliquez sur Générer.
- 4. Si vous souhaitez construire la chaîne de connexion en utilisant un DSN Fichier, cliquez sur l'onglet Source de données de fichier. Sinon, cliquez sur l'onglet Source de données de l'appareil. (Les DSN Système et DSN Utilisateur sont connus comme sources de données « Appareil ».)
- 5. Sélectionnez le fichier .dsn requis, puis cliquez sur OK.
#### Pour se connecter en utilisant une chaîne de connexion préparée :

- 1. Lancer l'assistant de connexion de base de données <sup>(60)</sup>.
- 2. Cliquez sur ODBC Connections.
- 3. Sélectionnez Créer une chaîne de connexion.
- 4. Collez la chaîne de connexion dans la fenêtre fournie, puis cliquez sur **Se connecter**.

### 4.2.1.5.1 Consulter les pilotes ODBC disponibles

Vous pouvez consulter les pilotes ODBC disponibles sur votre système d'exploitation dans l'Administrateur de source de données ODBC. Vous pouvez accéder à l'Administrateur de source de données ODBC (**Odbcad32.exe**) à partir du Panneau de configuration Windows, sous **Outils d'administration**. Dans les systèmes d'exploitation 64-bit, vous trouverez deux versions de ce programme d'exécution :

- La version 32-bit du fichier Odbcad32.exe est située dans le répertoire C:\Windows\SysWoW64 (en partant du principe que C: est votre disque système).
- La version 64-bit du fichier Odbcad32.exe est située dans le répertoire C:\Windows\System32.

Tout pilote de base de données 32-bit installé sera visible dans la version 32-bit de l'Administrateur de source de données ODBC, alors que les pilotes 64-bit seront visibles dans la version 64-bit. Veuillez donc vous assurer que vous contrôlez les pilotes de base de données depuis les versions pertinentes de l'Administrateur de source de données ODBC.

Pilotes ODBC	Traçage	Groupement de con	nexions	À propo	S
Pilotes ODBC installés	sur votre système	:			_
Nom		Version	Soc	iété	^
Microsoft Paradox D	river (*.db )	6.01.7601	.17632 Mic	rosoft Corpo	
Microsoft Paradox-Treiber (*.db )		6.01.7601	.17632 Mic	rosoft Corpo	
Microsoft Text Drive	r (*.bd; *.csv)	6.01.7601	.17632 Mic	rosoft Corpo	
Microsoft Text-Treiber (*.txt; *.csv)		6.01.7601	.17632 Mic	rosoft Corpo	
Microsoft Visual FoxPro Driver		1.00.02.0	0 Mic	rosoft Corpo	
Microsoft Visual FoxPro-Treiber		1.00.02.0	0 Mic	rosoft Corpo	Ξ
Oracle dans OraClient11g_home1		11.01.00.	07 Ora	cle Corpora	-
SQL Server		6.01.7601	.17514 Mic	rosoft Corpo	Ŧ
•				+	
Un pilote ODBC permet aux programmes compatibles ODBC d'obtenir des informations sur d'autres sources de données ODBC. Pour installer de nouveaux pilotes, utilisez le programme d'installation du pilote.					

Administrateur de source de données ODBC

Si le pilote de votre base de données cible n'existe pas dans la liste, ou si vous souhaitez ajouter un autre pilote, vous devrez le télécharger auprès du distributeur de base de données (voir <u>Aperçu des pilotes de base de données</u><sup>(102)</sup>). Une fois que le pilote ODBC est disponible sur votre système, vous êtes prêt à créer des connexions ODBC (voir <u>Configurer une connexion ODBC</u><sup>(173)</sup>).

# 4.2.1.6 Configurer une connexion JDBC

JDBC (Java Database Connectivity) est une interface d'accès à des bases de données, elle fait partie de la plateforme logicielle Java d'Oracle. Les connexions JDBC nécessitent généralement plus de ressources que les connexions ODBC, mais elles peuvent fournir des fonctions qui ne sont pas disponibles par le biais d'ODBC.

### Prérequis

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(100)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) corresponde à celle du JRE/JDK.

- Les pilotes JDBC du distributeur de base de données doivent être installés. Il se peut que ce soient des pilotes JDBC installés en tant que partie de l'installation client de la base de données, ou de bibliothèques JDBC (.jar files) téléchargées séparément, si disponible ou pris en charge par la base de données, voir aussi <u>Exemples de connexion à la base de données</u><sup>[99]</sup>.
- La variable d'environnement CLASSPATH doit inclure le chemin du pilote JDBC (un ou plusieurs fichiers .jar) sur votre système d'exploitation Windows. Lorsque vous installez des clients de base de données, l'installateur peut configurer cette variable automatiquement. Voir aussi <u>Configuration de</u> <u>CLASSPATH</u><sup>185</sup>.

## Se connecter à SQL Server par le biais de JDBC avec des identifiants Windows

Si vous vous connectez à SQL Server par le biais de JDBC avec des identifiants Windows (sécurité intégrée), veuillez noter :

- Le fichier sqljdbc\_auth.dll contenu dans le package du pilote JDBC doit être copié dans un répertoire qui se trouve dans la variable d'environnement du système PATH. Il y a deux fichiers de ce type, un pour la plateforme x86 et un pour la plateforme x64. Veuillez vous assurer que vous ajoutez à PATH celui qui correspond à votre plateforme JDK. De même, veuillez vous assurer que vous redémarrez MapForce (ou le programme qui exécute le mappage, le cas échéant) après avoir changé la variable d'environnement.
- Le string de connexion JDBC doit inclure la propriété integratedSecurity=true. Vous pouvez ajouter cette propriété depuis plusieurs endroits :
  - o depuis l'assistant à la connexion de base de données, voir ci-dessous
  - o depuis les <u>paramètres de composant</u><sup>281</sup> de base de données
  - le cas échéant, en éditant le string de connexion à la base de données dans du <u>code Java</u> <u>généré</u><sup>108</sup>.

Pour plus d'informations, veuillez consulter la documentation *Microsoft JDBC driver for SQL Server*, <u>https://docs.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url</u>.

### Configurer une connexion JDBC

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur **Connexions JDBC**.
- 3. En option, saisir une liste de chemins d'accès de fichiers .jar séparées par des points-virgules dans la zone de texte « Chemins de classe ». Les bibliothèques .jar qui sont saisies ici seront chargées dans l'environnement en plus de celles déjà définies dans la variable d'environnement CLASSPATH. Une fois que vous avez terminé d'éditer la fenêtre de texte "Chemins de classe", tout pilote JDBC trouvé dans les librairies de source .jar sont automatiquement ajoutées à la liste « Pilote » (voir l'étape suivante).

Classpaths:	C:\jdbc\instantclient_12_1\ojdbc7.jar
Driver:	oracle.jdbc.driver.OracleDriver 🗸
Username:	johndoe
Password:	•••••
Database URL:	jdbc:oracle:thin:@//ora12c:1521:orcl12c
	<u>C</u> onnect Close

4. À côté de « Pilote », choisir un pilote JDBC depuis la liste des pilotes ou saisir un nom de classe Java. Veuillez noter que cette liste contient tous les pilotes JDBC configurés par la variable d'environnement CLASSPATH (voir <u>Configuration de CLASSPATH</u><sup>105</sup>) ainsi que celles trouvées dans la zone de texte « Chemins de classe ».

Les chemins de pilote JDBC définis dans la variable CLASSPATH, ainsi que dans tout chemin de fichier .jar saisis directement dans la boîte de dialogue de connexion de base de données sont tous fournis sur la Java Virtual Machine (JVM). La JVM décide ensuite quels pilotes utiliser pour établir une connexion. Il est recommandé de garder une trace des classes Java chargées dans la JVM pour ne pas créer de conflits de pilotes JDBC potentiels et éviter des résultats inattendus lors d'une connexion à la base de données.

- 5. Saisir le nom d'utilisateur et le mot de passe de la base de données dans les fenêtres correspondantes.
- 6. Dans la zone de texte URL de base de données, saisir le string URL de connexion JDBC dans le format spécifique à votre type de base de données. La table suivante décrit la syntaxe des URL de connexion JDBC (chaînes) pour les types de base de données communs.

Base de données	URL de connexion JDBC	
Firebird	jdbc:firebirdsql:// <host>[:<port>]/<database or<br="" path="">alias&gt;</database></port></host>	
IBM DB2	jdbc:db2://hostName:port/databaseName	

Base de données	URL de connexion JDBC
IBM DB2 for i	jdbc:as400://[host]
IBM Informix	jdbc:informix- sqli:// <b>hostName:port</b> / <b>databaseName</b> :INFORMIXSERVER= <b>myserver</b>
MariaDB	jdbc:mariadb://hostName:port/databaseName
Microsoft SQL Server	jdbc:sqlserver://hostName:port;databaseName=name
MySQL	jdbc:mysql://hostName:port/databaseName
Oracle	jdbc:oracle:thin:@ <b>hostName:port:SID</b> jdbc:oracle:oci:@// <b>hostName:port:service</b>
Oracle XML DB	jdbc:oracle:oci:@//hostName:port:service
PostgreSQL	jdbc:postgresql://hostName:port/databaseName
Progress OpenEdge	jdbc:datadirect:openedge:// <b>host:port</b> ;databaseName= <b>db_name</b>
Sybase	jdbc:sybase:Tds: <b>hostName:port/databaseName</b>
Teradata	jdbc:teradata:// <b>databaseServerName</b>

- Note : des variations de syntaxe pour les formats cités ci-dessus sont également possibles (par exemple, l'URL de base de données peut exclure le port ou peut inclure le nom d'utilisateur et le mot de passe de la base de données). Vérifiez la documentation du distributeur de la base de données pour plus de détails.
  - 7. Cliquez sur **Se connecter**.

# 4.2.1.6.1 Configuration de CLASSPATH

La variable d'environnement CLASSPATH est utilisée par Java Runtime Environment (JRE) ou le Kit de développement Java (JDK) pour situer les classes Java et d'autres fichiers de ressource sur votre système d'exploitation. Lorsque vous vous connectez à une base de données par le biais de JDBC, cette variable doit être configurée pour contenir le chemin vers le pilote JDBC sur votre système d'exploitation, et, dans certains cas, le chemin vers des fichiers de librairie supplémentaires spécifiques au type de base de données que vous utilisez.

La table suivante recense les chemins de fichier échantillon qui sont généralement inclus dans la variable CLASSPATH. Attention ! Vous devrez éventuellement ajuster cette information sur la base de l'emplacement du pilote JDBC sur votre système, du nom de pilote JDBC, et de la version JRE/JDK présente sur votre système d'exploitation. Pour éviter tout problème de connectivité, contrôlez les instructions d'installation et les étapes de configuration de pré-installation ou de post-installation applicables au pilote JDBC installé sur votre système d'exploitation.

Base de données	Entrées CLASSPATH échantillons	
Firebird	C:\Program Files\Firebird\Jaybird-2.2.8-JDK_1.8\jaybird-full- 2.2.8.jar	
IBM DB2	C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar;C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar;	
IBM DB2 for i	C:\jt400\jt400.jar;	
IBM Informix	C:\Informix_JDBC_Driver\lib\ifxjdbc.jar;	
Microsoft SQL Server	C:\Program Files\Microsoft JDBC Driver 4.0 for SQL Server\sqljdbc_4.0\enu\sqljdbc.jar	
MariaDB	<installation directory="">\mariadb-java-client-2.2.0.jar</installation>	
MySQL	mysql-connector-java-Version-bin.jar;	
Oracle	ORACLE_HOME \jdbc\lib\ojdbc6.jar;	
Oracle (avec XML DB)	ORACLE_HOME\jdbc\lib\ojdbc6.jar;ORACLE_HOME\LIB\xmlparserv2.jar; ORACLE_HOME\RDBMS\jlib\xdb.jar;	
PostgreSQL	<installation directory="">\postgresql.jar</installation>	
Progress OpenEdge	<pre>%DLC%\java\openedge.jar;%DLC%\java\pool.jar;</pre>	
	Note : Si le Progress OpenEdge SDK est installé sur la machine, %DLC% est le répertoire où OpenEdge est installé.	
Sybase	C:\sybase\jConnect-7_0\classes\jconn4.jar	
Teradata	<installation directory="">\tdgssconfig.jar;<installation directory&gt;\terajdbc4.jar</installation </installation>	

- Un changement de la variable CLASSPATH peut influer sur le comportement des applications Java sur votre machine. Afin de comprendre les implications possibles avant de commencer, veuillez vous référer à la documentation Java.
- Les variables d'environnement peuvent être utilisateur ou système. Pour changer les variables d'environnement système, vous aurez besoin de droits administratifs sur le système d'exploitation.
- Après avoir changé la variable d'environnement, redémarrez les programmes en cours pour que la configuration puisse prendre effet. En alternative, déconnectez-vous ou redémarrez votre système d'exploitation.

### Pour configurer CLASSPATH sur Windows 7 :

- 1. Ouvrir le menu **Démarrer** et cliquer avec le bouton de droite sur **Ordinateur**.
- 2. Cliquer sur **Propriétés**.
- 3. Cliquer sur Configurations de système avancées.
- 4. Dans l'onglet Avancé, cliquer sur Variables d'environnement,

- 5. Localiser la variable CLASSPATH sous les variables d'environnement utilisateur ou système, puis cliquer sur **Éditer**. Si la variable CLASSPATH n'existe pas, cliquer sur **Nouveau** pour la créer.
- 6. Éditer la valeur de variable pour inclure le chemin sur votre système d'exploitation où se trouve le pilote JDBC. Pour séparer le chemin de pilote JDBC des autres chemins qui pourraient déjà se trouver dans la variable CLASSPATH, utiliser un point-virgule en tant que séparateur (; ).

#### Pour configurer CLASSPATH sur Windows 10 :

- 1. Cliquer sur la touche Windows, puis saisir "variables d'environment".
- 2. Cliquer sur Éditer les variables d'environnement de système.
- 3. Cliquer sur Variables d'environnement.
- 4. Localiser la variable CLASSPATH sous les variables d'environnement utilisateur ou système, puis cliquer sur **Éditer**. Si la variable CLASSPATH n'existe pas, cliquer sur **Nouveau** pour la créer.
- 5. Éditer la valeur de variable pour inclure le chemin sur votre système d'exploitation où se trouve le pilote JDBC. Pour séparer le chemin de pilote JDBC des autres chemins qui pourraient déjà se trouver dans la variable CLASSPATH, utiliser un point-virgule en tant que séparateur (; ).

# 4.2.1.7 Configurer une connexion PostgreSQL

Les connexions aux bases de données PostgreSQL peuvent être configurées soit en tant que connexions natives, soit en tant que connexions par le biais de ODBC, JDBC et d'autres pilotes. L'avantage d'une configuration native est qu'il n'est pas nécessaire d'installer d'autres pilotes sur votre système. Aussi, si vous souhaitez déployer des fichiers pour exécution sur un serveur Linux ou macOS, il n'est pas requis d'installer les pilotes égfalement sur le serveur cible.

Si vous préférez établir une connexion par le biais d'un pilote non natif, voir les sections suivantes :

- <u>Configurer une connexion JDBC<sup>182</sup>
  </u>
- <u>Connexion à PostgreSQL (ODBC)</u><sup>234</sup>

Sinon, si vous souhaitez configurer une connexion native à PostgreSQL, veuillez suivre les étapes suivantes. Avant de pouvoir continuer, il vous faudra les éléments suivants : nom d'hôte, port, nom de base de données, nom d'utilisateur et mot de passe.

#### Pour configurer une connexion PostgreSQL native :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur Connexions PostgreSQL.
- 3. Saisir l'hôte (*localhost*, si PostgreSQL est exécuté sur la même machine), le port (en général 5432, cette étape est optionnelle), le nom de la base de données, le nom d'utilisateur et le mot de passe dans les fenêtres correspondantes.

Se connecter à PostgreSQL		
Veuillez saisir les pa 'Se connecter' pour	aramètres requis pour la base de données Postgre. Puis diquer sur poursuivre.	
Hôte :	DBSERV	
Port:	5432 [en option]	
Base de données :	200	
Nom d'utilisateur :	dbuser	
Mot de passe :	•••••	
	< Retour Se connecter Fermer	

4. Cliquez sur Se connecter.

Si le serveur de base de données PostgreSQL se trouve sur une machine différente, veuillez noter les points suivants :

- Le serveur de base de données PostgreSQL doit être configuré de manière à accepter les connexions provenant des clients. En particulier, le fichier pg\_hba.conf doit être configuré de manière à permettre des connexions non locales. Ensuite, le fichier postgresql.conf doit être configuré de manière à écouter des adresses IP et des ports spécifiés. Pour plus d'information, consultez la documentation PostgreSQL (https://www.postgresgl.org/docs/9.5/static/client-authentication-problems.html).
- Le serveur doit être configuré de manière à accepter les connexions sur le port désigné (en général 5432) par le biais du pare-feu. Par exemple, sur un serveur de base de données exécuté sur Windows, une règle peut être créée pour permettre des connexions sur le port 5432 par le biais du pare-feu, depuis Panneau de configuration > Pare-feu Windows > Fonctions avancées > Règles entrantes.

# 4.2.1.8 Configurer une connexion SQLite

<u>SQLite</u> est un type autonome de base de données sur base de fichier, idéal pour des scénarios mettant l'accent sur la portabilité et la simplicité de configuration. Puisque les bases de données SQLite sont prises en charge nativement par MapForce, vous n'aurez pas besoin d'installer des pilotes pour les connecter.

## Notes de prise en charge de la base de données SQLite

- Sur Linux, le timeout pour l'exécution d'instruction n'est pas pris en charge pour les bases de données SQLite.
- La recherche plein texte n'est pas prise en charge pour les tables.
- SQLite permet des valeurs de types de données différents dans chaque ligne d'une table donnée. Toutes les valeurs traitées doivent être compatibles avec le type de colonne déclaré ; c'est pourquoi les valeurs non attendues peuvent être extraites et les erreurs d'exécution peuvent se produire si votre base de données SQLite a des valeurs de ligne qui ne sont pas les mêmes que le type de colonne déclaré.
- Si votre mappage doit écrire des donnés sur une base de données SQLite et si vous ne possédez pas déjà le fichier de base de données cible, vous devrez le créer séparément. Dans ce cas, vous soit le créer avec un outil comme <u>DatabaseSpy</u> soit télécharger le shell de ligne de commande SQLite depuis le site Internet officiel, puis créer le fichier de base de données depuis la ligne de commande (voir aussi <u>Exemple : Données de mappage de XML à SQLite</u><sup>(33)</sup>). Pour une référence complète à la syntaxe de commande SQLite, veuillez vous référer à la documentation SQLite officielle.
- Les bases de données SQLite sont prises en charge dans le langage de transformation BUILT-IN MapForce (soit lorsque vous prévisualisez le mappage, soit lorsque vous exécutez un fichier d'exécution MapForce Server).
- Les bases de données SQLite ne sont pas prises en charge dans les fonctions définies par les utilisateurs (UDF).

### Important

Il est recommandé de créer des tables avec le mot-clé STRICT afin de garantir un comportement plus prévisible de vos données. Autrement, les données ne sont pas écrites ou lues correctement lorsque les valeurs de différents types sont mélangées dans une colonne. Pour en savoir plus sur les tables STRICT, voir la <u>documentation SQLite</u>.

## 4.2.1.8.1 Connexion à une base de données SQLite existante

### Pour se connecter à une base de données SQLite existante :

- 1. Exécuter l'assistant de connexion de base de données (voir <u>Lancer l'Assistant de la connexion de la</u> base de données <sup>(60)</sup>).
- 2. Choisir SQLite, puis cliquer sur Suivant.
- Chercher le fichier de base de données SQLite, ou saisir le chemin (soit relatif ou absolu) à la base de données. La touche Se connecter s'active une fois que vous avez saisi le chemin vers un fichier de base de données SQLite.
- 4. Cliquer sur **Se connecter**.

# 4.2.1.9 Ressources globales

Après avoir créé une base de données en tant que ressource globale, ses détails de connexion sont stockés et peuvent être utilisés dans tous les produits Altova installés sur votre appareil.

### Créer une base de données en tant que ressource globale

Pour créer une base de données en tant que ressource globale, suivez les étapes suivantes

- 1. Dans le menu Outils de MapForce, cliquez sur les Ressources globales.
- 2. Cliquez sur Ajouter, puis cliquez sur Base de données.
- 3. Saisissez un nom pour la ressource globale dans le champ Resource Alias.
- 4. Cliquer sur **Choose Database**. L'<u>assistant de connexion</u><sup>160</sup> apparait.
- 5. Utiliser l'assistant de connexion pour ajouter une connexion de base de données, tel que décrit cidessus.

### Utiliser une base de données ressource globale

Pour utiliser une base de données qui a été créée en tant que ressource globale (*voir-ci-dessus*), suivez les étapes suivantes :

- 1. Lancer l'assistant de connexion de base de données, tel que décrit ci-dessus.
- Sélectionnez les Ressources globales. Toutes les bases de données qui ont été créées en tant que ressources globales seront recensées par leurs noms dans le volet des Ressources globales (voir la capture d'écran ci-dessous).

Ressources globales		
Nom de ressource globale		
Alias2		
AdventureWorks2014		
Alias4		
	Se connecter	Fermer

3. Sélectionnez la ressource globale de votre choix. Astuce : déplacez le curseur sur une ressource globale dans la liste afin de voir l'information sur les bases de données.

# 4.2.1.10 Exemples de connexion à la base de données

Cette section comprend des exemples de connexion à une base de données provenant de MapForce par le biais de ADO, ODBC ou JDBC. Les exemples de connexion ADO.NET sont recensés séparément, voir <u>Chaînes de connexion échantillons ADO.NET</u><sup>179</sup>. Pour consulter des instructions concernant la mise en place d'une connexion native sur PostgreSQL et SQLite, voir <u>Configurer une connexion PostgreSQL</u><sup>187</sup> et <u>Configurer une connexion SQLite</u><sup>189</sup>, respectivement.

Noter :

 Les instructions peuvent différer si votre configuration Windows, l'environnement de réseau et le client de base de données ou le logiciel de serveur ne sont pas les mêmes que ceux décrits dans chaque exemple. • En ce qui concerne la plupart des types de base de données, il est possible de se connecter en utilisant plus d'une technologie d'accès aux données (ADO, ADO.NET, ODBC, JDBC) ou pilote. La performance de la connexion de la base de données, ainsi que ses fonctions et ses limitations dépendront du pilote sélectionné, du logiciel client de base de données (le cas échéant), et de tout autre paramètre de connectivité que vous êtes susceptible d'avoir configuré outre MapForce.

## 4.2.1.10.1 Firebird (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données Firebird par le biais de JDBC.

#### Conditions préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(103)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) corresponde à celle du JRE/JDK.
- Le pilote Firebird JDBC doit être disponible sur votre système d'exploitation (il prend la forme d'un fichier .jar qui offre une connectivité à la base de données). Le pilote peut être téléchargé depuis le site web Firebird (<u>https://www.firebirdsql.org/</u>). Cet exemple utilise *Jaybird 2.2.8*.
- Vous disposez des détails de connexion pour la base de données suivante : l'hôte, le chemin de base de données ou alias, le nom d'utilisateur et le mot de passe.

#### Pour se connecter à Firebird par le biais de JDBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquer sur JDBC Connections.
- 3. À côté de « Chemins de classe », saisir le chemin menant au fichier .jar qui fournit la connexion à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par des points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : C:\jdbc\firebird\jaybird-full-2.2.8.jar. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le chemin d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi <u>Configuration du CLASSPATH</u><sup>185</sup>).
- 4. Dans le champ « Pilote », choisir org.firebirdsql.jdbc.FBDriver. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Chemins de classe :	C:\jdbc\firebird\jaybird-full-2.2.8.jar
Pilote :	org.firebirdsql.jdbc.FBDriver ~
Nom d'utilisateur :	prod_admin
Mot de passe :	•••••
URL base de DB :	jdbc:firebirdsql://firebirdserv/COMPANY
	×
	Se connecter Fermer

- 5. Saisir le nom d'utilisateur et le mot de passe de la base de données dans les champs correspondants.
- 6. Saisir la chaîne de connexion du serveur de base de données dans le champ de saisie URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

jdbc:firebirdsql://<host>[:<port>]/<database path or alias>

7. Cliquer sur Se connecter.

### 4.2.1.10.2 Firebird (ODBC)

Cet exemple illustre comment se connecter à une base de données Firebird 2.5.4 fonctionnant sur un serveur

Linux.

#### Conditions préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(100)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Le pilote Firebird ODBC doit être installé sur votre système d'exploitation. Cet exemple utilise le pilote Firebird ODBC version 2.0.3.154 téléchargé depuis le site web Firebird (<u>https://www.firebirdsql.org/</u>).
- Le client Firebird doit être installé sur votre système d'exploitation. Veuillez noter qu'aucun installateur autonome n'est disponible pour le client Firebird 2.5.4 ; le client fait partie du paquet d'installation du serveur Firebird. Vous pouvez télécharger le paquet d'installation du serveur Firebird depuis le site web Firebird (<u>https://www.firebirdsql.org/</u>), chercher « Windows executable installer for full Superclassic/Classic or Superserver ». Pour n'installer que les fichiers client, choisir « Minimum client install no server, no tools » lorsque vous suivrez les instructions de l'assistant.

#### Important :

- La plateforme du pilote Firebird ODBC et du client (32-bit ou 64-bit) doit correspondre à celle de MapForce.
- La version du client Firebird doit correspondre à la version du serveur Firebird auquel vous vous connectez.
- Vous disposez des détails de connexion à la base de données suivantes : le nom d'hôte du serveur ou l'adresse IP, le chemin de base de données (ou alias) sur le serveur, le nom d'utilisateur et le mot de passe.

#### Pour se connecter à Firebird via ODBC :

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquer sur **Connexions ODBC**.
- Choisir DSN utilisateur (ou DSN système, si vous disposez de privilèges d'administrateur), puis cliquez sur Ajouter .

Cn	éer un DSN ODBC
	Choisir un pilote puis cliquer soit sur Utilisateur soit sur Système pour déterminer le genre de DSN que vous souhaitez créer.
	IBM DB2 ODBC DRIVER
	DSN utilisateur DSN système Annuler

4. Choisir le pilote Firebird, puis cliquer sur **DSN utilisateur** (ou **DSN système**, selon ce que vous aurez sélectionné dans l'étape précédente). Si le pilote Firebird n'est pas disponible dans la liste, veuillez

vous assurer qu'il est installé dans votre système d'exploitation (voir aussi <u>Consulter les pilotes ODBC</u> <u>disponibles</u> ).

Data Source Name (DSN)		Driver
firebird dsn		IscDbc 🗸
Description		
Connects to the Firebird databas	se on a remote Linux sei	rver.
Database		
firebirdserv:products		Browse
Client		
C:\Program Files\Firebird\Firebir	rd_2_5\bin\fbclient.dll	Browse
Database Account	Password	Role
PROD_ADMIN	•••••	
Character Set	Services	Test connection
Options Transaction read (default write) nowait (default wait) Lock Timeout Dialect Safe thread	Extende V quot sens auto Set nul	ed identifier properties ted identifiers sitive identifier oquoted identifier II field SCHEMA

5. Saisir les détails de connexion de base de données comme suit :

Nom de source de données (DSN)	Saisir un nom descriptif pour la source de données que vous êtes en train de créer.
Base de données	Saisir le nom d'hôte du serveur ou l'adresse IP, suivi d'un double point, suivi d'un alias de base de données (ou chemin). Dans cet exemple, le nom d'hôte est firebirdserv, et l'alias de base de données est products, comme suit :
	firebirdserv:products
	L'utilisation d'une base de données assume que, du côté serveur, l'administrateur de la base de données a configuré les <i>produits</i> d'alias pour pointer vers le fichier de base de données Firebird (.fdb) sur le serveur (voir la documentation Firebird pour plus de détails).
	Vous pouvez également utiliser l'adresse IP du serveur à la place du nom d'hôte et un chemin à la place d'un alias ; donc n'importe laquelle des chaînes de connexion d'échantillon est valide :

	firebirdserver:/var/Firebird/databases/butterflies.fdb 127.0.0.1:D:\Misc\Lenders.fdb	
	Si la base de données se trouve sur la machine Windows locale, cliquer sur <b>Parcourir</b> et sélectionner directement le fichier de base de données Firebird (.fdb).	
Client	Saisir le chemin vers le fichier <b>fbclient.dll</b> . Par défaut, il s'agit du sous-répertoire bin du répertoire d'installation Firebird.	
Compte de base de données	Saisir le nom d'utilisateur de base de données fourni par l'administrateur de la base de données (dans cet exemple, PROD_ADMIN).	
Mot de passe	Saisir le mot de passe de la base de données fourni par l'administrateur de la base de données.	

#### 6. Cliquer sur OK.

### 4.2.1.10.3 IBM DB2 (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données IBM DB2 par le biais de JDBC.

### Conditions préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(105)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) corresponde à celle du JRE/JDK. Cet exemple utilise l'OpenJDK 11.0 64-bit d'Oracle, et, par conséquent, la version 64-bit de MapForce.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui fournissent une connectivité dans la base de données) doit être disponible sur votre système d'exploitation. Cet exemple utilise le pilote JDBC disponible après avoir installé la version 10.1 (64-bit) **IBM Data Server Client**. Pour installer les pilotes JDBC, choisir une installation **Typical**, ou sélectionner cette option explicitement sur l'assistant d'installation.

提 DB2 Setup - IBM Data Server Client - DB2COPY1	×
Select the features to install	
To customize which features are installed, dick on icons in the follow	ing list.
D D IN Data Carrier Chart	Feature Description
IBM Data Server Clent       IBM Data Server Clent        IBM Data Server Clent        Image: Spatial Extender Clent	JDBC Support allows Java samples, not containing embedded SQL (SQLJ), to be built and run using the JDBC driver.
Testallation folder	
Installation folder	
Directory C:\Program Files\IBM\SQLLIB\	<u>C</u> hange
Space required: 707 MB	Disk Space
InstallShield	
< <u>B</u> ack	Next > Cancel Help

Si vous n'avez pas changé le chemin d'installation par défaut, les fichiers .jar requis se trouvent dans le répertoire **C:\Program Files\IBM\SQLLIB\java** après l'installation.

• Vous nécessitez les détails de connexion de base de données suivants : hôte, port, nom d'utilisateur et mot de passe.

#### Pour se connecter à IBM DB2 par le biais de JDBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquer sur JDBC Connections.
- 3. À côté de « Classpaths », saisir le chemin menant au fichier .jar qui fournit la connexion à la base de données. Cet exemple se réfère à C:\Program Files\IBM\SQLLIB\java\db2jcc.jar. Vous devrez éventuellement vous référer au pilote db2jcc4.jar, selon la version du serveur de base de données. Pour une compatibilité de pilote, se référer à la documentation IBM (<u>http://www-01.ibm.com/support/docview.wss?uid=swg21363866</u>). Veuillez noter que vous pouvez laisser le champ de texte "Classpaths" vide si vous avez ajouté les chemins de fichier .jar dans la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configurer le CLASSPATH<sup>(III)</sup>).
- 4. Dans la case "Driver", choisir **com.ibm.db2.jcc.DB2Driver**. Cette entrée devient disponible uniquement si un chemin de fichier .jar valide a été trouvé soit dans le champ de texte "Classpaths", soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Classpaths:	C:\Program Files\IBM\SQLLIB\java\db2jcc.jar
Driver:	com.ibm.db2.jcc.DB2Driver
Username:	usemame
Password:	•••••
Database URL:	jdbc:db2://dbserver:50000/dbname

- 5. Saisir le nom d'utilisateur et le mot de passe de l'utilisateur de la base de données dans le champ de texte correspondant.
- Saisir le string de connexion JDBC dans le champ de texte de l'URL de base de données. Veuillez vous assurer de remplacer les détails de connexion avec ceux applicable dans votre serveur de base de données.

jdbc:db2://hostName:port/databaseName

7. Cliquer sur **Se connecter**.

### 4.2.1.10.4 IBM DB2 (ODBC)

Cet exemple illustre comment se connecter à une base de données IBM DB2 via ODBC.

#### Conditions préalables :

- L'IBM Data Server Client doit être installé et configuré sur votre système d'exploitation (cet exemple utilise IBM Data Server Client 9.7). Pour travailler avec des instructions d'installation, veuillez consulter la documentation fournie avec votre logiciel IBM DB2. Après avoir installé IBM Data Server Client, vérifiez si les pilotes ODBC sont disponibles sur votre machine (voir <u>Consulter les pilotes ODBC</u> <u>disponibles</u><sup>(181</sup>).
- Créer un alias de base de données. Vous pouvez procéder de plusieurs manières :
  - Depuis l'Assistant de configuration IBM DB2
  - o Depuis le IBM DB2 Command Line Processor
  - Depuis l'assistant de source de données ODBC (dans ce cas, les instructions sont affichées cidessous)
- Vous disposez des détails de connexion de base de données suivants : hôte, base de données, port, nom d'utilisateur et mot de passe.

#### Pour se connecter à IBM DB2 :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup> et sélectionner **IBM DB2 (ODBC/JDBC)**.
- 2. Cliquer sur **Suivant**.

JDBC contre ODBC
JDBC Connexion basée sur Java qui permet de prendre en charge des fonctionnalités plus modernes de votre base de données non disponibles via ODBC. Le revers de ces fonctionnalités est une possible perte de performance.
ODBC Une connexion ODBC sera généralement plus rapide et nécessitera moins de ressources qu'une connexion JDBC, mais ne permet pas de prendre en charge des fonctionnalités de base de données plus modernes (comme les types natifs XML).
< Retour Suivant > Fermer

Choisir ODBC et cliquer sur Suivant. Si vous êtes invité à éditer la liste des pilotes connus pour la base de données, sélectionnez les pilotes de base de données applicables pour IBM DB2 (voir les <u>Conditions préalables</u><sup>188</sup>), et cliquer sur Suivant.

Sé	electionner les pilotes :	
Éd	iter la liste des pilotes connus pour la base de données.	
	Pilote	•
	SQL Server	
F	Microsoft ODBC for Oracle	
	Microsoft Access Driver (*.mdb)	-
	Microsoft Access-Treiber (*.mdb)	=
	Driver do Microsoft Access (*.mdb)	
	Microsoft dBase Driver (*.dbf)	
	Microsoft dBase-Treiber (*.dbf)	
	Driver do Microsoft dBase (*.dbf)	
	Microsoft Excel Driver (*.xls)	
	Microsoft Excel-Treiber (*.xls)	
	Driver do Microsoft Excel(*.xls)	
	Microsoft Paradox Driver (*.db )	
	Microsoft Paradox-Treiber (*.db)	
	Driver do Microsoft Paradox (*.db )	Ŧ
	< Retour Fermer	

4. Choisir le pilote IBM DB2 depuis la liste et cliquer sur **Se connecter**. (Pour éditer la liste des pilotes disponibles, cliquer sur **Éditer pilotes**, puis cocher ou décocher les pilotes IBM DB2 que vous souhaitez ajouter ou supprimer, respectivement.)

Se connecter à IBM DB2
Où trouver les pilotes IBM DB2 ?
Choisir vos options de connexion la base de données puis cliquer Se connecter. O Créer un nouveau nom de source de données (DSN) avec le pilote :
IBM DB2 ODBC DRIVER
O Utiliser un nom de source de données existant :         Image:
Ignorer l'étape de configuration de l'assistant
< Retour Se connecter Fermer

5. Saisir un nom de source de données (dans cet exemple, **DB2DSN**), puis cliquer sur Ajouter.

Sélectionnez l'alias de la base de données DB2 à enregistrer comme source de données ODBC ou cliquez sur Ajouter pour en créer un. Vous pouvez modifier le nom et la description de la source de données ou accepter les valeurs par défaut.			
Nom de la source de données	DB2DSN		
Alias de la base de données	✓ Ajouter		
Description			
	OK Annulation		

6. Dans l'onglet **Source de données**, saisir le nom d'utilisateur et le mot de passe de la base de données.

Source de données TCP/IP Optie	ions de sécurité Paramètres avancés	
Nom de la source de données Description	DB2DSN	
ID utilisateur Mot de passe	iohn_doe •••••• Sauvegarder le mot de passe	
	OK Annuler Appliquer Aide	

7. Dans l'onglet **TCP/IP**, saisir le nom de base de données, un nom pour l'alias, le nom d'hôte et le numéro de port, puis cliquer sur OK.

Source de données TCP/IP Options de	sécurité Paramètres avancés
Nom de base de données	database 1
Alias de la base de données	alias1
Nom d'hôte	host1
Numéro de port	50000
🔲 La base de données réside physiqu	uement sur un hôte ou un système OS/400.
Connexion directe au serveur	
Connexion au serveur via la pa	sserelle
Paramètres DCS	
INTERRUPT_ENABLE	D
Optimisation pour l'application	▼
	OK Annuler Appliquer Aide

8. Saisir à nouveau le nom d'utilisateur et le mot de passe puis cliquer sur **OK**.

Alias de la base de données	DB2DSN -	Ajouter
ID utilisateur	john_doe	
Mot de passe	•••••	
Modification du mot de passe		
Nouveau mot de passe		
Confirmation du nouveau mot de		
Mode de connexion		
Partage	🔘 Exclusif	
		OK Annulation

### 4.2.1.10.5 IBM DB2 pour i (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données **IBM DB2 for i** par le biais de JDBC.

#### Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(105)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) corresponde à celle du JRE/JDK. Cet exemple utilise l'OpenJDK 11.0 64-bit d'Oracle, et, par conséquent, la version 64-bit de MapForce.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui offrent une connectivité à la base de données) doit être disponible sur votre système d'exploitation. Cet exemple utilise la source ouverte Toolbox for Java/JTOpen version 9.8 (<u>http://jt400.sourceforge.net/</u>). Après avoir installé le package et l'avoir décompressé vers un répertoire local, les fichiers .jar requis seront disponibles dans le sous-répertoire lib.
- Vous nécessitez les détails de connexion de base de données suivants : hôte, port, nom d'utilisateur et mot de passe.

#### Pour se connecter à IBM DB2 pour i par le biais de JDBC :

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquez sur JDBC Connections.
- À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : C: \jdbc\jtopen\_9\_8\jt400.jar. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH<sup>(BS)</sup>).
- 4. Dans la case « Driver », choisir com.ibm.as400.access.AS400JDBCDriver. Cette entrée devient disponible uniquement si un chemin de fichier .jar valide a été trouvé soit dans le champ de texte « Classpaths », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Classpaths:	C:\jdbc\jtopen_9_8\lib\jt400.jar
Driver:	com.ibm.as400.access.AS400JDBCDriver
Username:	usemame
Password:	•••••
Database URL:	jdbc:as400://192.0.2.0

- 5. Saisissez le nom d'utilisateur et le mot de passe de l'utilisateur de la base de données dans la zone de texte correspondante.
- 6. Saisissez le string de connexion JDBC dans la zone de texte **Base de données URL** Assurez-vous de remplacer l'hôte par le nom de l'hôte ou l'adresse IP de votre serveur de base de données.

jdbc:as400://**host** 

7. Cliquez sur **Connexion**.

## 4.2.1.10.6 IBM DB2 pour i (ODBC)

Cet exemple illustre comment se connecter à une base de données IBM DB2 pour i par le biais d'ODBC.

#### Exigences préalables :

 IBM System i Access pour Windows doit être installé sur votre système d'exploitation (cet exemple utilise IBM System i Access pour Windows V6R1M0). Pour les instructions d'installation, vérifiez la documentation fournie avec votre logiciel IBM DB2 pour i. Après l'installation, vérifier si le pilote ODBC est disponible sur votre appareil (voir <u>Consulter les pilotes ODBC disponibles</u><sup>181</sup>).

Name	Version	Company	1
Driver do Microsoft Paradox (*.db )	6.03.9600.16384	Microsoft Corporation	
BM DB2 ODBC DRIVER	9.07.301.326	International Business Machines Co	πp
BM DB2 ODBC DRIVER - DB2COPY1	9.07.301.326 International Business Machines Corp.		φ
Series Access ODBC Driver	12.00.00.00	IBM Corporation	
Microsoft Access Driver (*.mdb)	6.03.9600.16384	Microsoft Corporation	
Microsoft Access Driver (*.mdb, *.accdb)	15.00.4569.1501	Microsoft Corporation	
Microsoft Access Text Driver (*.bd, *.csv)	15.00.4569.1501	Microsoft Corporation	
Microsoft Access-Treiber (*.mdb)	6.03.9600.16384	Microsoft Corporation	
Microsoft dBase Driver (*.dbf)	6.03.9600.16384	Microsoft Corporation	`
C			>
An ODBC driver allows ODBC-enable new drivers, use the driver's setup pro	d programs to get inform ogram.	ation from ODBC data sources. To ins	stall

- Vous disposez des détails de connexion à la base de données suivants : l'adresse IP du serveur de la base de données, le nom d'utilisateur de la base de données et le mot de passe.
- Exécuter System i Navigator et suivez l'assistant pour créer une nouvelle connexion. Sur invitation à spécifier un système, saisissez l'adresse IP du serveur de la base de données. Après avoir établi la connexion, il est recommandé de la vérifier (cliquez sur la connexion, et choisissez Fichier >

**Diagnostics** > **Vérifier connexion**). Si vous obtenez des erreurs de connectivité, contacter l'administrateur de serveur de la base de données.

#### Pour se connecter à IBM DB2 for i:

- 1. Lancer l'assistant de connexion de base de données<sup>(10)</sup>.
- 2. Cliquez sur connexions ODBC.
- 3. Cliquez sur**DSN Utilisateur** (en alternative, cliquez sur **DSN Système** ou **DSN Fichier**, dans un tel cas, les instructions suivantes seront semblables).
- 4. Cliquez sur Ajouter 📌 .
- 5. Sélectionnez le **pilote iSeries Access ODBC** depuis la liste et cliquez sur **DSN Utilisateur** (ou **DSN Système**, le cas échéant).

Créer un DSN ODBC	<
Choisir un pilote puis cliquer soit sur Utilisateur soit sur Système pour déterminer le genre de DSN que vous souhaitez créer.	
iSeries Access ODBC Driver 🗸 🗸	
DSN utilisateur DSN système Annuler	

 Saisissez un nom de source de données et sélectionnez la connexion depuis la zone de liste du Système. Dans cet exemple, le nom de source de données est iSeriesDSN et le Système est 192.0.2.0.

eneral	Server	Data Types	Packages	Performance	Language	Catalog	Conversions	Diagnostic
	001101	Data Types	raditagee	1 circinia ico	Language	outdrog	Controlocito	Citignetic
Data si	ource na	me:						
iSeries	sDSN							
Descrip	ption:							
System	n i Acces	s for Windows	ODBC data	source				
System	1:							
192.0.	2.0		~	Connectio	n Options			
				OK	Can	cel	Apply	Help
				on	Juli		Check Control	map

- **Note :** en ajoutant une source de données ODBC pour une base de données *IBM DB2 pour i*, un indicateur par défaut est défini qui active des timeouts de requête. Ce paramètre doit être désactivé pour MapForce afin de charger correctement des fichiers de mappage. Pour désactiver le paramètre, sélectionnez l'onglet **Performance**, cliquez sur **Avancé**, et videz la case à cocher **Permettre un timeout de requête**.
  - 7. Cliquez sur les options de connexion, sélectionnez **Utiliser l'ID utilisateur spécifiée ci-dessous** et saisissez le nom de l'utilisateur de la base de données (dans cet exemple, **DBUSER**).

<ul> <li>Use Windows user name</li> </ul>	
Use the user ID specified below	
DBUSER	
None	
OUse System i Navigator default	
O Use Kerberos principal	
Signon dialog prompting	
Prompt for SQLConnect if needed	d
Never prompt for SQLConnect	
Security	
Do not use Secured Sockets La	ver (SSL)
O Use Secured Sockets Layer (SS	L)
Use same security as System i N	avigator connection
	-

- 8. Cliquez sur OK. La nouvelle source de données devient disponible dans la liste des DSN.
- 9. Cliquez sur **Connexion**.
- 10. Saisissez le nom utilisateur et le mot de passe de la base de données quand vous y êtes invité, puis cliquez sur **OK**.

## 4.2.1.10.7 IBM Informix (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données IBM Informix par le biais de JDBC.

#### Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(100)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) correspond à celle du JRE/JDK.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui offrent une connectivité à la base de données) doit être disponible sur votre système d'exploitation. Dans cet exemple, le pilote IBM Informix JDBC version 3.70 est utilisé. En ce qui concerne les instructions d'installation du pilote, voir la documentation fournie avec le pilote ou bien le manuel « IBM Informix JDBC Driver Programmer's Guide »).

• Vous disposez des détails de connexion à la base de données suivants : hôte, nom du serveur Informix, base de données, port, nom d'utilisateur et mot de passe.

#### Pour se connecter à IBM Informix par le biais de JDBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur JDBC Connections.
- 3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : C:\Informix\_JDBC\_Driver\lib\ifxjdbc.jar. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le chemin d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi <u>Configuration du CLASSPATH</u><sup>165</sup>).
- 4. Dans le champ « Pilote », sélectionnez **com.informix.jdbc.lfxDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Classpaths:	C:\jdbc\Informix_JDBC_Driver\lib\ifxjdbc.jar;
Driver:	com.informix.jdbc.lfxDriver
Username:	dbuser
Password:	•••••
Database URL:	jdbc:informix- sqli://host:port/MyDatabase:INFORMIXSERVER=MyServerN ame
	<u>C</u> onnect Close

- 5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
- Saisir la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

jdbc:informix-sqli://hostName:port/databaseName:INFORMIXSERVER=myserver;

7. Cliquez sur **Connexion**.

## 4.2.1.10.8 MariaDB (ODBC)

Cet exemple indique comment se connecter à un serveur de base de données MariaDB par le biais d'ODBC.

#### Exigences préalables :

- Le connecteur/ODBC MariaDB (<u>https://downloads.mariadb.org/connector-odbc/</u>) doit être installé.
- Vous disposez des détails de connexion à la base de données suivants : hôte, base de données, port, nom utilisateur et mot de passe.

#### Pour se connecter à MariaDB avec ODBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Choisissez MariaDB (ODBC) et cliquez sur Suivant.

Where can I find MariaDB drivers?
the database and dick Connect. with the driver:
Edit Drivers

- Choisissez Créer un nouveau nom de source de données (DSN) avec le pilote et choisissez le pilote MariaDB ODBC 3.0. Si ce pilote n'est pas disponible dans la liste, cliquez sur Éditer pilotes, et choisissez un des pilotes MariaDB disponibles (la liste contient tous les pilotes ODBC installés dans votre système d'opération).
- 4. Cliquez sur Se connecter.

-

Create a new Data So	rce to MariaDB ×				
	Welcome to the MariaDB ODBC Data Source Wizard!				
	This wizard will help you to create an ODBC data source that you can use to connect to a MariaDB server.				
	What name do you want to use to refer to your data source ?				
	Name: mariadb_connection				
	How do you want to describe the data source ?				
	Description:				
	< Previous Next > Cancel Help				

5. Saisir le nom et, éventuellement, une description qui vous aidera à identifier cette source de données ODBC à l'avenir.

Create a new Data So	Data Source to MariaDB				
	How do you want to connect to MariaDB				
	TCP/IP	Server Name:	demoserver		
	O Named Pipe	Port:	3306		
	Please specify a us	er name and password	d to connect to MariaDB		
	Decement.				
	Password:	•••••		Test DSN	
	Please specify a us	er name and passwor	d to connect to MariaDB		
	Database:	mydatabase	~		
	< Pr	revious Next >	Cancel	Help	

 Remplissez les identifiants de connexion dans la base de données (Serveur TCP/IP, Utilisateur, Mot de passe), choisissez une base de données et cliquez sur **Tester DSN**. Une fois la connexion établie, un message apparaît :



- Cliquez sur Suivant et terminez l'assistant. D'autres paramètres seront éventuellement nécessaires, selon les cas (par exemple, des certificats SSL si vous vous êtes connecté à MariaDB par le biais d'une connexion sécurisée).
- **Note :** si le serveur de base de données est à distance, il doit être configuré par l'administrateur du serveur pour accepter les connexions à distance depuis l'adresse IP de votre appareil.

### 4.2.1.10.9 Microsoft Access (ADO)

Une manière simple de se connecter à une base de données Microsoft Access est de suivre les instructions de l'assistant et de parcourir le fichier de base de données, tel que montré dans la section <u>Connexion à une base de données Microsoft Access existante</u><sup>168</sup>. Une approche alternative est de définir explicitement une connexion ADO, tel que montré dans cette section. Cette approche est utile si votre base de données est protégée par un mot de passe.

Il est également possible de se connecter à Microsoft Access par le biais d'une connexion ODBC, mais ce scénario a des limites, il vaut donc mieux l'éviter.

#### Pour se connecter à une base de données Microsoft Access protégée par un mot de passe :

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquez sur Connexions ADO.
- 3. Cliquer sur Générer.



4. Sélectionnez le fournisseur Microsoft Office 15.0 Access Database Engine OLE DB, puis cliquez sur Suivant.

Foumisseur	Connexion	Propriétés avancées	Toutes			
Informations	requises pour	la connexion à ces do	nnées :			
1. Entrez	la source de o	données et/ou l'emplac	ement des données :			
Source	e de donnée	s :				
Empla	acement :					
2. Entrez le	<ol> <li>Entrez les informations pour la connexion au serveur :</li> <li>Utiliser la sécurité intégrée de Windows NT</li> </ol>					
⊚ Ut sp	Utiliser un nom d'utilisateur et un mot de passe spécifiques :					
N	om d'utilisateu	ur : Admin				
M	lot de passe :					
Mot de passe vide mot de passe						
3. Entrez le catalogue initial à utiliser :						
· · · · · · · · · · · · · · · · · · ·						
Tester la connexion						
OK Annuler Aide						

- Dans la case de source de données, saisissez le chemin vers le fichier Microsoft Access dans le format UNC, par exemple \\myserver\\mynetworkshare\Reports\Revenue.accdb, où myserver est le nom du serveur et mynetworkshare est le nom du réseau partagé.
- 6. Dans l'onglet **Toutes**, double-cliquez la propriété **Jet OLEDB:Database Password** et saisissez le mot de passe de la base de données en tant que valeur de propriété.

Description de la propriété				
Jet OLEDB:Database Password				
Valeur de la propriété				
•••••				
Rétablir la valeur	OK Annuler			

Note : si vous ne parvenez toujours pas à vous connecter, localisez le fichier d'information du groupe de travail (System.MDW) applicable à votre profil d'utilisateur, et réglez la valeur de la propriété Jet OLEDB : System database sur le chemin d'accès du fichier System.MDW.

## 4.2.1.10.10 Microsoft SQL Server (ADO)

Cet exemple indique comment se connecter à une base de données SQL Server avec ADO. Ces instructions sont applicables quand vous utilisez **Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL)** recommandé, qui est disponible pour le téléchargement au lien <u>https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15</u>.

Avant de suivre ces instructions, assurez-vous avoir téléchargé et installé le fournisseur ci-dessus sur votre poste de travail. Le fournisseur ADO doit correspondre à la plate-forme de MapForce (32-bit ou 64-bit).

Si vous souhaitez utiliser d'autres fournisseurs ADO tels que **SQL Server Native Client (SQLNCLI)** ou **Microsoft OLE DB Provider for SQL Server (SQLOLEDB)**, les instructions sont semblables, mais ces fournisseurs sont dépréciés et donc pas recommandés. Aussi, pour que le connexion réussisse avec un fournisseur, vous allez devoir définir des propriétés de connexion additionnelles telles que décrites dans <u>Configurer les propriétés de SQL Server Data Link</u><sup>(16)</sup>.

**Microsoft OLE DB Provider for SQL Server (SQLOLEDB)** est connu pour avoir des soucis avec la liaison de paramètre de requêtes complexes comme Common Table Expressions (CTE) et des déclarations imbriquées SELECT.

#### Pour se connecter à SQL Server :

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- Choisissez Microsoft SQL Server (ADO), puis cliquez sur Suivant. La liste des fournisseurs ADO disponibles est affichée. Dans cet exemple, le Microsoft OLE DB Driver for SQL Server est utilisé. S'il ne figure pas dans la liste, assurez-vous qu'il est installé sur votre ordinateur, tel que mentionné cidessus.



3. Cliquez sur Next. Le dialogue Data Link Properties apparaît.

💮 Data Link Properties	$\times$
Provider Connection Advanced All	
1. Select or enter a server name:	
	<ul> <li>✓ <u>R</u>efresh</li> </ul>
2. Enter information to log on to the server:	
SQL Server Authentication	$\sim$
Ser⊻er SPN:	
User name:	
Password:	
☑ Blank password	Allow saving password
Use strong encryption for data	Trust server certificate
3. ( Select the <u>d</u> atabase:	~
Attach a database file as a database n	ame:
Using the filename:	
Change Password	Test Connection
ОК С	Cancel Help

- Choisissez ou saisissez le nom du serveur de base de données, par exemple SQLSERV01. Si vous vous connectez à une instance SQL Server nommée, le nom du serveur ressemblera à SQLSERV01\SOMEINSTANCE.
- 5. Si le serveur de la base de données a été configuré pour permettre des connexions depuis des utilisateurs authentifiés dans le domaine Windows, choisissez Windows Authentication. Sinon, sélectionnez SQL Server Authentication, décochez la case Blank password et saisissez les identifiants de base de données dans les champs pertinents.
- 6. Sélectionnez la case **Autoriser enregistrer mot de passe** et la base de données à laquelle vous vous connectez (dans cet exemple, « Nanonull »).
| Data Link Properties                          | $\times$ |
|---|----------|
| Provider Connection Advanced All              |          |
| 1. Select or enter a server name:             |          |
| ✓ <u>R</u> efresh                             |          |
| 2. Enter information to log on to the server: |          |
| SQL Server Authentication $\sim$              |          |
| Ser <u>v</u> er SPN:                          |          |
| User name:                                    | -        |
| Password:                                     | =        |
| ✓ Blank password Allow saving passw           | vord     |
| Use strong encryption for data                |          |
| 3.  Select the database:                      |          |
| O Attach a database file as a database name:  | _        |
|   |          |
| Using the filename:                           | _        |
|   |          |
| Change Password Test Connection               |          |
| OK Cancel Help                                | >        |

- 7. Pour tester la connexion à cet endroit, cliquez sur **Tester la connexion**. Il s'agit d'une étape optionnelle recommandée.
- 8. Cliquez sur OK.

# 4.2.1.10.11 Microsoft SQL Server (ODBC)

Cet exemple indique comment se connecter à une base de données SQL Server avec ODBC.

## Exigences préalables :

 Pour télécharger et installer Microsoft ODBC Driver for SQL Server depuis le site web Microsoft, voir https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server. Cet exemple utilise le Pilote Microsoft ODBC 17 pour SQL Server pour se connecter à une base de données SQL Server 2016. Vous pouvez tout à fait télécharger une version de pilote ODBC différente, selon la version du Serveur SQL auquel vous souhaitez vous connecter. Pour plus d'informations concernant les versions de pilote ODBC prises en charge par votre base de données Serveur SQL, veuillez consulter les exigences de système du pilote.

## Pour se connecter à SQL Server avec ODBC :

- 1. Lancer l'assistant de connexion de base de données <sup>160</sup>.
- 2. Cliquez sur ODBC Connections.
- 3. Sélectionnez DSN Utilisateur (ou DSN Système si vous avez des privilèges d'administrateur), puis cliquez sur Ajouter +
- 4. Sélectionnez le pilote depuis la liste. Veuillez noter que le pilote apparaît dans la liste uniquement après qu'il ait été installé.

Create an ODBC DSN	×
Select a Driver and click on either User or System to determine what kind of D you want to create.	SN
ODBC Driver 17 for SQL Server	~
User DSN System DSN Cance	el

5. Cliquez sur DSN Utilisateur (ou System DSN si vous créez un DNS Système).

La création d'un **DNS Système** exige que MapForce soit exécuté en tant qu'administrateur. C'est pourquoi, afin de créer un **DNS Système**, annulez l'assistant, assurez-vous d'exécuter MapForce en tant qu'administrateur et effectuez les étapes ci-dessus à nouveau.

6. Saisissez un nom et, en option, une description pour identifier cette connexion, puis sélectionner le serveur SQL depuis la liste sur lequel vous voulez vous connecter (**SQLSERV01** dans cet exemple).

Microsoft SQL Server DS	SN Configuration		<
SQL Server	This wizard will help y SQL Server. What name do you w	you create an ODBC data source that you can use to connect to want to use to refer to the data source?	
	Na <u>m</u> e:	odbc_for_sql_server	]
λ	How do you want to	describe the data source?	_
	Description:		
	Which SQL Server d	lo you want to connect to?	_
SC.	<u>S</u> erver:	SQLSERV01 ~	
			_
	[	Finish <u>N</u> ext > Cancel Help	

7. Si le serveur de la base de données a été configuré pour permettre des connexions depuis les utilisateurs authentifiés sur le domaine Windows, sélectionnez Avec l'authentification Windows intégrée. Sinon, sélectionnez une des autres options, le cas échéant. Cet exemple utilise Avec authentification SQL Server..., qui requiert que le nom utilisateur et le mot de passe soient saisis dans les cases pertinentes.

Create a New Data Source to SQL Server		Х
Ser.	How should SQL Server verify the authenticity of the login ID?	
SQL Server	O <u>With Integrated Windows authentication</u> . SPN (Optional):	
	◯ With Azure Active Directory Integrated authentication.	
	$\textcircled{\begin{tabular}{ll} \label{eq:constraint} \begin{tabular}{ll} With \underline{S} QL & Server authentication using a login ID and password entered by the user. \end{tabular}$	
	O With Azure <u>A</u> ctive Directory Password authentication using a login ID and password entered by the user.	
Protocol	O With Azure Active Directory Interactive authentication using a login ID entered by the user.	
	Login ID: dbuser	
	Password:	
	< <u>B</u> ack <u>N</u> ext > Cancel Help	

8. En option, cochez la case **Changer la base de données par défaut en** et saisissez le nom de la base de données à laquelle vous vous connectez (dans cet exemple, **Sandbox**).

Create a New Data Sour	ce to SQL Server	×
SQL Server	Change the default database to: Sandbox Mirror server:	~
	SPN for mirror server (Optional):	
	Use ANSI quoted identifiers. Use ANSI nulls, paddings and warnings. Application intent:	~
	Multi-subnet failover.	·
	Use FMTONLY metadata discovery.	
	< <u>B</u> ack <u>N</u> ext > Cancel Help	

9. Cliquez sur Suivant et, en option, configurez des paramètres supplémentaires pour cette connexion.

Create a New Data Source to SQL Server		$\times$
SQL Server	Change the language of SQL Server system messages to:     (Default)	~
	Use strong encryption for data. Trust server certificate.	
8	Perform translation for character data.	
	Use regional settings when outputting currency, numbers, dates and times.	
	Save long running queries to the log file:	
	C:\Users\altova\AppData\Local\Temp\QUERY.LO( Browse	
personal contractions	Long query time (milliseconds): 30000	
	Log ODBC driver statistics to the log file:	
	C:\Users\altova\AppData\Local\Temp\STATS.LOG Browse	
	Connect retry count: 1	
	Connect retry interval (seconds): 10	
	< <u>B</u> ack Finish Cancel Help	

10. Cliquez sur **Terminer**. Un dialogue de confirmation recensant les détails de connexion s'ouvre.

ODBC Microsoft SQL Server Setup	×
A new ODBC data source will be created with the following configuration	c
Microsoft ODBC Driver for SQL Server Version 17.05.0002 Data Source Name: ds Data Source Description: Server: Use Integrated Security: No Database: (Default) Language: (Default) Data Encryption: No Trust Server Certificate: No Multiple Active Result Sets(MARS): No Translate Character Data: Yes Log Long Running Queries: No Log Driver Statistics: No Use Regional Settings: No Use ANSI Quoted Identifiers: Yes Use ANSI Null, Paddings and Warnings: Yes	^
Test Data Source OK Cancel	

11. Cliquez sur **OK**. La source de données apparaît maintenant dans la liste de sources de données **Utilisateur** ou **Système**, tel que configuré, par exemple :

ODBC Connection	S
System DSN     User DSN     File DSN	O Build a connection string
Data Source Name	Driver

## 4.2.1.10.12 MySQL (ODBC)

Cet exemple illustre comment se connecter à un serveur de base de données MySQL depuis une machine Windows par le biais du pilote ODBC. Le pilote ODBC MySQL n'est pas disponible sur Windows, il doit donc être téléchargé et installé séparément. Cet exemple utilise MySQL Connecteur/ODBC 8.0.

## Exigences préalables :

- Le pilote ODBC MySQL doit être installé sur votre système d'exploitation. Vérifiez la documentation MySQL pour la version du pilote recommandée de votre version de serveur de base de données (voir <u>https://dev.mysql.com/downloads/connector/odbc/</u>).
- Vous disposez des détails de connexion à la base de données suivants : hôte, base de données, port, nom utilisateur et mot de passe.

Si vous avez installé MySQL Connector/ODBC pour la plateforme 64-bit, assurez-vous d'installer aussi MapForce pour la plateforme 64-bit.

#### Pour se connecter à MySQL via ODBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Sélectionnez MySQL (ODBC), puis cliquez sur Suivant.

Where can I find MySQL drivers?
with the driver:
5 H D L

 Sélectionnez Créer un nouveau Nom de source de données (DSN) avec le pilote, puis sélectionnez un pilote MySQL. Si aucun pilote MySQL n'est disponible dans la liste, cliquez sur Éditer les pilotes, puis sélectionnez tout pilote MySQL disponible (la liste contient tous les pilotes ODBC installés sur votre système d'exploitation).

Si vous avez installé MapForce 64-bit, alors les pilotes ODBC 64-bit sont affichés dans la liste. Sinon, les pilotes ODBC 32-bit sont affichés. Voir aussi <u>Consulter les pilotes ODBC</u> <u>disponibles</u><sup>(181)</sup>.

4. Cliquez sur **Connexion**.

MySQL Connector/ODBC I	Data Source Configuration X
Mysqu Connector/ODBC	
Connection Parameters	
Data Source Name:	MySQL Data Source
Description:	
TCP/IP Server:	server01 Port: 3306
O Named Pipe:	
User:	dbuser
Pass <u>w</u> ord:	•••••
Data <u>b</u> ase:	dbproducts $\checkmark$ <u>T</u> est
Details >>	OK <u>C</u> ancel <u>H</u> elp

- 5. Dans le champ du nom de la source de données, saisissez un nom descriptif qui vous permettra d'identifier cette source de données ODBC à l'avenir.
- 6. Remplir les identifiants de connexion de base de données (Serveur TCP/IP, Utilisateur, Mot de passe), sélectionnez une base de données et cliquez sur **OK**.
- **Note :** si le serveur de base de données est à distance, il doit être configuré par l'administrateur du serveur pour accepter les connexions à distance depuis l'adresse IP de votre appareil. De même, si vous cliquez sur **Détails>**, vous trouverez plusieurs paramètres supplémentaires disponibles pour la configuration. Vérifiez la documentation du pilote avant de changer leurs valeurs par défaut.

# 4.2.1.10.13 Oracle (JDBC)

Cet exemple montre comment vous connecter à un serveur de base de données Oracle à partir d'une machine client, à l'aide de l'interface JDBC. La connexion est créée en tant qu'une connexion Java pure, par le biais de **Oracle Instant Client Package (Basic)** disponible sur le site Internet Oracle. L'avantage de ce type de connexion est qu'elle ne nécessite que l'environnement Java et les bibliothèques .jar fournies par l'Oracle Instant Client Package, vous évitant d'installer et de configurer un client de base de données plus complexe.

## Exigences préalables :

 JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(050)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.

- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) correspond à celle du JRE/JDK.
- L'Oracle Instant Client Package (Basic) doit être disponible sur votre système d'exploitation. Le paquet peut être téléchargé depuis le site Internet officiel d'Oracle. Cet exemple utilise l'Oracle Instant Client Package version 12.1.0.2.0, pour Windows 32-bit et par conséquent, Oracle JDK 32-bit.
- Vous disposez des détails de connexion à la base de données suivants : hôte, port, nom de service, nom d'utilisateur et mot de passe.

## Pour se connecter à Oracle par le biais de Instant Client Package :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur JDBC Connections.
- 3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : C:\jdbc\instantclient\_12\_1\ojdbc7.jar. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH<sup>185</sup>).
- 4. Dans le champ « Pilote », sélectionnez soit oracle.jdbc.OracleDriver, soit oracle.jdbc.driver.OracleDriver. Veuillez noter que ces saisies sont disponibles si un chemin d'accès .jar valid est trouvé soit dans le champ de saisie « Classpath », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir étape précédente).
- 5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.

Classpaths:	C:\jdbc\instantclient_12_1\ojdbc7.jar
Driver:	oracle.jdbc.driver.OracleDriver
Username:	johndoe
Password:	•••••
Database URL:	jdbc:oracle:thin:@//ora12c:1521:orcl12c
	-
	<u>C</u> onnect Close

 Saisissez la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

jdbc:oracle:thin:@//host:port:service

7. Cliquez sur Connexion.

# 4.2.1.10.14 Oracle (ODBC)

Cet exemple illustre un scénario commun au cours duquel vous vous connectez depuis MapForce à un serveur de base de données Oracle sur une machine de réseau par le biais d'un client de base de données Oracle installé sur le système d'exploitation local.

L'exemple comprend des instructions pour configurer une source de données ODBC (DSN) en utilisant l'assistant de connexion de base de données dans MapForce. Si vous avez déjà créé un DSN, ou si vous préférez le créer directement depuis l'**administrateur de ODBC Data Source** dans Windows, vous pouvez le faire, puis le sélectionner si vous êtes invité à le faire par l'assistant. Pour plus d'informations concernant les sources de données ODBC, voir <u>Configurer une connexion ODBC</u><sup>(178)</sup>.

## Exigences préalables :

- Le client de base de données Oracle (qui inclut le pilote ODBC Oracle) doit être installé et configuré sur votre système d'exploitation. Pour obtenir des instructions d'installation et de configuration d'un client de base de données Oracle, veuillez vous référer à la documentation fournie avec votre logiciel Oracle.
- Le fichier **tnsnames.ora** situé dans le répertoire personnel Oracle contient une entrée qui décrit les paramètres de connexion de la base de données dans un format semblable au suivant :

```
ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = server01)(PORT = 1521))
    )
    (CONNECT_DATA =
        (SID = orcl)
        (SERVER = DEDICATED)
    )
   )
```

Le chemin d'accès menant au fichier **tnsnames.ora** dépend de l'emplacement où le répertoire personnel Oracle a été installé. Pour le client de base de données Oracle 11.2.0, le chemin d'accès du répertoire personnel Oracle par défaut pourrait être comme suit :

C:\app\username\product\11.2.0\client\_1\network\admin\tnsnames.ora

Vous pouvez ajouter de nouvelles entrées au fichier **tnsnames.ora** soit en collant les détails de connexion et en enregistrant le fichier, ou en exécutant l'assistant d'Oracle *Net Configuration Assistant* (si disponible). Si vous voulez que ces valeurs apparaissent dans des listes déroulantes durant le processus de configuration, alors vous allez éventuellement devoir ajouter le chemin au dossier admin en tant que variable d'environnement**TNS\_ADMIN**.

# Pour se connecter à Oracle avec ODBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Sélectionnez Oracle (ODBC / JDBC), puis cliquez sur Suivant.

JDBC contre ODBC
JDBC Connexion basée sur Java qui permet de prendre en charge des fonctionnalités plus modernes de votre base de données non disponibles via ODBC. Le revers de ces fonctionnalités est une possible perte de performance.
ODBC Une connexion ODBC sera généralement plus rapide et nécessitera moins de ressources qu'une connexion JDBC, mais ne permet pas de prendre en charge des fonctionnalités de base de données plus modernes (comme les types natifs XML).
< Retour Suivant > Fermer

3. Sélectionnez ODBC.

Se connecter à Oracle
Où puis-je trouver les pilotes Orade
Choisir vos options de connexion la base de données puis diquer Se connecter.
Microsoft ODBC for Oracle
<ul> <li>Utiliser un nom de source de données existant :</li> <li>DSN utilisateur</li> <li>DSN système</li> <li>Éditer pilotes</li> </ul>
Nom de la source de données
Ignorer l'étape de configuration de l'assistant
< Retour Ajouter Fermer

4. Cliquez sur Éditer les pilotes.

Édi	ter la liste des pilotes connus pour la base de données.	
	Pilote	
	Driver do Microsoft Excel(*.xls)	
	Microsoft Paradox Driver (*.db )	
	Microsoft Paradox-Treiber (*.db )	
	Driver do Microsoft Paradox (*.db )	
	Microsoft Text Driver (*.txt; *.csv)	
	Microsoft Text-Treiber (*.txt; *.csv)	
	Driver da Microsoft para arquivos texto (*.txt; *.csv)	
	Microsoft Visual FoxPro Driver	
	Microsoft FoxPro VFP Driver (*.dbf)	
	Microsoft dBase VFP Driver (*.dbf)	
	Microsoft Visual FoxPro-Treiber	
	Driver para o Microsoft Visual FoxPro	
	Microsoft Access Driver (*.mdb, *.accdb)	=
	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	
	Microsoft Access Text Driver (*.txt, *.csv)	
•	Oracle dans OraClient11g_home1	
	IBM DB2 ODBC DRIVER - DB2COPY1	
	IBM DB2 ODBC DRIVER	
	SQL Server Native Client 11.0	Ŧ

- Sélectionnez les pilotes Oracle que vous souhaitez utiliser (dans cet exemple, Oracle dans OraClient11g\_home1). La liste affiche les pilotes disponibles sur votre système après l'installation du client Oracle.
- 6. Cliquez sur **Retour**.
- 7. Sélectionner **Créer un nouveau nom de source de données (DSN) avec le pilote**, puis sélectionner le pilote Oracle choisi dans l'étape .4

Se connecter à Oracle
Où puis-je trouver les pilotes Oracle
Choisir vos options de connexion la base de données puis cliquer Se connecter. Oréer un nouveau nom de source de données (DSN) avec le pilote :
Oracle dans OraClient11g_home1
<ul> <li>Utiliser un nom de source de données existant :</li> <li>DSN utilisateur</li> <li>DSN système</li> <li>Éditer pilotes</li> </ul>
Ignorer l'étape de configuration de l'assistant
Retour     Ajouter   Fermer

Évitez d'utiliser le pilote fourni par Microsoft appelé **Microsoft ODBC for Oracle**. Microsoft recommande d'utiliser le pilote ODBC fourni par Oracle (voir <u>http://msdn.microsoft.com/en-us/library/ms714756%28v=vs.85%29.aspx</u>)

8. Cliquez sur **Se connecter**.

Oracle ODBC Driver Configu	ration
Data Source Name	Oracle DSN 1
Description	
TNS Service Name	ORCL   Help  Tab Councilian
User ID	Test Connection
Application Oracle Wor Enable Result Sets	Read-Only Connection
Enable Closing Cursors	Enable Thread Safety
Batch Autocommit Mode	Commit only if all statements succeed
Numeric Settings	Use Oracle NLS settings

- 9. Dans le champ de saisie Nom de source de données, saisissez un nom pour identifier la source de données (dans cet exemple, **Oracle DSN 1**).
- 10. Dans le champ TNS Service Name, saisissez le nom de connexion tel qu'il est défini dans le fichier tnsnames.ora (voir <u>conditions préalables</u><sup>223</sup>). Dans cet exemple, le nom de connexion est ORCL. *Note :* Si vous souhaitez avoir une liste déroulante de la combo box peuplée avec les valeurs du fichier tnsnames.ora, alors vous allez éventuellement devoir ajouter le chemin au dossier admin en tant que variable d'environnement TNS\_ADMIN.
- 11. Cliquez sur OK.

Oracle ODBC Driver Connect	×
Service Name ORCL	
User Name	
john_doe	Cancel
Password	About
••••••	

12. Saisissez le nom d'utilisateur et le mot de passe de la base de données, puis cliquez sur OK.

# 4.2.1.10.15 PostgreSQL (ODBC)

Cet exemple illustre comment se connecter à un serveur de base de données PostgreSQL depuis un appareil Windows machine par le biais des pilotes ODBC. Le pilote ODBC PostgreSQL n'est pas disponible sur Windows, il doit donc être téléchargé et installé séparément. Cet exemple utilise le pilote psqlODBC (version 11.0) téléchargé depuis le site web officiel (voir aussi <u>Aperçu des pilotes de base de données</u><sup>(62)</sup>).

**Note :** vous pouvez aussi vous connecter à un serveur de base de données PostgreSQL directement (sans le pilote ODBC), voir <u>Configurer une connexion PostgreSQL</u><sup>187</sup>.

## Exigences préalables :

- Le pilote *psqlODBC* doit être installé sur votre système d'exploitation.
- Vous disposez des détails de connexion à la base de données suivants : serveur, port, base de données, nom d'utilisateur et mot de passe.

## Pour établir une connexion à PostgreSQL avec ODBC :

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquez sur **Connexions ODBC**.
- 3. Sélectionnez l'option DSN Utilisateur.
- 4. Cliquez sur **Créer un nouveau DSN** is et sélectionnez le pilote depuis la liste déroulante. Si aucun pilote PostgreSQL n'est disponible dans la liste, veuillez vous assurer que le pilote ODBC PostgreSQL est installé sur votre système d'exploitation, comme mentionné dans les conditions préalables cidessus.

réer un DSN ODBC	C	:
Choisir un pilote puis cliquer s	soit sur Utilisateur soit sur Système pour détern haitez créer	niner le
genie de DOIV que vous sou		
SOL Server		

5. Cliquez sur DSN Utilisateur.

PostgreSQL Ur	icode ODBC Driver (	psqIODBC) Setup		×
<u>D</u> ata Source	PostgreSQL35W	Des <u>c</u> ription		
Data <u>b</u> ase	zoodb	SS <u>L</u> Mode	allow	$\sim$
<u>S</u> erver	my-postgresql-server	Port	5432	
<u>U</u> ser Name	dbs-user	Pass <u>w</u> ord	•••••	
				_
Options				Test
Datasource	Global	Manage DSN	Save	Cancel

6. Remplissez les identifiants de connexion à la base de données (ils doivent être fournis par le propriétaire de la base de données), puis cliquez sur **Enregistrer**.

La connexion est désormais disponible dans la liste des connexions ODBC. Pour se connecter à la base de données, vous pouvez soit double-cliquer sur la connexion ou la sélectionner, puis cliquer sur **Connexion**.

3 Se connecte	er à une source de données		×
	Connexions ODBC		
Assistant connexion	ODSN système ODSN utilisateur	Créer une chaîne de connexion	
Connexions existantes	DSN fichier      Nom de la source de données	Pilote	I
100			
Connexions ADO			
Connexions ODBC			

# 4.2.1.10.16 Progress OpenEdge (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données Progress OpenEdge 11.6 par le biais de JDBC.

## Prérequis

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK, soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(1050)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Veuillez vous assurer que la plateforme de MapForce (32-bit, 64-bit) correspond à celle du JRE/JDK.
- La variable d'environnement PATH du système d'exploitation doit inclure le chemin vers le répertoire bin du répertoire d'installation JRE ou JDK, par exemple C:\Program Files (x86) \Java\jre1.8.0\_51\bin.
- Le pilote Progress OpenEdge JDBC doit être disponible sur votre système d'exploitation. Dans cet exemple, la connectivité JDBC est fournie par les fichiers de composant de pilote openedge.jar et pool.jar disponibles dans C:\Progress\OpenEdge\java en tant que partie intégrante de l'installation OpenEdge SDK.
- Vous disposez des détails de connexion à la base de données suivants : hôte, port, nom de la base de données, nom d'utilisateur et mot de passe.

## Pour se connecter à OpenEdge par le biais de JDBC

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquez sur JDBC Connections.
- 3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, les chemins de fichier .jar exigés se trouvent sous : C:\Progress\OpenEdge\java\openedge.jar;C:\Progress\OpenEdge\java\pool.jar;. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le chemin d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH<sup>(165)</sup>).
- 4. Dans le champ « Driver », choisir com.ddtek.jdbc.openedge.OpenEdgeDriver. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Chemins de classe :	C:\Progress\OpenEdge\java\openedge.jar;C:\Progress\OpenE
Pilote :	~
Nom d'utilisateur :	dbusest
Mot de passe :	•••••
URL base de données :	jdbc:datadirect.openedge://localhost:8910;databaseName=
	~

- 5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
- 6. Saisissez la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

jdbc:datadirect:openedge://host:port;databaseName=db\_name

7. Cliquez sur **Connexion**.

# 4.2.1.10.17 Progress OpenEdge (ODBC)

Cet exemple illustre comment se connecter à un serveur de base de données Progress OpenEdge par le biais du pilote Progress OpenEdge 11.6 ODBC.

## Exigences préalables :

• Le pilote ODBC Connector for Progress OpenEdge doit être installé sur votre système d'exploitation. Le pilote Progress OpenEdge ODBC peut être téléchargé depuis le site web du vendeur (voir aussi <u>Aperçu des pilotes de base de données</u><sup>(162)</sup>). Veillez à télécharger le pilote 32-bit lorsque vous exécutez la version 32-bit de MapForce, et le pilote 64-bit lorsque vous exécutez la version 64-bit. Après l'installation, vérifiez si le pilote ODBC est disponible sur votre appareil (voir aussi <u>Aperçu des pilotes ODBC disponibles</u><sup>(161)</sup>).

ODBC Data Source Administrator		
User DSN System DSN File DSN Drivers Tr	acing Connection F	Pooling About
ODBC Drivers that are installed on your system:		
Name	Version	Company
Microsoft Access Driver (*.mdb, *.accdb)	15.00.4695.1000	Microsoft Corpo
Microsoft Access Text Driver (*.txt, *.csv)	15.00.4695.1000	Microsoft Corpo
Microsoft Excel Driver (*xls. *xlsx. *xlsm, *xlsb)	15.00.4695.1000	Microsoft Corpo
Progress OpenEdge 11.6 Driver	7.10.05.167	Progress Softwa
SQL Server	6.01.7601.17514	Microsoft Corpo
SQL Server Native Client 11.0	2011.110.3000.00	Microsoft Corpo
< [		Þ
An ODBC driver allows ODBC-enabled ODBC data sources. To install new d program.	d programs to get info rivers, use the drivers	ermation from s setup
OK	Apply	Help

• Vous disposez des détails de connexion de base de données suivants : nom d'hôte, numéro de port, nom de base de données, ID utilisateur et mot de passe.

## Pour se connecter à Progress OpenEdge par le biais d'ODBC

- 1. Lancer l'assistant de connexion de base de données<sup>100</sup>.
- 2. Cliquez sur **Connexions ODBC**.
- 3. Cliquez sur**DSN Utilisateur** (en alternative, cliquez sur **DSN Système** ou **DSN Fichier**, dans un tel cas, les instructions suivantes seront semblables).
- 4. Cliquez sur Ajouter 📌 .
- 5. Sélectionnez le **pilote Progress OpenEdge** depuis la liste et cliquez sur **DSN Utilisateur** (ou **DSN Système**, le cas échéant).

С	réer un DSN ODBC	×
	Choisir un pilote puis cliquer soit sur Utilisateur soit sur Système pour déterminer le genre de DSN que vous souhaitez créer.	
	Progress OpenEdge 11.6 Driver 🔹	
	DSN utilisateur DSN système Annuler	

6. Remplissez les identifiants de connexion à la base de données (base de données, serveur, port, nom utilisateur, mot de passe) et cliquez sur **OK**. Pour vérifier la connectivité avant d'enregistrer les données saisies, cliquez sur **Tester la connexion**.

ODBC Progress OpenEdge Wire Protocol Driver Setup				
General Advanced Securit	y Failover About			
Data Source Name:	my_progress_dsn	Help		
Host Name:	localhost			
Port Number:	8910			
Data <u>b</u> ase Name:	oebpsdev			
<u>U</u> ser ID:	altova			
Test Connect	OK Cancel	Apply		

7. Cliquer sur OK. La nouvelle source de données apparait désormais dans la liste des sources de données d'ODBC.

Connexions ODBC	
O DSN système	O Créer une chaîne de connexion
OSN utilisateur	
O DSN fichier	$\sim$
Nom de la source de données my_progress_dsn	Pilote Progress OpenEdge 11.6 Driver
+ * Ø D	
	Se connecter Fermer

8. Cliquez sur Se connecter.

# 4.2.1.10.18 Sybase (JDBC)

Cet exemple illustre comment se connecter à un serveur de base de données Sybase par le biais de JDBC.

## Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK, soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(105)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Assurez-vous que la plateforme de MapForce (32-bit, 64-bit) correspond à celle du JRE/JDK.
- Le composant Sybase *jConnect* doit être installé sur votre système d'exploitation (dans cet exemple, *jConnect 7.0* est utilisé, installé en tant que partie intégrante de l'installation Sybase Adaptive Server Enterprise PC Client). En ce qui concerne les instructions d'installation du client de base de données, référez-vous à la documentation Sybase.

• Vous disposez des détails de connexion à la base de données suivants : hôte, port, nom de la base de données, nom d'utilisateur et mot de passe.

## Connexion à Sybase par le biais de JDBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur JDBC Connections.
- 3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le chemin de fichier .jar exigé se trouve sous : C: \sybase\jConnect-7\_0\classes\jconn4.jar. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi Configuration du CLASSPATH<sup>165</sup>).
- 4. Dans la case "Driver", choisir **com.sybase.jdbc4.jdbc.SybDriver**. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Classpaths:	C:\sybase\jConnect-7_0\classes\jconn4.jar;	
Driver:	com.sybase.jdbc4.jdbc.SybDriver	•
Username:	dbuser	
Password:	•••••	
Database URL:	jdbc:sybase:Tds:SYBASE12:2048/PRODUCTSDB	*
		~
	Connect	Close

- 5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
- 6. Saisissez la chaîne de connexion au serveur de la base de données dans la zone de texte URL de base de données, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

jdbc:sybase:Tds:hostName:port/databaseName

7. Cliquez sur **Connexion**.

# 4.2.1.10.19 Teradata (JDBC)

Cet exemple indique comment se connecter à un serveur de base de données Teradata avec JDBC.

## Exigences préalables :

- JRE (Java Runtime Environment) ou Java Development Kit (JDK) doit être installé. Il peut s'agir soit d'Oracle JDK, soit d'un build open source comme Oracle OpenJDK. MapForce déterminera le chemin vers la Java Virtual Machine (JVM) depuis les emplacements suivants, dans l'ordre suivant : a) le chemin personnalisé JVM que vous avez éventuellement défini dans l'application **Options**, voir <u>Paramètres Java</u><sup>(1050)</sup>; b) le chemin JVM trouvé dans le registre Windows ; c) la variable d'environnement JAVA\_HOME.
- Assurez-vous que la plateforme de MapForce (32-bit, 64-bit) correspond à celle du JRE/JDK.
- Le pilote JDBC (un ou plusieurs fichiers .jar qui fournissent une connectivité à la base de données) doit être installé sur votre système d'exploitation. Dans cet exemple, le pilote JDBC Teradata 16.20.00.02 est utilisé. Pour plus d'informations, voir <u>https://downloads.teradata.com/download/connectivity/jdbc-driver</u>.
- Vous disposez des détails de connexion à la base de données suivants : hôte, base de données, port, nom utilisateur et mot de passe.

## Connexion à Teradata par le biais de JDBC :

- 1. Lancer l'assistant de connexion de base de données<sup>160</sup>.
- 2. Cliquez sur JDBC Connections.
- 3. À côté de « Classpaths », saisissez le chemin menant au fichier .jar qui fournit une connectivité à la base de données. Si nécessaire, vous pouvez aussi saisir une liste séparée par points-virgules de chemins d'accès de fichiers .jar. Dans cet exemple, le fichier .jar exigé se trouve sous le chemin suivant : C:\jdbc\teradata\. Veuillez noter que vous pouvez laisser en blanc le champ « Chemins de classe » si vous avez ajouté le(s) chemin(s) d'accès du fichier .jar à la variable d'environnement CLASSPATH du système d'exploitation (voir aussi <u>Configuration du CLASSPATH</u><sup>105</sup>).
- 4. Dans le champ « Pilote », choisir com.teradata.jdbc.TeraDriver. Veuillez noter que cette entrée est disponible si un chemin d'accès de fichier .jar valide est trouvé soit dans le champ « Chemins de classe », soit dans la variable d'environnement CLASSPATH du système d'exploitation (voir l'étape précédente).

Enter a connectio 'Connect' to proce	n string and select (or enter manually) a valid JDBC drive ed.	ar. Click on
Classpaths:	C:\jdbc\teradata\terajdbc4.jar;C:\jdbc\teradata\t	dgssconfig.jar
Driver:	com.teradata.jdbc.TeraDriver	~
Username:	demouser	
Password:	•••••	
Database URL:	jdbc:teradata://demodatabase	^
		~

- 5. Saisissez le nom d'utilisateur et le mot de passe de la base de données dans les champs de saisie correspondants.
- Saisissez la chaîne de connexion du serveur de la base de données dans le champ de saisie Database URL, en remplaçant les valeurs soulignées avec celles applicables à votre serveur de base de données.

//databaseServerName	rName	
----------------------	-------	--

7. Cliquez sur **Connexion**.

## 4.2.1.10.20 Teradata (ODBC)

Cet exemple indique comment se connecter à un serveur de base de données Teradata avec ODBC.

## Exigences préalables :

- Le pilote Teradata ODBC doit être installé (voir <u>https://downloads.teradata.com/download/connectivity/odbc-driver/windows</u>. Cet exemple utilise le pilote Teradata ODBC pour Windows version 16.20.00.
- Vous nécessitez les détails de connexion de base de données suivants : hôte, port, nom d'utilisateur et mot de passe.

## Connexion à Teradata par le biais de ODBC :

 Appuyez sur la clé Windows, saisissez « ODBC » et sélectionnez Configurer les sources de données ODBC (32-bit) depuis la liste de suggestions. Si vous disposez d'un pilote 64-bit, sélectionnez Configurer les sources de données ODBC (64-bit) et utilisez MapForce 64-bit aux étapes suivantes.

ODBC Data Source Administrator (32-bit)	×
User DSN System DSN File DSN Drivers Tracing Connect	ion Pooling About
<u>S</u> ystem Data Sources:	
Name Platform Driver	A <u>d</u> d
	Remove
	Configure
An ODBC System data source stores information about A System data source is visible to all users on this mach	how to connect to the indicated data provider. nine, including NT services.
ОК	Cancel Apply Help

2. Cliquez sur System DSN, puis cliquez sur Ajouter

Create New Data Source		×
	Select a driver for which you want to set up a data Name Microsoft Paradox Driver (*.db ) Microsoft Paradox-Treiber (*.db ) Microsoft Text Driver (*.txt; *.csv) Microsoft Text-Treiber (*.txt; *.csv) SQL Server SQL Server SQL Server Native Client 11.0 Teradata Database ODBC Driver 16.10	v ^ 1 1 1 1 1 2 1 2 1 2 1 2
	< <u>B</u> ack Finish	Cancel

3. Choisissez le pilote Teradata Database ODBC et cliquez sur Terminer.

ODBC Driver S	etup f	or Teradata Database		×
Data Source				OK
Name:	my	_teradata_source		Cancel
Description:				Cancer
Teradata Serv	ver Infr			Help
Name or IP address	der	noserver		$\hat{}$
Authentication	n ated Se	curity		
Parameter:			¥	Change
Username:	demo	user		
<u>Password</u>	••••	••••		
O Teradata <u>V</u>	<u>V</u> allet S	tring		
Optional D <u>e</u> fault Datab	ase:			
Account String	g: [			Options >>
Session Charac	ter Set			_
UTF8				~

- 4. Saisir le nom et, éventuellement, une description qui vous aidera à identifier cette source de données ODBC à l'avenir. De même, saisissez les identifiants de connexion de base de données (serveur de base de données, utilisateur, mot de passe), et, en option, choisissez une base de données.
- 5. Cliquez sur **OK**. La source de données apparaît maintenant dans la liste.

S ODBC	Data Source A	dministr	ator (32-bi	t)						×
User DSN	System DSN	File DSN	N Drivers	Tracing	Connection	Pooling	About			_
<u>S</u> ystem D	ata Sources:									
Name		Platform	Driver					A <u>d</u> d		
my_tera	data_source	32-bit	Teradata [	)atabase (	DBC Driver	16.10		<u>R</u> emo	ve	
								<u>C</u> onfigu	ire	
<							>			
	An ODBC Sy A System dat	stem data ta source i	source stor s visible to a	es informal all users of	ion about ho this compute	w to conr r, includin	nect to th ng NT se	ne indicated data rivices.	a provider.	
					OK	Cano	cel	Apply	Help	

- 6. Exécutez MapForce et lancez l'assistant de connexion à la base de données<sup>(60)</sup>.
- 7. Cliquez sur Connexions ODBC.

ODBC Connection	s
System DSN	O Build a connection string
O User DSN	
⊖ File DSN	$\sim$
Data Source Name	Driver
my_teradata_source	Teradata Database ODBC Driver 16.10
C 🛇 🗱 🕂	

- 8. Cliquez sur **System DSN**, sélectionnez la source de données créée précédemment, puis cliquez sur **Se connecter**.
- Note : si vous obtenez l'erreur suivante : « Le pilote a retourné (ou n'a pas pu retourné) SQL\_DRIVER\_ODBC\_VER: 03.80 invalide », veuillez vous assurer que le chemin menant au client ODBC (par exemple, C:\Program Files\Teradata\Client\16.10\bin, si vous l'avez installé à cet endroit, par exemple) existe bien dans votre variable d'environnement PATH de votre système. Si ce chemin est manquant, ajoutez-le manuellement.

# 4.2.2 Introduction aux mappages de base de données

Cette section est une introduction pour travailler avec des bases de données dans MapForce. Elle vous montre comment travailler avec une base de données une fois qu'une connexion a été établie avec succès (voir <u>Connexion à une base de données</u><sup>153</sup>). Cela inclut la sélection des objets de base de données que vous souhaitez voir apparaître sur le mappage, gérer les relations de base de données, et de configurer les paramètres de base de données applicables au processus de mappage. Des exemples pour atteindre des

objectifs spécifiques lorsque les données de mappage de ou vers les composants de base de données sont aussi inclus.

# 4.2.2.1 Ajouter des bases de données vers le mappage

Avant d'ajouter une base de données au mappage, veuillez vous assurer de choisir un langage de transformation qui prend en charge les mappages de base de données. Cela peut être un langage de transformation BUILT-IN, ou un des langages suivants : C++, C#, Java (voir aussi <u>Choisir un langage de transformation</u>). Veuillez noter que si vous avez l'Intention de déployer le mappage sur FlowForce Server ou de l'exécuter avec MapForce Server, ou d'utiliser des fonctions comme Bulk Transfer et des procédures stockées, BUILT-IN doit être sélectionné en tant que langage de transformation.

Une fois que le langage de transformation désiré a été sélectionné, vous pouvez ajouter une base de données au mappage d'une des manières suivantes :

- Dans le menu Insérer, cliquer sur Base de données.
- Cliquer sur la touche de la barre d'outils **Insérer base de données** ( <sup>0</sup><sup>1</sup> ).

En choisissant une de ces actions, un assistant de connexion de base de données apparaît, vous guidant à travers les étapes requises pour se connecter à la base de données.

🙄 Choisir une base	de données	×
Assistant connexion	Assistant connexion Veuillez choisir une base de données source puis cliquer sur Suivant.	
connexions existantes Connexions ADO Connexions ADO Connexions ODBC Connexions JDBC Connexions JDBC Connexions ADO.NET Ressources globales	Veuillez choisir une base de données source puis cliquer sur Suivant. Si le fournisseur de base de données avec lequel vous souhaitez travailler n'est pas listé ci-dessous, veuillez créer une connexion utilisant soit ADO soit directement ODBC. ( Microsoft Access (ADO) Serveur Microsoft SQL (ADO) Oracle (ODBC / JDBC) MySQL (ODBC) IBM DB2 (ODBC / JDBC) Sybase (ODBC) PostgreSQL Firebird (ODBC / JDBC) SQLite	
Connexions PostgreSQL	Sauter l'étape de configuration pour l'assistant	
	Suivant > Fermer	

Assistant de connexion à la base de données

**Note :** Dans certains scénarios avancés, les bases de données peuvent être ajoutées au mappage en tant que variables (voir <u>Utiliser des Variables</u><sup>(22)</sup>). Lorsque vous choisissez d'ajouter une structure de base de données en tant que variable, le même assistant de connexion de base de données que l'assistant mentionné ci-dessus apparaît.

Pour plus d'instructions concernant comment l'établissement d'une connexion à une des bases de données prises en charge par MapForce, voir <u>Connexion à une base de données</u><sup>153</sup>.

Une fois que la connexion à la base de données a été établie avec succès, vous êtes invité à choisir les objets de base de données qui doivent apparaître dans le mappage. Voir <u>Ajouter, éditer et supprimer des objets de base de données</u><sup>254</sup> pour plus d'informations.

# 4.2.2.2 Exemple : ajouter la base de données "altova.mdb" dans le mappage

Cet exemple vous montre comment ajouter un échantillon de base de données Microsoft Access à un mappage. La base de données échantillon est appelée **altova.mdb** et se trouve dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\**. La base de données **altova.mdb** prend en charge plusieurs actions et concepts liés à la base de données décrits dans cette documentation.

## Pour ajouter la base de données altova.mdb dans le mappage :

 Dans le menu Insérer, cliquer sur Base de données. En alternative, cliquer sur la touche de la barre d'outils Insérer base de données (<sup>01</sup>).

🙁 Choisir une base	e de données	×
Assistant	Assistant connexion	
connexion	Veuillez choisir une base de données source puis cliquer sur Suivant.	
Connexions existantes Connexions ADO Connexions ODBC Connexions JDBC Connexions	Si le fournisseur de base de données avec lequel vous souhaitez travailler n'est pas listé ci-dessous, veuillez créer une connexion utilisant soit ADO soit directement ODBC. Microsoft Access (ADO) Progress (ODBC) Serveur Microsoft SQL (ADO) Teradata (JDBC) Oracle (ODBC / JDBC) MariaDB (ODBC) MySQL (ODBC) IBM DB2 (ODBC / JDBC) IBM Informix (ODBC / JDBC) Sybase (ODBC) PostgreSQL Firebird (ODBC / JDBC) SQLite	
Connexions ADO.NET Ressources globales Connexions PostgreSQL	Sauter l'étape de configuration pour l'assistant           Suivant >         Fermer	

2. Cliquer sur Microsoft Access (ADO), puis cliquer sur Suivant.
| 😢 Choisir une bas                            | e de données          |   | × |
|--|-----------------------|---|---|
| Assistant                                    | Connexions            | Se connecter à MS Access  |   |
| connexion                                    | existantes            | Cliquer sur 'Parcourir' et choisir une base de données Microsoft Access. Cliquer ensuite<br>sur 'Suivant' pour se connecter à la base de données. |   |
| Connexions ADO                               | Connexions<br>ODBC    |   |   |
| Connexions<br>JDBC<br>Ressources<br>globales | Connexions<br>ADO.NET | nts\Altova\MapForce2019\MapForceExamples\Tutorial\altova.mdb Parcourir  |   |
|  |                       | < Retour Se connecter Fermer  |   |

- 3. Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb disponible dans le dossier <br/>
  Chercher le fichier altova.mdb da
- 4. Sur invitation de choisir les objets de base de données, sélectionner Tables.

🛱 Insérer objets de base de données					
Veuillez choisir les Tables, Aperçus, Procédures stockées ou créez vos propres instructions SQL SELECT à insérer dans votre mappage. Les instructions SQL SELECT définies par l'utilisateur sont stockées localement dans le fichier mfd.					
Source					
💎 🗟 🛉 Ajouter/Modifier l'instruction SELECT Ajouter/Modifier relations (0) Ajouter/Éditer structures des jeux d'enregistrements					
म् 🖉 altova					
E 🖓 🔽 📴 altova					
I Address					
₩ MSysACEs					
₩ MSysObjects					
In MSysQueries					
₩ MSysRelationships					
Afficher aperçu >> Utiliser des noms d'objet par rapport au schéma par défaut	Annuler				

# 4.2.2.3 Ajouter, éditer et supprimer des objets de base de données

Certaines bases de données peuvent contenir un grand nombre d'objets (comme les schémas, tables, modes, etc.). Cette rubrique vous montre comment n'intégrer au mappage que les objets de base de données qui sont requis pour des objectifs de mappage. Ci-dessous, nous allons utiliser une base de données échantillon Access ; les instructions sont semblables pour les autres types de base de données.

- 1. Dans le menu Insérer, cliquer sur Base de données.
- 2. Cliquer sur Assistant de connexion, puis cliquer sur Microsoft Access (ADO).
- 3. Cliquer sur **Suivant**, et chercher **altova.mdb** disponible dans le dossier <**Documents**\**Altova\MapForce2023\MapForceExamples**\.

Un dialogue apparaît, vous permettant de choisir les objets de base de données que vous souhaitez voir inclus dans le mappage.

💼 Insérer objets de base de données						
Veuillez choisir les Tables, Aperçus, Procédures stockées ou créez vos propres instructions SQL SELECT à insérer dans votre mappage. Les instructions SQL SELECT définies par l'utilisateur sont stockées localement dans le fichier mfd.						
😴 🔽 🛧 Aigutar/Madifiar l'instruction SELECT – Aigutar/Madifiar relations (0) – Aigutar/Éditar structures des jaux d'apregistrements						
Address						
Altova						
Imposed MoysAccessODJects Market Ma						
₩SysObjects						
──── 🖉 赢 MSysRelationships						
Afficher apercu >> Utiliser des noms d'obiet par rapport au schéma par défaut						

Le nœud supérieur dans la structure indique la connexion à la base de données. La structure qui s'ensuit varie selon le type de la base de données. Par exemple, les bases de données Oracle et IBM DB2 ont un nœud de schéma to sous le nœud de connexion, alors que d'autres types de base de données ont un nœud de catalogue (base de données). Des caractères gras indiquent que le catalogue par défaut (base de données) ou le schéma est applicable.

Si votre compte d'utilisateur de base de données a accès à plusieurs bases de données ou schémas sur le serveur, vous pouvez passer à la base de données nécessaire en cliquant sur l'icône . L'exemple ci-dessous illustre bien le serveur SQL ; la structure peut varier dans d'autres types de base de données.



Afin d'inclure un objet de base de données (par exemple, une table) dans le mappage, cliquer sur la case à côté. Dans le but de cet exemple, cliquer sur la case située à côté de **Tables d'utilisateur**.

La touche de **Localisateur d'objet** ( ) vous permet de trouver des items de base de donnés spécifiques. Sélectionner un objet particulier (ou saisir son nom) dans la liste de choix qui apparaît dans la zone inférieure du dialogue.

La touche **Filtre** () vous permet de filtrer des objets par leur nom. Une fois que vous cliquez sur la touche Filtre, une icône de filtre est disponible à côté des objets qui prennent en charge les filtres (dans cet exemple, "Tables"). Cliquer sur l'icône de filtre pour choisir si le nom d'objet doit commencer avec, terminer avec, être égal à, ou contenir le texte de recherche. Maintenant, vous pouvez saisir le texte de recherche situé à côté du filtre (dans cet exemple, "A"):



La touche Afficher uniquement les objets activés ( 🗹 ) affiche les items où une case à cocher est active.

La touche **Ajouter/Modifier une instruction SELECT** vous permet d'ajouter ou d'éditer des instructions SELECT personnalisées pour la base de données actuelle. Les données retournées par ces instructions deviennent disponibles en tant que source de mappage. Pour plus d'informations, voir <u>Instructions SELECT</u> <u>SQL en tant que tables virtuelles</u><sup>(32)</sup>.

La touche **Ajouter/Modifier relations** vous permet de définir des relations de clé primaire locale et de clé étrangère entre les champs dans la base de données, en plus de celles qui peuvent déjà être présentes. Pour plus d'informations, voir <u>Définir des relations locales</u><sup>270</sup>.

La touche **Ajouter/Modifier structures des jeux d'enregistrements** s'applique aux bases de données qui prennent en charge les procédures stockées. Elle est uniquement activée si une procédure stockée est

actuellement sélectionnée depuis l'arborescence de la base de données. Pour plus d'informations, voir <u>Procédures stockées</u><sup>336</sup>.

La touche **Afficher aperçu** vous permet d'apercevoir rapidement les données de la table ou du mode sélectionné actuellement. Notez que vous pouvez aussi choisir et requêter une base de données indépendamment du processus de mappage, en utilisant le Navigateur de base de données. Pour plus d'informations, voir <u>Parcourir et interroger des bases de données</u><sup>332</sup>.

La case à cocher **Retirer les noms de schéma depuis les noms de table** est activée uniquement pour les bases de données qui prennent en charge les objets de qualification avec un nom relatif. Cela est utile si vous prévoyez de passer à une autre base de données plus tard. Pour plus d'informations, voir <u>Changer de bases de données et de schémas</u><sup>257</sup>.

Lorsque vous êtes prêt à ajouter des objets de base de données dans le mappage, cliquer sur **OK**. Seuls les tables, modes sélectionnés, etc. apparaîtront sur le composant de base de données et vous pouvez dessiner des connexions de mappage de ou vers eux de la manière standard de MapForce.

🖯 altova	
þ⊞ III Address	Į,
¦∋⊞ III Altova	þ
≥ ⊞ III Department	
⇒ ⊞ Office	
¦>⊞ III Person	

Pour changer à tout moment les objets de base de données, cliquer avec la touche de droite sur le composant, et choisir **Ajouter/Supprimer/Éditer des objets de base de données**.

Image: altowa         Image: altowa <t< th=""><th>計 よ 電 であ ×</th><th>Ajoute Actual Reche Coupe Copie <u>C</u>oller <u>S</u>uppr</th><th>er/Suppr iiser rche er r imer</th><th>rimer les</th><th>objets c</th><th>le base di</th><th>e données. S S S S</th><th>itrg+F itrg+X trg+C trg+V Entf</th></t<>	計 よ 電 であ ×	Ajoute Actual Reche Coupe Copie <u>C</u> oller <u>S</u> uppr	er/Suppr iiser rche er r imer	rimer les	objets c	le base di	e données. S S S S	itrg+F itrg+X trg+C trg+V Entf
	~	A <u>l</u> igne Ali <u>g</u> ne <u>P</u> ropri	er arbore er arbore iétés	escence à escence à	à gauche à droite	2		

# 4.2.2.4 Changer de Base de données et de schémas

Pour changer un composant de base de données sur le mappage afin qu'il utilise une nouvelle connexion de base de données, ouvrir les <u>paramètres de composant de base de données</u><sup>(28)</sup>, cliquer sur **Modifier**, et suivre les instructions de l'assistant pour vous connecter à la nouvelle base de données. Si vous souhaitez uniquement modifier les objets de la base de données qui sont affichés dans le mappage, cliquer avec la touche de droite sur la barre de titre du composant de base de données et choisir **Ajouter/Éditer/Supprimer des objets de base de données** depuis le menu contextuel, voir aussi <u>Ajouter, Éditer, Supprimer des objets de base de données</u><sup>(28)</sup>.

Veuillez noter que, par défaut, une modification de la base de données ne remappe pas toutes les connexions de mappage existantes de ou vers le composant de base de données. Avec la plupart des types de base de données, néanmoins, il est possible de préserver les connexions de mappage après être passé à une autre base de données, si vous prenez la peine de respecter les étapes de préparation décrites dans cette rubrique.

En sélectionnant des objets de base de données comme décrit dans <u>Ajouter</u>, <u>Éditer</u>, <u>Supprimer des objets de</u> <u>base de données</u><sup>(26)</sup>, vous pouvez choisir en option si des noms d'objets de base de données doivent être considérés comme relatifs par rapport à un schéma par défaut. Dans le contexte des bases de données, un "schéma" est un regroupement logique des objets de base de données, généralement à des fins de sécurité comme de l'attribution des permissions. Les schémas ont un comportement légèrement différent selon le type de base de données.

Dans MapForce, traiter les noms d'objet de la base de données en tant que relatif à un schéma est important si vous prévoyez de changer plus tard le mappage vers une base de données différente (par exemple dans une base de données de production). Cela est également utile si le schéma de base de données a été renommé sur le serveur à un certain moment ; vous devrez donc mettre à jour le mappage. Si le nouveau schéma a la même structure que celle utilisée au moment du design de mappage, vous pouvez y passer sans devoir

changer manuellement toutes les connexions de mappage. Cela est uniquement possible si vous configurez les noms d'objet pour qu'ils soient relatifs à un schéma par défaut tel que décrit ci-dessous.

Veuillez noter :

- Utiliser les noms d'objet relatifs à un schéma par défaut est possible pour les types de base de données qui prennent en charge les schémas : IBM DB2, IBM Informix, IBM Db2 for i (iSeries), Oracle, PostgreSQL, Progress OpenEdge, SQL Server et Sybase.
- Cette fonction est prise en charge en commençant avec la version de MapForce 2020.
- Il n'est pas possible d'utiliser des noms relatifs si le composant de base de données inclut des relations locales<sup>270</sup> ou des instructions SELECT en tant que tables virtuelles<sup>362</sup>.

Pour traiter des noms d'objet comme relatifs au schéma par défaut :

- Cliquer avec la touche de droite sur la barre de titre du composant de base de données sur le mappage et sélectionner Ajouter/Supprimer/Éditer des objets de base de données depuis le menu contextuel.
- 2. Sélectionner un ou plusieurs objets qui appartiennent au schéma par défaut, ou pour le catalogue par défaut (base de données) et le schéma, selon le cas. La base de données par défaut et le schéma sont affichés en gras. Dans l'exemple ci-dessous, le catalogue par défaut est **Sandbox** et le schéma par défaut est **user**. Cette structure est spécifique pour les bases de données SQL Server et peut varier dans d'autres types de base de données. Pour plus d'informations concernant la structure affichée, voir Ajouter, Éditer, Supprimer des objets de base de données

🛱 Add/Remove/Edit Database Objects					
Please choose Tables, Views, Stored Procedures, or create your own SQL SELECT statements to insert into your mapping. SQL SELECT statements defined by the user are stored locally in the mfd file.					
V          Add/Edit SELECT Statement         Add/Edit Relations (0) Add/Edit Recordset Structures          Image: Connection          Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Image: Connection          Image: Connection           Sandbox         4 Sandbox					
□ 1: test					
Show Preview >>  Use object names relative to default schema	Cancel				

3. Cocher la case **Utiliser les noms d'objet relatifs au schéma par défaut**. Veuillez noter que cette case à cocher est grisée si la base de données ne prend pas en charge les noms d'objet relatifs.

Si les objets que vous requérez dans le mappage se trouvent dans un schéma différent (de celui par défaut), vous disposez des alternatives suivantes :

- Se connecter en tant qu'un autre utilisateur de base de données qui a accès au schéma par défaut requis.
- Reconfigurer les serveurs de base de données de manière à changer le schéma par défaut de l'utilisateur de base de données existante, si vous disposez des privilèges requis.

Prenons l'exemple d'un Serveur SQL, un administrateur de base de données peut changer le schéma par défaut d'un utilisateur de base de données avec une déclaration comme celle ci-dessous (qui part du principe que le catalogue actuel est "Sandbox" et que l'utilisateur et le schéma existent déjà).

USE [S	Sandbo	[xc				
GO						
ALTER	USER	[test_user]	WITH	DEFAULT_	_SCHEMA=[test_so	chema]
GO						

Les objets de base de données qui sont relatifs au schéma par défaut sont affichés dans le mappage avec un nom relatif. Veuillez considérer les différences entre les deux composants de base de données :

Connection		Connection1
þ⊡ III table	₽	Þ⊟ III test.table Þ
p 🔤 🕅	⊳	¢ [¶id ¢
val	⊳	⊳ 🖬 val 🛛 🖒
b user_data	⊳	🔉 🛄 user_data 🛛 🖒
	4	

Dans le composant de gauche, la case **Utiliser les noms d'objet relatifs au schéma par défaut** a été cochée, c'est pourquoi la table est affichée avec un nom relatif. Dans le composant de droite, la case n'a pas été cochée, et le nom de table contient le nom de schéma.

Lorsque les objets de base de données sont référencés de manière relative, il devient possible de faire passer le mappage dans une nouvelle base de données ou schéma, *sans perdre les connexions de mappage*, comme suit :

- Ouvrir les <u>paramètres de composant de base de données</u><sup>231</sup> et cliquer sur **Changer**. Suivre les instructions pour vous connecter à la nouvelle base de données *en tant que nouvel utilisateur*. Si vous avez qualifié les objets d'un nom relatif, et si la nouvelle base de données a la même structure, toutes les connexions dans le mappage seront mises à jour automatiquement. Plus particulièrement, elles correspondront au *catalogue et schéma par défaut du nouvel utilisateur de base de données*.
- Si vous souhaitez effectuer le passage de manière récurrente, il est recommandé de définir la connexion de base de données en tant que <u>Ressource Globale</u> <sup>(971)</sup>. Par exemple, la Ressource globale peut présenter deux configurations : une configuration par défaut pour la base de données de développement et une configuration de production. En partant du principe que ces deux base de données ont la même structure, vous pouvez passer entre les bases de données en choisissant la configuration requise depuis une liste déroulante :

÷	Default	
	Derudit	 а.

Pour plus d'informations concernant ce scénario, voir <u>Exemple: Changer de base de données</u><sup>682</sup>.

Si des objets de base de données apparaissent en rouge après avec changé de base de données, cela indique qu'ils n'existent pas dans le nouveau schéma de base de données.

### Génération de code

La case à cocher **Utiliser des noms d'objets relatifs au schéma par défaut** touche le code de programmation C#, C++, ou Java généré. Si cette case est cochée, toutes les références de base de données deviennent également relatives dans le code généré. L'exemple suivant est un exemple d'une déclaration SQL générée (en C#) lorsque la case n'est pas cochée. Veuillez noter que le nom de la table inclut le nom de schéma.

```
var3_NewStatement = (Altova.Db.Dbs.NewStatement(closure.var1_Connection_Instance, 1,
"SELECT [id], [val], [user_data] FROM [user].[table]"));
```

L'exemple suivant est un exemple d'une déclaration SQL générée (en C#) lorsque la case est cochée:

```
var3_NewStatement = (Altova.Db.Dbs.NewStatement(closure.var1_Connection_Instance, 1,
"SELECT [id], [val], [user_data] FROM [table]"));
```

**Note :** Il existe encore un autre moyen de supprimer les noms de schéma dans le code généré, en sélectionnant la case **Supprimer noms de schéma des noms de table** dans les <u>paramètres de</u> <u>composant de base de données</u><sup>281</sup>. Cette approche est uniquement prise en charge pour une rétrocompatibilité et doit être évitée.

## 4.2.2.5 Gérer des relations de base de données

Les bases de données relationnelles, comme l'indique leur nom, présentent généralement des relations définies entre leurs tables. Prenons comme exemple la base de données **altova.mdb** trouvée dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\.** On y trouve plusieurs relations, par exemple :

- L'entreprise d'échantillon (correspondant à la table "Altova") consiste en un ou plusieurs bureaux, (par exemple, à Brenton et à Vereno). Dans la terminologie de base de données, il existe une relation one-to-many entre les tables "Altova" et "Office". En d'autres termes, pour chaque entrée PrimaryKey dans la table "Altova", il peut y avoir plusieurs entrées ForeignKey dans la table "Office". Toute entrée "Office" où la valeur ForeignKey correspond à la valeur PrimaryKey dans "Altova" devrait donc être considéré être un bureau de "Altova".
- Chaque bureau comprend un ou plusieurs départements (par exemple "Marketing", "IT", "Development"). Ici, on retrouve une relation one-to-many entre les tables "Office" et "Department".
- Enfin, chaque département consiste en une ou plusieurs personnes. d'où la relation one-to-many entre les tables "Department" et "Person".



Relations des tables dans la base de données altova.mdb (mode "Relationships" Microsoft Access)

Les relations entre les tables de base de données sont importantes pour des raisons de mappage. MapForce garde une trace de ces relation de base de données lorsque vous ajoutez une base de données au mappage. Cela vous permet de préserver les relations de base de données lorsque vous mappez des données de ou vers une base de données. Afin de mieux comprendre ce concept, ajouter la base de données **altova.mdb** au mappage (en utilisant la commande de menu **Insérer | Base de données**). Nous appellerons chacune des tables ci-dessous une table "root" :

🏮 altova	
⇒⊞ III Address	1
Description: D	l k
þ. ∰ Department	l k
Description: The second se	l l
þ⊞ III Person	l l

Tables "Root"

En agrandissant une table "root", vous voyez apparaître en-dessous toutes les tables liées dans une vue arborescente. Par exemple, si vous agrandissez la table **Office**, veuillez noter comment la hiérarchie de la table liée est affichée :

- Une flèche orientée vers la gauche (←) devant une table montre qu'il s'agit d'une table enfant. Par exemple, Address est un enfant d'Office. Department est aussi un enfant d'Office, de même qu'une table "frère" d'Address, les deux ont donc le même niveau de retrait. Comme vous pouvez le voir, la relation dans le mappage correspond au diagramme ci-dessus.
- Une flèche orientée vers la droite (→) devant une table montre qu'il s'agit d'une table parent. Par exemple, Altova est un parent d'Office.



Relations de tables dans MapForce (base de données altova.mdb)

Cette représentation hiérarchique des tables vous aide à préserver les relations de base de données existantes lorsque votre mappage est lu depuis ou écrit dans une base de données. Par exemple, partons du principe que vous souhaitez obtenir tous les enregistrements de la table **Person** dans un fichier XML, regroupé par département. Plus spécifiquement, votre fichier XML devrait relier chaque personne à un département, comme pour la base de données **altova.mdb** utilisée dans cet exemple :

				Depart	ment				_	×
2	F	PrimaryKey 👻 Fo	reignKey 👻	Name 🚽						-
[	Ę	1	1 Ad	ministratior						
	-	🖉 PrimaryKey 👻	EMail 👻	First	<ul> <li>Last</li> </ul>	*	PhoneExt		-	
		1	v.callaby@nan	Vernon	Callaby		5	82 Office Man	age	
		2	f.further@nan	Frank	Further		4	71 Accounts R	ece	
		3	I.matise@nand	Loby	Matise		9	63 Accounting	g Ma	
	) +	*								
6	٦L	2	1 Ma	rketing						
	42	🖉 PrimaryKey 👻	EMail 👻	First	<ul> <li>Last</li> </ul>	-	PhoneExt	<ul> <li>Title</li> </ul>	-	
		4	j.firstbread@n	Joe	Firstbre	ad	6	21 Marketing	Mar	
		5	s.sanna@nano	Susi	Sanna		7	53 Art Directo	r	
	) +	¥								
E	₹_	3	1 En	gineering						
	4	🖉 PrimaryKey 👻	EMail 👻	First	<ul> <li>Last</li> </ul>		PhoneExt	<ul> <li>Title</li> </ul>	-	
		6	f.landis@nano	Fred	Landis		9	51 Program M	ana	
		7	m.landis@nan	Michelle	Butler		6	54 Software E	ngir	
		8	t.little@nanon	Ted	Little		8	52 Software E	ngir	
		9	a.way@nanon	Ann	Way		9	51 Technical V	Vrit	
		10	l.gardner@nar	Liz	Gardner		7	53 Software E	ngir	
		11	p.smith@nanc	Paul	Smith		3	34 Software E	ngir	
	3	*								
6	7_	4	1 IT 8	& Technical	_				_	 _
		PrimaryKey -	EMail 👻	First	- Last	*	PhoneExt	▼ Title	Ψ.	
		12	a.martin@nan	Alex	Martin		7	78 IT Manager		
		13	g.hammer@na	George	Hamme	r	2	23 Web Deve	ope	
		14	j.band@nanon	Jessica	Bander		2	41 Support En	gine	
		15	I.king@nanon	Lui	King		3	45 Support En	gine	
Pac	) orde	₩  1   1 of 3	N No. V No. Fil	ter Search						
Rec	oru		X NO FI	Bearch						

Comme illustré ci-dessus, le département "Administration" contient trois personnes, "Marketing" contient deux personnes, "Engineering" contient six personnes, etc.

Lorsque vous mappez des données depuis cette base de données, si vous souhaitez que chaque personne soit distribuée dans le département correct, il est important d'utiliser **Department** en tant que table "root", puis que vous mappiez depuis la table **Person** table qui est un enfant de **Department**:



Le mappage ci-dessus est un DB\_Altova\_Hierarchical.mfd modifié depuis le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Lorsque vous prévisualisez le mappage, le résultat est que chaque personne sera regroupée par département, ce qui était le résultat escompté. Donc, "Administration" contient trois personnes, "Marketing" contient deux personnes, "Engineering" contient six personnes, etc.

1	<	?xml version="1.0" encoding="UTF-8"?>							
2	<b>-</b> <	Altova xsi:noNamespaceSchemaLocation="file:///C:							
3	0	<office></office>							
4	0	<department></department>							
5		<name>Administration</name>							
6		<person> </person>							
12		<person> </person>							
18		<person> </person>							
24	- H								
25	0	<department></department>							
26		<name>Marketing</name>							
27		<person> </person>							
33		<person> </person>							
39	- H								
40	0	<department></department>							
41		<name>Engineering</name>							
42		<person> </person>							
48		<person> </person>							
54		<person> </person>							
60		<person> </person>							
66		<person> </person>							
72		<person> </person>							
78	- E								
79		<department> </department>							
106		<department> </department>							
121		<department> </department>							
130		<department> </department>							
151	-								
152	L <	/Altova>							

À présent, observons le mappage légèrement modifié ci-dessous, où des connexions ont été tirées délibérément de manière à ce que **Department** et **Person** sont des tables "root".



Cette fois, lorsque vous prévisualisez le mappage, toutes les personnes (quelque soit leur département de source) sont groupées sous chaque département de cible, ce qui n'était pas le résultat escompté. Donc, "Administration" contient 21 personnes, "Marketing" contient 21 personnes, "Engineering" contient 21 personnes, etc.

1	xml version="1.0" encoding="UTF-8"?
2	Altova xsi:noNamespaceSchemaLocation="file:///C:/l
3	
4	Operation of the second sec
5	<name>Administration</name>
6	erson>
12	erson>
18	erson>
24	erson>
30	erson>
36	erson>
42	erson>
48	erson>
54	erson>
60	erson>
66	erson>
72	erson>
78	erson>
84	erson>
90	erson>
96	erson>
102	erson>
108	erson>
114	erson>
120	erson>
126	erson>
132	
133	Operation of the second sec
262	Operation of the second sec
391	Operation of the second sec
520	Operation of the second sec
649	Operation of the second sec
778	Operation of the second sec
907	<ul> <li></li> </ul>

908 </Altova>

Dans le deuxième exemple, les relations de base de données sont ignorées, en raison de la manière dont les connexions ont été établies.

Therefore, lorsque vous souhaitez préserver des relations de base de données, make sure that connections are drawn to or from the same "root" table, which contains the child tables whose relationships you want to preserve. This works in the same way for both source and target databases. For examples of database mappings which preserve relationships, see the **DB\_Altova\_Hierarchical.mfd** and **Altova Hierarchical DB.mfd** files available in the

< Documents>\Altova\MapForce2023\MapForceExamples\ folder (see also Inserting Data into Multiple Linked Tables<sup>200</sup>).

There might also be cases when you do not want to preserve relations de base de données. For example, let's assume that you want to export all data from the **altova.mdb** database to a flat XML file adhering to the SQL/XML specification (Part 14 of the Structured Query Language (SQL) specification). This kind of mapping is illustrated by the **DB\_Altova\_SQLXML.mfd** sample, available in the

<Documents>\Altova\MapForce2023\MapForceExamples\ folder. The goal of the mapping is to get database data as flat XML file. The target SQL/XML schema was generated with XMLSpy, using the Convert | Create XML Schema from DB Structure menu command.



DB\_Altova\_SQLXML.mfd

Comme illustré ci-dessus, chaque table de base de données possède un élément correspondant dans le XML cible. Lorsque vous prévisualisez le résultat de mappage, vous pouvez constater que les lignes de base de données de chaque table sont écrites dans les éléments "row" dans la cible.

1	xml version="1.0" encoding="UTF-8"?		
2	<altova xsi:noNamespaceSchemaLocation="file:///</p>		
3	🖕 <altova></altova>		
4	Address>		
5	<pre>crow&gt; </pre>		
12			
13			
21			
22	⊖ <altova></altova>		
23	row>		
27	<ul> <li></li> </ul>		
28	Operation of the second sec		
29	ews/sums/rows		
34	<pre>crow&gt; </pre> //row>		
20			
35			
44	<pre>viows</pre>		
49	e <row></row>		
54	row>		
59	row>		
64	-		
65	Office>		
66	<pre>crow&gt;</pre>		
76	<pre>srow&gt;l</pre>		
86	-		
87	Person		
00			
00			
97	<pre></pre>		
106	<pre></pre>		
115	<pre>evaluation = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 =</pre>		
124	e <row></row>		
133	row>		
142	<row> </row>		
151	row>		
160	row>		
169	even with the second		
178	<pre>crow&gt;</pre>		
187	<pre>crow&gt; </pre> /row>		
196			
205			
205			
214	<row> </row>		
223	<pre>evaluation = = = = = = = = = = = = = = = = = = =</pre>		
232	<pre>erow&gt; </pre>		
241	row>		
250	row>		
259	row>		
268	row>		
277			
278	-		
279			
	a second data		

Comme le montre la sortie XML, aucune hiérarchie n'existe entre les éléments XML ; il s'agit d'une structure SQL/XML plat. Les relations de base de données ont été ignorées parce que nous avons intentionnellement mappé des données depuis plusieurs tables "root".

# 4.2.2.6 Définir des relations locales

Lorsque des tables de base de données ne possèdent pas de relations explicitement définies entre elles, vous pouvez définir ce type de relation localement dans MapForce. En particulier, vous pouvez créer, depuis MapForce, des relations de clé primaire et étrangère entre des colonnes de tables différentes, sans aucunement affecter la base de données. Toute colonne de base de données peut être utilisée en tant que clés primaire ou étrangère. De plus, de nouvelles relations peuvent être créées, en plus de celles existant dans la base de données. Les relations définies localement sont enregistrées ensemble avec le mappage.

Ces relations "on-the-fly" sont appelées **Relations locales** dans MapForce. La table suivante recense tous les champs possibles entre lesquels vous pouvez définir des relations locales. Des relations mixtes sont possibles (par exemple, mapper la sortie d'une procédure stockée dans une colonne de base de données). Les champs participant à la relation doivent avoir les mêmes types de données ou un type de données compatible.

Clé primaire/unique	Clé étrangère
<ul> <li>La colonne d'une table de base de données ou d'un mode</li> <li>Paramètre de sortie ou valeur de retour d'une procédure stockée, voir aussi <u>Procédures</u> <u>stockées</u></li> <li>La colonne d'un recordset retournée par une procédure stockée*</li> <li>La colonne d'une instruction SELECT définie par l'utilisateur, voir aussi <u>Instructions</u> <u>SELECT SOL en tant que tables</u> <u>virtuelles</u></li> </ul>	<ul> <li>La colonne d'une table de base de données ou d'un mode</li> <li>Paramètre d'entrée d'une procédure stockée</li> <li>Paramètre d'entrée d'une instruction SELECT définie par l'utilisateur</li> </ul>

\* Applicable si la procédure stockée est appelée soit en tant que source de données (sans paramètres) ou en tant que fonction (avec paramètres d'entrée et de sortie). Pour que le recordset devienne disponible pour la sélection, vous devez exécuter la procédure stockée une fois, pour extraire le recordset.

La base de données **altova-no-relation.mdb** utilisée dans cet exemple est une version simplifiée de la base de données **altova.mdb** fournie avec MapForce. Les tables "Person" et "Address", ainsi que toutes les relations de table ont été supprimées dans Microsoft Access. Comme indiqué dans le graphique ci-dessous, aucune des tables visibles dans l'arborescence **altova-no-relation** ont des tables enfants ; toutes les tables se trouvent au même niveau "root". Le contenu de chaque table est limité aux champs qu'elle contient.



Structure de base de données sans relation explicite

Le but de l'exemple est d'afficher les bureaux de "Altova" et montrer les départements dans chaque bureau. Veuillez noter que, dans **altova-no-relation.mdb**, les relations de clé primaire et étrangères n'existent pas explicitement, comme indiqué ci-dessus. Elles existent uniquement logiquement (implicitement), nous allons donc les recréer localement dans MapForce pour atteindre l'objectif du mappage.

Les relations locales peuvent être définies en ajoutant une base de données au mappage, ou en cliquant avec la touche de droite sur un composant de mappage existant et en choisissant **Ajouter/Supprimer/Éditer objets de base de données** du menu contextuel, comme illustré dans les étapes ci-dessous.

- 1. Dans le menu Insérer, cliquer sur Base de données.
- 2. Dans l'assistant de connexion, cliquer sur Microsoft Access (ADO), puis cliquer sur Suivant.
- 3. Chercher la base de données altova-no-relation.mdb disponible dans le dossier <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\, et cliquer sur Connexion.
- 4. Choisir la case à cocher Tables d'utilisateur.



5. Cliquer sur la touche Ajouter/Éditer Relations dans la barre d'icône.

6. Le dialogue "Ajouter/Éditer relations de table" s'ouvre. Cliquer sur **Ajouter relation**.

Add/Edit Relations			
In this dialog you can define relations in addition to those defined in the database. Please select for each relation a primary key object and a foreign key object and at least one column pair with matching datatypes.			
Primary/Unique Key Object	Foreign Key Object	ОК	
E [select object]	[select object]	Cancel	
[select object first]	[select object first]		
		Add Relation	
		Delete Relation	
		Add Column	
		Delete Column	
		-	
L			

7. Choisir des valeurs depuis les deux listes déroulantes afin de créer une relation de clé primaire et étrangère entre les tables "Altova" et "Office", comme illustré ci-dessous. Les deux listes déroulantes vous permettent de sélectionner les tables ou les objets de base de données pour lesquels vous souhaitez créer des relations. La liste de gauche spécifie l'objet qui stocke la clé primaire/unique, alors que la liste de gauche spécifie l'objet de clé étrangère. L'objet de Clé Primaire/Unique sera l'objet parent dans MapForce, et l'objet de Clé étrangère sera affiché en tant qu'enfant dans le composant de base de données (voir aussi <u>Gérer des relations de base de données</u>).

	Add/Edit Relations				<b>—</b>
lr re	n this dialog you can define relatior slation a primary key object and a f	ns in ar oreign	ddition to those defined in the data key object and at least one colur	abase. nn pair	Please select for each with matching datatypes.
	Primary/Unique Key Object		Foreign Key Object		ок
G	III Altova	•	Office	<b>•</b>	
	📲 PrimarvKey (INT)	•	ForeignKey (INT)	<b>T</b>	Cancel
					Add Relation Delete Relation
					Add Column
					Delete Column
L				-	

8. Cliquer sur **OK** pour terminer la définition de relation locale, et puis cliquer sur la touche **Insérer** pour insérer la base de données dans la zone de mappage.

À ce niveau, vous avez créé une relation locale entre la colonne **PrimaryKey** de la table "Altova" et la colonne **ForeignKey** de la table "Office". Comme illustré ci-dessous, la table de racine "Altova" est maintenant un parent de la table "Office". Concrètement, la table est indiquée en tant que table liée en-dessous de la table Altova avec son icône d'extension.

🖯 altova-no-relation	
🖓 🖽 Altova	ե
👌 🔤 🕎 PrimaryKey	ե
👌 🔤 🔟 xmins	ե
🔁 🔟 ipo	ե
🖒 🔟 xsi	ե
🖒 🔤 🔟 schemaLocation	ե
🖒 📖 🕅 Name	ե
¦} <sup>i</sup> ←⊞ <b>⊞ Office</b> ForeignKe	y 🖁
¦≥⊞ III Department	, s
¦⊳⊞ IIII Office	la la

Néanmoins, l'objectif de mappage n'est pas encore atteint. Pour atteindre l'objectif de mappage, utiliser la même méthode pour créer une relation entre les tables **Office** et **Department**, comme indiqué ci-dessous.



Pour ouvrir à nouveau le dialogue "Ajouter/Éditer les relations", cliquer avec la touche de droite dans le composant de base de données, et choisir **Ajouter/Supprimer/Éditer des objets de base de données** du menu de contexte.

Enfin, ajouter le schéma de cible dans le mappage comme suit :

- 1. Dans le menu Insérer, cliquer sur Insérer Schéma XML/Fichier.
- Chercher le fichier Altova\_Hierarchical.xsd disponible dans le dossier >Chercher le fichier Altova\MapForce2023\MapForceExamples\. Lorsque vous êtes invité à fournir un fichier XML d'échantillon, cliquer sur Sauter. Lorsque vous êtes invité à choisir un élément racine, choisir "Altova".

Veuillez noter que, afin de préserver les relations entre les tables dans le XML cible, toutes les connexions ont été tirées de la même table "root", hiérarchiquement (dans ce cas, "Altova"). Pour plus d'informations, voir <u>Gérer des relations de base de données</u><sup>(260)</sup>.



Une fois avoir défini le mappage indiqué ci-dessus, cliquer sur l'onglet **Sortie**, pour consulter le résultat. Le résultat de mappage affiche les éléments de département imbriqués sous chaque bureau respectif, ce qui était le but recherché de ce mappage.



# 4.2.2.7 Exécuter des mappages qui modifient des bases de données

Lorsqu'un mappage modifie des données de base de données de quelque manière que ce soit (par exemple, en insérant, en mettant à jour ou en supprimant des enregistrements), les modifications seront appliquées à la base de données par le moteur qui exécute le mappage. Le moteur qui exécute le mappage peut être MapForce, MapForce Server (tous deux autonomes ou sous gestion de FlowForce Server), ou l'environnement d'exécution du code généré pour C++, C# ou Java.

Lorsque vous prévisualisez le résultat de mappage directement dans MapForce (en cliquant sur l'onglet **Sortie**), un script de mise à jour s'affiche. Le script ne sera exécuté par rapport à la base de données qu'au moment où vous effectuerez explicitement cette action ; elle est uniquement disponible à la prévisualisation.

1	/*
2	The following SQL statements are only for preview and may not be executed in another SQL query too!
3	To execute these statements use function "Run SQL-script" from menu "Output".
4	Connect to database using the following connection-string:
5	Data Source=C:/Users/altova/Documents/Altova/MapForce2016/MapForceExamples/AltovaTarget.mdb;Provider=Microsoft.Jet.OLEDB.4.0
6	*/
7	
8	begin transaction
9	
10	DELETE FROM [Address]
11	
12	DELETE FROM [Person]
13	
14	DELETE FROM [Department]
15	
16	DELETE FROM [Office]
17	
18	DELETE FROM [Altova]
19	
20	SELECT IIF(MAX([Altova].[PrimaryKey]) IS NULL,0,MAX([Altova].[PrimaryKey]))+1 AS [PrimaryKey] FROM [Altova]
21	>>> %PrimaryKey1%
22	
23	INSERT INTO [Altova] ([Name], [PrimaryKey]) VALUES ('Organization Chart', '%PrimaryKey1%')

Aperçu de sortie d'un mappage qui modifie une base de données (Altova\_Hierarchical\_DB.mfd)

Le script montre des instructions pseudo-SQL uniquement à des fins informations ; il ne reflète pas directement les instructions qui doivent être exécutées. Par exemple, si plusieurs actions sont définies par rapport à une table (par exemple, "Update if... Insert Rest") seule la première action est affichée dans l'aperçu, puisque la deuxième action est exécutée conditionnellement.

Le script pseudo-SQL ne doit pas être appliqué manuellement à la base de données en utilisant d'autres outils SQL que les moteurs d'exécution mentionnés plus haut. Le script peut contenir le formatage de valeurs qui ne sont pas "comprises" par des éditeurs SQL externes.

Si vous souhaitez appliquer les modifications du mappage dans la base de données directement depuis MapForce, cliquer sur la commande **Exécuter Script SQL** disponible dans le menu **Sortie**. Attention, ne pas oublier que cette action modifiera en fait la base de données de manière immédiate.

Lorsque le mappage est exécuté avec MapForce Server (aussi bien la version autonome que sous gestion FlowForce Server), les modifications de la base de données seront appliqués immédiatement. La même chose se produit dans le code généré : les modifications de la base de données sont appliquées lorsque vous compilez et exécutez le code (par exemple en cliquant sur la commande **Exécuter** dans Visual Studio).

Votre installation MapForce comprend plusieurs bases de données d'exemple (fichiers Microsoft Access ou SQLite) disponibles dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples**\. Il est recommandé de ne pas appliquer les changements de la base de données depuis MapForce, en utilisant la commande **Exécuter script SQL**, par rapport à une des bases de données d'exemple fournies par MapForce ; cela pourrait fausser les exemples et les rendre inutilisables. Un moyen simple d'éviter d'écraser les données originales est de sauvegarder tout le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\ avant d'y mettre à jour des données.

Pour plus d'informations concernant l'exécution des mappages dans les autres environnements d'exécution que MapForce, voir :

- <u>Déployer des mappages dans FlowForce Server</u><sup>960</sup>
- Compiler des mappages dans les fichiers d'exécution MapForce Server<sup>957</sup>
- Générateur de Code<sup>1065</sup>

# 4.2.2.8 Remplacer des caractères spéciaux

Lors de la transformation de données de la base de données, vous allez éventuellement devoir supprimer des caractères spéciaux spécifiques, tels que des caractères chariot/saut de ligne (CR/LF), du chariot sources de données. Pour ce faire, vous pouvez utiliser une des approches suivantes :

- Définir une fonction de nœud pour le champ de base de données spécifique (ou même plusieurs champs) dont vous avez besoin. La fonction de nœud recevra la valeur du champ de base de données en tant qu'entrée, y appliquera des processus puis retournera le résultat au mappage. Pour plus d'informations concernant cette technique, voir <u>Défauts et fonctions de nœud</u><sup>600</sup>.
- 2. Traiter les valeurs de base de données avec l'aide des fonctions intégrées MapForce. Par exemple, pour identifier des caractères spécifiques, y compris les caractères de commande, vous pouvez utiliser la fonction <u>char-from-code</u> depuis la bibliothèque "core". Pour remplacer des valeurs, utiliser la fonction <u>replace</u> depuis la bibliothèque "lang".

Cette rubrique illustre la seconde approche. La base de données utilisée dans cet exemple est Microsoft Access, mais la même stratégie peut être appliquée avec d'autres base de données également.

Considérer une base de données Microsoft Access consistant en une table "Lignes" qui a deux colonnes : "ID" et "Description".

T	Lines			
	ID 👻	Description 👻	Click to Add	Ŧ
	1	This is our new company policy.		
	2	It will be implemented immediately.		
*	(New)			

L'objectif est d'extraire chaque description d'un fichier CSV (une description par ligne) ; ainsi un mappage pour atteindre cet objectif pourrait ressembler à l'exemple suivant :



Néanmoins, étant donné que chaque ligne "Description" dans Access contient plusieurs lignes séparées par des caractères CR/LF, la sortie de mappage contient aussi des sauts de ligne, ce qui n'est pas le résultat intenté :

1	"This is
2	our new company policy."
3	"It will be
4	implemented immediately."
5	

Afin de surmonter ce problème, nous allons ajouter au mappage les fonctions <u>char-from-code</u> et <u>replace</u> depuis la bibliothèque intégrée MapForce (voir aussi <u>Ajouter une fonction au mappage</u>). Chaque description doit être traitée de manière à ce que, quels que soient les caractères rencontrés ci-dessus, ils doivent être remplacés par un caractère d'espace.

Dans le graphique Unicode (<u>http://www.unicode.org/charts/</u>), les caractères LF et CR correspondent aux caractères **hex 0A | dec 10** et **hex 0D | dec 13**, respectivement. C'est pourquoi le mappage doit être modifié pour convertir les valeurs Unicode décimales 13 et 10 en un string, afin de permettre un autre traitement par la fonction replace.



Si vous faites un aperçu du mappage maintenant, vous verrez que les caractères CR/LF se trouvant dans le champ de base de données ont chacun été remplacés par un espace.

```
    This is our new company policy.
    It will be implemented immediately.
```

## 4.2.2.9 Gérer des valeurs nulles

Pour contrôler pendant l'exécution d'un mappage si un champ de base de données est null, utiliser les fonctions MapForce de bibliothèque <u>is-null</u><sup>(23)</sup> et <u>is-not-null</u><sup>(23)</sup>. Pour voir depuis MapForce si une table a zéro champs, la requêter en utilisant le navigateur de base de données (voir <u>Chercher et requêter les bases de données</u><sup>(32)</sup>).

Pour régler un champ de base de données sur zéro, utiliser la fonction <u>set-null</u><sup>724</sup>.

Pour remplacer les valeurs de base de données nulles par un string, utiliser la fonction <u>substitute-null</u><sup>(24)</sup>. Un échantillon de mappage qui illustre cet exemple est **DB\_ApplicationList.mfd**, disponible dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\**.

Pour plus d'informations concernant la gestion des comparaisons de valeur NULL dans les mappages qui mettent à jour les bases de données, voir <u>Gérer des nulls dans des actions de table de base de données</u><sup>327</sup>.

Pour plus d'informations concernant la gestion des nulls lors du mappage de bases de données de ou vers des documents XML, voir <u>Valeurs Nulles/ Nillables</u><sup>128</sup>.

# 4.2.2.10 Générer des valeurs séquentielles et uniques

Lors de l'insertion de données ou de la mise à jour d'une base de données, vous devrez parfois créer des valeurs séquentielles ou uniques "on-the-fly" pour les champs de base de données qui n'ont pas d'entrée depuis la source mappée. Dans ces cas, utiliser les fonctions de bibliothèque intégrées MapForce suivantes :

- <u>auto-number</u><sup>660</sup> (disponible dans la bibliothèque "core | generator functions"). Cette fonction est généralement <u>utilisée pour générer des valeurs</u> de clé primaire pour un champ numérique.
- <u>create-guid</u><sup>[47]</sup> (disponible dans la bibliothèque "lang | generator functions"). Cette fonction crée un identifiant globalement unique (en tant que string encodé hex) pour le champ spécifique.

Veuillez noter que les valeurs pour les champs de base de données peuvent aussi être écrits en utilisant des valeurs générées par la base de données. Cette option est disponible dans le dialogue Actions de table de base de données (voir <u>Paramètres d'actions de la table de base de données</u><sup>(30)</sup>) et est particulièrement utile lors de la génération de clés primaires.

## 4.2.2.11 Suggestions de remplissage automatique SQL

Lorsque vous saisissez des instructions SQL dans certains contextes, MapForce peut automatiquement suggérer des entrées de texte. L'auto-remplissage est disponible dans les contextes suivants :

Éditeur SQL (voir <u>Parcourir et interroger des bases de données</u><sup>382</sup>)

- Fenêtre "SQL personnalisé" dans le dialogue "Actions de table de base de données" (voir <u>Paramètres</u> d'actions de la table de base de données<sup>(330)</sup>)
- Dialogue "Saisir une instruction SELECT SQL" (voir <u>Créer des instructions SELECT</u><sup>363</sup>)



Remplissage automatique

Utiliser les touches de clavier **Haut** et **Bas** pour naviguer parmi la liste de suggestions. Pour choisir une entrée suggérée, cliquer dessus ou appuyer sur **Entrée**.

#### Pour désactiver les suggestions de remplissage automatique :

- 1. Dans le menu **Outils**, cliquer sur **Options** (ou appuyer sur **Ctrl+Alt+O**).
- 2. Sous Base de données, cliquer sur Éditeur SQL.
- 3. Sous Assistants à la saisie, décocher la case Ouvrir automatiquement.

Pour invoquer manuellement les suggestions de remplissage automatique comme et quand vous le souhaitez :

• Appuyer sur **Ctrl+Espace**.

# 4.2.2.12 Paramètres de composant de base de données

Une fois avoir ajouté un composant de base de données à la zone de mappage, vous pourrez configurer les paramètres applicables depuis le dialogue Paramètres de composant. Vous pouvez ouvrir le dialogue Paramètres de composant d'une des trois manières suivantes :

- Choisir le composant dans le mappage, puis, dans le menu Composant, cliquer sur Propriétés.
- Double-cliquer sur la barre de titre du composant.
- Cliquer avec la touche de droite sur le composant, puis cliquer sur Propriétés.

Component Setting	5	×	
Database Data source: Connection name: Database kind: Connection string:	Nanonull.sqlite Nanonull SQLite SQLite native connection to Database	Change Wanonull.sqlite	
Login settings User: Password: JDBC-specific settin JDBC driver: Database URL:	gs org.sqlite.JDBC jdbc:sqlite:Nanonull.sqlite	Generation settings         Strip schema names from table names         Timeout for statement execution         Timeout:       60         seconds       Infinite         Database transaction handling	
ADO/OLEDB-specifi Data Source: Catalog: Provider: Add. options:	Image: Settings       Nanonull.sqlite	✓ Use transactions         When an error occurs:       rollback top transaction and stop         Traces         Trace level:       Disabled         Trace file:       Browse	
Save all file paths relative to MFD file OK Cancel			

Fenêtre de dialogue Paramètres de composant de base de données

Les paramètres disponibles sont les suivants.

### Base de données

Ce groupe affiche des informations de connexion de base de données. Cliquer sur **Changer** pour sélectionner une base de données différente ou pour redéfinir les objets de la base de données dans le composant de base de données existant. Les connecteurs vers les tables portant le même nom seront retenus. Vous pouvez aussi changer les tables dans le composant, en cliquant avec la touche de droite sur un composant de base de données et en choisissant **Ajouter/Supprimer/Éditer des objets de base de données**.

Source de données	Spécifie le nom de la source de données actuelle. Pour les bases de données basées sur fichier, cela peut être un chemin dans le système du fichier.
	Utiliser ce paramètre pour déterminer si une base de données basée sur fichier a été ajoutée au mappage en utilisant un chemin absolu ou relatif. Dans le cas des chemins relatifs, "Source de données" contient un chemin ; dans les cas des chemins absolus, il contient uniquement le nom de fichier de la base de données.

Nom de connexion	Spécifie le nom de la connexion. Ce nom est généré automatiquement par MapForce. Généralement, il s'agit du même nom que celui du nom de source de données, mais il peut s'agir d'un nom d'"alias" si vous vous connectez à des <u>Resources</u> <u>Globales Altova</u> <sup>(971)</sup> . S'il y a plusieurs composants de base de données portant le même nom de connexion sur le mappage, le nom prendra la forme de " <connection1>", "<connection2>", etc.</connection2></connection1>
Type de base de données	Spécifie le type de la base de données.
String de connexion	Affiche le string de connexion de la base de données actuel. Ce champ en lecture seule est généré sur la base de l'information que vous fournissez lors de la création ou de la modification de la connexion de base de données.

### Paramètres de Login

Les paramètres de login sont utilisés pour toutes les cibles de génération de code et le moteur d'exécution intégré.

Utilisateur	Vous permet de changer de nom d'utilisateur pour vous connecter à la base de données. Obligatoire si la base de données nécessite un nom d'utilisateur pour pouvoir se connecter.
<i>Mot de passe</i>	Vous permet de changer le mot de passe vous connecter à la base de données. Obligatoire si la base de données nécessite un mot de passe pour pouvoir se connecter.

### Paramètres spécifiques à JDBC

Ces paramètres sont utilisés pour se connecter à la base de données si le mappage contient une connexion JDBC et est exécuté par un code Java généré ou par MapForce Server.

**Note :** Les connexions ADO, ADO.NET et ODBC sont converties en JDBC (et les paramètres JDBC cidessous s'appliquent) lorsque le mappage est exécuté sur un appareil Linux ou macOS, voir <u>Mappages de base de données dans plusieurs environnement d'exécution</u><sup>157</sup>.

Pilote JDBC	Affiche le pilote actuellement actif pour le composant de base de données. Le pilote par défaut est saisi automatiquement lorsque vous définissez le composant de base de données. Vous pouvez modifier le pilote saisi ici selon vos besoins. Assurez-vous que la syntaxe de l'entrée dans le champ URL de base de données est conforme au pilote spécifique que vous avez choisi.
URL de base de données	URL de la base de données actuellement sélectionnée. Assurez- vous que cette entrée soit conforme à la syntaxe du pilote JDBC du pilote spécifique saisi dans le champ du pilote JDBC.

### Paramètres spécifiques à ADO/OLEDB

Ces paramètres sont utilisés pour se connecter à la base de données si le mappage contient une connexion ADO et est exécuté par un code C# ou C++, généré ou par MapForce Server exécuté sous Windows, voir <u>Mappages de base de données dans plusieurs environnements d'exécution</u><sup>(157)</sup>. Les paramètres **Source de données** et **Catalogue** ne sont pas utilisés par le moteur d'exécution intégré.

Source de données	Affiche le nom de la source de données ADO.
Catalogue	Affiche le nom du catalogue ADO.
Fournisseur	Affiche le fournisseur actuellement actif pour le composant de base de données.
Options supplémentaires	Affiche toute option de base de données supplémentaire.

### Paramètres de Génération

Les paramètres de Génération s'appliquent à toutes les cibles de génération de code ainsi qu'à l'appareil d'exécution intégré.

Supprimer les noms de schéma des noms de table	Vous permet de supprimer des noms de schéma de base de données du code généré, et en ne retenant que les noms de table pour une plus grande flexibilité.
	Veuillez noter que cette option ne fonctionne que pour les instructions Select SQL générées par MapForce. Les instructions SQL définies par l'utilisateur, lors de la création de <u>tables</u> <u>virtuelles</u> , ne seront pas modifiées.
	Cette option est prise en charge pour la compatibilité inversée doit être évitée. Afin de traiter les noms d'objet de table de données en tant que relatives au schéma par défaut, utiliser l'approche décrite dans <u>Changer de bases de données et de</u> <u>schémas</u> <sup>257</sup> .

### Timeout pour exécution d'instruction

Lorsqu'une base de données est utilisée en tant que composant de cible, des timeouts d'exécution peuvent se produire en raison de la disponibilité du serveur, du trafic, des déclencheurs à longue durée, et d'autres facteurs. Ce paramètre vous permet de définir la longueur de la période de timeout avant que la connexion de base de données ne se ferme. Le paramètre prend effet lors de la requête des données de base de données et dans du code généré par C#, Java et C++.

Timeout	Définit la période en secondes, pendant laquelle le moteur d exécution doit attendre pour une réponse de la base de données avant d'abandonner l'exécution de l'instruction de la base de données. Le paramètre par défaut pour le timeout d'exécution est
	de 60 secondes.

Infini	Si elle est activée, cette option indique au moteur d'exécution de
	ne jamais effectuer de time out.

Note : Le timeout pour l'exécution d'instruction n'est pas applicable pour les bases de données SQLite.

#### Gestion de la transaction de base de données

Utiliser des transactions	Permet le processus de transaction lorsque vous utilisez une base de données en tant que cible. Le processus de transaction est activé pour toutes les tables du composant de base de
	données lorsque vous sélectionnez cette option. Pour plus d'informations, voir <u>Gérer les exceptions de base de données</u> <sup>423</sup> .

### Traçages

Niveau de traçage	Applicable pour les composants de base de données cible. Si le traçage est activé, les actions effectuées par le mappage par rapport à la base de données sont consignées dans un fichier de traçage. Vous pouvez choisir de journaliser toutes les actions, uniquement les erreurs ou désactiver le traçage complètement. Pour plus d'informations, voir <u>Traçage de base de données et journalisation d'erreur</u> <sup>435</sup> .
Fichier de traçage	Spécifie le fichier sur lequel les informations de traçage de base de données seront écrites lorsque le mappage est exécuté. Le fichier de traçage est en format XML, voir <u>Structure du fichier de</u> <u>traçage</u> <sup>433</sup> . Ce chemin peut être soit absolu soit relatif, et est influé par la case <b>Enregistrer tous les chemins relatifs au</b> <b>fichier MFD</b>

### Enregistrer tous les chemins de fichier relatifs au fichier MFD

Lorsque cette option est activée, MapForce enregistre les chemins de fichier affichés dans le dialogue Paramètres de composant relatifs à l'emplacement du fichier MapForce Design (.mfd). Utiliser des chemins relatifs si vous avez l'intention d'exécuter le mappage avec MapForce Server sur un système d'exploitation différent. Voir aussi <u>Utiliser des chemins relatifs sur un composant</u><sup>83</sup>.

# 4.2.3 Mapper des données dans des bases de données

Cette section fournit des instructions et exemples de transfert de données de toute source de mappage appuyée par MapForce (par exemple, un fichier XML) vers une base de données cible. Utiliser la feuille de route pour un résumé des options disponibles.

Je voudrais	Lire cette rubrique
Insérer des données dans une table de base de données cible basée sur les données fournis par le mappage	<ul> <li>Insérer des données dans une table<sup>286</sup></li> </ul>

Je voudrais	Lire cette rubrique
Contrôler comment les valeurs clé primaires seront créées	<ul> <li>Insérer des données dans une table<sup>286</sup></li> <li>Insérer des données dans plusieurs tables liées<sup>230</sup></li> </ul>
Exécuter une instruction SQL "préliminaire" à être exécutée avant qu'une table soit modifiée par le mappage (par exemple, supprimer tous les enregistrements dans la table, ou une instruction SQL personnelle)	<ul> <li>Insérer des données dans plusieurs tables liées</li> <li>Paramètres Actions de table de base de données</li> </ul>
Préserve les relations hiérarchiques des enregistrements dans des tables liés par des clés étrangères	Insérer des données dans plusieurs tables liées <sup>230</sup>
Mettre à jour une table de manière conditionnelle	<u>Mettre à jour une table</u> <sup>295</sup>
Fusionner les enregistrements dans une table de la base de données (mettre à jour des enregistrements, et aussi insérer certains autres enregistrements dans la même table), sur la base de la condition	<ul> <li><u>Action « Mettre à jour si Insérer Rest</u><sup>238</sup></li> <li><u>Instructions MERGE</u><sup>305</sup></li> </ul>
Préserver l'intégrité de base de données lors de la mise à jour des tables qui sont liées à d'autres tables par le biais de relations de clé étrangère	<ul> <li><u>Options pour tables enfant lors de la mise à jour</u> <u>d'une table parent</u><sup>307</sup></li> </ul>
Définir plusieurs actions par rapport à la même table (par exemple, supprimer un enregistrement si une condition est satisfaite, sinon insérer un nouvel enregistrement)	<ul> <li><u>"Supprimer si" Action<sup>315</sup></u></li> <li><u>« Ignorer si » Action</u><sup>319</sup></li> </ul>
Préserver l'intégrité des données en cas d'exécution de mappage échouée	<u>Gérer des exceptions de base de données</u> <sup>423</sup>
Insérer de nombreux enregistrements dans une table de base de données dans bulk (combiner de multiples instructions INSERT dans une requête)	Insertions Bulk (MapForce Server) <sup>323</sup>
Éviter des résultats involontaires lors du mappage de données dans des tables de base de données cible qui contiennent des valeurs null	<u>Gérer les Nulls dans les actions de table de base</u> <u>de données</u>
Consulter un exemple pour découvrir comment créer un mappage qui met à jour une base de données.	<u>Exemple : Mapper des données de XML vers</u> <u>SQLite</u>

# 4.2.3.1 Insérer des données dans une table

Un mappage peut insérer des données dans une table de base de données depuis un des composants de source pris en charge par MapForce, y compris d'autres bases de données. Vous pouvez configurer de manière très flexible comment la clé primaire des enregistrements insérés récemment doit être créée. Par exemple, la clé primaire peut être prélevée depuis le mappage, généré par la base de donnée ou calculée sur la base des valeurs de clé existante dans la table de base de données.

Cet exemple vous montre comment insérer de nouveaux enregistrements dans une table de base de données existante depuis un fichier XML. Vous allez aussi configurer comment générer la clé primaire. L'exemple utilise les fichiers suivants :

- altova-cmpy.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- altova.mdb la base de données cible à mettre à jour.

Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet vers les fichiers seront omis, pour des raisons de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

L'objectif du mappage est d'insérer des entreprises trouvées dans **Altova-cmpy.xml** en tant que nouveaux enregistrements dans la table "Altova" de la base de données **altova.mdb**. Si vous ouvrez le fichier XML de source, vous remarquerez qu'il contient une seule entreprise, appelée "Microtech OrgChart". C'est pourquoi, le mappage doit ajouter un nouvel enregistrement dans la table "Altova" portant le nom "Microtech OrgChart". De même, une nouvelle clé primaire doit être générée.

Pour obtenir l'objectif de mappage, nous allons suivre les étapes suivantes.

### Étape 1 : Insérer le composant XML source

• Dans le menu **Insérer**, cliquer sur **Schéma/Fichier XML**, et chercher **Altova\_Hierarchical.xsd**. Lorsque vous êtes invité à fournir un fichier d'instance, chercher **Altova-cmpy.xml**.

### Étape 2 : Insérer la base de données cible

Dans le menu Insérer, cliquer sur Database, et suivre les étapes d'assistant pour se connecter à altova.mdb (voir <u>Exemple : ajouter la base de données "altova.mdb" dans le mappage</u><sup>251</sup>).

### Étape 3 : Tirer les connexions

• Tirer les connexions de mappage comme indiqué ci-dessous.



**Note :** Si des connexions non désirées sont tracées automatiquement pour des items descendants, l'option "Connecter automatiquement des enfants" est active. Dans ce cas, pour défaire la dernière action, choisir l'option de menu **Édition | Annuler**. Pour désactiver l'option de connexion automatique, choisir l'option de menu **Connexion | Auto connexion des enfants correspondants**.

### Étape 4 : Configurer l'action Insérer

- Dans le composant cible, veuillez noter la touche Action: Insérer (<u>A:In</u>). Cette touche apparaît pour chaque table qui dispose d'une connexion depuis le mappage (dans ce cas, la table "Altova"). Cliquer sur cette touche pour configurer avec plus de détails l'action de base de données à exécuter (dans ce cas, l'action Insérer). Le dialogue des Actions de base de données de table apparaît.
- 2. Dans le dialogue des Actions de base de données de table, sous **Insérer tout**, à côté de **PrimaryKey**, choisir l'option **max() + 1**.

Action on record	NULL equal Insert All	•	~	Append Action
💵 PrimaryKey	max() + 1	<b>•</b>		
Name	mapped value	<b>_</b>	×	Insert Action
Delete data in child tables			~	
gnore input child data				
III Office				
			$\sim$	Delete Action

Les options disponibles dans cette liste auront les significations suivantes :

Option	Description
mapped value	Autorise les données de source à être mappées directement dans le champ de base de données et il s'agit du paramètre standard pour tous les champs de base de données. Il est également possible d'utiliser une procédure stockée pour fournir une valeur de clé en définissant une relation, voir <u>Utiliser des procédures stockées pour générer des clés</u> primaires <sup>419</sup> .
max() + 1	Génère les valeurs de clé basées sur les clés existantes dans la base de données. Par exemple, si la table comporte trois enregistrements, avec la clé primaire 1, 2 et 3, puis <b>max() + 1</b> est 4.
	Dans cet exemple, la table "Altova" a uniquement un enregistrement avec la clé primaire 1, donc <b>max() + 1</b> est 2, qui est la valeur attendue de la nouvelle clé primaire.
--------------	---
DB-generated	La base de données utilise la <b>fonction Identité</b> pour générer des valeurs clé.

L'option **valeur mappée** située à côté de "Name" signifie que cette colonne obtiendra la valeur directement depuis le mappage. For reference to other options available on the Database Table Actions dialog box, voir <u>Paramètres Actions de table de base de données</u><sup>(30)</sup>.

### Étape 5 : Consulter le mappage et mettre à jour la base de données

Cliquer sur l'onglet **Sortie** pour consulter le mappage. Un script SQL est généré, contenant les actions à exécuter par rapport à la base de données. Le script n'a pas encore modifié la base de données ; il n'est valable que pour la prévisualisation.

2 The following SQL statements are only for preview and may not be executed in another SQL query tool! 3 To execute these statements use function "Run SQL-script" from menu "Output". 4 Connect to database using the following connection-string: 5 Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Insert\attova.mdb;Provider=Microsoft.Jet.OLEDB.4.0 6 \*/ 7 8 SELECT IF(MAX([Altova].[PrimaryKey]) IS NULL,0,MAX([Altova].[PrimaryKey]))+1 AS [PrimaryKey] FROM [Altova] 9 -->>> %PrimaryKey1% 10 11 INSERT INTO [Altova] ([Name], [PrimaryKey]) VALUES ('Microtech OrgChart', '%PrimaryKey1%') 12

Pour exécuter le script par rapport à la base de données :

- Dans le menu Sortie, cliquer sur Exécuter script SQL.
- **Note :** L'exécution du script SQL directement depuis MapForce n'est qu'un des moyens mis à disposition pour mettre à jour la base de données, voir aussi <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

Pour voir le résultat, ouvrir la base de données **altova.mdb** dans DatabaseSpy ou Access. Veuillez noter qu'un nouvel enregistrement "Microtech OrgChart" a été ajouté à la table "Altova" avec la nouvelle clé primaire 2. Les données pour cet enregistrement provenaient de l'instance XML d'entrée.

			Altova		_		×			
2		PrimaryKey 👻	Name	-						
	÷	1	Organization Chart							
	÷	2	Microtech OrgChart							
*										
Record: H 4 3 of 3										

Vous avez à présent terminé de créer un mappage qui insère des données dans une table de base de données. Pour consulter un exemple de mappage de données qui insère les données aussi bien dans la table actuelle que dans une table enfant dépendante, voir <u>Insérer des données dans plusieurs tables liées</u><sup>(23)</sup>.

## 4.2.3.2 Insérer des données dans plusieurs tables liées

Une table de base de données peut être une table "parent" ; c'est à dire, elle peut être référée par d'autres tables dans la base de données par le biais des relations de clé étrangères. Dans ce type de scénarios, vous pouvez configurer le mappage pour insérer des enregistrements non seulement dans la table parent, mais aussi dans des tables enfant dépendantes. Par exemple, lors de l'insertion d'un nouvel enregistrement "company" dans la table de base de données, vous pouvez aussi insérer des enregistrements pour des bureaux liés à cette entreprise, ainsi que pour leurs départements, personnes, etc. enfants.

Cet exemple vous montre comment insérer des données dans des tables tout en préservant les relations de la base de données. Il s'agit d'une version un peu plus élaborée de l'exemple précédent, <u>Insérer des données</u> <u>dans une table</u><sup>286</sup>. L'exemple est accompagné par un échantillon de mappage, et utilise les fichiers suivants :

- Altova\_Hierarchical.mfd le fichier de mappage.
- Altova\_Hierarchical.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- AltovaTarget.mdb la base de données à mettre à jour.

Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet vers les fichiers sera omis, par mesure de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

L'objectif du mappage est de remplacer des données dans la base de données cible (AltovaTarget.mdb) avec des données provenant d'un fichier XML de source. La structure de fichier XML correspond à peu près à la structure hiérarchique des tables dans la base de données. Il s'agit d'un graphique organisationnel et structuré comme suit : l'élément supérieur est une entreprise qui contient deux bureaux. Chaque bureau contient des départements et chaque département contient des personnes. La même hiérarchie existe dans

**AltovaTarget.mdb**, où la table "Altova" correspond à l'entreprise. Cette table est liée, par le biais de relations de clé étrangères, aux enregistrements dans la table "Office". De même, les liens "Office" vers "Department", et les liens "Department" vers "Person". Pour consulter un diagramme de relation de la base de données **AltovaTarget.mdb**, l'ouvrir dans le mode "Relationships" d'Access (voir aussi <u>Gérer des relations de base de données</u>).

Pour aboutir à l'objectif de mappage, nous allons suivre les étapes ci-dessous.

### Étape 1: Insérer le composant XML de source

• Dans le menu **Insérer**, cliquer sur **Schéma XML/Fichier**, et chercher **Altova\_Hierarchical.xsd**. Lorsque vous êtes invité à fournir un fichier d'instance, chercher **Altova\_Hierarchical.xml**.

### Étape 2: Insérer la base de données cible

Dans le menu Insérer, cliquer sur Base de données, et suivre les étapes de l'assistant pour vous connecter à AltovaTarget.mdb. Les instructions pour se connecter à cette base de données sont les mêmes que pour altova.mdb (voir Exemple : ajouter la base de données "altova.mdb" dans le mappage<sup>[25]</sup>).

### Étape 3: Tirer les connexions

• Tirer les connexions de mappage comme affiché ci-dessous. Veuillez noter que les clés primaires et étrangères ne sont pas mappées ; elles seront générées immédiatement, comme affiché ci-dessous.



**Note :** Si des connexions non désirées sont tracées automatiquement pour des items descendants, l'option "Connecter automatiquement des enfants" est active. Dans ce cas, pour défaire la dernière action, choisir l'option de menu **Édition | Annuler**. Pour désactiver l'option de connexion automatique, choisir l'option de menu **Connexion | Auto connexion des enfants correspondants**.

## Étape 4: Configurer les actions Insérer

Dans le composant cible, cliquer sur la touche Action: Insérer ( A:In ) à côté de la table "Altova" et configurer le paramètre max() + 1 de la clé primaire comme affiché ci-dessous. Ce paramètre a été expliqué de manière plus détaillée dans l'exemple précédent, voir Insérer des données dans une table 283.

Database Table Actions - Altova X								
SQL statement to execute befo	re first record also delete all rec	cords in all child ta	bles					
Actions to execute for each rec All input data are compared to Select the "NULL equal" check If all comparisons are true, th	ord the DB table data box to treat NULL en the specific act	a, using the opera , values as equal, tion is executed.	ators defined here.					
Action on record	NULL equal	Insert All	<u> </u>	<u>^</u>	Append Action			
		max() + 1	<u>•</u>		Insert Action			
Delete data in child tables Ignore input child data				~	Inservacion			
				~	Delete Action			

De plus, veuillez noter que l'option **SUPPRIMER tous les enregistrements** est activée. Cela permet de supprimer tous les enregistrements existants depuis la table, avant que de nouveaux enregistrements soient saisis, ce qui constitue le comportement désiré dans cet exemple. Si vous désactivez cette option, de nouveaux enregistrements (avec une clé primaire) seront ajoutés à la base de données en plus de celles qui existent, à chaque fois que vous exécutez le mappage, ce qui n'est pas le comportement désiré.

Pour le besoin de cet exemple, l'option **supprime aussi tous les enregistrements des tables enfants** est aussi activée. Cela garantit que non seulement les enregistrements depuis la table "Altova" sont supprimés, mais aussi tous les enregistrements, dans les tables qui sont liés à la table "Altova" par le biais de la relation de la clé étrangère. Si les tables enfant possèdent leur propres tables enfants, ceux-ci seront aussi supprimés, etc., jusque la dernière table dans l'arborescence de dépendance. Si vous tentez de supprimer uniquement des enregistrements depuis la table racine "Altova", cela violerait l'intégrité de la base de données, et l'exécution du mappage échouerait.

Pour des références à d'autres options disponibles dans le dialogue Actions de la table de base de données, voir <u>Paramètres Actions de table de base de données</u><sup>330</sup>.

- 2. Cliquer sur **OK** pour fermer le dialogue. Veuillez noter que, dans la surface de mappage, l'apparence de la touche a changé en <u>DEL, A:n</u>. Cela indique qu'une instruction "Supprimer" est configurée pour avoir lieu avant l'action "Insérer".
- 3. Cliquer sur la touche A:In située à côté de la table "Office" et configurer le paramètre **max() + 1** de la clé primaire.

😢 Database Table Actions - Office								
Actions to execute for each record All input data are compared to the DB table dat Select the "NULL equal" check box to treat NULL If all comparisons are true, then the specific ac	a, using the oper L values as equal tion is executed.	ators	s defined here.					
Action on record NULL equal	Insert All max() + 1 foreign key mapped value mapped value mapped value mapped value mapped value mapped value		~	Append Action				
Delete data in child tables Ignore input child data Address Department			∧	Delete Action				

4. Effectuer l'étape 3 pour chaque table descendant depuis "Office", concrètement : "Address", "Department", et "Person". Veuillez vous assurer que toutes ces tables sont immédiatement descendantes de la table de racine "Altova". Pour une explication de ce qui est une table "root" et pourquoi elle est nécessaire, voir <u>Gérer des relations de base de données</u><sup>260</sup>.

### Étape 5: Consulter le mappage et la mise à jour de la base de données

Cliquer sur l'onglet **Sortie** pour consulter le mappage. Un script SQL est généré, contenant les actions à exécuter par rapport à la base de données. Le script n'a pas encore modifié la base de données, elle est uniquement destinée à la prévisualisation.

Pour exécuter le script par rapport à la base de données :

- Dans le menu Sortie, cliquer sur Exécuter Script SQL.
- **Note :** L'exécution du script SQL directement depuis MapForce est juste un des moyens mis à disposition pour mettre à jour la base de données, voir aussi <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

Pour voir le résultat, ouvrir la table "Altova" dans Microsoft Access, et observer comment les relations depuis le fichier XML ont maintenant été propagées dans la base de données, depuis la table "Altova" jusqu'à la table "Person".



Vous avez maintenant terminé la création d'un mappage qui insère des données dans plusieurs tables de base de données, tout en préservant les relations d'intégrité de la table.

# 4.2.3.3 Mettre à jour une table

Cet exemple vous montre comment mettre à jour les données d'une table de base de données existante avec des données provenant d'une source XML. L'exemple utilise les fichiers suivants :

- altova-cmpy.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- altova.mdb la base de données cible à mettre à jour.

Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet y menant sera omis, pour des raisons de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

L'objectif du mappage est de mettre à jour tous les enregistrements dans la table "Person" avec des instances de "Person" depuis le document XML. Chaque personne dans le fichier XML a un élément enfant PrimaryKey. Chaque personne dans la table "Person" a une colonne PrimaryKey. Seuls les enregistrements où une PrimaryKey d'une personne dans le fichier XML correspond à la PrimaryKey d'une personne dans la base de données doivent être mis à jour.

Pour atteindre l'objectif de mappage, nous allons suivre les étapes suivantes.

### Étape 1: Insérer le composant XML source

• Dans le menu **Insérer**, cliquer sur **Schéma XML/Fichier**, et chercher **Altova\_Hierarchical.xsd**. Lorsque vous êtes invité à fournir un fichier d'instance, chercher **altova-cmpy.xml**.

### Étape 2: Insérer la base de données cible

Dans le menu Insérer, cliquer sur Base de données, et suivre les étapes de l'assistant pour vous connecter à altova.mdb (voir <u>Exemple : Ajouter la base de données "altova.mdb" dans le mappage</u><sup>(251)</sup>).

### Étape 3: Tirer les connexions

• Tirer les connexions de mappage comme affiché ci-dessous.



## Étape 4: Configurer l'action de Mise à jour

1. Dans le composant, cliquer sur la touche Action: Insérer ( A:In ) à côté de la table "Person".

- 3. À côté de **PrimaryKey**, choisir la valeur **equal**, comme affiché ci-dessous. Cela définit la condition de mise à jour : c'est à dire que l'enregistrement de la base de données sera uniquement mise à jour lorsque sa valeur **PrimaryKey** est égale à la valeur **PrimaryKey** provenant du mappage.

Actions to execute for each record									
All input data are compared	All input data are compared to the DB table data, using the operators defined here.								
Select the "NULL equal" che If all comparisons are true.	eck box to treat NUL then the specific ac	L values as equi tion is executed	al.						
Thair companisons are a de,	alen ale specifie de								
Action on record	NULL equal	Update if	<b>.</b>		~	Append Action			
🛙 🕅 PrimaryKey		equal	<b>T</b>						
🛙 ForeignKey			-			Insert Action			
EMail			-						
First			-						
🔲 Last			-						
PhoneExt			-						
I Title			-						
						Delete Astien			
					×	Delete Action			

Dans cet exemple, l'opérateur d'égalité est appliqué au champ **PrimaryKey**, ce qui constitue un scénario probable lors d'une mise à jours de bases de données. Veuillez noter que des conditions peuvent aussi être définies dans d'autres champs qui ne sont pas nécessairement des clés primaires. Par exemple, en choisissant **equal** à côté des champs **First** et **Last**, vous ne mettriez à jour que les enregistrements pour lesquels le prénom et le nom sont égaux à ceux dans le XML source.

En option, cocher la case **NULL equal**. Dans cet exemple, cette case à cocher n'est pas pertinente car ni le champ **PrimaryKey** se trouvant dans le XML de source ni le champ **PrimaryKey** dans la base de données ne contient de valeurs null. Néanmoins, vous devriez cocher cette case si vos données contiennent des valeurs null, et si vous souhaitez traiter ces valeurs comme égales ; sinon, vous risquez d'obtenir des résultats indésirables. Pour plus d'informations, voir <u>Gérer des valeurs null dans les actions de table de base de</u> données<sup>327</sup>.

4. Cliquer sur OK pour fermer le dialogue. Veuillez noter que, une fois que vous serez retourné au mappage, la touche Action: Insérer A:In s'est transformée en une touche Action: Mise à jour ( A:Up ). Cela indique qu'une action de mise à jour est configurée pour se avoir lieu pour cette table.

## Étape 5: Consulter le mappage et mettre à jour la base de données

Cliquer sur l'onglet **Sortie** pour consulter le mappage. Un script SQL est généré, contenant des actions à exécuter par rapport à la base de données. Le script n'a pas encore modifié la base de données ; elle est uniquement destinée à la prévisualisation.

1	/*
2	The following SQL statements were executed during "Generate output" function.
3	Every single result is written right to the ">>>" string.
4	These statements are only for preview and may not be executed in another SQL query too!
5	The database was connected using the following connection-string:
6	Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Update
	ifvaltova.mdb;Provider=Microsoft.Jet.OLEDB.4.0
7	*/
8	
9	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'A.Aldrich@microtech.com', [First] = 'Albert', [Last] = 'Aldrich',
	[PhoneExt] = 582, [Title] = 'Manager' WHERE ([Person].[PrimaryKey] = 1)
10	>> OK. 1 row(s).
11	
12	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'b.bander@microtech.com', [First] = 'Bert', [Last] = 'Bander',
	[PhoneExt] = 471, [Title] = 'Accounts Receivable' WHERE ([Person].[PrimaryKey] = 2)
13	>>> OK. 1 row(s).
14	
15	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.clovis@microtech.com', [First] = 'Clive', [Last] = 'Clovis',
	[PhoneExt] = 963, [Title] = 'Accounting Manager' WHERE ([Person].[PrimaryKey] = 3)
16	>>> OK. 1 row(s).
17	
18	UPDATE [Person] SET [ForeignKey] = 2, [EMail] = 'd.Durnell@microtech.com', [First] = 'Dave', [Last] = 'Durnell',
	[PhoneExt] = 621, [Title] = 'Marketing Manager Europe' WHERE ([Person].[PrimaryKey] = 4)
19	>>> OK. 1 row(s).

Pour exécuter le script par rapport à la base de données :

- Dans le menu Sortie, cliquer sur Exécuter Script SQL
- **Note :** L'exécution du script SQL directement depuis MapForce est juste un des moyens mis à disposition pour mettre à jour la base de données, voir aussi <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

# 4.2.3.4 Action "Mettre à jour si... Insérer Rest"

Parfois, il est nécessaire non seulement de mettre à jour des enregistrements existants, mais aussi d'insérer de nouveaux enregistrements dans la même table de base de données. Dans ces cas, MapForce propose une action "Mettre à jour si... Insérer Rest". Elle fonctionne comme suit :

- Si la condition **Mettre à jour si** est vraie, alors l'enregistrement de base de données existant est mis à jour avec les données provenant du mappage.
- Si la condition **Mettre à jour si** est fausse, et qu'une condition **Insérer Rest** existe, alors un nouvel enregistrement est inséré.
- Si des enregistrements existent dans la base de données sans contre-partie dans le fichier de source, alors ces enregistrements restent inchangés.

### Note / MariaDB ODBC

Si la base de données cible est MySQL ou MariaDB through ODBC, l'option **Retourner des lignes correspondantes au lieu des lignes affectées** doit être activée dans l'onglet **Cursor/Results** de MySQL ODBC Connector. En alternative, si vous saisissez le string de connexion manuellement par le biais de l'assistant Connexion de base de données, ajouter Option=2 au string de connexion, par exemple : Dsn=mydsn;Option=2;

Pour activer cette option depuis le Connecteur ODBC MySQL :

- 1. Appuyer sur la touche Windows et commencer à saisir "ODBC".
- Exécuter l'Administrateur de sources de données ODBC (soit 32-bit soit 64-bit, selon la plateforme du Connecteur ODBC MySQL installé).
- 3. Cliquer sur la Data Source Name (DSN) utilisée par le mappage MapForce, puis cliquer sur **Configurer**.

S ODBC I	Data Source A	dministra	tor (32-bi	t)						×
User DSN	System DSN	File DSN	Drivers	Tracing	Connection	Pooling	About			
<u>U</u> ser Data	Sources:									
Name		Platform	Driver					A <u>d</u> d.		
Excel Fi MS Acc	Excel Files     32-bit     Microsoft Excel Driver (*xls, *xlsx, *xlsm, MS Access Database       MS Access Database     32-bit     Microsoft Access Driver (*.mdb, *.accdb)       32-bit     32-bit     Microsoft Access Driver (*.mdb, *.accdb)					, *xlsm, * accdb)	.xlsb)	<u>R</u> emo	ve	
mysqr_d	SIT	32-01	MySQLOI	000 0.0 0	nicode Dilvei			<u>C</u> onfigu	re	
<							>			
	An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.									
					OK	Cano	cel	Apply	Help	

En option, cocher la case **NULL equal**. Dans cet exemple, cette case à cocher n'est pas pertinente car ni le champ **PrimaryKey** se trouvant dans le XML de source ni le champ **PrimaryKey** dans la base de données ne contient de valeurs null. Néanmoins, vous devriez cocher cette case si vos données contiennent des valeurs null, et si vous souhaitez traiter ces valeurs comme égales ; sinon, vous risquez d'obtenir des résultats indésirables. Pour plus d'informations, voir <u>Gérer des valeurs null dans les actions de table de base de</u> <u>données</u><sup>(27)</sup>.

- 4. Cliquer sur **Détails >>** pour rendre les options avancées disponibles.
- 5. Cliquer sur l'onglet **Curseurs/Résultats**, et puis sélectionner la case à cocher **Retourner lignes** correspondantes au lieu des lignes touchées.

### Exemple

L'exemple suivant vous montre comment fusionner (aussi bien Mettre à jour qu'Insérer) des données depuis une source XML dans une table de base de données. L'exemple utilise les fichiers suivants :

- altova-cmpy-extra.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- altova.mdb la base de données cible à mettre à jour.

#### Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet y menant sera omis, pour des raisons de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

L'objectif du mappage est de joindre tous les enregistrements provenant d'un document XML de source dans une table cible "Person". Concrètement, pour chaque enregistrement dans les données XML de source, le mappage doit effectuer les choses suivantes :

- Si la PrimaryKey de la personne dans le fichier XML correspond à une PrimaryKey de personne dans la base de données, alors mettre à jour l'enregistrement.
- Tout enregistrement existant dans la table Person qui ne satisfait pas à la condition ci-dessus ne doit pas être touché.
- Si la PrimaryKey de la personne dans le fichier XML ne dispose pas d'une correspondance dans la table de base de données de cible, alors ajouter un nouvel enregistrement à la table de la base de données.

Pour atteindre l'objectif de mappage, nous allons suivre les étapes suivantes.

### Étape 1: Insérer le composant XML de source

• Dans le menu **Insérer**, cliquer sur **Schéma XML/Fichier**, et chercher **Altova\_Hierarchical.xsd**. Lorsque vous êtes invité à fournir un fichier d'instance, chercher **altova-cmpy-extra.xml**.

### Étape 2: Insérer la base de données cible

Dans le menu Insérer, cliquer sur Base de données, et suivre les étapes de l'assistant pour vous connecter à altova.mdb (voir <u>Exemple : Ajouter la base de données "altova.mdb" dans le mappage</u><sup>(251)</sup>).

### Étape 3: Tirer les connexions

• Tirer les connexions de mappage comme affiché ci-dessous.



## Étape 4: Configurer les actions "Mettre à jour si... Insérer Rest"

- 1. Dans le composant de cible, cliquer sur la touche **Action: Insérer** (<u>A:In</u>) située à côté de la table "Person".
- 2. À côté de **Action sur enregistrement**, choisir **Mettre à jour si...** Cela change l'action de table de la base de données en une action de mise à jour conditionnelle. C'est à dire que l'enregistrement actuel sera uniquement mis à jour lorsqu'une condition sera satisfaite (voir étape suivante).
- 3. À côté de **PrimaryKey**, choisir la valeur **equal**, comme indiqué ci-dessous. Cela définit la condition de mise à jour : c'est à dire, l'enregistrement de la base de données sera mise à jour uniquement lorsque sa valeur **PrimaryKey** sera égale à la valeur **PrimaryKey** provenant du mappage.

Action on record	NULL equal	Update if	•
🗊 PrimaryKey		equal	•
🗊 ForeignKey			•
III EMail			•
First			•
Last			•
PhoneExt			•
Title			•

4. Cliquer sur **Apposer action**. Cela permet d'ajouter une nouvelle action à la droite de l'action existante **Mettre à jour si**. Configurer la nouvelle action en tant que **Insérer Rest :** 

ll input data are compare	d to the DB table dat	ta, using the ope	erator	s defined here.			
elect the "NULL equal" ch	eck box to treat NUL	L values as equ	al.				
fall comparisons are true,	then the specific ac	tion is executed					
Action on record	NULL equa	Update if	T	Insert Rest	•	~	Append Action
💵 PrimaryKey		equal	•	mapped value	-		
🕼 ForeignKey			•	mapped value	•		Insert Action
💷 EMail			•	mapped value	-		
First			•	mapped value	•		
🔲 Last			•	mapped value	-		
PhoneExt			•	mapped value	-		
Title			•	mapped value	•		

Dans l'image ci-dessus, les actions de la table de base de données ont été configurées conformément avec les objectifs du mappage. C'est à dire que seule si la condition **Mettre à jour si**... est satisfaite, l'enregistrement sera mis à jour ; sinon, il sera inséré. L'option "valeur mappée" spécifie que des valeurs provenant du mappage seront utilisées pour remplir tous les champs de l'enregistrement.

Il est également possible de définir plus de deux actions par rapport à la même table de base de données (pas nécessairement dans cet exemple néanmoins). Au moment du mappage, des actions sont exécutées de gauche à droite. La dernière action **Insérer** est considérée être finale ; tout autre action ajoutée ultérieurement sera ignorée.

Veuillez noter que la touche **Apposer Action** dans le dialogue ajoute la nouvelle action *après* l'action sélectionnée. L'**Action Insérer** ajoute la nouvelle action *avant* l'action sélectionnée. Pour supprimer une action existante, cliquer n'importe où à l'intérieur puis cliquer sur **Action Supprimer**.

5. Cliquer sur OK pour fermer le dialogue. Veuillez noter que, de retour au mappage, la touche Action: Insérer <u>A:In</u> est maintenant transformée en une touche Action: Mettre à jour; Insérer (<u>A:Up,In</u>). Cela indique que l'action de mise à jour et d'insertion est configurée pour avoir lieu pour cette table.

## Étape 5: Consulter le mappage et mettre à jour la base de données

Cliquer sur l'onglet **Sortie** pour consulter le mappage. Un script SQL est généré, contenant des actions à exécuter par rapport à la base de données. Le script n'a pas encore modifié la base de données ; elle est uniquement destinée à la prévisualisation.

1	
2	The following SQL statements are only for preview and may not be executed in another SQL query too!
3	To execute these statements use function "Run SQL-script" from menu "Output".
4	Connect to database using the following connection-string:
5	Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Update if Insert
	Rest\altova.mdb;Provider=Microsoft.Jet.OLEDB.4.0
6	*/
7	
8	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'A.Aldrich@microtech.com', [First] = 'Albert', [Last] = 'Aldrich',
	[PhoneExt] = 582, [Title] = 'Manager' WHERE ([Person].[PrimaryKey] = 1)
9	
10	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'b.bander@microtech.com', [First] = 'Bert', [Last] = 'Bander',
	[PhoneExt] = 471, [Title] = 'Accounts Receivable' WHERE ([Person].[PrimaryKey] = 2)
11	
12	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.clovis@microtech.com', [First] = 'Clive', [Last] = 'Clovis',
	[PhoneExt] = 963, [Title] = 'Accounting Manager' WHERE ([Person].[PrimaryKey] = 3)
13	
14	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.Cicada@microtech.com', [First] = 'Camilla ', [Last] = 'Cicada',
	[PhoneExt] = 765, [Title] = 'HR' WHERE ([Person].[PrimaryKey] = 30)

Script SQL (vue partielle) avant de mettre à jour la base de données

Vous noterez qu'aucune instruction INSERT n'est visible dans le script de prévisualisation. Il s'agit d'un comportement normal, car les enregistrements sont insérés conditionnellement et les instructions INSERT dépendent du résultat de l'action **Mettre à jour si** (ce qui n'est pas connu avant que le mappage soit exécuté).

Note : Pour certains types de base de données, MapForce crée des instructions MERGE au lieu des instructions de UPDATE. Pour plus d'informations, voir <u>instructions MERGE</u><sup>305</sup>.

Pour exécuter le script par rapport à la base de données :

• Dans le menu Sortie, cliquer sur Exécuter Script SQL.

Maintenant que le mappage a été exécuté et que le script est appliqué par rapport à la base de données, veuillez noter que les instructions INSERT sont visibles dans l'onglet **Sortie**.

1	1*
2	The following SQL statements were executed during "Generate output" function.
3	Every single result is written right to the ">>>" string.
4	These statements are only for preview and may not be executed in another SQL query too!
5	The database was connected using the following connection-string:
6	Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Update if Insert
	Rest\altova.mdb;Provider=Microsoft.Jet.OLEDB.4.0
7	*/
8	
9	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'A.Aldrich@microtech.com', [First] = 'Albert', [Last] = 'Aldrich',
	[PhoneExt] = 582, [Title] = 'Manager' WHERE ([Person].[PrimaryKey] = 1)
10	>>> OK. 1 row(s).
11	
12	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'b.bander@microtech.com', [First] = 'Bert', [Last] = 'Bander', [PhoneExt]
	= 471, [Title] = 'Accounts Receivable' WHERE ([Person].[PrimaryKey] = 2)
13	>>> OK. 1 row(s).
14	
15	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.clovis@microtech.com', [First] = 'Clive', [Last] = 'Clovis', [PhoneExt] = 963. [Title] = 'Accounting Manager' WHERE ([Person] [PrimaryKey] = 3)
16	->>> OK 1 row(s)
17	
18	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.Cicada@microtech.com', [First] = 'Camilla ', [Last] = 'Cicada',
	[PhoneExt] = 765, [Title] = 'HR' WHERE ([Person].[PrimaryKey] = 30)
19	>>> OK. 0 row(s).
20	
21	INSERT INTO [Person] ([PrimaryKey], [ForeignKey], [EMail], [First], [Last], [PhoneExt], [Title]) VALUES (30, 1,
	'c.Cicada@microtech.com', 'Camilla ', 'Cicada', 765, 'HR')
22	>>> OK. 1 row(s).

Script SQL (vue partielle) après avoir mis à jour la base de données

**Note :** L'exécution du script SQL directement depuis MapForce est juste un des moyens mis à disposition pour mettre à jour la base de données, voir aussi <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

Si vous ouvrez la table "Person" dans l'onglet de requête de BD de MapForce (voir <u>Parcourir et interroger des</u> bases de données<sup>(32)</sup>), vous pouvez voir le résultat du mappage comme suit :

- Tous les enregistrements de base de données qui avaient des clés primaires correspondantes dans le fichier XML ont été mis à jour. Dans l'exemple, des enregistrements avec la clé primaire 1, 2, 3, 4 et 5.
- Tous les enregistrements de base de données qui n'ont pas de clés correspondantes dans le fichier XML restent non-affectés. Dans l'exemple, des enregistrements avec la clé primaire 6, 7, 8 et 9.
- De nouveaux enregistrements ont été insérés dans la table "Person" (pour lesquels la clé n'existe pas encore dans la base de données). Dans l'exemple, des enregistrements avec la clé primaire 30 et 31.

	PrimaryKey •	ForeignKey •	EMail •	First •	Last •	PhoneExt •	Title •
1	1	1	A.Aldrich@microtech.com	Albert	Aldrich	582	Manager
2	2	1	b.bander@microtech.com	Bert	Bander	471	Accounts Receivable
3	3	1	c.clovis@microtech.com	Clive	Clovis	963	Accounting Manager
4	4	2	d.Durnell@microtech.com	Dave	Durnell	621	Marketing Manager Europe
5	5	2	e.ellas@microtech.com	Eve	Ellas	753	Art Director
6	6	3	f.landis@nanonull.com	Fred	Landis	951	Program Manager
7	7	3	m.landis@nanonull.com	Michelle	Butler	654	Software Engineer
8	8	3	t.little@nanonull.com	Ted	Little	852	Software Engineer
9	9	3	a.way@nanonull.com	Ann	Way	951	Technical Writer
10	10	3	l.gardner@nanonull.com	Liz	Gardner	753	Software Engineer
11	11	3	p.smith@nanonull.com	Paul	Smith	334	Software Engineer
12	12	4	a.martin@nanonull.com	Alex	Martin	778	IT Manager
13	13	4	g.hammer@nanonull.com	George	Hammer	223	Web Developer
14	14	4	n.newbury@microtech.com	Nira	Newbury	241	Support Engineer
15	15	4	o.origone@microtech.com	Olanda	Origone	345	Support Engineer
16	16	5	s.meier@microtech.com	Paul	Ponetti	114	Office Manager
17	17	5	t.bone@microtech.com	Quang	Qui	331	Accounts Receivable
18	18	6	m.nafta@microtech.com	Rita	Rotter	122	PR & Marketing Manager US
19	19	7	v.bass@microtech.com	Silla	Semua	716	IT Manager
20	20	7	c.franken@microtech.com	Tom	Tinker	147	Support Engineer
21	21	7	m.redgreen@microtech.com	Uri	Ugoner	152	Support Engineer
22	30	1	c.Cicada@microtech.com	Camilla	Cicada	765	HR
23	31	1	c.corrigan@microtech.com	Carol	Corrigan	629	Admin

La table "Person" après mettre à jour la base de données

# 4.2.3.5 Instructions MERGE

Pour certains mappages qui mettent à jour et insèrent des données dans une table de base de données (voir aussi <u>Action "Mettre à jour si... Insérer Rest"</u><sup>233</sup>), MapForce génère des instructions MERGE à exécuter par rapport à la base de données au moment de la marche du mappage. Le moteur d'exécution n'est pas forcément MapForce, voir <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

Les instructions MERGE sont pris en charge par les types de base de données suivantes :

- Serveur SQL 2008 et plus
- Oracle
- DB2
- Firebird

Les instructions MERGE réduisent le nombre d'appels de serveur de base de données, puisqu'elles combine les instructions INSERT et UPDATE en un. De même, en cas d'instructions MERGE, la vérification de cohérence est effectuée par la base de données.

Pour voir si le mappage exécutera des instructions MERGE de base de données lors de la marche (contrairement à l'application d'une combinaison des instructions INSERT et UPDATE) :

- 1. Créer un mappage qui utilise une action **Mettre à jour si...** ainsi que **Insérer Rest**. Pour un exemple, voir <u>Action "Mettre à jour si... Insérer Rest"</u><sup>[238]</sup>.
- 2. Consulter le mappage, en cliquant l'onglet Sortie.

Si MERGE est pris en charge par le type de base de données, le script SQL généré contient des instructions MERGE, par exemple :

1	/-	-
2	The following SQL statements are only for preview and may not be executed in another SQL query tool!	_
3	To execute these statements use function "Run SQL-script" from menu "Output".	
4	Connect to database using the following connection-string:	
5	Data Source= ;Initial	
	Catalog=DB MERGE;MultipleActiveResultSets=True;Password=•••••;Persist Security Info=True;User	
	ID=	
6 7	*/	
8	SET QUOTED IDENTIFIER ON	
9		
10	UPDATE [dbo].[Users] SET [FirstName] = (CAST('Despine' AS nvarchar(50))), [LastName] =	
	<pre>(CAST('Buttler' AS nvarchar(50))) WHERE ([dbo].[Users].[UserID] = 1)</pre>	
11		
12	SELECT [UserID] FROM [dbo].[Users] WHERE ([UserID] = 1)	
13	>>> %UserID1%	
14		
15	LELE FROM [doo].[Addresses] WHERE EXISIS(SELECT * FROM [doo].[Users] WHERE [doo].[Users].[UserID] =	
16	[db0].[Addresses].[oserib] AND ([db0].[oserib].[oserib] = %oseribi%))	
17	MERGE INTO [dbo].[Addresses] AS I USING ( VALUES ( '%UserID1%', 1, 1, (CAST('Home' AS pyarchar(20))).	
	(CAST('Louisville' AS nvarchar(50))). (CAST('Elm Street' AS nvarchar(50))). (CAST('12' AS	
	<pre>nvarchar(20))) ) AS 5 ( [UserID], [IsShipping], [IsBilling], [AddressType], [City], [Street],</pre>	
	[Number] ) ON ( (S.[UserID] = T.[UserID] ) WHEN MATCHED THEN UPDATE SET [ISShipping] = S.	
	[IsShipping], [IsBilling] = S.[IsBilling], [AddressType] = S.[AddressType], [City] = S.[City],	
	[Street] = S.[Street], [Number] = S.[Number] WHEN NOT MATCHED THEN INSERT ( [UserID], [IsShipping],	
	[IsBilling], [AddressType], [City], [Street], [Number] ) VALUES ( S.[UserID], S.[IsShipping], S.	
	[IsBilling], S.[AddressType], S.[City], S.[Street], S.[Number] );	
18		
		•
Mappir	ng DB Query Output	
		_

Si MERGE n'est pas pris en charge par le type de base de données, le script SQL généré contient uniquement des instructions UPDATE. Aucune instruction INSERT n'est visible pour une préconsultation, puisque ceux-ci doivent être exécutés uniquement si la condition **Mettre à jour si...** n'est pas satisfaite (et cela n'est pas connue avant l'exécution de mappage).

### Notes

- MapForce crée automatiquement des instructions MERGE lorsqu'il détecte un type de base de données pris en charge ; il n'est pas possible d'influer manuellement si MapForce doit créer une instruction MERGE.
- Si vous mettez à jour plusieurs tables ayant des relations parent-enfant, les fusions sont créée uniquement pour les tables "leaf". Une table "leaf" est la table enfant la plus basse qui est mappée.
   Par exemple, dans le mappage ci-dessous, les actions Update If... Insert Rest... ont été définies pour

la table Users et la table Addresses. Néanmoins, les instructions MERGE seront uniquement générées pour la table leaf, Addresses. La table parent, Users, recevra les instructions UPDATE ou INSERT au lieu de MERGE.



 avec des instructions MERGE, l'option "Bulk Transfer" (voir <u>Insertions Bulk (MapForce Server</u>)<sup>623</sup> est prise en charge uniquement pour des connexions de base de données ODBC et JDBC.

## 4.2.3.6 Options pour tables enfants lors de la mise à jour d'une table parent

Lorsque le mappage met à jour une table qui est une table "parent" (c'est à dire qu'elle a des relations étrangères avec d'autres tables), vous pouvez configurer comment les enregistrements dépendants doivent être traités aussi bien dans les données de source que la table de cible. Par exemple, partons du principe que vous souhaitez mettre à jour la table "Department" dans la base de données **altova.mdb**. Étant donné que chaque personne est liée à un département par le biais d'une clé étrangère, vous allez probablement souhaiter prendre aussi une action envers la table "Person" (qui pourrait être insertion, mise à jour ou supprimer). Ainsi, vous contribuez à conserver l'intégrité de la base de données et évitez les erreurs de mappage.

Cette rubrique traite des options disponibles pour la table "Person" lorsque vous mettez à jour la table parent "Department". L'exemple utilise les fichiers suivants :

- altova-cmpy-extra.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- **altova.mdb** la base de données cible à mettre à jour.

Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet y menant sera omis, pour des raisons de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement

recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

D'abord, ajouter le schéma XML de source et d'instance ainsi que la base de données cible dans le mappage (voir <u>Exemple : ajouter la base de données "altova.mdb" dans le mappage</u><sup>(251)</sup>). Suivre les mêmes étapes que dans <u>Action "Mettre à jour si... Insérer Rest"</u><sup>(233)</sup>. Ensuite, tirer les connexions de mappage comme indiqué cidessous :



Comme illustré ci-dessus, le mappage met à jour la table "Department" dans la base de données cible. La table "Department" est choisie en tant que table "root". Pour plus d'informations concernant ce qu'est une table racine et pourquoi elle est nécessaire, voir <u>Gérer des relations de base de données</u><sup>200</sup>. L'action à prendre par rapport à la table enfant "Person" est le sujet de cette rubrique.

Les tables suivantes illustrent plusieurs options de configuration et le résultat de mappage correspondant. Ces options peuvent être sélectionnées depuis le dialogue Actions de table de base de données de la table parent "Department" et la table enfant "Person".

# Configuration A

Paramètres			Résultat de mappage		
Action on record  PrimaryKey  ForeignKey  Name  Delete data in child tables  Ignore input child data  Person  Table "Department"	Update if		<ul> <li>Met à jour les enregistrements Department dans lequel PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Les Insertions, provenant de l'instance XML d'entrée, tous les enregistrements Person qui n'existent pas déjà dans la base de données.</li> </ul>		
Action on record	Insert All		Supprime des données enlant (enregistrements Person) parmi les enregistrements Department qui		
	foreign key		satisfont la condition <b>Mettre à jour si</b>		
EMail	mapped value	-			
I First	mapped value	•			
Last	mapped value	•			
PhoneExt	mapped value	-			
Title	mapped value	<b>•</b>			
Table "Person"					

# Configuration B

Paramètres		Résultat de mappage
Action on record PrimaryKey SroreignKey Name Delete data in child tables	Update if  equal	Le mappage échoue avec une erreur d'exécution SQL. La raison est que le mappage tente d'insérer de nouveaux enregistrements Person avec la même clé primaire que les enregistrements de Person existants. Si vous souhaitez insérer des enregistrements depuis le XML d'entrée dans ceux
Ignore input child data		se trouvant déjà dans la base de données, voir l'option suivante.
Table "Department"	Insart All 🗾	
	manned value	
ForeignKey	foreign key	
III EMail	mapped value	
I First	mapped value	
🛛 Last	mapped value	
PhoneExt	mapped value	
Title	mapped value	
Table "Person"		

# Configuration C

Paramètres			Résultat de mappage
Action on record       Update if         PrimaryKey       equal         ForeignKey       in         Name       in         Delete data in child tables       in         Ignore input child data       in         Person       in         Table "Department"			<ul> <li>Met à jour les enregistrements Department lorsque PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Des enregistrements New Person (avec des clés primaires générées) sont insérés dans la table Person en plus des enregistrements existants.</li> </ul>
Action on record	Insert All	•	
🗊 PrimaryKey	max() + 1	•	
💵 ForeignKey	foreign key		
EMail	mapped value	<b>T</b>	
I First	mapped value	<b>T</b>	
Last	mapped value	-	
PhoneExt	mapped value	-	
III Title	mapped value	•	
Table "Person"			

# Configuration D

Paramètres			Résultat de mappage
Action on record       Update if         ImaryKey       equal         ImaryKey       equal         ImaryKey       ImaryKey         ImaryKey       ImaryKey <td< td=""><td></td><td><ul> <li>Met à jour les enregistrements Department lorsque PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Aucun enregistrement n'est inséré dans la table Person par ce que l'option Ignorer les données enfant d'entrée est activée à partir de la table parent Departments.</li> </ul></td></td<>			<ul> <li>Met à jour les enregistrements Department lorsque PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Aucun enregistrement n'est inséré dans la table Person par ce que l'option Ignorer les données enfant d'entrée est activée à partir de la table parent Departments.</li> </ul>
Action on record	Insert All		
PrimaryKey	mapped value	<b>_</b>	
III ForeignKey	foreign key		
III EMail	mapped value	<b>•</b>	
III First	mapped value	<b>•</b>	
III Last	mapped value	<b>T</b>	
PhoneExt	mapped value	<b>T</b>	
1 Title	mapped value	<b>T</b>	
Table "Person"			

# Configuration E

Paramètres		Résultat de mappage
Action on record	Update if	<ul> <li>Met à jour les enregistrements Department lorsque PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Supprime tous les enregistrements Person liés à une Department qui dispose d'une PrimaryKey correspondante dans le XML de source. La raison est que l'option Supprimer des données dans des tables enfant est activée pour la table parent Department.</li> <li>Les enregistrements Person liés à un département qui ne satisfont pas à la condition Mettre à jour si restent dans la base de données.</li> <li>Aucun enregistrement n'est mis à jour dans la table Person.</li> </ul>

# Configuration F

Paramètres			Résultat de mappage		
Action on record  PrimaryKey  ForeignKey  Name  Delete data in child tables  Ignore input child data  Person  Table "Department"	Update if		<ul> <li>Met à jour les enregistrements Department lorsque PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Supprime tous les enregistrements Person liés à une Department qui dispose d'une PrimaryKey correspondante dans le XML de source. La raison est que l'option Supprimer des données dans</li> </ul>		
Action on record	Delete if	<b>_</b>	des tables enfant est activée pour la table parent		
Di PrimaryKey	equal	<b>•</b>	Les enregistrements Person liés à un département		
	foreign key	-	qui ne satisfont pas à la condition <b>Mettre à jour</b>		
T First		- -	si restent dans la base de données.		
Last		<b>•</b>			
PhoneExt		•			
I Title		•			
Table "Person"					

## **Configuration G**

Paramètres		Résultat de mappage
Action on record	Update if	<ul> <li>Met à jour les enregistrements Department lorsque PrimaryKey dans le XML de source correspond à la PrimaryKey dans la table de base de données.</li> <li>Ne met pas à jour les enregistrements Department existants qui n'ont pas de contrepartie dans le fichier XML d'entrée (aucune clé de ce type n'existe dans la source).</li> <li>Supprime tous les enregistrements Person qui satisfont aux deux conditions suivantes :</li> <li>a. L'enregistrement Person est lié à un</li> </ul>
Action on record	Delete if	Department qui possède une PrimaryKey correspondante dans le XML de source, et
Image: ForeignKey         Image: EMail         Image: First         Image: Last         Image: PhoneExt         Image: Table "Person"	foreign key	b. L'enregistrement Person possède une PrimaryKey correspondante dans le XML de source.

# 4.2.3.7 Action "Supprimer si..."

L'action de table **Supprimer si..** est utilisée pour supprimer des données depuis une table de base de données de manière conditionnelle. Vous pouvez définir cette action depuis le dialogue Actions de table de base de données (voir <u>Paramètres Actions de table de base de données</u><sup>(30)</sup>). Par exemple, lorsque vous mappez des données depuis un XML de source dans une base de données de cible, vous pouvez configurer une condition **Supprimer si...** pour vérifier si un certain champ dans le XML de source est égal au champ dans l'enregistrement de base de données cible (généralement une valeur de clé primaire). Si la condition **Supprimer si...** est vraie (donc, si les deux champs sont égaux), l'enregistrement de base de données sera supprimé lorsque le mappage est exécuté.

Note: L'action de table **Supprimer si...** ne doit pas être confondue avec l'option **Supprimer des données dans des tables enfants** disponible dans le dialogue Actions de table de base de données. L'action de table **Supprimer si...** ne touche que la table pour laquelle l'action est définie ; aucune autre table n'est touché.

Cet exemple vous montre comment supprimer des données depuis une table de base de données de manière conditionnelle, et insère aussi des enregistrements dans la même table de base de données dans le résultat de la condition de suppression.

L'exemple utilise les fichiers suivants :

- altova-cmpy-extra.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- altova.mdb la base de données cible à mettre à jour.

#### Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet y menant sera omis, pour des raisons de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

L'objectif du mappage est le suivant :

- Si des enregistrements de personne avec la même PrimaryKey existent aussi bien dans le XML de source que la table Person de cible, ils doivent être supprimés de la table Person.
- Tous les autres enregistrements du XML de source doivent être insérés dans la table Person.

Pour atteindre l'objectif de mappage, nous allons suivre les étapes suivantes.

### Étape 1: Insérer le composant XML de source

• Dans le menu **Insérer**, cliquer sur **Schéma XML/Fichier**, et chercher **Altova\_Hierarchical.xsd**. Lorsque vous êtes invité à fournir un fichier d'instance, chercher **altova-cmpy-extra.xml**.

### Étape 2: Insérer la base de données cible

Dans le menu Insérer, cliquer sur Base de données, et suivre les étapes de l'assistant pour vous connecter à altova.mdb (voir <u>Exemple : Ajouter la base de données "altova.mdb" dans le mappage</u><sup>(251)</sup>).

### Étape 3: Tirer les connexions

Tirer les connexions de mappage comme affiché ci-dessous.



## Étape 4: Configurer les actions "Mettre à jour si... Insérer Rest"

- 1. Dans le composant de cible, cliquer sur la touche **Action: Insérer** (<u>A:In</u>) située à côté de la table "Person".
- À côté de PrimaryKey, choisir la valeur equal, comme indiqué ci-dessous. Cela définit la condition de mise à jour : c'est à dire, l'enregistrement de la base de données sera supprimé uniquement lorsque sa valeur PrimaryKey sera égale à la valeur PrimaryKey provenant du mappage.

Action on record	NULL equal	Delete if	•
📲 PrimaryKey		equal	-
🗊 ForeignKey			-
EMail			-
First			-
🔲 Last			-
PhoneExt			-
II Title			-

En option, cocher la case **NULL equal**. Dans cet exemple, cette case à cocher n'est pas pertinente car ni le champ **PrimaryKey** se trouvant dans le XML de source ni le champ **PrimaryKey** dans la base de données ne contient de valeurs null. Néanmoins, vous devriez cocher cette case si vos données contiennent des valeurs null, et si vous souhaitez traiter ces valeurs comme égales ; sinon, vous risquez d'obtenir des résultats indésirables. Pour plus d'informations, voir <u>Gérer des valeurs null dans les actions de table de base de</u> <u>données</u><sup>327</sup>.

4. Cliquer sur **Apposer action**. Cela permet d'ajouter une nouvelle action à la droite de l'action existante **Supprimer si**. Configurer la nouvelle action en tant que **Insérer Rest** :

Actions to execute for each record All input data are compared to the DB table data, using the operators defined here. Select the "NULL equal" check box to treat NULL values as equal. If all comparisons are true, then the specific action is executed.									
Action on record	NULL equal	Delete if	•	Insert Rest	-	~	Append Action		
🛛 🕅 PrimaryKey		equal	•	mapped value	•				
I ForeignKey			•	mapped value	•		Insert Action		
EMail			-	mapped value	-				
I First			•	mapped value	•				
Last			•	mapped value	-				
PhoneExt			•	mapped value	-				
1 Title			•	mapped value	•				
						~	Delete Action		

Dans l'image ci-dessus, les actions de la table de base de données ont été configurées conformément avec les objectifs du mappage. C'est à dire que seule si la condition **Supprimer si**... est satisfaite, l'enregistrement sera mis à jour ; sinon, il sera inséré. L'option "valeur mappée" spécifie que des valeurs provenant du mappage seront utilisées pour remplir tous les champs de l'enregistrement.

 Cliquer sur OK pour fermer le dialogue. Veuillez noter que, de retour au mappage, la touche Action: Insérer A:In est maintenant transformée en une touche Action: Supprimer ; Insérer ( A:De,In ). Cela indique que l'action de suppression et d'insertion est configurée pour avoir lieu pour cette table.

### Étape 5: Consulter le mappage et mettre à jour la base de données

Cliquer sur l'onglet **Sortie** pour consulter le mappage. Un script SQL est généré, contenant des actions à exécuter par rapport à la base de données. Le script n'a pas encore modifié la base de données ; elle est uniquement destinée à la prévisualisation :

• Dans le menu Sortie, cliquer sur Exécuter Script SQL.

**Note :** L'exécution du script SQL directement depuis MapForce est juste un des moyens mis à disposition pour mettre à jour la base de données, voir aussi <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

Si vous ouvrez la table "Person" dans l'onglet de requête de BD de MapForce (voir <u>Parcourir et interroger des</u> bases de données<sup>(312)</sup>), vous pouvez voir le résultat du mappage comme suit :

- Tous les enregistrements de base de données qui avaient des clés primaires correspondantes dans le fichier XML ont été supprimés. Dans l'exemple, des enregistrements avec la clé primaire 1, 2, 3, 4 et 5.
- Tous les enregistrements de base de données qui n'ont pas de clés correspondantes dans le fichier XML restent non-affectés. Dans l'exemple, des enregistrements avec la clé primaire 6, 7, 8, 9, 10, 11, 12 et 13.
- De nouveaux enregistrements ont été insérés dans la table "Person" (lorsque la clé n'existait pas encore dans la base de données). Dans l'exemple, des enregistrements avec la clé primaire 30 et 31.

	PrimaryKey •	ForeignKey •	EMail •	First •	Last •	PhoneExt •	Title •
1	6	3	f.landis@nanonull.com	Fred	Landis	951	Program Manager
2	7	3	m.landis@nanonull.com	Michelle	Butler	654	Software Engineer
3	8	3	t.little@nanonull.com	Ted	Little	852	Software Engineer
4	9	3	a.way@nanonull.com	Ann	Way	951	Technical Writer
5	10	3	l.gardner@nanonull.com	Liz	Gardner	753	Software Engineer
6	11	3	p.smith@nanonull.com	Paul	Smith	334	Software Engineer
7	12	4	a.martin@nanonull.com	Alex	Martin	778	IT Manager
8	13	4	g.hammer@nanonull.com	George	Hammer	223	Web Developer
9	30	1	c.Cicada@microtech.com	Camilla	Cicada	765	HR
10	31	1	c.corrigan@microtech.com	Carol	Corrigan	629	Admin

La table "Person" après mise à jour de la base de données

# 4.2.3.8 Action "Ignorer si..."

L'action de table **Ignorer si...** est utilisée pour empêcher certains enregistrements dans une table de base de données d'être mis à jour, sur la base d'une condition définie. L'action **Ignorer si...** est uniquement significative lorsqu'elle est utilisée en combinaison avec une autre action de table de base de données (comme l'action **Insérer Rest**). Par exemple, lors du mappage de données depuis un XML de source dans une base de données cible, vous pouvez configurer une condition **Ignorer si...** pour vérifier si un certain champ dans le XML de source est égal à un champ dans l'enregistrement de base de données de cible (généralement, une valeur de clé primaire). Si la condition **Ignorer si...** st vraie (donc, si les deux champs sont égaux), l'enregistrement de base de données sera ignoré lorsque le mappage est exécuté, et que la prochaine action définie (**Insérer Rest**, par exemple) sera exécutée.

Cet exemple vous montre comment insérer des enregistrements dans une table de base de données basée sur le résultat de la condition **Ignorer si...** L'exemple utilise les fichiers suivants :

- altova-cmpy-extra.xml contient les données de source à insérer dans la base de données.
- Altova\_Hierarchical.xsd le schéma utilisé pour valider le fichier d'instance ci-dessus.
- **altova.mdb** la base de données cible à mettre à jour.

Tous les fichiers sont disponibles dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\. Ci-dessous, le chemin complet y menant sera omis, pour des raisons de simplicité.

Le mappage dans cet exemple modifie un fichier de base de données d'échantillon. Il est fortement recommande de sauvegarder la base de données originale et de commencer avec une nouvelle copie avant de suivre les étapes ci-dessous. Cela garantit que les exemples originaux ne sont pas contournés et que vous obtenez les mêmes résultats ci-dessous. Pour plus d'informations, voir <u>Exécuter des mappages qui</u> modifient les bases de données<sup>276</sup>.

L'objectif du mappage est le suivant :

- Si des enregistrements de personne avec la même PrimaryKey existent aussi bien dans le XML de source que la table Person de cible, aucune action ne doit être prise contre elles (donc, elles doivent être ignorées).
- Si un des enregistrements de personne qui ne satisfait pas à la condition mentionnée ci-dessus existe dans la table Person, aucune action ne doit être entreprise non plus.
- Les enregistrements provenant du XML de source qui ne possèdent pas de contrepartie (aucune clé primaire) dans la table Person doivent être traités en tant que Nouveau et Inséré dans la table Person avec une clé primaire.

Pour atteindre l'objectif de mappage, nous allons suivre les étapes suivantes.

### Étape 1 : Insérer le composant XML de source

Dans le menu **Insérer**, cliquer sur **Schéma XML/Fichier**, et chercher **Altova\_Hierarchical.xsd**. Lorsque vous êtes invité à fournir un fichier d'instance, chercher **altova-cmpy-extra.xml**.

### Étape 2 : Insérer la base de données cible

Dans le menu Insérer, cliquer sur Base de données, et suivre les étapes de l'assistant pour vous connecter à altova.mdb (voir <u>Exemple : Ajouter la base de données "altova.mdb" dans le mappage</u>)<sup>251</sup>.

### Étape 3 : Tirer les connexions

• Tirer les connexions de mappage comme affiché ci-dessous.



### Étape 4 : Configurer les actions "Ignorer si... Insérer Rest"

- 1. Dans le composant de cible, cliquer sur la touche **Action: Insérer** (<u>A:In</u>) située à côté de la table "Person".
- 2. À côté de **Action sur enregistrement**, choisir **Ignorer si...** Cela change l'action de table de la base de données en une action d'ignorer. C'est à dire que l'enregistrement actuel sera uniquement ignoré lorsqu'une condition sera satisfaite (voir étape suivante).
- 3. À côté de **PrimaryKey**, choisir la valeur **equal**, comme indiqué ci-dessous. Cela définit la condition ignorer : c'est à dire, que l'enregistrement de la base de données sera ignoré uniquement lorsque sa valeur **PrimaryKey** sera égale à la valeur **PrimaryKey** provenant du mappage.

Action on record	NULL equal	lgnore if	-
🛙 🕅 PrimaryKey		equal	-
💵 ForeignKey			-
EMail			-
First			-
Last			-
PhoneExt			-
I Title			-

En option, cocher la case **NULL equal**. Dans cet exemple, cette case à cocher n'est pas pertinente car ni le champ **PrimaryKey** se trouvant dans le XML de source ni le champ **PrimaryKey** dans la base de données ne contient de valeurs null. Néanmoins, vous devriez cocher cette case si vos données contiennent des valeurs null, et si vous souhaitez traiter ces valeurs comme égales ; sinon, vous risquez d'obtenir des résultats indésirables. Pour plus d'informations, voir <u>Gérer des valeurs null dans les actions de table de base de</u> données<sup>627</sup>.

 Cliquer sur Apposer action. Cela permet d'ajouter une nouvelle action à la droite de l'action existante Ignorer si. Configurer la nouvelle action en tant que Insérer Rest, avec une clé primaire définie sur max() + 1, comme indiqué ci-dessous :

Action on record	NULL equal	Ignore if	•	Insert Rest	•	$\sim$	Append Action
🗊 PrimaryKey	✓	equal	•	max() + 1	-		
🕼 ForeignKey			-	mapped value	-		Insert Action
EMail			-	mapped value	-		
First			•	mapped value	-		
Last			•	mapped value	-		
PhoneExt			-	mapped value	•		
Title			-	mapped value	-		
						$\sim$	Delete Action

Dans l'image ci-dessus, les actions de la table de base de données ont été configurées conformément avec les objectifs du mappage. C'est à dire que seule si la condition **Ignore If**... est satisfaite, l'enregistrement sera sauté ; sinon, il sera inséré. L'option "valeur mappée" spécifie que des valeurs provenant du mappage seront utilisées pour remplir tous les champs de l'enregistrement. L'option **max() + 1** génère une valeur clé primaire unique pour l'enregistrement.

5. Cliquer sur **OK** pour fermer le dialogue. Veuillez noter que, de retour au mappage, la touche **Action: Insérer** A:In est maintenant transformée en une touche **Action: Ignorer ; Insérer** () Cela indique que l'action ignorer et insérer est configurée pour avoir lieu pour cette table

## Étape 5: Consulter le mappage et mettre à jour la base de données

Cliquer sur l'onglet **Sortie** pour consulter le mappage. Un script SQL est généré, contenant des actions à exécuter par rapport à la base de données. Le script n'a pas encore modifié la base de données ; elle est uniquement destinée à la prévisualisation :

- Dans le menu Sortie, cliquer sur Exécuter Script SQL.
- **Note :** L'exécution du script SQL directement depuis MapForce est juste un des moyens mis à disposition pour mettre à jour la base de données, voir aussi <u>Exécuter des mappages qui modifient des bases de données</u><sup>276</sup>.

Si vous ouvrez la table "Person" dans l'onglet de requête de BD de MapForce (voir <u>Parcourir et interroger des</u> bases de données<sup>(332)</sup>), vous pouvez voir le résultat du mappage comme suit :

- Tous les enregistrements de base de données qui avaient des clés primaires correspondantes dans le fichier XML ont satisfait la condition **Ignorer si...** et restent non-touchés. Des exemples sont des enregistrements avec une clé primaire 1, 2, 3, 4 et 5.
- Tous les enregistrements de base de données qui n'avaient pas de clés correspondantes dans le fichier XML n'ont pas satisfait la condition **Ignorer si...** mais restent tout de même non-touchés. Dans l'exemple, des enregistrements avec la clé primaire 6, 7, 8, 9, 10, 11, 12 et 13.
- De nouveaux enregistrements ont été insérés dans la table "Person" (lorsque la clé n'existait pas encore dans la base de données). Dans l'exemple, des enregistrements avec la clé primaire 30 et 31

	PrimaryKey •	ForeignKey •	EMail •	First •	Last •	PhoneExt •	Title •
1	1	1	v.callaby@nanonull.com	Vernon	Callaby	582	Office Manager
2	2	1	f.further@nanonull.com	Frank	Further	471	Accounts Receivable
3	3	1	l.matise@nanonull.com	Loby	Matise	963	Accounting Manager
4	4	2	j.firstbread@nanonull.com	Joe	Firstbread	621	Marketing Manager Europe
5	5	2	s.sanna@nanonull.com	Susi	Sanna	753	Art Director
6	6	3	f.landis@nanonull.com	Fred	Landis	951	Program Manager
7	7	3	m.landis@nanonull.com	Michelle	Butler	654	Software Engineer
8	8	3	t.little@nanonull.com	Ted	Little	852	Software Engineer
9	9	3	a.way@nanonull.com	Ann	Way	951	Technical Writer
10	10	3	l.gardner@nanonull.com	Liz	Gardner	753	Software Engineer
11	11	3	p.smith@nanonull.com	Paul	Smith	334	Software Engineer
12	12	4	a.martin@nanonull.com	Alex	Martin	778	IT Manager
13	13	4	g.hammer@nanonull.com	George	Hammer	223	Web Developer
14	14	4	j.band@nanonull.com	Jessica	Bander	241	Support Engineer
15	15	4	l.king@nanonull.com	Lui	King	345	Support Engineer
16	16	5	s.meier@nanonull.com	Steve	Meier	114	Office Manager
17	17	5	t.bone@nanonull.com	Theo	Bone	331	Accounts Receivable
18	18	6	m.nafta@nanonull.com	Max	Nafta	122	PR & Marketing Manager US
19	19	7	v.bass@nanonull.com	Valentin	Bass	716	IT Manager
20	20	7	c.franken@nanonull.com	Carl	Franken	147	Support Engineer
21	21	7	m.redgreen@nanonull.com	Mark	Redgreen	152	Support Engineer
22	22	1	c.Cicada@microtech.com	Camilla	Cicada	765	HR
23	23	1	c.corrigan@microtech.com	Carol	Corrigan	629	Admin

dans le fichier XML de source. Ceux-ci étaient insérés dans la base de données avec la nouvelle clé primaire 22 et 23, respectivement.

La table "Person" après mise à jour de la base de données

# 4.2.3.9 Insertions Bulk (MapForce Server)

L'option **Utiliser Bulk Transfer** vous permet d'insérer des données à des vitesses très élevées depuis un composant MapForce (TXT, CSV, DAT, etc.) dans une table de base de données. Le recours à cette option accélère considérablement le processus Insérer, étant donné qu'il suffit d'exécuter une seule instruction au lieu de plusieurs.

L'option **Utiliser Bulk Transfer** peut être activée dans MapForce, au moment du design de mappage, comme indiqué ci-dessous. Un mappage pour lequel cette option est activée peut être exécutée dans MapForce, mais aucune insertion bulk ne s'applique à ce niveau. Le transfert bulk réel des données se produit lorsque le mappage est exécuté par MapForce Server.

Le transfert Bulk est pris en charge lorsque les conditions suivantes sont vraies :

- Le langage de transformation de mappage est configuré sur BUILT-IN. Pour plus d'information, voir Langages de transformation<sup>25</sup>.
- Le mappage est exécuté par MapForce Server (soit autonome soit sous gestion FlowForce Server). Cela signifie que le mappage doit être compilé soit en format .mfx ou déployé sur FlowForce Server. Pour plus d'informations, voir <u>Compiler des mappages dans les fichiers d'exécution MapForce</u> <u>Server</u><sup>G57</sup> et <u>Déployer des mappages vers FlowForce Server</u><sup>G60</sup>.
- La licence de MapForce Server ne se limite pas à "single thread execution" sur un appareil multi-core. C'est-à-dire que la case à cocher **Limiter à l'exécution de thread unique** dans l'onglet "Gestion de serveur" d'Altova LicenseServer doit être inactive.
- L'action de la base de données est "Insérer tout", voir aussi <u>Paramètres Actions de table de base de</u> <u>données</u><sup>330</sup>.
- La table dans laquelle les données doivent être chargées en bloc doit être une table "leaf", c'est à dire, dans la hiérarchie la plus basse de la base de données. Il ne doit pas y avoir de tables liées, modes ou procédures stockées, en référençant la table dans le mappage.
- Le pilote de base de données prend en charge l'insertion bulk dans les conditions WHERE.

La table suivante résume la prise en charge pour les insertions bulk selon le type de base de données et le pilote utilisé.

	ADO	ODBC	JDBC	ADO.NET	Natif
Access	Non	Non	n/a	n/a	n/a
DB2	Non	Oui	Oui	Oui	n/a
Firebird	n/a	Oui	Oui	Non	n/a
Informix	Non	Oui	Oui	Oui	n/a
iSeries	Non	Oui	Oui	Oui	n/a
MariaDB	Non	Oui	Oui	Non	n/a
MySQL	n/a	Oui*	Oui	Non	n/a
Oracle	Non	Oui	Oui	Oui	n/a
PostgreSQL	n/a	Oui	Oui	n/a	Oui
Progress	n/a	Oui	Oui	n/a	n/a
Serveur SQL	Oui	Oui	Oui	Oui	n/a
SQLite	n/a	n/a	n/a	n/a	Non
Sybase	Non	Oui	Oui	n/a	n/a
Teradata	n/a	Oui	Oui	n/a	n/a

\* MySQL version 5 ou plus est nécessaire.

**Note :** pour permettre la prise en charge de bulk insert pour MySQL et MariaDB via JDBC, utiliser l'option de connexion rewriteBatchedStatements=true.
### Exemple de bulk insert

Cet exemple vous montre comment créer un mappage qui charge en bloc des données depuis un fichier d'échantillon **source.txt** dans une base de données de cible. L'exemple utilise SQL Server 2014 et la base de données AdventureWorks 2014. Ce dernier peut être téléchargé depuis la page modèle AdventureWorks sur GitHub (<u>https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks</u>).

Location A,15.3,39 Location B,46,34 Location C,56.33,0 Location D,0,399 Location E,0,97.43

source.txt

### Pour définir une insertion bulk :

- 1. Configurer le langage de transformation sur BUILT-IN (
- Se connecter à la base de données AdventureWorks 2014 et ajouter la table "Production.Location" dans le mappage. Pour plus d'informations, voir <u>Ajouter des bases de données vers le mappage</u><sup>249</sup>.

	O AdventureWorks2014	
Ľ	Production.Location Location	[
Ľ	> ······ DILL smallint NOT NULL	C
Ľ	> 🔟 Name nvarchar(50) NOT NULL	C
Ľ	CostRate smallmoney NOT NULL	C
Ľ	Availability decimal(8,2) NOT NULL	C
Ľ	ModifiedDate datetime NOT NULL	C

Dans le menu Insérer, cliquer sur Texte, et ajouter un fichier de texte de source (comme dans l'échantillon source.txt ci-dessus) dans le mappage. Pour plus d'information, voir <u>Fichiers CSV et Texte</u>
 Assurez-vous que les types de données et les deux composants source et cible sont compatibles. Les types de données sont visibles sur le composant lorsque la touche de barre d'outils

source	
File: source.txt File/String str	ing [>
I → I → Rows [0∞]	¢
Field1 string [01]	¢
Field2 decimal [01]	¢
Field3 decimal [01]	¢

Afficher types de données ( 1) doit être activée.

O AdventureWorks2014	
Production.Location Location	þ
> 🗊 LocationID smallint NOT NULL	¢
> 🔟 Name nvarchar(50) NOT NULL	¢
CostRate smallmoney NOT NULL	¢
> III Availability decimal(8,2) NOT NULL	¢
> III ModifiedDate datetime NOT NULL	ķ

4. Tracer les connexions de mappage comme indiqué ci-dessous. Veuillez noter que la touche **Actions de base de données** A:In est maintenant visible à droite du nom de la table.

source		O AdventureWorks2014	
File: source.txt File/String string	7	P Troduction.Location A:In Location	Ç
[>		> I LocationID smallint NOT NULL	¢
Field1 string [01]	•	Name nvarchar(50) NOT NULL	¢
Field2 decimal [01]	•	CostRate smallmoney NOT NULL	¢
Field3 decimal [01]	•	Availability decimal(8,2) NOT NULL	¢
		NOT NULL	k

5. Cliquer sur la touche Actions de base de données ( A:In ), cocher la case Utiliser transfert Bulk, et cliquer sur OK pour confirmer.

😢 Database Table Actions	- Production.Loca	tion					$\times$
SQL statement to execute None DELETE all records Custom SQL:	before first record	ecords in all child t	ables				
Actions to execute for each All input data are compare Select the "NULL equal" ch If all comparisons are true	n record ed to the DB table dai neck box to treat NUL e, then the specific ac	ta, using the oper L values as equal tion is executed.	ators defined	l here.			
Action on record	NULL equa	Insert All	•		~	Append Action	
LocationID		DB-generated	•				_
Name		mapped value	•			Insert Action	
CostRate		mapped value	<b>•</b>				
Availability		mapped value	-				
ModifiedDate		mapped value	<b>•</b>				
Database transaction ha	andling				~	Delete Action	
When an error occurs:	rollback top transa	iction and stop				~	
Traces							
naces							
Trac <u>e</u> level:	Use component se	ttings				Fields	
☑ Use <u>b</u> ulk transfer (Map ('Use Transactions' can be	Force Server only) eselected after deact	Batch tivating 'Use Bulk'	n size:	1000 records			
					OK	Cancel	
							1

Dans le dialogue ci-dessus, veuillez noter que "Action sur enregistrement" est "Insérer tout". Le champ **Taille de batch** définit le nombre d'enregistrements à insérer par action.

Note : Lorsque l'option Utiliser transfert Bulk est activée, l'option Utiliser Transactions est désactivée et vice-versa. Si vous souhaitez activer le traitement de transaction, cliquer pour supprimer la case à cocher Utiliser transfert Bulk.

Maintenant que l'insertion bulk est activée, l'étape suivante permet d'exécuter le mappage dans MapForce Server (soit en version autonome ou sous gestion FlowForce Server). Pour plus d'informations, voir <u>Compiler</u> <u>des mappages dans les fichiers d'exécution MapForce Server</u> et <u>Déployer des mappages vers FlowForce</u> <u>Server</u><sup>(60)</sup>.

# 4.2.3.10 Gérer des nulls dans des actions de table de base de données

Lorsqu'un mappage met à jour une base de données cible par le biais d'actions de table comme "Ignorer si", "Mettre à jour si", "Supprimer si", MapForce compare les données de source par rapport aux données cible et, en résultat, génère des requêtes de mise à jour de base de données internes. Ces requêtes internes sont disponibles à la consultation dans le volet **Sortie** de MapForce, voir <u>Exécuter des mappages qui modifient des</u> <u>bases de données</u><sup>276</sup>. Les requêtes générées reflètent les conditions de comparaison qui ont été définies depuis le dialogue "Actions de table de base de données"

😢 Database Table Actions	- dbo.TARGET			×
SQL statement to execute b <ul> <li>None</li> <li>DELETE all records</li> <li>Custom SQL:</li> </ul>	efore first record also delete all records in all child	tables		
Actions to execute for each All input data are compared Select the "NULL equal" che If all comparisons are true,	record d to the DB table data, using the ope eck box to treat NULL values as equi then the specific action is executed	erators defined here. al. I.		
Action on record	NULL equal Ignore if	▼ Insert Rest ▼	^	Append Action
1 firstname	equal	▼ mapped value ▼		Insert Action
lastname	equal	▼ mapped value ▼		
email	equal	▼ mapped value ▼	~ [	Delete Action

Dialogue Actions de table de base de données

Si vos données de source ou de cible contiennent des champs nullables, vous disposez de deux options pour comparer les valeurs null depuis la source avec celles dans les données de cible :

- 1. traiter les valeurs null en tant qu'égal
- 2. traiter les valeurs null pas comme égal

Pour ce faire, le dialogue affiche une case à cocher **NULL equal** à côté de certains champs. Le résultat de mappage change si vous cochez cette case à cocher ou pas, cette option sera discutée dans cette rubrique. Veuillez noter que la case à cocher peut uniquement être sélectionnée pour les champs qui sont nullables, et si au moins une action de table possède une condition "equal" ou "equal (ignorer casse)".

Pour éviter tout résultat indésirable, vous devriez cocher la case **NULL equal** si toutes les conditions suivantes sont vraies :

- 1. Le dialogue "Actions de table de base de données" contient les actions "Ignorer si", "Mettre à jour si", "Supprimer si", et
- 2. si ces actions sont prises par rapport à des enregistrements qui peuvent contenir des valeurs null et
- si votre mappage exige que des valeurs null dans la source doivent être traitées en tant qu'égal avec des valeurs null dans la cible.

Par défaut, la case à cocher **NULL equal** n'est pas cochée. Si les conditions ci-dessus sont vraies et que la case à cocher n'est pas cochée, la table de base de données cible peut ne pas être mise à jour comme prévu (par exemple, plus de lignes seraient insérées que nécessaire). Cela se produit parce que des valeurs null touchent la comparaison de données. Par exemple, supposons que dans l'image ci-dessus, un enregistrement a un **email** null aussi bien dans la source que dans la cible. Si vous cochez la case **NULL equal**, cet enregistrement satisfera la condition **Ignorer si...** et sera ignoré (sauté). Néanmoins, si vous ne cochez pas la case à cocher **NULL equal**, l'enregistrement ne satisfera plus la condition **Ignorer si...** et sera insérée dans la base de données.

### Exemple

Afin de mieux comprendre la sensibilité null dans les mappages, analysons un exemple dans lequel se produit une comparaison de données null. Cet exemple prend une base de données Microsoft SQL Server ; néanmoins, il s'applique aussi à tout autre type de base de données. En option, si vous disposez d'un Microsoft SQL Server, vous pouvez créer les tables et les données utilisées dans cet exemple en exécutant le script de base de données suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\CreateNullableFields.sql.

Les tables de base de données sont affichées ci-dessous. Les deux tables stockent les données des personnes et ont les mêmes colonnes. De même la colonne email peut contenir des données null dans les deux tables.

++   id   :	+ firstname	lastname	email
1   1	Toby	Hughey	t.hughey@nanonull.com
2   I	Mia	Dahill	NULL
3   2	Fred	Weinstein	f.weinstein@nanonull.com

La table SOURCE

+		+	+	+
Ì	id	firstname	lastname	email
+	1	Mia	Dahill	

2   Fred	Weinstein	f.weinstein@nanonull.com
++	+	++

#### La table TARGET

Supposons que votre tâche n'est plus de fusionner des données depuis la table SOURCE dans la table TARGET. Seuls les nouveaux enregistrements doivent être insérés dans la table TARGET (dans cet exemple, "Tobie Hughey"). Les enregistrements qui existent dans les deux tables ("Mia Dahill" et "Fred Weinstein") doivent être ignorés.

La tâche peut être accomplie comme suit.

- Dans le menu Insertion, choisir Base de données. Suivre les étapes de l'assistant pour se connecter à la base de données (voir aussi <u>Connexion à une base de données</u><sup>(53)</sup>). Lorsque vous serez invité à ajouter des objets de base de données, choisir la table SOURCE.
- 2. Dans le menu **Insertion**, choisir **Base de données**. Se connecter à nouveau à la base de données et ajouter la table TARGET au mappage.
- 3. Tirer les connexions de mappage entre les composants de source et de cible.



4. Cliquer sur la touche Action:Insérer A:In et configurer les actions de table de base de données comme suit :

Actions to execute for each re	ecord					
All input data are compared to Select the "NULL equal" check If all comparisons are true, to	to the DB table dat k box to treat NULI hen the specific ac	a, using the op L values as equ tion is execute	perator Jal. d.	s defined here.		
Action on record	NULL equal	lgnore if	-	Insert Rest	•	Ī
💵 id			-	DB-generated	•	
firstname		equal	-	mapped value	•	
Iastname		equal	-	mapped value	•	
🔲 email		equal	•	mapped value	<b>•</b>	1

Comme illustré ci-dessus, une combinaison des actions "Ignorer si.. Insérer Rest" est définie. Cette configuration signifie que, pour chaque enregistrement, le mappage vérifie si :

- firstname dans la source est égal à firstname dans la cible, ET
- lastname dans la source est égal à lastname dans la cible, ET
- email dans la source est égal à email dans la cible.

Si toutes les conditions ci-dessus sont vraies, l'enregistrement est ignoré (conformément aux exigences). Sinon, un nouvel enregistrement est inséré dans la table cible. L'id du nouvel enregistrement est généré par la base de données, alors que les autres champs (firstname, lastname, email) sont remplis avec des valeurs mappées depuis la source.

Attention: la case à cocher située à côté d'email active ou désactive la comparaison sensible à NULL pour ce champ. Il est nécessaire de choisir cette case à cocher parce que email peut contenir des valeurs null (concrètement, "Mia Dahill" a une adresse e-mail null). Pour voir le rôle joué par cette case à cocher, essayez de mettre à jour la base de données deux fois : la première fois en cochant la case et une deuxième fois en décochant la case.

Pour mettre à jour la base de données, cliquer sur l'onglet **Sortie** et exécuter la commande de menu **Sortie** | **Exécuter script SQL**.

Si la case à cocher est sélectionnée, MapForce est soumis à l'indication explicite que vous devez traiter les valeurs null en tant qu'égaux. C'est pourquoi, l'enregistrement "Mia Dahill" n'est pas inséré dans la table de cible, ce qui est le résultat escompté.

Si la case à cocher n'est pas sélectionnée, l'enregistrement "Mia Dahill" est inséré dans la table cible (malgré le fait qu'il existe déjà), ce qui n'est pas le résultat attendu. La raison est qu'aucune indication explicite n'a été donnée à MapForce que vous souhaitez traiter les valeurs null en tant qu'égales. Une condition semblable se produirait si vous exécutiez la requête suivante par rapport à la base de données (cette requête n'extrait aucun enregistrement parce que la valeur null est comparée à l'opérateur "=" de manière à ce qu'il ne soit pas sensible à NULL):

```
SELECT firstname, lastname, email FROM TARGET WHERE firstname = 'Mia' AND lastname =
'Dahill' AND email = NULL;
```

Pour qu'elle soit sensible à NULL, la requête ci-dessus doit être réécrite comme suit :

```
SELECT firstname, lastname, email FROM TARGET WHERE firstname = 'Mia' AND lastname =
'Dahill' AND email IS NULL;
```

**Note :** Les requêtes ci-dessus ont un but purement illustratif et ne reflètent pas la syntaxe réelle des requêtes internes générées par MapForce. MapForce adapte la syntaxe des requêtes générées conformément au type de base de données (puisque plusieurs fournisseurs de base de données on des approches différentes pour gérer des comparaisons de type null).

### 4.2.3.11 Paramètres Actions de table de base de données

Lorsque vous créez une connexion de mappage dans une table de base de données, une touche Actions de base de données apparaît à côté de la table touchée. Cliquer sur cette touche pour ouvrir un dialogue Actions de table de base de données, à partir duquel vous pouvez configurer les actions insertion, mise à jour et supprimer, ainsi que d'autres options.

😢 Database Table Actions ·	Person								Х
SQL statement to execute be <u>None</u> DELETE all records <u>C</u> ustom SQL:	efore first record	cords in all child t	tables	3					
Actions to execute for each All input data are compared Select the "NULL equal" che If all comparisons are true,	record I to the DB table dat ck box to treat NULI then the specific act	a, using the ope . values as equa tion is executed.	eratore al.	s defined here.					
Action on record	NULL equal	Update if	•	Insert Rest	•		~	Append Action	
PrimaryKey		equal	•	mapped value	-		L		
💵 ForeignKey			•	mapped value	-			Insert Action	
III EMail			•	mapped value	-				
First			•	mapped value	•				
Last			•	mapped value	•				
PhoneExt			•	mapped value	-				
1 Title			•	mapped value	▼				
Database transaction har	ndling rollback top transac	tion and stop					× [	Delete Action	
Traces									.
Trac <u>e</u> level:	Use component set	tings					$\sim$	Fields	
Use <u>b</u> ulk transfer (MapF Bulk transfer is unsupported	orce Server only) d by the database of	Batcl	h size	; 100	1 00	records			
							OK	Cancel	

Dialogue Actions de table de base de données

Ci-dessous, vous trouverez une description des paramètres disponibles dans le dialogue Actions de table de base de données.

### Instruction SQL pour une exécution avant le premier enregistrement

Dans cette fenêtre de groupe, vous pouvez définir des instructions SQL qui sont exécutées avant toute action définie sous **Actions à exécuter pour chaque enregistrement**. Sélectionner le bouton radio désiré :

• Aucun — Aucune action n'est effectuée. Il s'agit du paramètre par défaut.

- SUPPRIMER tous les enregistrements Tous les enregistrements provenant de la table sélectionnée sont supprimés avant que toute action de table spécifique définie dans la fenêtre de groupe Actions à exécuter pour tous les enregistrements soit effectuée. Cocher la case supprimer aussi tous les enregistrements dans toutes les tables enfants si vous souhaitez vous débarrasser des données stockées dans les tables enfant de la table sélectionnée. Pour consulter un exemple, voir Insérer des données dans plusieurs tables liées
- SQL personnalisé Écrire une instruction SQL personnalisée pour toucher la table tout entière.
   Veuillez noter que la prise en charge pour plusieurs instructions SQL dans une requête dépend de la base de données, de la méthode de connexion et du pilote utilisé.

### Actions à exécuter pour chaque enregistrement

Ce groupe de paramètres spécifie les actions de base de données qui doivent être effectuées par rapport à cette table lors de l'exécution du mappage. Pour gérer les actions de table, cliquer sur les touches **Apposer Action**, **Insérer Action**, ou **Supprimer Action**. Plusieurs actions peuvent être définies le cas échéant (par exemple, une action "Mettre à jour si..." suivie par une action "Insérer Rest".

Les actions de table définies sont traitées de gauche à droite. Dans l'exemple ci-dessus, l'action "Mettre à jour si..." est traitée en premier. Si la condition de mise à jour n'est pas satisfaite, l'action suivante sera traitée (dans cet exemple, l'action "Insérer Rest"). Veuillez noter les points suivants :

- Toutes les conditions définies d'une action doivent être satisfaites pour que l'action de table soit exécutée. Lorsque cela est le cas, tous les champs pour lesquels une connexion existe entre les items de source et de cible dans le mappage sont mis à jour. Toutes action de table suivante (à la droite d'une action dont la condition est correspondante) sont ignorées pour cet enregistrement.
- Si la condition définie n'est pas satisfaite, l'action de table est sautée et l'actions suivante (à droite) est traitée.
- Si aucune des conditions n'est satisfaite, aucune action de table n'a lieu.

Toute action de table définie après les actions "Insérer tout" ou "Insérer Rest" ne sera jamais exécutée car aucune condition de colonne n'existe pour les actions d'insertion. Un dialogue apparaît lorsque cela est le cas indiquant que les colonnes d'action de table suivantes seront supprimées.

La case à cocher "NULL Equal", située à côté de chaque enregistrement, indique à MapForce que les valeurs null dans l'enregistrement de source doivent être traitées de manière égale avec des valeurs nulles dans l'enregistrement cible. Le fait de ne pas cocher cette case à cocher peut entraîner des résultats indésirables, voir <u>Gérer des nulls dans des actions de table de base de données</u> 2020).

Lorsque le mappage met à jour une table qui possède des relations de clé étrangères vers d'autres tables, les options suivantes peuvent être utilisées :

Supprimer des données dans les tables enfants	Cette option est pertinente lorsque vous choisissez l'action "Mettre à jour si" pour une table parent. Il peut s'avérer nécessaire si le nombre d'enregistrement dans le fichier source est différent du nombre d'enregistrements dans la base de données de cible, et que vous souhaitez que la base de données reste synchronisée (pas de données orphelines dans les tables enfants). Voir aussi <u>Options pour tables</u> <u>enfants lors de la mise à jour d'une table parent</u> <sup>307</sup> .
	souhaitez que la base de données reste synchronisée (pas de données orphelines dans les tables enfants). Voir aussi <u>Options pour tables</u> enfants lors de la mise à jour d'une table parent <sup>307</sup> .

Ignorer les données enfant	Utiliser cette option lorsque vous souhaitez mettre à jour une table de
d'entrée	parent cible, sans toucher une des tables/enregistrements enfants de
	cette table. Voir aussi Options pour tables enfants lors de la mise à jour
	<u>d'une table parent</u> <sup>307</sup> .

Pour consulter des exemples qui illustrent plusieurs combinaisons d'actions, voir :

- Insérer des données dans une table<sup>280</sup>
- Insérer des données dans plusieurs tables liées<sup>200</sup>
- Mettre à jour une table<sup>295</sup>
- Options pour tables enfants lors de la mise à jour d'une table parent 307
- <u>Action "Mettre à jour si... Insérer Rest"</u><sup>299</sup>
- Action "Supprimer si..."<sup>315</sup>
- <u>Action "Ignorer si..." 319</u>

### Utiliser des transactions

Active des transactions de base de données pour cette action de table particulière. Pour plus d'informations, voir <u>Gérer les exceptions de la base de données</u><sup>423</sup>.

### Utiliser le Transfert Bulk

Active le transfert bulk (plusieurs instructions INSERT en tant qu'une seule requête). Les transferts Bulk sont pris en charge si le mappage est exécuté par MapForce Server et l'action de base de données est "Insérer tout". Pour plus d'informations, voir <u>Insertions Bulk (MapForce Server)</u><sup>633</sup>.

# 4.2.3.12 Exemple : Mapper des données de XML vers SQLite

Cet exemple vous guide à travers les étapes nécessaires pour créer un mappage MapForce qui lit des données depuis un fichier XML et les rédige dans une base de données SQLite. L'exemple s'accompagne d'un fichier d'échantillon de design de mappage (.mfd). Si vous souhaitez consulter le fichier d'échantillon avant de vous lancer dans cet exemple, vous pouvez l'ouvrir depuis le chemin suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\XMLtoSQLite.mfd.

Le but de cet exemple est d'insérer des données depuis un fichier XML dans une base de données SQLite. Pour atteindre l'objectif de cet exemple, il vous faudra une base de données SQLite vide dans laquelle des données seront rédigées. Comme illustré ci-dessous, vous pouvez créer et explorer la base de données SQLite soit avec Altova DatabaseSpy, ou avec le shell de ligne de commande disponible depuis le site web SQLite officiel.

#### Pour créer la base de données SQLite :

Si DatabaseSpy est installé sur votre ordinateur (soit autonome, soit en tant que partie intégrante d'Altova MissionKit), vous pouvez créer la nouvelle base de données SQLite comme suit :

- 1. Exécuter DatabaseSpy.
- 2. Dans le menu Fichier, cliquer sur Créer une connexion de base de données.
- 3. Cliquer sur Assistant de connexion, puis cliquer sur SQLite.

- 4. CCliquer sur Créer une nouvelle base de données SQLite, saisir c:\sqlite\articles.sqlite en tant que chemin, puis cliquer sur Se connecter.
- 5. Lorsque vous êtes invité à définir un nom de source de données, ne pas modifier le nom par défaut.
- 6. Ouvrir un nouvel éditeur SQL (Ctrl+N) et exécuter la requête suivante par rapport à la base de données

create table articles (number smallint, name varchar(10), singleprice real);

Sinon, suivre les étapes suivantes, pour créer la base de données :

- Télécharger le shell de ligne de commande SQLite pour Windows depuis la page de téléchargement SQLite (<u>http://www.sqlite.org/download.html</u>) et dé comprimer l'archive .zip dans un répertoire sur votre appareil local (dans le cas de notre exemple, utiliser c:\sqlite).
- 2. Exécuter c:\sqlite\sqlite3.exe et saisir l'instruction suivante :

create table articles (number smallint, name varchar(10), singleprice real);

Cela crée la table articles dans la base de données en mémoire. La table articles consiste en trois colonnes : *number*, *name*, et *singleprice*. L'objectif de ces colonnes est de stocker des données depuis les éléments avec le même nom défini dans le schéma XML de source. Chaque colonne est déclarée avec le type de données pertinent pour les données censées être stockées dans cette colonne.

3. Exécuter la commande :

.save articles.sqlite

Cela permet d'enregistrer la base de données en mémoire dans le chemin de travail actuel : c: \sqlite\articles.sqlite. Veuillez noter que vous devrez vous référer à ce chemin dans les étapes suivantes.

Vous avez à présent achevé de créer l'échantillon de base de données SQLite nécessaire pour cet exemple.

#### Pour créer le design de mappage XML en SQLite :

- 1. Exécuter MapForce et s'assurer que le langage de transformation est configuré sur BUILT-IN (utiliser la commande de menu **Sortie | Moteur d'exécution Built-in**).
- Ajouter à la surface de mappage le fichier Articles.xml situé dans le dossier
   <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial (utiliser la commande de menu Fichier | Insérer le schéma XML/Fichier).
- Ajouter à la surface de mappage la base de données articles.sqlite créée dans des étapes précédentes (utiliser la commande de menu Fichier | Insérer base de données), puis sélectionner SQLite.
- 4. Cliquer sur **Se connecter**. Lorsque vous serez invité à choisir les objets de la base de données, choisir la table articles.

🗊 Insert Database Objects
Please choose Tables, Views, Stored Procedures, or user are stored locally in the mfd file.
Source
▼ ▼
□ ♥ 📴 articles
User Tables
Show <u>P</u> review >>

5. Tirer les connexions comme indiqué ci-dessous :



6. Cliquer sur la touche A:In dans le composant de base de données et choisir l'option Supprimer tous les enregistrements. Cela Cela permet de garantir que, à chaque fois que le mappage est exécuté, toutes les lignes de base de données existantes sont supprimées en premier, pour empêcher toute duplication.

😢 Database Table Actions - artic	Database Table Actions - articles								
SQL statement to execute before f <ul> <li><u>N</u>one</li> <li><u>DELETE all records</u> als</li> <li><u>C</u>ustom SQL:</li> </ul>	first record so delete all records in all child t	ables			j				
Actions to execute for each record All input data are compared to the Select the "NULL equal" check box If all comparisons are true, then t	Actions to execute for each record All input data are compared to the DB table data, using the operators defined here. Select the "NULL equal" check box to treat NULL values as equal. If all comparisons are true, then the specific action is executed.								
Action on record	NULL equal Insert All	<b>_</b>	~	Append Action					
	mapped value	<b>_</b>							
name	mapped value	<u> </u>		Insert Action					
singleprice	mapped value	<b>_</b>							
			~	Delete Action	1				

7. Cliquer sur l'onglet **Sortie** de la fenêtre de mappage principale. MapForce exécute le mappage en utilisant le moteur d'exécution built-in et affiche la requête créer SQL dans la fenêtre Sortie.



 Exécuter le script SQL pour remplir la base de données (utiliser la commande de menu Sortie | Exécuter script SQL). Si MapForce ne rencontre pas d'erreur de marche, les enregistrements sont insérés dans la base de données SQLite.

The following SQL statements were executed during "Generate output" function. 2 3 Every single result is written right to the "-->>>" string. 4 These statements are only for preview and may not be executed in another SQL query tool! 5 The database was connected using the following connection-string: 6 C:/sqlite/articles.sqlite 7 \*/ 8 9 PRAGMA foreign\_keys = ON; 10 -->>> OK. 0 row(s). 11 12 DELETE FROM "articles" 13 -->>> OK. 0 row(s). 14 15 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (1, 'T-Shirt', 25) 16 -->>> OK. 1 row(s). 17 18 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (2, 'Socks', 2.3) 19 -->>> OK. 1 row(s). 20 21 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (3, 'Pants', 34) 22 -->>> OK. 1 row(s). 23 24 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (4, 'Jacket', 57.5) 25 -->>> OK. 1 row(s). 26

#### Pour contrôler si les données ont été insérées correctement dans la base de données SQLite :

- 1. Cliquer sur l'onglet **Requête DB** de MapForce.
- 2. Choisir l'entrée articles depuis la liste déroulante située en haut du volet Requête DB.
- 3. Saisir la requête suivante dans l'éditeur SQL :

select \* from articles;

En alternative, suivre les étapes suivantes :

1. Exécuter le fichier c:\sqlite\sqlite3.exe et ouvrir la base de données avec la commande :

.open articles.sqlite

2. Exécuter l'instruction select suivante :

```
select * from articles;
```

Quelle que soit l'approche que vous choisissez pour sélectionner les données de table (shell MapForce ou SQLite), la requête doit à présent retourner quatre lignes. Cela correspond au nombre d'enregistrements dans le fichier XML de source, ce qui était l'objectif recherché de cet exemple.

# 4.2.4 Joindre des données de base de données

Dans des mappages qui lisent des données depuis des bases de données, vous pouvez joindre des objets de base de données comme des tables ou des vues en ajoutant un composant Join dans le mappage. Par exemple, vous pouvez combiner des données provenant de deux ou plusieurs tables liées par des relations de clé étrangère, ce qui constitue la méthode typique pour stocker des données dans des base de données relationnelles. Le résultat est le même que si vous exécutez une requête SQL par rapport à la base de données lorsque deux ou plus tables sont jointes par le biais d'une opération INNER JOIN (ou LEFT JOIN, le cas échéant).

Selon le type de données connectées au composant join, l'opération join peut se produire soit dans le mode standard (non-SQL), soit dans le mode SQL. Les jointures dans le mode non-SQL sont effectués par MapForce, alors que les jointures dans le mode SQL sont effectués par la base de données à partir de laquelle le mappage lit les données.

Des jointures en mode non-SQL sont plus flexibles car elles prennent en charge plus de types de composant en tant qu'entrée (par exemple, la jointure peut être entre des tables provenant de bases de données différentes, ou entre des structures XML et des tables de base de données). Pour consulter un exemple d'une jointure non-SQL, voir <u>Exemple : Joindre des structures XML</u><sup>[511]</sup>. D'un autre côté, une jointure non-SQL oblige le moteur de mappage a effectuer des comparaisons coûteuses en mémoire (parce que le nombre total de comparaisons représente la jointure croisée, ou le produit cartésien, de toutes les structures jointes). Généralement, ce processus a lieu très rapidement et est négligeable dans des mappages qui ne nécessitent pas trop de données ; néanmoins, si les sources de données jointes consistent en un nombre important d'enregistrements, le mappage prendra un temps considérable à s'exécuter. Si vos mappages doivent traiter un grand nombre d'enregistrements, nous vous recommandons de considérer la mise sous licence de MapForce Server Advanced Edition, qui contient une optimisation join dédiée pour accélérer l'exécution de mappage.

Une jointure dans le mode SQL n'accepte que les objets de base de données éligibles en tant qu'entrée (comme des tables ou des vues), elle n'est donc pas aussi flexible qu'une jointure non-SQL. Néanmoins, elle offre de meilleures performances de mappage parce qu'elle est exécutée de manière native par la base de données. Pour plus d'informations voir <u>Joins dans le Mode SQL</u><sup>(33)</sup>.

Note : L'utilisation d'un composant Join n'est pas le seul moyen de joindre des tables ou des affichages de base de données. Les jointures applicables aux bases de données peuvent aussi être effectués en utilisant des instructions SELECT SQL, voir <u>Instructions SELECT SQL en tant que tables virtuelles</u> <sup>622</sup>. Une différence principale entre les instructions SELECT SQL et les composants Join est que les premières sont écrites à la main, ce qui leur confère une plus grande flexibilité. Les composants Join sont une alternative plus simple si vous ne souhaitez pas rédiger des instructions SQL à la main.

### Pour ajouter un composant Join :

- Définir le langage de transformation de mappage sur BUILT-IN (pour ce faire, cliquer soit sur la touche de la barre outils , soit utiliser la commande de menu Sortie | Moteur d'exécution Built-In).
- Dans le menu Insérer, cliquer sur Join. En alternative, cliquer sur la touche de la barre outils Join (

). Le composant Join apparaît dans le mappage. Par défaut, il accepte des données provenant de deux structures, il a donc deux entrées nodes/rows. Si nécessaire, vous pouvez ajouter de nouvelles entrées dans la jointure en cliquant sur la touche **Ajouter entrée** ( ), voir <u>Joindre trois structures ou</u>

plus<sup>510</sup>.



- 3. Connecter les structures qui doivent être jointes dans l'item nodes/rows du composant join.
- 4. Ajouter la condition pour la jointure (ou plusieurs conditions). Pour ce faire, cliquer avec la touche de droite dans le composant Join et choisir **Propriétés**. Les conditions Join peuvent aussi être ajoutées directement depuis le mappage, en connectant le résultat Booléen de certaines fonctions à l'item condition du composant Join. Dans certains cas. lorsque des tables de base de données sont jointes, la condition (ou les conditions) de jointure peut être créée automatiquement par MapForce. Pour plus d'informations, voir <u>Ajouter des conditions Join</u><sup>672</sup>.

Notes :

- Les composants Join sont pris en charge lorsque la langue cible du mappage est configurée sur BUILT-IN. La génération de code dans C#, C++, ou Java n'est pas prise en charge.
- Lorsqu'une structure n'est pas une source d'entrée valide ou prise en charge pour la jointure, MapForce affiche des indices soit immédiatement et directement dans le mappage, soit dans la fenêtre Messages, lorsque vous validez le mappage (voir <u>Valider des mappages</u><sup>103</sup>).
- Les composants Join ne doivent pas être connectés aux paramètres d'entrée ou aux résultats des fonctions définies par l'utilisateur inline. Si de telles connexions existent, des erreurs de validation se produiront au cours de la validation de mappage.
- Lorsque vous vous connectez à des composants de base de données éligibles (comme des tables ou des modes) directement dans un composant Join, une touche mode SQL ( <sup>SQL</sup> ) apparaît automatiquement en haut à droite du composant Join. Une fois activée, cette touche offre des fonctions SQL spéciales applicables à l'opération de jointure (voir <u>Joins dans le mode SQL</u> <sup>(SQL</sup>).
- Il n'est pas possible de connecter la sortie de l'item joined à un autre composant Join. Néanmoins, le cas échéant, vous pouvez connecter un résultat partiel à une jointure d'un autre.

# 4.2.4.1 À propos des joins dans le Mode SQL

Lorsque vous connectez des composants de base de données éligibles (comme des tables ou des vues) directement à un composant join, un bouton **mode SQL** ( <sup>SQL</sup> ) apparaît automatiquement dans le coin supérieur droit du composant join. Lorsque le mode SQL est activé, l'opération join est entreprise par la base de données à partir de laquelle le mappage lit les données. En d'autres termes, MapForce enverra une requête en interne vers la base de données avec la syntaxe SQL appropriée pour sélectionner et combiner des données depuis toutes les tables qui participent à la jointure. Attention: vous n'avez pas besoin d'écrire de SQL ; la requête désirée est produite sur la manière dont vous avez conçu visuellement le composant Join sur le mappage, comme vous le verrez dans les exemples suivants.

Pour que le mode SQL soit possible, les conditions suivantes doivent être satisfaites :

- 1. Les deux objets (tables ou vues) qui doivent être joints doivent provenir de la même base de données.
- 2. Les deux objets qui doivent être joints doivent provenir du même composant MapForce. (Veuillez noter que vous pouvez rapidement ajouter/supprimer des objets de base de données dans un composant

comme suit : cliquer avec la touche de droite sur le composant de base de données, et choisir Ajouter/Supprimer/Éditer des objets de base de données depuis le menu contextuel.)

- La condition Join (ou les conditions) doivent uniquement être définies depuis les propriétés de composant (en cliquant avec la touche de droite sur l'en-tête du composant join, et choisir Propriétés), et pas sur le mappage (voir aussi <u>Ajouter des conditions Join</u><sup>507</sup>).
- **Note :** Lorsque des tables de base de données sont jointes dans le mode SQL, MapForce créera la condition Join (ou les conditions) automatiquement, sur la base des relations de clé étrangère détectée entre les tables. Pour que des conditions Join automatiques se produisent, les tables de base de données doivent se trouver dans une relation enfant-parent dans le composant MapForce (c'est à dire, qu'une table doit être "parent" ou "child" d'un autre dans le composant), voir <u>Exemple: Joindre des tables dans le mode SQL</u><sup>[342]</sup>.
  - 4. Toutes les tables de base de données ne doivent pas encore être se trouver dans le contexte cible actuel. Lorsque le résultat join est utilisé dans un composant cible, aucune des tables réunies ne peut être connectée directement ou indirectement vers des nœuds de parent cible. Pour plus d'informations concernant la manière dont un mappage est exécuté, voir <u>Règles et stratégies de mappage</u><sup>372</sup>.

Vous pouvez visualiser ou contrôler le mode SQL par le biais de la touche **SQL** ( SQL ) situé dans le coin supérieur droit du composant join, comme suit :



SQL

Le mode SQL est désactivé (join sera exécuté par MapForce (ou, si applicable, par MapForce Server).

Le mode SQL est activé (join sera exécuté par la base de données).

Si la touche set manquante, cela signifie que le mode SQL ne sera pas significatif ou pas pris en charge pour les données qui sont jointes.

Dans certains cas, le mode SQL doit être désactivé explicitement ( SQL ), par exemple:

- Lorsque votre mappage nécessite des conditions join en-dehors des propriétés de composant join (c'est à dire, des conditions définies sur le mappage et connectées à l'item condition du composant join).
- Lorsque vous souhaitez joindre des tables depuis des base de données variées. Utiliser un joint standard (non-SQL) si vous souhaitez joindre des tables depuis des base de données différentes.

### Changer le mode Join

Lorsque le composant Join se trouve dans le **mode SQL** sque, vous pouvez joindre les tables de base de données ou les vues des manières suivantes :

- INNER JOIN Seuls les enregistrements qui satisfont la condition dans les deux ensembles d'entrée sont retournés par le composant Join.
- LEFT OUTER JOIN Le composant Join inclut tous les enregistrements provenant de la table "à l'extrême gauche" (dans MapForce, il s'agit de la table supérieure d'un composant Join), plus les enregistrements provenant de la table réunie par la suite qui satisfait la condition join.

Pour consulter le mode join d'une table ou consulter le composant Join, observer l'icône affichée devant la table réunie ou la vue. Une des icônes suivantes peut s'afficher pour toute table réunie ou vue sauf la première:

- Inner Join ⋈
- Left Join א

Pour afficher une infobulle contenant des détails concernant le join, déplacer le curseur sur l'icône :



Pour modifier le mode join, :

- Cliquer sur l'icône Inner Join ⋈ ou Left Join ⋈ devant chaque table ou vue réunie et choisir Inner Join ou Left Outer Join depuis le menu contextuel.
- Cliquer avec la touche de droite sur la deuxième (ou troisième, quatrième, etc) table ou vue réunie dans le composant Join, et choisissez Join Type | Inner Join, ou Join Type | Left Outer Join depuis le menu contextuel.



Noter :

- Si vous avez changé le mode join en LEFT OUTER JOIN, la table ou la vue supérieure représentera le côté "gauche" du join.
- Le fait de modifier le mode join affecte les données retournées par le composant join de la même manière que INNER JOIN ou LEFT JOIN impacte le résultat d'une requête SQL dans une base de données.

### Noms d'alias

Il arrive souvent que des tables ou des vues de base de données jointes contiennent des noms de champ identiques dans les deux structures jointes. Lorsque le mode SQL est activé, ces items apparaissent sur le composant préfixé par le mot-clé "AS". Par exemple, si les deux tables jointes contiennent un champ "id", ce champ apparaît comme "id" sur la première table jointe et en tant que "id AS id2" sur la deuxième table jointe. Les tables jointes peuvent aussi produire des noms d'alias, par exemple, si la même table est jointe à elle-même.

Le champ alias ou les noms de table sont importants si vous souhaitez vous y référer dans un mappage ultérieurement. Par exemple, imaginez le cas où vous souhaitez filtrer ou trier le résultat de la jointure. Pour y parvenir, la sortie du composant join peut être connectée à un composant WHERE/ORDER SQL, où vous pourrez saisir les clauses SQL WHERE et ORDER BY.

Pour se référer à un champ depuis la clause WHERE, écrire le nom de la table, suivi d'un point (.), suivi par le nom de champ. Pour vous référer à un alias de table, utiliser le nom d'alias comme il apparaît dans le composant Join. Dans la clause ORDER BY, vous pouvez utiliser soit la même technique (table.field), ou écrire juste le nom de champ d'alias (le nom qui apparaît après "AS").

Pour un mappage d'exemple qui utilise des clauses WHERE/ORDER SQL, voir <u>Exemple : Tables Join dans le</u> <u>mode SQL</u><sup>342</sup>.

**Note :** Les composants WHERE/ORDER SQL ne sont pas autorisés entre une table de base de données et le composant join ; Ils peuvent uniquement être ajoutés après (et non pas avant) un composant join. Pour plus d'informations concernant les composant WHERE/ORDER SQL, voir <u>Filtrer et trier les</u> données de base de données (WHERE/ORDER SQL)<sup>(35)</sup>.

# 4.2.4.2 Exemple : Joindre des tables dans le mode SQL

Cet exemple illustre comment joindre des données provenant de deux tables de base de données, en utilisant un composant join de MapForce. L'opération join est effectuée dans le mode SQL, comme décrit dans <u>Joins</u> dans le Mode SQL<sup>600</sup>. Veuillez noter que la jointure de trois tables ou plus fonctionne de manière très semblable, voir aussi <u>Exemple : Créer un rapport CSV à partir de plusieurs tables</u><sup>61</sup>.

L'exemple est accompagné d'un échantillon de mappage disponible dans le chemin suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\JoinDatabaseTables.mfd.



JoinDatabaseTables.mfd

L'objectif du mappage ci-dessus est de combiner des données à partir de deux tables de base de données de source dans un seul fichier CSV cible. Comme illustré dans le diagramme de base de données ci-dessous, la première table (users) stocke les adresses des personnes et la deuxième table (addresses) stocke les noms et les adresses des personnes. Les deux tables sont liées par un champ commun (id dans users correspond à user\_id dans addresses). Dans la terminologie de la base de données, ce type de relation est aussi connu comme une "relation de clé étrangère".



Pour plus de commodité, l'image ci-dessous illustre les données actuelles dans les deux tables.

id 🔹	first_name 🔹	last_name 🔹	email	l	٠	crea	ated_at	•	upd	lated_at 🔹				
1	Marquita	Bailey	m.bai	iley@nanonul	l.com	201	6-12-29 14:37	7:14	[NU	NULL]				
2	Sharda	Junker	s.jun	junker@nanonull.com		201	2016-12-29 14:37:14 [NUL		LL]	ers				
3	Totie	Rea	t.rea(	@nanonull.co	m	201	6-12-29 14:37	7:14	[NU	LL]	n			
4	Tobie	Hughey	t.hug	hey@nanonu	II.com	201	6-12-29 14:37	7:14	[NU	LL]				
5	Eadith	Lafreniere	e.lafr	eniere@nand	onull.com 201		6-12-29 14:37	7:14	[NU	LL]				
6	Yehudi	Sponga	y.spo	nga@nanoni	ill.com 2016-12-29 14:37:14		7:14	[NULL]						
7	Laurianne	Huisman	l.huis	man@nanon	ull.com	2016-12-29 14:37:14		2016-12-29 14:37:14 [NULL]						
8	Fred	Weinstein	f.wei	instein@nand	null.com	201	6-12-29 14:37	7:14	[NU	LL]				
9	Mia	Dahill	id 🔹	user_id •	is_shippin	g •	is_billing •	type	e •	city	•	street •	number •	
10	June	Leiker	1	1	1		0	wo	rk	Bridgedell		Maple Lane	1	es
11	Benedick	Kocyk	2	1	0		1	hon	ne	Bridgedell		Olive Street	6	ess
12	Andrej	Hildebrand	3	3	1		1	hon	ne	Roseford		Evergreen Lane	34	ddr
13	Ariel	Phelan	4	4	1		1	wo	rk	Beardale		Route 44	9	n l
14	Matthaeus	Hulick	5	6	1		1	hon	ne	Johnson C	ity	Franklin Avenue	11	
15	Lotta	Mendes	6	7	1		1	hon	ne	North Kings	stown	Beach Alley	5	
16	Jessey	Decelles	7	8	1		1	hon	ne	Merrowme	adow	Freybeach Street	85	
17	Hilda	Lees	8	10	1		1	wo	rk	Barrowedg	)e	Penn Street	8	
18	Mark	Marzolla	9	12	1		1	hon	ne	Elfville		Creek Road	3	
19	Dannie	Vignola	10	13	1		1	hon	ne	Roseford		Bowman Ave.	853	
20	Lanita	Krysiak	11	14	1		1	wo	rk	Beardale		Iroquois Street	98	
			12	17	1		1	hon	ne	Bridgedell		Smith Road	7	
			13	18	1		0	hon	ne	Roseford		Wood Street	7	
			14	18	0		1	wo	rk	Johnson C	ity	Thorne Lane	9677	
			15	20	1		1	hon	ne	Mechanics	ville	Vine Street	9065	

Chaque enregistrement d'utilisateur dans la table users peuvent avoir zéro ou plusieurs adresses dans la table addresses. Par exemple, un utilisateur peut avoir une adresse de type "home", ou deux adresses (une adresse de type "maison" et une autre adresse de type "travail"), ou aucune adresse.

L'objectif du mappage est d'extraire les données complètes (prénom, nom de famille, e-mail, ville, rue, numéro) de tous les utilisateurs ayant au moins une adresse dans la table d'adresses. Il devrait également être possible d'extraire aisément uniquement les adresses d'un certain type (par exemple, uniquement les adresses "maison" ou les adresses "travail"). Le type d'adresses à extraire ("maison" ou "travail") doit être fourni en tant que paramètres du mappage. Les enregistrements extraits des personnes doivent être triés alphabétiquement par le nom de famille.

Les exigences de mappage seront réalisées à l'aide d'un composant Join, comme illustré dans les étapes cidessous.

Note : L'utilisation d'un composant Join n'est pas le seul moyen de joindre des tables ou des affichages de base de données. Les jointures applicables aux bases de données peuvent aussi être effectués en utilisant des instructions SELECT SQL, voir <u>Instructions SELECT SQL en tant que tables virtuelles</u> <sup>322</sup>. Une différence principale entre les instructions SELECT SQL et les composants Join est que les premières sont écrites à la main, ce qui leur confère une plus grande flexibilité. Les composants Join sont une alternative plus simple si vous ne souhaitez pas rédiger des instructions SQL à la main.

# Étape 1: Ajouter la base de données de source

- Dans le menu Insérer, cliquer sur Base de données. (En alternative, cliquer sur la touche de la barre d'outils Insérer la base de données <sup>[]</sup>
- 2. Sélectionner "SQLite" en tant que type de base de données, et cliquer sur Suivant.
- 3. Chercher le fichier Nanonull.sqlite disponible dans le dossier : <br/>
  <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\, et cliquer sur Se connecter.
- 4. Sur invitation, choisir les tables addresses et users.



# Étape 2: Ajouter le composant join

- Dans le menu Insérer, cliquer sur Join. (En alternative, cliquer sur touche de la barre d'outils Join 2.
   ).
- 2. Tirer la connexion depuis la table users vers la première entrée du composant join.
- 3. Agrandir la table users et tirer une connexion depuis la table addresses (enfant de users) vers la deuxième entrée du composant join. La touche vous permet d'ajouter plus de tables si nécessaire ; néanmoins, dans cet exemple, seules deux tables sont jointes.



- **Note :** Il est également possible d'ajouter la connexion directement depuis la table addresses (celui qui n'est pas enfant d'users) ; néanmoins, dans ce cas, les conditions join devront être définies manuellement, comme décrit dans <u>Ajouter les conditions Join</u><sup>507</sup>. Dans l'objectif de cet exemple, veuillez vous assurer de créer les connexions comme indiqué ci-dessus. Cela garantit que la condition join requise est créée automatiquement.
  - 4. Cliquer sur la touche **Définir la condition Join** is disponible sur le composant join. Veuillez noter que la condition join a été créée automatiquement (users.id = addresses.user\_id).

😢 Define Join Condition		×
Join structures when all of the following conditions are true:		<u>A</u> dd Condition
🗊 users.id	= 🛯 addresses.user_id	^
		~
,		
The selected condition is from a foreign key relation and cannot be	modified.	
Structure 1:	Structure 2:	
🖸 🛄 users	^	$\sim$ 4 F
Inst_name	in addresses user_io	^
🗊 email	user_id = users.id	
····· III created_at	is_shipping	
III updated_at	III is_billing	
	ity city	
	······ II street	
	🗊 number	

# Étape 3: Ajouter le composant CSV cible

- 1. Dans le menu **Insérer**, cliquer sur **Fichier texte**. (En alternative, cliquer sur la touche de la barre outils**Insérer fichier de texte**).
- 2. Lorsque vous êtes invité à choisir un mode de traitement de texte, choisir Utiliser le traitement simple pour le CSV standard...
- 3. Cliquer sur **Apposer champ** plusieurs fois pour créer sept champs CSV. Laisser les autres paramètres tels qu'ils sont.

Field1		Field2		Field3		Field4		Field5		Field6		Field7	
string	-	string	•	string	-	string	-	string	<b>T</b>	string	-	string	-
S													2
Append Fie	eld	l <u>n</u> sert Field	Rem	nove Field	<<	>>							

4. Double-cliquer sur la cellule de titre de chaque champ pour lui donner un nom descriptif (cela rendre votre mappage plus facile à lire).

.

. <b>▼</b> stri	ng 💌	string	<b>▼</b> 5	tring 💌	string 👤	string	Ŧ
eld <u>R</u> emove	Field <<	>>					
1	ìeld <u>R</u> emove	ield <u>R</u> emove Field <<	ield <u>R</u> emove Field << >>				

5. Tirer les connexions de mappage entre le composant Join et le composant CSV comme indiqué cidessous. La connexion entre l'item joined du composant join et l'item Rows du composant de cible "créer autant d'enregistrements (lignes) dans la cible étant donné qu'il y a des enregistrements qui satisfont à la condition join".



### Étape 4: Ajouter la condition WHERE/ORDER SQL et le paramètre d'entrée

- 1. Cliquer avec la touche droite dans la connexion entre l'item joined du composant Join et l'item Rows du composant CSV cible, et choisir **Insérer WHERE/ORDER SQL**.
- 2. Saisir les clauses WHERE et ORDER BY comme indiqué ci-dessous.

SQL-WHERE/ORDER Properties							
To create a parameter, enter a ':' followed by its name in the WHERE condition. Sample: Name = :Name AND Age >= :ValidAge SELECT (fields) FROM addresses, users WHERE							
addresses.type	= :address_type	^					
Darameters:		~					
Parameter	Тура						
address type	string	<b>T</b>					
ORDER BY							
last_name		^					
		$\sim$					
	ОК Са	ncel					

3. Dans le mappage, ajouter un composant d'entrée (en utilisant la commande de menu **Insérer | Insérer entrée**) et connecter sa sortie au paramètre address\_type créé dans l'étape précédente.



4. Double-cliquer sur le composant d'entrée et le configurer comme indiqué ci-dessous. Une valeur de design-time est nécessaire (dans ce cas, "home") pour consulter la sortie de mappage dans MapForce. Si vous souhaitez que l'aperçu extraie les adresses de travail, remplacer cette valeur par "work".

🙁 Edit In	put X
<u>N</u> ame:	input
<u>D</u> atatype:	string ~
	Input is required
Design-ti	me Execution
✓ Spec	ify value
<u>V</u> alue:	home
	OK Cancel

### L'explication de mappage

La condition join créée automatiquement dans l'étape 2 garantit que seuls des enregistrements qui satisfont à la condition join users.id = addresses.user\_id sont copiés dans la cible. La condition join est ajoutée automatiquement parce que les deux tables sont liées par une relation étrangère clé et les connexions de mappage ont été tirées de manière appropriée. Pour plus d'informations concernant les relations de table, voir <u>Gérer des relations de base de données</u><sup>260</sup>. Puisque cet exemple a été fait en utilisant les relations de table déjà existantes, vous n'avez pas dû définir de conditions join manuellement, voir <u>Exemple : Créer un rapport CSV depuis</u><sup>351</sup>.

Les deux tables de source proviennent de la même base de données et du même composant, ce qui permet de réunir des avantages du mode SQL ( <sup>SQL</sup> ). Puisque le mode SQL est activé, l'opération join est entreprise par la base de données, et non pas par MapForce. En d'autres termes, une instruction INNER JOIN est générée internement par MapForce et est envoyée à la base de données pour l'exécution. Le type du join (INNER JOIN) est indiqué par l'icône **Join intérieur** en face de la table **addresses** dans le composant Join. Vous pouvez aussi modifier le type join en LEFT OUTER JOIN », comme décrit dans <u>Changer le Mode Join</u> <sup>(40)</sup>. Noter, néanmoins que la modification du mode join n'a pas d'effet sur la sortie de cet exemple.

Le composant WHERE/ORDER SQL ajouté dans l'étape 4 permet le filtrage (afin d'extraire soir les adresses home (maison) soit work (travail)) et trier les enregistrements. Veuillez noter que la clause WHERE a créé un paramètre **:address\_type** de type string. Ce paramètre permet de fournir le type d'adresse (home (maison) soit work (travail) depuis le mappage. Pour plus d'information à propos de WHERE/ORDER SQL, voir <u>Filtrer et trier les données de base de données (SQL WHERE/ORDER)</u><sup>355</sup>.

Enfin, le composant d'entrée permet de fournir la valeur de paramètre réelle lorsque le mappage est exécuté. Veuillez noter que, lorsque le mappage est exécuté en-dehors de MapForce (par exemple, lorsqu'il est exécuté par MapForce Server sur un autre appareil), l'entrée doit être fournie au moment de l'exécution du mappage en tant que paramètre de ligne de commande, afin que la valeur design-time mentionnée ci-dessus soit ignorée. Pour plus d'informations, voir Fournir des paramètres au mappage<sup>477</sup>.

# 4.2.4.3 Exemple : Créer un rapport CSV depuis plusieurs tables

Cet exemple montre comment joindre plusieurs tables de base de données afin d'extraire des données dans un seul rapport en format CSV. La base de données utilisée dans cet exemple est appelée **Nanonull.sqlite** et est disponible dans le chemin suivant : **<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\**. Cette base de données stocke des informations à propos d'un commerce fictif (comprenant des commandes, des produits, des utilisateurs et leurs adresses). Comme cela est souvent le cas avec des base de données relationnelles, l'information est normalisée et distribuée sur plusieurs tables. Par exemple, la table users stocke des données personnelles d'utilisateur (qui contiennent le prénom, le nom de famille et l'e-mail). La base de donnée stocke également des informations concernant des produits commandés par les utilisateurs, dans deux tables différentes : orders (qui inclut l'ID unique de la commande et la date à laquelle elle a été passée) et orderedproducts (qui inclut une liste de produits commandés, et leurs quantités). De plus, les noms des produits eux-mêmes sont stockés dans une table séparée appelée products.

Le but de cet exemple est de produire un rapport sur la base des données extraites depuis des tables variées, afin de montrer clairement qui a commandé certains produits, quand et en quelles quantités. Pour atteindre l'objectif du mappage, suivre les étapes suivantes :

- 1. Dans le menu Insérer, cliquer sur Base de données.
- Lorsque vous êtes invité à choisir un type de base de données, cliquer sur SQLite, puis cliquer sur Suivant.
- 3. Chercher la base de données Nanonull.sqlite mentionnée ci-dessus et cliquer sur Se connecter.
- 4. Sur invitation, choisir les tables orderedproducts, orders, products, et users, et cliquer sur OK.



- 5. Ajouter un composant Join au mappage et créer quatre items nodes/rows en cliquant sur la touche **Ajouter entrée** ( ).
- 6. Connecter les quatre tables depuis le composant de base de données dans les items entrée correspondants du composant Join.



**Note :** Dans un autre scénario, vous pouvez connecter la table orderedproducts au composant Join, ensuite la table orders (celle qui est imbriquée en-dessous, pas celle se trouvant au même niveau), etc. afin

que toutes les tables jointes sont imbriquées sous la même table "racine", voir aussi <u>Gérer des</u> relations de base de données<sup>200</sup>. Le résultat de mappage serait le même que si vous joignez les tables de cette manière. La différence est que dans cet exemple, les conditions join sont créées manuellement, alors que dans le scénario alternatif les conditions join sont créées automatiquement par MapForce. Pour consulter un exemple de jointure des tables sans devoir définir des conditions join manuellement, voir <u>Exemple</u>: Joindre des tables en mode SQL<sup>342</sup>. Un autre mappage où toutes les tables jointes se trouvent sous la même table "racine" est disponible dans le chemin suivant : <**Documents-\Altova\MapForce2023\MapForceExamples\DB\_Denormalize.mfd**.

Dans cet exemple, les tables connectées au composant Join sont classées dans l'ordre suivant :

- 1. orderedproducts
- 2. orders
- 3. products
- 4. users

Cet ordre touche la manière dont les structures respectives sot affichées dans le dialogue "Définir une condition Join", lorsque vous cliquez sur la touche **Définir la Condition Join** ( ). Concrètement, la première table (orderedproducts) apparaît par défaut sous **Structure 1**, et la table juste après (orders) apparaît sous **Structure 2**.

😢 Define Join Condition			×
Join structures when all of the following conditions are true:			Add Condition
🥝 <undefined></undefined>	= (	oundefined>	<b>X</b>
			$\sim$
To define a condition, select an item from each structure to compare	e for equa	lity.	
Structure 1:		Structure 2:	
Im ordered products	<u>^</u>	orders	~ + F
□\" order_id		Generation	^
y product_ia			
		user_iu	
		In undated at	

Pour définir la première condition join, cliquer sur l'item order\_id dans le volet gauche et l'item id dans le volet de droite. À présent, les champs orderedproducts.order\_id et orders.id. sont jumelés :

😢 Define Join Condition		×
Join structures when all of the following conditions are true:		<u>A</u> dd Condition
orderedproducts.order_id	= 🗊 orders.id AS id2	X

Jusqu'à présent, seules deux tables ont été jointes. Afin de définir des conditions join qui impliquent une troisième table, sélectionner la table désirée depuis la liste déroulante disponible au-dessus du volet de droite. Dans ce cas, le panneau de gauche affiche toutes les tables qui se produisent *avant* dans le composant Join.

Par exemple, si vous choisissez products dans le côté droit, le volet de gauche affiche orderedproducts et orders (étant donné que ces tables se produisent avant products dans le composant Join). Vous pouvez maintenant jumeler des paires de tables products avec des champs de tables les précédant (dans ce cas, orderedproducts.products.id).

😢 Define Join Condition		×
Join structures when all of the following conditions are true:		Add Condition
orderedproducts.order_id	= 🗓 orders.id AS id2	XA
AND 🕎 orderedproducts.product_id	= 🛙 📲 products.id AS id3	X
To define a condition, select an item from each structure to compare for	equality.	V
Structures 1 and 2:	Structure 3:	
Image: ordered products       ^         Image: order_id       Image: order_id         Image: order_id       Image: order_id	products	

Pour joindre une quatrième table (users), choisir la table users depuis la liste déroulante. Vous pouvez maintenant jumeler les champs orders.user\_id et users.id.

😢 Define Join Condition		×
Join structures when all of the following conditions are true:           If orderedproducts.order_id           AND         If orderedproducts.product_id           AND         If orders.user_id	Add = II orders.id AS id2 = II products.id AS id3 = II users.id AS id4	Condition
To define a condition, select an item from each structure to compare for each	quality.	
Contactor 1 to 51	Users	
< >	<	>
Show <u>a</u> nnotations	OK	Cancel

Maintenant que toutes les conditions join requises ont été définies, les items du composant Join peuvent être mappés dans un composant de cible. Pour terminer le mappage, ajouter un composant CSV (voir <u>CSV et</u> <u>fichiers de texte</u><sup>440</sup>), et connecter les items depuis le composant Join vers le composant CSV cible comme illustré ci-dessous :



Le mappage illustré ci-dessus produit un rapport (en format CSV) compilé à partir des quatre tables comprises dans la jointure, comme suit :

- ID de la commande (prélevée dans la table orderedproducts)
- Quantité des items commandés (prélevée dans la table orderedproducts)
- Moment auquel la commande a été passée (prélevé dans la table orders)
- Nom du produit commandé (prélevé dans la table products)
- Prénom et nom de famille de l'utilisateur qui a commandé le produit (prélevés dans la table users).

Toutes les tables présentées dans cet exemple sont jointes en utilisant le mode INNER JOIN. Pour plus d'information concernant la modification du mode join en LEFT OUTER JOIN, voir <u>Changer le mode join</u><sup>340</sup>.

# 4.2.5 Filtrer et trier les données de base de données (WHERE/ORDER SQL)

Lorsque vous souhaitez filtrer et trier des données de base de données, utilisez un composant **SQL/NoSQL-WHERE/ORDER**. Ceci vous permet de saisir manuellement une clause SQL WHERE qui filtre les données. En

option, vous pouvez aussi spécifier une clause ORDER BY si vous souhaitez trier les enregistrements par un champ de base de données particulier, dans un ordre ascendant ou descendant.

Le composant **SQL/NoSQL-WHERE/ORDER** doit être connecté à une table ou un champ du composant de mappage de base de données. Il est également possible de connecter un composant **SQL/NoSQL-WHERE/ORDER** avec un composant **Join** si vous avez besoin de filtrer un ensemble rejoint ou des enregistrements. Pour plus d'informations, voir <u>Rejoindre les données de base de données</u><sup>333</sup>.

### Ajouter un composant SQL/NoSQL-WHERE/ORDER

Pour ajouter un composant SQL/NoSQL-WHERE/ORDER au mappage, suivez les instructions suivantes :

1. Allez au menu **Insérer** and cliquez sur **SQL/NoSQL-WHERE/ORDER**. Par défaut, le composant a la structure suivante :

	🔀 sql-wher	e/order
Ľ	>table/field	result

2. Connecter une table ou un champ de base de données de source à l'item table/field du composant SQL/NoSQL-WHERE/ORDER. Vous pouvez trouver un mappage échantillon FilterDatabaseRecords.mfd (voir la capture d'écran ci-dessous) dans le dossier suivant : Composant SQL/NoSQL-WHERE/ORDER prend les données de la table source utilisateurs, filtre tous ses enregistrements et sélectionne uniquement ceux où le last name commence par la lettre M (voir l'explication dans la sous-section ci-dessous).



 Double-cliquez sur l'en-tête du composant SQL/NoSQL-WHERE/ORDER. De manière alternative, cliquez avec la touche droite sur la barre de l'en-tête et sélectionnez Propriétés depuis le menu contextuel. Cela ouvre la boîte de dialogue Propriétés SQL/NoSQL-WHERE/ORDER.

🕲 Proprie	étés SQL/NoSQL-WHERE/ORDER	×
Pour créer un Modèle : Nom SELECT (cham ast_name LIK	paramètre, saisir un ':' suivi par son nom dans la condition WHERE. = :Name ET Age >= :ValidAge ps) FROM users WHERE E :sqlparam	
<		>
Paramètres :		
Paramètre	Туре	
sqlparam	string	•
ORDER BY		
last_name DE	SC	
<		>
	OK · Annule	r

4. Saisir la clause WHERE SQL dans le champ de saisie en haut. Dans notre exemple, la clause SQL Where est la suivante : last\_name LIKE :sqlparam. En option, saisir la clause ORDER BY. L'image ci-dessus illustre les clauses WHERE et ORDER BY définies dans le mappage FilterDatabaseRecords.mfd (ces paramètres sont expliqués plus bas). Pour plus d'exemples, voir Créer des clauses WHERE et ORDER BY

### Paramètres dans les composants SQL/NoSQL-WHERE/ORDER

Le composant **SQL/NoSQL-WHERE/ORDER** est utilisé dans le mappage **FilterDatabaseRecords.mfd** a une clause WHERE suivante : **last\_name LIKE :sqlparam**, où last\_name se réfère au nom de champ de la base

de données dans la table connectée ; LIKE est un opérateur SQL ; :sqlparam crée un paramètre appelé sqlparam dans le mappage.

Les paramètres dans le composant **SQL/NoSQL-WHERE/ORDER** sont optionnels. Ils sont utiles si vous souhaitez passer une valeur dans la clause WHERE depuis le mappage. Sans paramètres, la clause WHERE ci-dessus pourrait avoir été écrite comme suit : Last LIKE "M%". Ceci extrairaient toutes les personnes dont le nom de famille commence par la lettre M. En vue de rendre cette requête même plus flexible, nous avons ajouté un paramètre au lieu de "M%". Cela permet de fournir d'autres lettres depuis le mappage, par ex., D, et donc d'extraire les personnes dont le nom de famille commence avec D simplement en changeant une constante, ou un paramètre d'entrée de mappage. Dans le mappage ci-dessus, la lettre d'entrée provient d'un simple composant d'entrée appelé input. Si vous double-cliquez sur la barre de titre de ce composant et que vous ouvrez ses propriétés, vous remarquerez que m est donné en tant que valeur d'exécution au moment du design (*voir la capture d'écran ci-dessous*) :

🕲 Créer Entrée 🛛 🛛 🗙		<
Nom :	input	
Type de données:	string	~
	✓ Entrée requise	
Exécution	n Design-time	
✓ Spécifier la valeur		
Valeur :	m	]
	OK Annuler	

Dans le mappage, le caractère générique % est fourni par un constant. Ce caractère générique est concaténé avec une valeur de paramètre à l'aide de la fonction **concat**. L'avantage étant que vous ne devez pas saisir de caractère générique dans la ligne de commande si le mappage est exécuté dans un autre environnement (par ex., le serveur MapForce Server).

### Apparence de composants SQL/NoSQL-WHERE/ORDER

Des composants **SQL/NoSQL WHERE/ORDER** changent leur apparence dépendant de leurs paramètres y définis. Ainsi, vous pouvez consulter rapidement directement depuis le mappage ce que le composant **SQL/NoSQL WHERE/ORDER** fait (*voir la table ci-dessous*)

table/field result	Une clause WHERE avec un paramètre a été définie.
e table/field result e	

♥ users table/field sqlparam	Une clause WHERE avec un paramètre a été définie. Le paramètre name est visible sous l'item table/field.
table/field	Une clause WHERE avec un paramètre a été définie. De plus, une clause ORDER
sqlparam	BY a été définie. Le tri est indiqué par l'icône de tri A-Z.

Placer le curseur de la souris sur l'en-tête **SQL/NoSQL WHERE/ORDER** ouvre une infobulle affichant les différentes clauses qui ont été définies.

# 4.2.5.1 Créer des clauses WHERE et ORDER BY

Une fois qu'un composant **SQL/NoSQL-WHERE/ORDER** est ajouté au mappage, celui-ci nécessite une condition WHERE (clause) par le biais duquel vous pouvez spécifier comment vous souhaitez filtrer les données connectées au composant. La condition WHERE doit être saisie dans le dialogue **Propriétés SQL/NoSQL-WHERE/ORDER** (*voir la section précédente*).

L'écriture d'une condition WHERE depuis MapForce est semblable à l'écriture de la même clause SQL en dehors de MapForce. Veuillez utiliser la syntaxe applicable au dialecte SQL de la base de données correspondante. Par exemple, vous pouvez utiliser des opérateurs, des caractères génériques et des sous-sélections ou des fonctions d'agrégat. Pour créer des paramètres que vous pouvez passer depuis le mappage, saisir un double-point (:) suivi par le nom du paramètre.

**Note :** Lorsque vous terminez l'écriture de la clause WHERE et cliquez sur OK, MapForce valide l'intégrité de l'instruction SQL finale. Un dialogue vous relance en cas d'apparition d'erreurs de syntaxe.

La table ci-dessous recense certains des opérateurs typiques qui peuvent être utilisés dans la clause WHERE :

Opérateur	Description
=	Égal
<>	Non égal
<	Inférieur à
>	Supérieur à
>=	Supérieur à/égal
<=	Inférieur à/égal
IN	Extrait une valeur connue d'une colonne
J'AIME	Cherche un motif particulier

Opérateur	Description
BETWEEN	Cherche dans une fourchette

Utilise le caractère générique % (pourcentage) pour représenter tout nombre de caractères dans un motif. Par exemple, pour extraire tous les enregistrements se terminant en "r" depuis un champ appelé lastname, utilisez l'expression suivante :

lastname = "%r"

Lorsque vous requêtez des bases de données qui prennent en charge le stockage et la requête des données de base de données XML (par exemple, IBM DB2, Oracle, SQL Server), vous pouvez utiliser des fonctions et des mots-clés XML applicables à cette base de données particulière, par exemple :

xmlexists('\$c/Client/Address[zip>"55116"]' passing USER.CLIENTS.CONTACTINFO AS "c")

Voir aussi Exemple : Extraire des données depuis les colonnes IBM DB2 de type XML<sup>379</sup>.

En option, si vous souhaitez trier les enregistrements par un champ particulier, ajoutez une clause ORDER BY dans le champ de texte correspondant du dialogue **Propriétés SQL/NoSQL-WHERE/ORDER**. Pour trier par plusieurs champs, séparez les noms de champ avec des virgules. Pour changer l'ordre du tri, utilisez les motsclés ASC et DESC. Par exemple, la clause ORDER BY suivante extrait des enregistrements par lastname, puis par firstname, dans l'ordre descendant :

lastname, firstname DESC

### Exemple 1

La condition WHERE suivante est attachée à la table utilisateurs du composant de base de données Nanonull.sqlite. Elle extrait les enregistrements dans lesquels last\_name sont supérieurs à la lettre "M". En d'autres termes, il extrait tous les noms depuis "Marzolla".

last\_name > "M"

Notez comment les connexions sont placées :

- La connexion vers table/field provient dans la table que vous souhaitez interroger, (utilisateurs dans ce cas).
- La sortie result est connectée à un item "parent" des champs qui sont requêtés/filtrés (dans ce cas l'item row).


#### Exemple 2

La condition WHERE suivante crée un paramètre param qui apparaît ensuite dans le composant WHERE/ORDER SQL dans le mappage.



Le composant de constante %S fournit la valeur du paramètre param. Le caractère générique % dénote un nombre quelconque de caractères. Le mappage cherche donc un motif dans la colonne last\_name (tous les noms de famille commençant par "M").

#### Exemple 3

La condition WHERE suivante crée deux paramètres, min et max, auxquelles les valeurs actuelles de quantity sont comparées. Les valeurs min et max sont fournies par deux composants constants du mappage.

```
quantity > :min and quantity < :max</pre>
```



La condition WHERE dans cet exemple peut aussi être écrite en utilisant l'opérateur BETWEEN :

quantity BETWEEN :min and :max

# 4.2.6 Instructions SELECT SQL en tant que tables virtuelles

MapForce prend en charge la création des instructions SELECT SQL avec des paramètres dans les composants de base de données. Il s'agit de structures ressemblant à une table qui contient les champs de l'ensemble de résultat généré par l'instruction SELECT. Ces structures peuvent ensuite être utilisées en tant que source de données de mappage, comme toute autre table ou vue définie dans la base de données.

- Lors de l'utilisation des **jointures** Intérieur/Extérieur dans l'instruction SELECT, les champs de toutes les tables sont inclus dans le composant.
- Les expressions avec des noms de corrélation (en utilisant le mot-clé "AS" SQL) apparaissent aussi en tant qu'items mappable dans le composant.
- Les vues de base de données peuvent aussi être utilisées dans la clause FROM.
- Des instructions SELECT peuvent contenir des paramètres qui utilisent la même syntaxe que le composant <u>WHERE/ORDER SQL</u><sup>355</sup>.

Les instructions SQL SELECT sont prises en charge dans les langages C++, C#, Java et BUILT-IN, tant qu'ils ne contiennent pas de paramètres d'entrée. Les instructions SQL SELECT avec paramètres d'entrée sont uniquement pris en charge dans le langage de transformation BUILT-IN.

Une fois que l'instruction SELECT a été ajoutée à un composant de base de données, les champs retournés sont disponibles pour le mappage, par exemple :

eltova-products	
∋⊞ III Orders	ľ
SELECT_Statement	SELECT *, (Quantity*UnitPrice) AS FROM Orders INNER JOIN Products
Price	Ľ
OrderID	Ľ
Orders.ProductID	Ľ
OrderDate	ľ
🔲 Shipping	ľ
🔲 Quantity	ľ
Products.Product	ID (
ProductName	ľ
UnitPrice	ľ
Unitin Stock	Ľ

Le nombre des lignes visibles de l'instruction SELECT est configurable. Pour définir le nombre de lignes que vous souhaitez voir dans le composant, choisir la commande de menu **Outils | Options**, cliquer sur l'onglet **Généralités** et saisir le nombre de lignes dans le groupe Vue de mappage.

# 4.2.6.1 Créer des instructions SELECT

Vous pouvez créer des instructions SELECT sur un mappage qui contient un composant de base de données. Si votre mappage ne contient pas encore une base de données, ajouter tout d'abord une base de données (voir <u>Se connecter à une base de données</u><sup>(53)</sup>). Dans le but de cet exemple, choisir la commande de menu **Insérer | Insérer base de données** et suivre les étapes de l'assistant pour se connecter au fichier **altovaproducts.mdb** disponible dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\.

Pour créer une instruction SELECT :

 Cliquer avec la touche de droite sur le titre du composant de la base de données, et choisir Ajouter/Supprimer/Éditer des objets de base de données. (En guise d'alternative, sélectionner le composant de base de données, et puis sélectionner la commande de menu Composant | Ajouter/Supprimer/Éditer des objets de base de données).

# Insert Database Objects Please choose Tables, Views, Stored Procedures, or create your own SQL SELET stored locally in the mfd file. Source Image: Source

- 2. Suivre une des étapes suivantes:
  - Pour générer l'instruction SELECT depuis une table existante, cliquer avec la touche de droite sur une table et choisir Générer et ajouter une instruction SQL depuis le menu contextuel. Vous pourrez éditer l'instruction générée ultérieurement.
  - Pour écrire une instruction SELECT personnalisée, cliquer sur la touche Ajouter/Éditer instruction SELECT.
- Éditer ou créer l'instruction comme requis. Par exemple, l'instruction SELECT ci-dessous est valide pour le fichier altova-products.mdb disponible dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\. Le champ Price est le produit des deux champs, Quantity et UnitPrice, et est déclaré en tant que nom de corrélation (As Price).

```
SELECT *, (Quantity*UnitPrice) AS Price
From Orders
INNER JOIN Products
ON Orders.ProductID = Products.ProductID
Where Orders.Quantity > 2
```

#### 💀 Enter a SQL SELECT statement

Type in a single SQL SELECT statement to use as a source database object. Please make sure to choose the correct root object.

Auto-completion can be invoked holding Ctrl+space keys.

```
SELECT *, (Quantity*UnitPrice) AS Price
FROM Orders
INNER JOIN Products
ON Orders.ProductID = Products.ProductID
WHERE Orders.Quantity > 2
```

4. Cliquer sur **Ajouter instruction SELECT**. Veuillez noter que l'instruction SELECT est maintenant visible en tant qu'objet de base de données, semblable à la manière dont les tables, les vues et les procédures sont visibles.

🕞 Add/Remove/Edit Database Objects				
Please choose Tables, Views, Stored Procedures, o user are stored locally in the mfd file.				
Source				
💎 🔽 🔶 🛛 Add/Edit SELECT Statement				
- 🖓 📄 User Tables				
Address				
Altova				
Department				
Office				
······ ✔ Ⅲ Orders				
Person				
Products				
🖽 🗌 💼 System Tables				
🕀 🔽 💼 SELECT Statements				
SELECT_Statement				

5. Cliquer sur OK. L'instruction SELECT est aussi affichée dans le composant de base de données, et vous pouvez mapper des données depuis un des champs retournés par la requête SELECT.



Notes importantes :

- Toutes les expressions calculées dans l'instruction SELECT doivent avoir un nom de corrélation unique (comme "AS Price" dans cet exemple) pour être disponible en tant qu'item mappable.
- Si vous vous connectez à une base de données Oracle ou IBM DB2 en utilisant JDBC, l'instruction SELECT ne doit pas comporter de point-virgule final.

Pour supprimer une instruction SELECT ajoutée précédemment :

- 1. Cliquer avec la touche de droite sur le titre du composant de la base de données, et choisir Ajouter/Supprimer/Éditer des objets de base de données.
- 2. Cliquer avec la touche de droite sur l'instruction SELECT que vous souhaitez supprimer et choisir supprimer une instruction Select.

# 4.2.6.2 Exemple : SELECT avec des paramètres

Cet exemple vous montre comment créer un mappage MapForce qui lit des données depuis une base de données Microsoft Access et le rédige dans un fichier CSV. En particulier, le mappage décrit dans cet exemple utilise une requête SELECT de base de données personnalisée avec un paramètre. L'instruction SELECT combine des données depuis plusieurs tables. Ensuite, les résultats sont fournis au mappage pour des traitements ultérieurs.

L'exemple s'accompagne d'un fichier de design de mappage (.mfd) disponible dans le chemin suivant : **Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\select-component.mfd**. Vous pouvez ouvrir ce fichier d'échantillon et l'analyser tout d'abord ou suivre les étapes ci dessous pour la créer en partant de zéro.

Bien que cet exemple utilise une base de données Microsoft Access, le processus fonctionne de la même manière pour d'autres types de base de données. Pour plus d'informations concernant la connexion vers d'autres bases de données, voir <u>Connexion à une base de données</u><sup>158</sup>.

Les objectifs sont les suivants :

- Nous souhaitons sélectionner à partir de la base de données uniquement les commandes dont le nombre d'objets commandés dépassent une valeur personnalisée. Cette valeur personnalisée doit être fournie en tant que paramètre dans le mappage. Pour atteindre cet objectif, nous allons créer une instruction SELECT de base de données personnalisée qui prend un paramètre d'entrée.
- 2. Dans la base de données Access, le format de date est YYYY-MM-DD HH-MI-SS. Dans le fichier CSV, la partie de l'heure doit être réduite pour que le format soit YYYY-MM-DD. Pour atteindre cet objectif, nous utilisons la fonction <u>date-from-datetime</u><sup>(732)</sup> disponible dans MapForce.
- 3. Le fichier CSV résultat doit porter le nom **OrdersReport.csv**.

#### Étape 1: Ajouter la structure SELECT

- 1. Dans le menu Insérer, cliquer sur Base de données.
- Choisir Microsoft Access (ADO), et suivre les étapes de l'assistant pour se connecter au fichier altova-products.mdb disponible dans le dossier
   Cocuments>\Altova\MapForce2023\MapForceExamples\Tutorial\.



3. Dans le dialogue Insérer objets de base de données, cliquer sur **Ajouter/Éditer instruction SELECT**, et saisir la requête suivante :

```
SELECT *, (Quantity * UnitPrice) AS Price
FROM Orders
INNER JOIN Products
ON Orders.ProductID = Products.ProductID
WHERE Orders.Quantity > :Quantity
```

Cette requête utilise une jointure entre les tables Orders et Products, et extrait tous les champs (\*), et une valeur calculée (As Price). La requête spécifie aussi le paramètre :**Quantity** dans la clause WHERE.

🛃 Enter a SQL SELECT statement	×
Type in a single SQL SELECT statement to use as a source database object. Please make sure to choose the correct root object.	
SELECT *, (Quantity * UnitPrice) AS Price FROM Orders	
INNER JOIN Products ON Orders.ProductID = Products.ProductID	
WHERE Orders.Quantity > :Quantity	
Parameters: To create a parameter, enter a '' followed by its name in the statement above.	
Sample: SELECT (fields) FROM (table) WHERE (field)=:myParam	
Parameter	Datatype
Quantity	auto-detect
	~
Root Object:	$\sim$
Name: SELECT_Statement	
-	
Hestore SUL	Add SELECT Statement Cancel

- 4. Cliquer sur Ajouter l'instruction SELECT.
- 5. Cliquer sur OK. Le composant altova-products a maintenant été ajouté à la surface de mappage.
- 6. Dans le composant altova-products, cliquer sur 🕮 et choisir Insérer appel avec des paramètres.

🖯 altova-products	
SELECT_Statement	SELECT *, (Quantity * UnitPrice)
	Show Nodes as <u>S</u> ource
	Show Nodes as Target
	Insert Call with Parameters
	Edit SELECT Statement

Une nouvelle structure (SELECT\_Statement) est maintenant disponible dans le mappage. Il est partagé en deux parties : la partie gauche fournit des connecteurs d'entrée et la partie de droite fournit des connecteurs de sortie. Veuillez noter que la partie de gauche comprend le paramètre **Quantity** défini précédemment.



# Étape 2: Ajouter le paramètre d'entrée

- 1. Dans le menu Insérer, cliquer sur Insérer entrée.
- 2. Saisir "Quantity" en guise de nom.
- Sous Design-time Execution, saisir une valeur de paramètre à utiliser pour exécuter le mappage pendant la phase de design (dans cet exemple, "2"). Pour plus d'informations, voir Fournir des paramètres au mappage<sup>477</sup>.

🙁 Create	Input X
<u>N</u> ame:	Quantity
<u>D</u> atatype:	string ~
	Input is required
Design-ti	me Execution
Speci	fy value
<u>V</u> alue:	2
	OK Cancel

Vous pouvez connecter le paramètre d'entrée dans la structure d'appel de base de données, comme indiqué ci-dessous.



## Étape 3: Ajouter le composant CSV cible

- 1. Dans le menu Insérer, cliquer sur Fichier texte.
- 2. Choisir Utiliser le traitement simple pour le CSV standard..., et cliquer sur Continuer.
- Dans le dialogue Paramètres de composant, cliquer sur Apposer champ et ajouter neuf nouveaux champs. Nous vous recommandons de donner aux champs CSV le même nom que les champs de base de données, comme indiqué ci-dessous. Cela vous aidera à gagner du temps lorsque vous établirez les connexions de mappage. Pour plus d'informations concernant ces paramètres, voir <u>Configurer les options CSV</u><sup>455</sup>.

Price		OrderID		Producti	D	OrderDa	ate	Shipping		Quantity		ProductN	la	UnitPrice	•	UnitInSt	tock
string	-	string	•	string	•	string	•	string	•	string	▼	string	-	string	•	string	
c																	

4. Créer une connexion en tre le nœud **result** de la structure SELECT et le nœud **Rows** du composant CSV.



Étant donné que la plupart des champs dans le composant CSV porte le même nom que leur base de données équivalente, les connexions de mappage seront probablement établies lorsque vous connectez **result** à **Rows**. Si cela se produit, choisir le menu **Connexion** et s'assurer que l'option **Auto connexion des enfants correspondants** est activée. Le seul item de mappage que vous devrez connecter manuellement est **ProductID**, étant donné qu'il n'y a pas de champ portant ce nom dans la structure SELECT.

# Étape 4: Convertir la date

Dans la fenêtre Bibliothèques, chercher la fonction <u>date-from-datetime</u> et la glisser dans la surface de mappage. Ensuite connecter son entrée et sa sortie comme indiqué ci-dessous.



# Étape 5: Définir le nom du fichier de sortie

Pour définir le nom du fichier de sortie sur **OrdersReport.csv**, double-cliquer sur le composant CSV. Le dialogue Paramètres de composant s'ouvre, dans lequel vous pouvez saisir la valeur dans le champ de saisir **Fichier de sortie**.

# 4.2.7 Mapper des données XML de / vers des champs de base de données

MapForce vous permet de mapper des données depuis ou vers des champs de base de données (colonnes) qui stockent du contenu XML. Cela signifie que les données XML stockées par le champ de base de données (colonne) peuvent être extraites et écrites dans une autre structure prise en charge par MapForce, et inversement. Vous pouvez mapper les données comme suit :

- 1. Depuis ou vers les champs d'un type de XML dédié (par exemple, xml dans SQL Server, XMLType dans Oracle). La lecture ou l'écriture XML de/vers des champs XML dédiés est applicable aux bases de données qui ont une prise en charge native de XML (comme IBM DB2, Oracle et SQL Server).
- Depuis ou vers des champs de texte stockant du contenu XML (par exemple, Text, Varchar). Cela s'applique à toute base de données où le champ de texte a une longueur suffisante pour stocker un document XML.

Dans tous les cas, un schéma XML valide doit exister pour chaque colonne de base de données de/vers laquelle vous souhaitez mapper des données. Lorsqu'une colonne de base de données stocke du XML, MapForce vous permet de choisir d'attribuer un schéma XML directement depuis la base de données (si pris en charge par la base de données), ou de sélectionner un schéma depuis un fichier externe. Vous pouvez attribuer un schéma XML par colonne de base de données. Si le schéma possède plusieurs éléments racine, vous pouvez sélectionner un élément racine individuel de ce schéma.

Lorsque du XML est stocké en tant que champ de string dans une base de données, l'encodage de caractère du document XML est celui du champ de string sous-jacent. Si le champ de base de données ne stocke pas le texte en tant qu'Unicode, certains caractères ne pourront pas être représentés.

Certaines bases de données prennent en charge l'encodage XML pour les champs XML (qui n'est pas forcément le même que celui de l'ensemble des caractères de base de données). S'il est pris en charge par la base de données, il est convenu que la déclaration d'encodage du document XML sera celle déclarée dans le champ XML. Pour plus d'informations concernant la prise en charge d'encodage XML fournie par diverses bases de données, veuillez vous référer à leur documentation.

# 4.2.7.1 Attribuer un Schéma XML dans un champ de base de données

Cette rubrique illustre comment attribuer un schéma dans un champ défini nativement en tant que type XML dans la base de données. Les instructions ci-dessous utilisent SQL Server 2014 et l'Adventure Works 2014 de base de données. Ce dernier peut être télécharger depuis la page des échantillons AdventureWorks sur GitHub (<u>https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks</u>). Veuillez noter que le mappage des données de ou vers les champs XML fonctionne de la même manière qu'avec d'autres types de base de données qui prennent en charge les champs XML.

#### Pour ajouter la base de données Adventure Works 2014 en tant que composant de mappage :

Dans le menu Insérer, cliquer sur la Base de données, et suivre l'assistant pour se connecter à la base de données en utilisant votre méthode préférée (ADO ou ODBC). Pour plus d'informations, voir <u>Connexion à Microsoft SQL Server (ADO)</u> et <u>Connexion à Microsoft SQL Server (ODBC)</u>. NOTE : si vous utilisez le pilote SQL Server Native Client, vous pouvez souhaitez définir la propriété Integrated Security dans un espace de caractère (voir <u>Configurer les propriétés de liaison de données SQL Server</u>).

2. Dans le dialogue **Insérer l'objet de base de données**, agrandir le schéma **Production**, puis sélectionner la table **ProductModel**.



3. Cliquer sur OK.

La table de base de données a maintenant été ajoutée à la surface de mappage. Veuillez noter que cette table a deux champs de type XML : **CatalogDescription** et **Instructions**:

	sql_serv_2014 [AdventureWorks2014]
ιþē	Im Production.ProductModel Product model classification
	ProductModeIID Primary key for ProductModel record
Þ	🔟 Name Product model description.
Þ	CatalogDescription Detailed product catalog information
Þ	Instructions Manufacturing instructions in xml format
Þ	🔤 🔟 rowguid ROWGUIDCOL number uniquely identifying t
Þ	III ModifiedDate Date and time the record was last upd
L	

Pour que la structure des champs XML apparaissent dans le mappage, le schéma XML du contenu de champ est requis. Cliquer avec la touche de droite sur le champ **Instructions** et choisir **Attribuer Schéma XML dans le champ** depuis le menu contextuel.

sql_serv_2014 [AdventureWorks2	014]						
ProductModelID Primary key for ProductModel recort							
Name Product model description.							
>							
INSTRUCTIONS Manufactur	Assign XML Schema to Field						
> DI ModifiedDate Date and t	Unassign XML Schema from Field						
	Query Database						
	Delete Connections						
	Create Variable Component						
	Create variable Component						
	Component •						

Dans cet exemple particulier, vous allez attribuer un schéma au champ **Instructions** directement depuis la base de données. Pour ce faire, choisir l'item **Production.ManuInstructionsSchemaCollection** à côté de l'option **Base de données**, puis cliquer sur OK.

🕑 Assign XML Schem	na to Field	×				
XML Schema						
<u> Database:</u>	Production.ManuInstructionsSchemaCollecti					
© <u>F</u> ile:	Browse Edit					
<u>R</u> oot Element:	Choose					
Embedded Target XM	IL-Instance					
Prefix for target nam	espace:					
Cast values to tar values at risk of w	Cast values to target types (disable to preserve formatting of numeric or date values at risk of writing invalid output)					
Ualidate target XN	/IL in database					
Encoding						
Encoding name:	Unicode UTF-8					
Byte <u>o</u> rder:	Little Endian  Time Include Byte order mark					
Enable input procession	essing optimizations based on min/maxOccurs					
✓ Save all file paths	relative to MFD file					
	OK Cance					

La structure du champ XML apparaît maintenant dans le composant. Vous pouvez maintenant établir des connexions (et mapper des données) de ou vers ce champ.

<b>0</b> A	dventureWorks2014					
\$ <b>^</b> 5	Production.ProductModel Product model classificatic					
¢	🖫 ProductModelID					
\$	🔟 Name 🗘					
\$	CatalogDescription					
	E SHL Instructions					
₿	🗆 🕀 🗘 t:root					
P	Abc text() [>					
P	E () t:Location					
P	= LocationID					
P	SetupHours					
ß	MachineHours					
ß						
5						
2						
1	He tout()					
1						
ŗ	() timaterial					
1	() thluenrint					
1	() transce					
1	() trdian					
1	rowquid					
<b>v</b>	I ModifiedDate					

# 4.2.7.2 Exemple : Écrire des données XML dans un champ SQLite

Cet exemple vous guide à travers les étapes nécessaires pour créer un mappage de MapForce qui lit des données provenant de plusieurs fichiers XML et qui les rédige dans une base de données SQLite. L'objectif du mappage est de créer, pour chaque fichier de source XML, un nouvel enregistrement de base de données dans la base de données SQLite. Chaque enregistrement stockera le document XML en tant qu'un champ TEXTE.

Tous les fichiers utilisés dans cet exemple sont disponibles dans le chemin suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\. Les noms de fichier sont comme suit :

Le fichier de design de mappage	XmlToSqliteField.mfd
Les fichiers XML de source	<ul> <li>bookentry1.xml</li> <li>bookentry2.xml</li> <li>bookentry3.xml</li> </ul>
Le schéma XML utilisé pour validation	• books.xsd

La base de données SQLite cible	• Library.sqlite
---------------------------------	------------------

Pour atteindre l'objectif du mappage, les mesures suivantes seront prises :

- 1. Ajouter le composant XML et le configurer pour lire depuis plusieurs fichiers.
- 2. Ajouter le composant de la base de données SQLite et attribuer un schéma XML dans le champ TEXTE cible.
- 3. Créer les connexions de mappage et configure l'action INSERT de la base de données.

#### Étape 1: Ajouter le composant XML

- Dans le menu Insérer, cliquer sur Schéma XML/Fichier et chercher le schéma books.xsd situé dans le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\. Lorsque vous êtes invité à fournir un fichier XML échantillon, cliquer sur Sauter. Lorsque vous êtes invité à choisir un élément racine, choisir Books.
- Double-cliquer sur l'en-tête du composant et saisir bookentry\*.xml dans le champ Fichier XML d'entrée. Cela indique à MapForce de lire tous les fichiers XML dont le nom commence avec "bookentry-" dans le répertoire de source. Pour plus d'informations concernant cette technique, voir Traiter plusieurs fichiers d'entrée ou de sortie dynamiquement<sup>®33</sup>.

🕑 Component Settin	ıgs		×
Component name:	books		
Schema <u>fi</u> le			
lapForce2016\MapF	ForceExamples\Tutorial\books.xsd	Browse	Edi <u>t</u>
Input XML File			
bookentry*.xml		Browse	Edit

#### Étape 2: Ajouter le composant SQLite

Dans le menu **Insérer**, cliquer sur **Base de données**, et suivre l'assistant pour se connecter au fichier de base de données **Library.sqlite** provenant du répertoire

**Cocuments>\Altova\MapForce2023\MapForceExamples\Tutorial\** (voir aussi <u>Connexion à une base de données SQLite existante</u>). Lorsque vous êtes invité à choisir les objets de base de données, choisir la table BOOKS.

7 🗸	¢	Add/Edit	SELECT S
📮 🗹 🎦	Libra	<b>ry</b> SQLite	
• 🗗 🗹	🖯 ma	iin	
- 🔁 🖪	2 向 1	Tables	
÷-Ę	I 🗸 💼	User Table	s
	····· 🗸	III BOOKS	
·· 🕀 🗌	۱ 🧰 ۱	Views	
	1 🗀 🤅	SELECT State	ements

Le champ de base de données où le contenu XML sera écrit s'appelle metadata. Pour attribuer un schéma XML à ce champ, cliquer avec la touche de droite et sélectionner **Attribuer Schéma XML dans le champ** depuis le menu contextuel.



Dans ce tutoriel, le schéma attribué au champ metadata est le même que celui utilisé pour valider les fichiers XML de source. Cliquer sur **Chercher** et choisir le schéma **books.xsd** depuis le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\** :

۲	Assign XML Schem	a to Field	<b>×</b>
<b>ر</b> ا	XML Schema		
(	问 <u>D</u> atabase:		
(	• <u>F</u> ile:	:e2016\MapForceExamples\Tutorial\books.xsd	Browse Edit

Le schéma **books.xsd** possède deux éléments avec une déclaration globale : book et books. Dans cet exemple, nous allons définir book en tant que l'élément racine du XML écrit dans le champ de base de données. Cliquer sur **Choisir**, et sélectionner book en tant qu'élément racine :

Select root element		×
Name	Namespace URI	*
book		
books		

#### Étape 3: Créer les connexions de mappage et configurer l'action de base de données INSERT

Créer les connexions de mappage comme suit :



Comme indiqué ci-dessus, la connexion provenant de book vers book est une connexion "Copy-All", étant donné qu'aussi bien la source et la cible utilisent le même schéma et que les noms des éléments enfants sont les mêmes. Pour plus d'informations concernant ce type de connexions, voir <u>connexions Copy-all</u><sup>95</sup>.

La connexion supérieure (books vers BOOKS) itère par chaque élément de livre dans la source et écrit un nouvel enregistrement dans la table BOOKS. Cliquer sur la touche A: In sur le composant de base de données et définit les paramètres de mise à jour de la base de données comme affiché ci-dessous :

😢 Database Table Actions - BOOKS					
SQL statement to execute befo	re first record				
<ul> <li><u>N</u>one</li> <li><u>DELETE all records</u></li> <li><u>C</u>ustom SQL:</li> </ul>					
Actions to execute for each record					
All input data are compared to the DB table data, using the If all comparisons are true, then the specific action is exe					
Action on input data Insert All					
D¶ id	max() + 1				
III metadata	mapped value				

L'option **SUPPRIMER tous les enregistrements** instruit MapForce à supprimer les contenus de la table BOOKS avant d'insérer des enregistrements.

Les actions **Insérer toutes** spécifient qu'une requête INSERT de base de données aura lieu. Le champ id est généré depuis la base de données elle-même, alors que le champ metadata sera rempli avec la valeur fournie par le mappage.

Veiller à enregistrer le mappage avant de l'exécuter.

Pour exécuter le mappage et consulter l'entrée générée, cliquer sur l'onglet **Sortie**. Veuillez noter que cette action ne met pas à jour la base de données immédiatement. Lorsque vous êtes prêt à exécuter le script de base de données généré, choisir la commande de menu **Sortie | Exécuter Script SQL** (ou cliquer sur la

touche 陸 dans la barre d'outils).

# 4.2.7.3 Exemple: Extraire des données depuis les colonnes de type XML IBM DB2

Cet exemple montre comment extraire des données depuis des colonnes de base de données IBM DB2 de type XML et les écrire dans un fichier CSV cible. Il illustre également comment utiliser des instructions XQuery intégrées dans SQL afin d'extraire du contenu XML de manière conditionnelle. L'exemple requiert un accès à une base de données IBM DB2 pour laquelle vous avez la permission de créer et de remplir des tables.

Premièrement, il s'agit de préparer la base de données afin qu'elle contienne réellement des données XML. Pour ce faire, soit utiliser un outils d'administration de base de données spécifique à votre base de données, soit directement dans MapForce. Pour ce faire, suivre les étapes suivantes :

- 1. Créer un nouveau mappage et cliquer sur l'onglet DBQuery.
- 2. Cliquer sur **Connexion rapide** ( ) et suivre les instructions de l'assistant pour créer une nouvelle connexion de base de données (voir aussi <u>Database Connection Examples</u><sup>191</sup>).
- 3. Coller le texte suivant dans l'éditeur SQL. Cette requête SQL crée une table de base de données appelée ARTICLES et la remplit avec des données.

```
-- Create the table
CREATE TABLE
   ARTICLES (
      id INTEGER NOT NULL,
       article XML ) ;
-- Populate the table
INSERT INTO ARTICLES VALUES
   (1, '<Article>
      <Number>1</Number>
      <Name>T-Shirt</Name>
      <SinglePrice>25</SinglePrice>
   </Article>'),
(2, '<Article>
      <Number>2</Number>
      <Name>Socks</Name>
      <SinglePrice>230</SinglePrice>
   </Article>'),
(3, '<Article>
      <Number>3</Number>
      <Name>Pants</Name>
      <SinglePrice>34</SinglePrice>
   </Article>'),
 (4, '<Article>
      <Number>4</Number>
      <Name>Jacket</Name>
      <SinglePrice>5750</SinglePrice>
   </Article>');
```

4. Cliquer sur la touche **Exécuter** ( ). Le résultat d'exécution de la requête est affiché dans la fenêtre Résultats de requête. Si la requête est exécutée correctement, quatre lignes sont ajoutées à la table nouvellement créée.

Ensuite, nous allons créer un mappage qui extrait des données XML depuis la table ARTICLES créées cidessus de manière conditionnelle. L'objectif est d'extraire depuis la colonne ARTICLES uniquement les articles dont le prix est supérieur à 100.

#### Étape 1: Ajouter la base de données

- 1. Cliquer sur l'onglet Mappage pour retourner au volet de mappage.
- 2. Dans le menu **Insérer**, cliquer sur **base de données**, et suivre les instructions de l'assistant pour se connecter à la base de données.
- 3. Lorsque vous êtes invité à choisir les objets de la base de données, choisir la table ARTICLES créée précédemment.



#### Étape 2: Attribuer le schéma au champ de type XML

1. Cliquer avec la touche de droite sur l'item ARTICLE du composant, et choisir Attribuer Schéma XML au champ depuis le menu contextuel.



2. Choisir Fichier, et chercher le schéma suivant : Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\DB2xsd.xsd.

#### Étape 3: Ajouter le composant WHERE/ORDER SQL

- 1. Dans le menu Insérer, cliquer sur WHERE/ORDER SQL.
- 2. Connecter la colonne ARTICLE de type XML à l'entrée du WHERE/ORDER SQL.



3. Dans le dialogue Propriétés WHERE/ORDER SQL, saisie le texte suivant :

XMLEXISTS('\$a/Article[SinglePrice>100]' PASSING ARTICLE as "a")

 $\times$ 😢 SQL-WHERE/ORDER Properties To create a parameter, enter a ':' followed by its name in the WHERE condition. Sample: Name = :Name AND Age >= :ValidAge SELECT ARTICLE FROM PRODUCTS. ARTICLES WHERE XMLEXISTS('\$a/Article[SinglePrice>100]' PASSING ARTICLE as "a")

Le texte ci-dessus représente la partie "WHERE" de la requête SQL. Lors de l'exécution du mappage, cette partie sera combinées avec la partie "SELECT" affichée dans le dialogue. Cette instruction utilise la fonction XMLEXISTS et la syntaxe spécifiques aux bases de données IBM DB2.

#### Étape 4: Ajouter le fichier CSV cible

- 1. Dans le menu Insérer, cliquer sur Fichier texte.
- 2. Sur invitation, choisir Utiliser traitement simple pour CSV standard..., et cliquer sur Continuer.
- 3. Cliquer sur **Apposer champ** trois fois pour ajouter trois champs qui stockeront le numéro, le nom et le prix de l'article. Ne pas toucher aux autres paramètres.
- 4. Tirer les connexions de mappage comme indiqué ci-dessous.



Vous pouvez maintenant consulter un aperçu du mappage en cliquant sur l'onglet **Sortie**. Comme prévu, seuls des articles dont le prix est supérieur à 100 sont affichés dans la sortie.

1	2,Socks,230
2	4, Jacket, 5750

# 4.2.8 Parcourir et interroger des bases de données

MapForce détient un volet Requête de base de données (aussi appelé **Requête BD**) qui vous permet de requêter une base de données indépendamment du processus de mappage. Ce type de requête directe n'est pas enregistré avec le fichier \*.mfd de mappage mais propose un moyen pratique pour chercher ou modifier les contenus d'une base de données directement de MapForce.

Un volet **Requête BD** séparé existe pour chaque mappage actif actuellement. Vous pouvez créer plusieurs connexions actives, dans les différentes bases de données, dans le cadre de chaque volet **Requête BD**. Veuillez noter que les connexions créées depuis le volet **Requête BD** ne font pas partie du mappage et ne sont donc pas préservés une fois que vous fermez MapForce, à moins que vous les définissiez en tant que <u>Ressources globales</u><sup>971</sup>.

		Database	Browser							
Г	0	altous		ltown		. 88	N .*	t n	~	
		altova altova	bles ess a rtment e n Tables	1 2 3 4	SELEC [cit FROM	III (PrimaryK T [PrimaryK y], [state] [Address];	ey], [ , [str	QL Ed	itor	>
ĺ	1	6		,						٦
I		PrimaryKey •	ForeignKey •	city •	state •	street		•	zip •	
I	1	1	1	Vereno	CA	119 Oakstreet,	Suite 487	6	29213	
I	2	2	2	Brenton	MA	9865 Millenium (	Center, S	uite 456	5985	
	0				-	Query resu	ilts			
ŀ	0	Finished Retrie	val	Ro	ws: 2, Cols	: 6 0.047 sec		15:45:2	0	_
ł	Re	suits Message								_
L	M	apping    DB	Query Outp	nut 📀	HTML	1 RTF	PDF	0 W	or •	•
	R	esults tab			Messa	gestab				

Le volet Requête de base de données consiste les parties suivantes :

- Navigateur de base de données, qui affiche des informations de connexion et des tables de base de données
- Éditeur SQL, dans lequel vous rédigez vos requêtes SQL
- Onglet Résultats, qui affiche les résultats de recherche dans une forme tabulaire
- Onglet Messages, qui affiche des avertissements ou messages d'erreurs.

La partie supérieure du volet Requête de base de données contient les commandes de connexion vous permettant de définir des bases de données de travail, ainsi que les schémas de connexion et de base de données.

Pour configurer des paramètres de requête de base de données, utilisez la section *Base de données* dans la boîte de dialogue **Options**. Pour plus d'information, voir <u>Paramètres de requête de base de données</u><sup>1051</sup>.

# 4.2.8.1 Choisir ou se connecter à une base de données

Pour chaque base de données que vous souhaitez interroger, une connexion à une base de données doit être créée. Si votre mappage inclut déjà un composant de base de données, vous pouvez choisir la connexion de base de données existante depuis la surface supérieure du volet **Requête BD** (par défaut, la connexion est "Offline") et commencez à explorer les objets de base de données et à exécuter des requêtes.

Si votre mappage ne contient pas de composant de base de données, ou si vous souhaitez vous connecter à une nouvelle base de données, cliquer sur **Connexion rapide** ( $\bigcirc$ ) et suivre les instructions de l'assistant pour créer une nouvelle connexion de base de données (voir <u>Exemples</u><sup>(19)</sup>). Vous pouvez aussi choisir une connexion de base de données existante depuis Ressources globales, si une BD a été définie en tant que telle (voir <u>Ressources globales</u><sup>(97)</sup>).

Une fois que vous êtes connecté à la base de données, vous pouvez créer des requêtes de base de données en utilisant une des méthodes suivantes :

- Importer la requête SQL dans le volet Éditeur SQL depuis un fichier SQL existant.
- Rédiger la requête dans le le volet Éditeur SQL .
- Cliquer avec la touche de droite sur un objet dans le volet Navigateur de base de données et générer une requête (généralement, SELECT).

Lorsque vous êtes prêt à exécuter la requête affichée dans le volet Éditeur SQL, cliquer sur la touche **Exécuter** 

Les données de base de données sont extraites et affichées dans l'onglet Résultats sous forme de tableau. Veuillez noter que la barre de statut affiche le message "Extraction terminée" ( Finished Retrieval ), et d'autres informations pertinentes à propos des résultats de requête.

Une fois que le message "Extraction terminée" s'affiche, vous pouvez chercher, trier ou copier les résultats dans le presse-papiers (voir<u>Copier, trier, et chercher les résultats de requête</u><sup>334</sup>).

# 4.2.8.2 Créer et éditer des instructions SQL

L'Éditeur SQL est utilisé pour écrire et exécuter des instructions SQL. Il affiche toute instruction SQL que vous pouvez avoir généré automatiquement, chargé depuis des scripts SQL existants, ou écrit manuellement. L'éditeur SQL prend en charge le remplissage automatique (voir <u>Remplissage automatique</u><sup>230</sup>), régions, et commentaires de ligne ou bloc.

AltovaTarget 🔹	AltovaTarget	💌 🗖 🔁 🎍 👛 🗠 🖙 🦓 幕 🙆
AltovaTarget  AltovaTarget  AltovaTarget ADO  AltovaTarget  AltovaTarget  AltovaTarget  AltovaTarget  Altova  Altova	AltovaTarget  1 2 3 4 5 6 7	<pre></pre>
·····································	8	<pre>SELECT [PrimaryKey], [ForeignKey], [EMail], [First], [Last], [PhoneExt], [Title] FROM [Person];</pre>

La barre d'outils Éditeur SQL dispose des touches suivantes :

Basculer Navigateur : allume et éteint le volet Navigateur.

	Basculer Résultat : allume et éteint le volet Résultat.		
	<b>Exécuter (F5)</b> : cliquer sur cette touche pour exécuter les instructions SQL actuellement sélectionnées. Si plusieurs instructions existent et qu'aucune n'est sélectionnée, toutes seront exécutées.		
ŝ	Annuler : Vous permet d'annuler un nombre illimité d'éditions dans la fenêtre SQL.		
2	<b>Rétablir</b> : Vous permet de rétablir les commandes annulées précédemment. Vous pouvez feuilleter en avant et en arrière dans l'historique d'annulation en utilisant ces deux commandes.		
<b>.</b>	Importer fichier SQL : Ouvre un fichier SQL dans l'Éditeur SQL qui peut ensuite être exécuté.		
4	Exporter fichier SQL : Enregistre les requêtes SQL pour une utilisation ultérieure.		
٦	Ouvrir le script SQL dans DatabaseSpy : Lance DatabaseSpy et ouvre le script dans l'Éditeur SQL.		
<b>F</b>	<b>Options</b> : Ouvre le dialogue Options vous permettant de définir des paramètres de requête ainsi que des paramètres d'Éditeur SQL.		

# 4.2.8.2.1 Générer des instructions SQL

Des instructions SQL peuvent être générées automatiquement à partir du Navigateur de la base de données, chargées depuis des scripts, ou saisies manuellement.

# Pour générer les instructions SELECT SQL à partir du Navigateur de base de données, suivre une des étapes suivantes :

 Cliquer sur un objet de base de données (comme une table ou une vue), ou un dossier, dans le Navigateur de base de données et le glisser dans l'Éditeur SQL.



• Cliquer avec la touche de droite sur un objet de base de données dans le Navigateur de base de données et choisir Afficher dans l'Éditeur SQL | Sélectionner.

#### Pour créer des instructions SQL manuellement :

1. Commencer la saisie de l'instruction SQL dans l'éditeur SQL. Si le remplissage automatique est configuré pour se produire automatiquement, une liste déroulante contenant des suggestions apparaît pendant que vous saisissez une instruction.

1	SEL	
	SELECT	÷

2. Utiliser les touches **Haut** et **Bas** pour choisir une suggestion, puis appuyer sur **Entrée** pour insérer l'option marquée (voir aussi <u>Suggestions de remplissage automatique SQL</u><sup>280</sup>).

#### 4.2.8.2.2 Exécuter des instructions SQL

Les instructions SQL qui apparaissent dans l'éditeur SQL peuvent être exécutées avec un effet immédiat par rapport à la base de données. Le résultat de la requête SQL et du nombre de lignes concernées sont affichés dans le volet **Messages** du volet **Requête BD**.

Lorsque plusieurs instructions SQL apparaissent dans l'éditeur SQL, seules les instructions sélectionnées seront exécutées. Vous pouvez sélectionner des instructions SQL individuelles comme suit :

- Maintenir appuyée la touche de la souris, glisser le curseur sur une instruction spécifique.
- Cliquer sur un numéro de ligne dans l'éditeur SQL.
- Triple-cliquer sur une instruction spécifique.

#### Pour exécuter une instruction SQL :

- 1. Saisir ou cliquer l'instruction SQL dans l'éditeur SQL (voir <u>Générer des instructions SQL</u><sup>383</sup>).
- 2. Cliquer sur la touche **Exécuter** (**b**).

#### 4.2.8.2.3 Importer et exporter des scripts SQL

Vous pouvez enregistrer tout SQL qui apparaît dans une fenêtre d'Éditeur SQL dans un fichier et réutiliser le script ultérieurement.

#### Pour exporter les contenus du volet Éditeur SQL dans un fichier :

• Cliquer sur **Exporter fichier SQL** ( ), et saisir un nom pour le script SQL.

#### Pour importer un fichier SQL précédemment enregistré :

• Cliquer sur Import fichier SQL (

#### 4.2.8.2.4 Ajouter et supprimer des commentaires SQL

L'Éditeur SQL vous permet de décommenter des instructions, des parties d'instructions ou des groupes d'instructions. Ces instructions ou leurs parties respectives sont sautées lorsque le script SQL est exécuté.

#### Pour décommenter une section de texte :

- 1. Sélectionner une instruction ou une partie d'instruction.
- 2. Cliquer avec la touche de droite sur l'instruction sélectionnée et choisir **Insérer / Supprimer** commentaire en bloc.

#### Pour décommenter du texte ligne par ligne :

 Cliquer avec la touche de droite sur la position que vous souhaitez décommenter et sélectionner Insérer / Supprimer commentaire en ligne. L'instruction est décommentée à partir de la position actuelle du curseur jusqu'à la fin de l'instruction.

#### Pour supprimer un commentaire en bloc ou un commentaire en ligne :

- Sélectionner la partie de l'instruction qui est décommentée. Si vous souhaitez supprimer un commentaire en ligne, il suffit de sélectionner uniquement les marques de commentaire -- situées avant le commentaire.
- 2. Cliquer avec la touche de droite et sélectionner Insérer / Supprimer commentaire en bloc (ou ligne).

#### 4.2.8.2.5 Utiliser des signets

Les signets sont utilisés pour marquer des items intéressants dans de longs scripts.

#### Pour ajouter un signet :

 Cliquer avec la touche de droite sur la ligne que vous souhaitez marquer et choisir Insérer/Supprimer signet depuis le menu contextuel. 1

```
SELECT [PrimaryKey], [ForeignKey], [city], [state],
[street], [zip] FROM [Address];
SELECT [PrimaryKey], [Name] FROM [Altova];
SELECT [PrimaryKey], [ForeignKey], [Name] FROM
[Department];
SELECT [PrimaryKey], [ForeignKey], [Desc], [EMail],
[Established], [Fax], [Name], [Phone] FROM [Office];
SELECT [PrimaryKey], [ForeignKey], [EMail], [First],
[Last], [PhoneExt], [Title] FROM [Person];
```

Une icône de signet 💭 est affichée dans la marge au début de la ligne que vous souhaitez marquer.

#### Pour supprimer un signet :

• Cliquer avec la touche de droite sur la ligne dont vous souhaitez ôter le signet et choisir Insérer/Supprimer signet depuis le menu contextuel.

#### Pour naviguer entre les signets :

- Pour déplacer le signet vers le signet suivant, cliquer avec la touche de droite et choisir **Se rendre au signet suivant**.
- Pour déplacer le signet vers le signet précédent, cliquer avec la touche de droite et choisir Se rendre au signet précédent.

#### Pour supprimer tous les signets :

• Cliquer avec la touche de droite et choisir Supprimer tous les signets.

#### 4.2.8.2.6 Insérer des régions

Les Régions sont des sections de texte que vous marquez et déclarez comme unité afin de structurer vos scripts SQL. Les régions peuvent être agrandies et réduites pour afficher ou dissimuler des parties de scripts SQL. Il est également possible d'imbriquer des régions à l'intérieur d'autres régions.

Lorsque vous insérez une région, une icône d'agrandissement/réduction et un commentaire --region sont insérés au-dessus du texte sélectionné.

**Note :** vous pouvez modifier le nom d'une région en joignant un texte descriptif au commentaire --region. Le terme « région » ne doit pas être supprimé, par exemple, --region DB2query.

#### Pour créer une région :

1. Dans l'Éditeur SQL, choisir les instructions que vous souhaitez transformer en région.

2. Cliquer avec la touche de droite et choisir **Insérer Région** depuis le menu contextuel. Les instructions choisies deviennent une région qui peut ensuite être agrandie ou réduite.

```
1
2
      SELECT [PrimaryKey], [ForeignKey], [city], [state],
       [street], [zip] FROM [Address];
3
    -- region
4
      SELECT [PrimaryKey], [Name] FROM [Altova];
5
      SELECT [PrimaryKey], [ForeignKey], [Name] FROM
      [Department];
6
      -- endregion
7
      SELECT [PrimaryKey], [ForeignKey], [Desc], [EMail],
       [Established], [Fax], [Name], [Phone] FROM [Office];
8
      SELECT [PrimaryKey], [ForeignKey], [EMail], [First],
      [Last], [PhoneExt], [Title] FROM [Person];
9
```

3. Cliquer sur le symbole + ou - pour agrandir ou réduire la région.

#### Pour supprimer une région :

• Supprimer les commentaires -- region et -- endregion.

# 4.2.8.3 Parcourir des objets de base de données

Lorsque vous êtes connecté à une ou plusieurs bases de données, le volet **Navigateur de base de données** donne un aperçu complet des objets contenus dans chaque base de données, y compris des tables, des vues, des procédures, etc., jusqu'au niveau le plus détaillé. Pour des bases de données prenant XML en charge, le **Navigateur de base de données** montre également des schémas XML enregistrés dans un dossier séparé.

Pour une navigation personnalisée dans les objets de la base de données, le volet **Navigateur de base de données** comprend plusieurs mises en page d'affichage de base de données prédéfinis. Les mises en page prédéfinies sont disponibles dans la partie supérieure du Navigateur de base de données.



Pour choisir une mise en page, cliquer sur la touche déroulante **Mise en page dossiers** () et choisir une entrée depuis la liste. Veuillez noter que la touche change avec la mise en page sélectionnée.

- La mise en page **Dossiers** organise des objets de base de données dans des dossiers sur la base du type d'objet dans une arborescence hiérarchique, il s'agit du paramètre par défaut.
- La mise en page Pas de Schémas est semblable à la mise en page Dossiers, sauf qu'il n'y a pas de dossiers de schéma de base de données ; les tables ne sont donc pas catégorisées par un schéma de base de données.
- La mise en page Pas de dossiers affiche des objets de base de données dans une hiérarchie sans utiliser de dossiers.
- La mise en page **Plate** divise les objets de base de données en type dans le premier niveau hiérarchique. Par exemple, au lieu de colonnes contenues dans la table correspondante, toutes les colonnes sont affichées dans un dossier Colonnes séparé.
- La mise en page **Dépendance de table** catégorise des tables conformément à leurs relations avec d'autres tables. Il existe des catégories pour des tables avec des clés étrangères, des tables référencées par des clés étrangères et des tables qui n'ont pas de relation avec d'autres tables.

Outre la navigation dans la mise en page, vous pouvez utiliser le **Navigateur de base de données** pour les tâches suivantes :

- Générer des instructions SQL (voir <u>Générer des instructions SQL</u><sup>385</sup>).
- Filtrer et chercher les objets de base de données affichés (voir <u>Filtrer et chercher des objets de base</u> <u>de données</u><sup>(330)</sup>).
- Trier les tables en tables "System" et "User".
- Réinitialiser l'objet racine de la source de données active.

#### Pour trier les tables dans des tables Utilisateurs et Système :

• Dans le **Navigateur de base de données**, cliquer avec la touche de droite dans le dossier "Tables", puis sélectionner **Trier dans Tables Utilisateur et Système**.

**Note :** Cette fonction est disponible lorsqu'une des mises en page suivantes est sélectionnée : **Dossiers, Pas de schémas** ou **Plate**.

#### Pour réactualiser l'objet racine de la source de données active :

• En haut du Navigateur de base de données, cliquer sur Réactualiser ( 5).

#### 4.2.8.3.1 Filtrer et chercher des objets de base de données

Vous pouvez filtrer tout objet de base de données (schémas, tables, vues, etc.) affiché dans le **Navigateur de base de données** par le nom ou une partie du nom. Les objets sont filtrés pendant que vous saisissez les caractères. Le filtre n'est pas sensible à la casse par défaut. Le filtre n'est pas pris en charge si vous avez choisi la mise en page "Pas de dossiers".

#### Filtrer des objets de base de données

1. En haut du Navigateur de base de données, cliquer sur **Filtrer contenus de dossier** (). Les icônes de filtre apparaissent à côté de tous les dossiers dans la mise en page sélectionnée actuellement.



2. Cliquer sur l'icône filtre située à côté du dossier que vous souhaitez filtrer et choisir l'option de filtre depuis le menu contextuel (par exemple, **Contains**).



 Dans le champ vide qui apparaît à côté de l'icône du filtre, saisir le texte de recherche (par exemple, "G"). Les résultats sont ajustés pendant que vous saisissez le texte.



#### Chercher des objets de base de données

Pour trouver un item de base de données spécifique par son nom, vous pouvez soit utiliser les fonctions de filtre suivantes ou le **Localisateur d'objet**. Pour trouver les éléments de base de données se servant du Localisateur d'objet :

- 1. En haut du Navigateur de base de données, cliquer sur Localisateur d'objet ( 🔶 ).
- 2. Dans la liste déroulante qui apparaît, saisir le texte de recherche (par exemple, "Off").



3. Cliquer sur un objet dans la liste pour le choisir dans le Navigateur de base de données.

#### 4.2.8.3.2 Options contextuelles dans le Navigateur de base de données

Les options de menu contextuel disponibles dans le **Navigateur de base de données** dépendent de l'objet que vous avez choisir, par exemple :

- Cliquer avec la touche de droite sur l'objet "racine" pour Réactualiser la base de données.
- Cliquer avec la touche de droite sur un dossier permet de présenter toujours les mêmes choix : Agrandir | Frères | Enfants et Réduire | Frères | Enfants.
- Cliquer avec la touche de droite sur un objet de base de données permet de révéler la commande **Afficher dans l'Éditeur SQL** et les items de sous-menu discutés ci-dessous.

Pour sélectionner plusieurs objets dans plusieurs bases de données, appuient soit sur Shift + Clic ou Ctrl + Clic.

**Note** : La syntaxe des instructions SQL peut varier selon la base de données que vous utilisez. La syntaxe ci-dessous s'applique à Microsoft SQL Server 2014.

Les options suivantes sont disponibles sous le menu contextuel **Afficher dans l'Éditeur SQL** pour l'objet racine :

- CREATE : Crée une instruction CREATE pour l'objet racine de base de données sélectionnée, par exemple :
- CREATE DATABASE [MYDB]
- DROP : Crée une instruction DROP pour l'objet racine de base de données sélectionnée, par exemple
   :
   DROP DATABASE [MYDB]

Les options suivantes sont disponibles sous le menu contextuel Afficher dans l'Éditeur SQL pour les tables et les vues :

• SELECT : Crée une instruction SELECT qui extrait des données depuis toutes les colonnes de la table de source, par exemple :

```
SELECT [DepartmentID], [Name], [GroupName], [ModifiedDate] FROM [MYDB].
[HumanResources].[Department]
```

- **Name**: Retourne le nom de la table.
- **Chemin**: Retourne le chemin complet des tables, dans le format DataSourceName.DatabaseName.SchemaName.TableName.
- Si vous choisissez plusieurs tables, les noms ou les chemins sont imprimés sur des lignes individuelles, séparées par des virgules.

Les options suivantes sont disponibles sous le menu contextuel Afficher dans l'Éditeur SQL pour les colonnes :

- SELECT : Crée une instruction SELECT qui extrait des données depuis les colonnes sélectionnées de la table de parent, par exemple : SELECT [DepartmentID] FROM [MYDB].[HumanResources].[Department]
- **Nom** : Retourne le nom de la colonne sélectionnée.
- **Chemin**: Retourne le chemin complet de la colonne, dans le format DataSourceName.DatabaseName.SchemaName.TableName.ColumnName.

Si vous sélectionnez plusieurs colonnes, les noms et les chemins sont imprimés sur des lignes individuelles, séparées par des virgules.

Les options suivantes sont disponibles sous le menu contextuel **Afficher dans l'Éditeur SQL** pour les contraintes :

- Nom : Retourne le nom de la contrainte sélectionnée.
- **Chemin**: Retourne le chemin complet de la contrainte, dans le format DataSourceName.DatabaseName.SchemaName.TableName.ConstraintName.
- Si vous sélectionnez plusieurs contraintes, les noms et les chemins sont imprimés sur des lignes individuelles, séparées par des virgules.

Les options suivantes sont disponibles sous le menu contextuel Afficher dans l'Éditeur SQL pour les indexes :

- Nom : Retourne le nom de l'index sélectionné.
- **Chemin**: Retourne le chemin complet de l'index, dans le format DataSourceName.DatabaseName.SchemaName.TableName.IndexName.
- Si vous sélectionnez plusieurs indexes, les noms et les chemins sont imprimés sur des lignes individuelles, séparées par des virgules.

Si la base de données prend en charge les Schémas XML, les options suivantes sont disponibles pour chaque schéma affiché sous le dossier "Schémas XML" :

- Afficher dans XMLSpy: Ouvre le schéma de base de données dans XMLSpy, si le dernier est installé.
- Gérer Schémas XML : Ouvre un dialogue dans lequel vous pouvez ajouter de nouveaux Schémas XML de bases de données ou abandonner des Schémas XML de bases de données existants.

фа,

Ð

# 4.2.8.4 Copier, Trier et Chercher les Résultats de requête

L'onglet **Résultats** du volet **Requête BD** montre les enregistrements extraits en tant que résultat d'une requête de base de données.

	PrimaryKey •	ForeignKey 🔹	EMail •	First •	Last •
1	1	1	v.callaby@nanonull.com	Vernon	Callaby
2	2	1	f.further@nanonull.com	Frank	Further
3	3	1	I.matise@nanonull.com	Loby	Matise
4	4	2	j.firstbread@nanonull.com	Joe	Firstbread
5	5	2	s.sanna@nanonull.com	Susi	Sanna
6	6	3	f.landis@nanonull.com	Fred	Landis
7	7	3	m.landis@nanonull.com	Michelle	Butler
8	8	3	t.little@nanonull.com	Ted	Little

Les touches de la barre d'outils permettent la navigation entre les résultats et les instructions SQL et facilite la recherche dans le cadre des résultats de requête.

**Recherche :** Cherche un texte spécifique dans le cadre des résultats affichés. Appuyer sur **F3** pour se rendre à l'occurrence suivante du terme de recherche.

Aller à l'instruction : Passe à l'Éditeur SQL et marque l'instruction SQL qui produit le résultat actuel. Cela peut être particulièrement utile lorsque l'Éditeur SQL contient plusieurs instructions.

#### Pour sélectionner des cellules depuis des résultats de requête :

- Cliquer sur un en-tête de colonne pour sélectionner toute la colonne
- Cliquer sur le numéro d'une ligne pour sélectionner toute la ligne
- Cliquer sur des cellules individuelles. Maintenir appuyée la touche Ctrl tout en cliquant vous permet d'effectuer plusieurs sélections. Si une colonne ou une cellule contient des données XML, ces données peuvent aussi être copiées.

Note : Le menu contextuel peut aussi être utilisé pour choisir des données, **Tri | Ligne | Colonne | Tout.** 

#### Pour copier les cellules sélectionnées dans le presse-papiers :

Cliquer avec la touche de droite et sélectionner Copier les cellules sélectionnées depuis le menu contextuel.

#### Pour trier des données :

- Cliquer avec la touche de droite n'importe où dans la colonne que vous souhaitez trier et choisir Tri | Par ordre croissant ou Par ordre décroissant
- Cliquer sur l'icône de tri dans l'en-tête de colonne

	ID	NAME
1	3227	KElla Kir

Les données sont triées conformément aux contenus de la colonne triée.

#### Pour restaurer l'ordre de tri par défaut :

• Cliquer avec la touche de droite n'importe où dans la table et choisir **Tri | Restaurer défaut** depuis le menu contextuel.

# 4.2.8.5 Consulter le statut des requêtes exécutées

L'onglet **Messages** du volet **Requête BD** offre des informations spécifiques concernant la dernière instruction SQL exécutée et les erreurs de rapports ou les message d'avertissement.



Vous pouvez utiliser plusieurs filtres pour personnaliser l'affichage de l'onglet **Messages** ou utiliser les touches **Suivant** et **Précédent** pour parcourir des données ligne par ligne. Les touches situées dans la partie supérieure sont utilisées pour parcourir les messages, copier du texte dans le presse-papiers et dissimuler certaines parties du message. Ces options sont aussi disponibles dans le menu contextuel, lorsque vous cliquez avec la touche de droite n'importe où dans l'onglet **Messages**.

**Filtre** : Ouvre un menu pop-up à partir duquel vous pouvez filtrer les types de message individuels (**Résumé**, **Succès**, **Avertissement**, **Erreur**, **Insertion automatique**, **Progrès**). "Insertion automatique" réfère au messages qui peuvent être déclenchés lorsque l'instruction SQL ou les structures SQL sont insérées dans l'Éditeur SQL. Les messages "Progrès" rapportent le résultat de connexion de base de données, et le résultat du parsage SQL et du chargement de la structure de données.

 $\nabla$ 

	Vous pouvez vérifier toutes ou aucune de ces options avec un seul clic de souris en choisissant soit <b>Cocher tout</b> ou <b>Décocher tout</b> depuis le menu pop-up.
-	Suivant : Passe au message suivant et le marque.
	Précédent : Passe au message précédent et le marque.
£	Copier le message choisi dans le presse-papiers
₿	Copier le message choisi y compris ses enfants dans le presse-papiers
윱	Copier tous les messages dans le presse-papiers
<b>#\$</b>	Rechercher : Ouvre le dialogue Recherche.
авр.	Chercher précédent : Passe à l'occurrence précédente du string spécifié dans le dialogue Recherche.
<b>da</b>	Chercher suivant : Passe à l'occurrence suivante du string spécifié dans le dialogue Recherche.
×	Supprimer : Supprime tous les messages de l'onglet Message de la fenêtre Éditeur SQL.

# 4.2.9 **Procédures stockées**

Les procédures stockées sont des programmes qui sont hébergés et exécutés sur un serveur de base de données. Les procédures stockées peuvent être appelées par les applications client et elles sont souvent rédigées dans un dialecte étendu de SQL. Certaines bases de données prennent aussi en charge des mises en place dans Java, .NET CLR, ou d'autres langages de programmation.

Les utilisations générales des procédures stockées incluent la requête d'une base de données et le renvoi de données à un client effectuant un appel, ou la réalisation de modifications au niveau de la base de données après la validation supplémentaire des paramètres d'entrée. Les procédures stockées peuvent aussi effectuer d'autres actions situées en-dehors de la base de données, per ex. envoyer des e-mails.

Une procédure stockée peut avoir zéro ou plusieurs paramètres d'entrée et de sortie, et peut retourner en option zéro ou plusieurs recordsets, en plus de la valeur retournée par défaut. Par conséquent, dans MapForce, vous pouvez appeler une procédure stockée de plusieurs manière :

- Appeler une procédure stockée pour obtenir des données, comme s'il s'agissait d'un composant de source dans le mappage. Cela est applicable pour des procédures qui ne prennent pas de paramètres d'entrée. Lorsque le mappage est exécuté, la procédure est appelée, et elle retourne certains recordset ou paramètres de sortie. Vous pouvez mapper de recordset, ou les paramètres de sortie, ou les deux, à tout autre type de données pris en charge par MapForce. Pour voir un exemple, voir <u>Procédures</u> <u>stockées en tant que source de données</u><sup>402</sup>.
- Appeler une procédure stockée en tant qu'appel semblable à une fonction, avec des paramètres. Dans ce cas, vous apportez tous les paramètres d'entrée requis depuis le mappage, et vous pouvez aussi mapper le recordset retourné, ou les paramètres de sortie, ou les deux, vers une autre cible prise en charge par MapForce. Pour voir un exemple, voir <u>Procédures stockées avec Entrée et Sortie</u><sup>405</sup>.
- Appeler une procédure stockée comme s'il s'agissait d'un composant cible sur le mappage. Le cas d'utilisation typique est d'appeler une procédure stockée avec des paramètres afin de modifier la base
de données (par exemple, insérer un enregistrement). Cette approche est convenable si vous ne nécessitez pas de sortie provenant de la procédure stockée. De même, dans cette approche, vous pouvez exécuter la procédure stockée dans le cadre d'une transaction de base de données qui peut être renversée en cas d'erreur. Pour voir un exemple, voir <u>Procédures stockées dans les Composants cible</u> <sup>409</sup>.

Il existe des cas pour lesquels vous pouvez avoir besoin d'appeler des procédures stockées ou d'effectuer des actions dans les tables de base de données dans un ordre spécifique (d'abord insérer, puis mettre à jour, etc.). Par exemple, si vous souhaitez passer le paramètre de sortie d'une procédure stockée dans une autre procédure stockée. Ou bien vous pouvez souhaiter combiner des données retournées par une procédure stockée stored avec des données provenant d'une table. Ce genre d'actions est possible avec l'aide des relations locales définies dans MapForce, même si la base de données sous-jacente n'impose pas de relations clés primaire/étrangère entre les tables. Pour voir un exemple, voir <u>Procédures stockées et Relations</u> <u>locales</u>

**Note :** Pour illustrer comment MapForce met en place des procédures de stockage, ce chapitre utilise Microsoft SQL Server 2016 et la base de données "AdventureWorks2016". Cette dernière peut être téléchargée depuis le <u>https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks</u>.

## Notes de prise en charge

- Les procédures stockées peuvent être utilisées uniquement dans le moteur d'exécution BUILT-IN. La génération de code en C++, C#, ou Java n'est pas prise en charge.
- Les types définis par l'utilisateur, les types de curseur, les types de variante et de nombreux autres types de données "exotiques" spécifiques à des bases de données (comme des tableaux, géométrie, types CLR) sont généralement pas pris en charge en tant que types de paramètres d'entrée ou de sortie.
- La surcharge de procédure et de fonction (plusieurs définitions de routines portant le même nom et des paramètres différents) n'est pas prise en charge.
- Certaines valeurs par défaut de prise en charge de bases de données dans les paramètres d'entrée, cela n'est pas pris en charge actuellement. Vous ne pouvez pas omettre des paramètres d'entrée dans le mappage pour utiliser la valeur par défaut.
- Les procédures stockées retournant des enregistrements multiples sont pris en charge selon la combinaison du pilote et de l'API de base de données (ODBC/ADO/ADO.NET/JDBC). Seules des procédures qui retournent le même nombre d'enregistrements avec une structure de colonne fixe sont prises en charge.
- Dès que possible, utiliser la dernière version du pilote natif de base de données maintenu par le fournisseur de base de données. Éviter d'utiliser des pilote de pont, comme ODBC vers ADO Bridge, ou ODBC vers JDBC Bridge.
- En option, vous pouvez activer des transactions de base de données pour les procédures stockées appelées en tant que source de données, voir <u>Procédures stockées dans les Composants cibles</u><sup>403</sup>. Les transactions ne sont pas prises en charge pour les procédures stockées appelées en tant que source de données (sans les paramètres d'entrée), ou celles qui sont appelées en tant qu'une fonction (avec l'entrée et la sortie).

La table suivante liste les notes de prises en charge spécifique à la base de données.

Base de données	Notes de prise en charge
Access	<ul> <li>Les procédures stockées dans les bases de données Microsoft Access disposent de fonctions très limitées et ne sont pas prises en charge dans MapForce.</li> </ul>

Base de données	Notes de prise en charge
DB2	<ul> <li>Pris en charge dans MapForce: procédures stockées, fonctions scalaires, fonctions table.</li> <li>Les valeurs de retour provenant des procédures de stockage de DB2 ne sont pas prises en charge parce qu'elles ne peuvent pas être lues par le biais des API de base de données utilisées dans MapForce.</li> <li>Les fonctions de ligne (RETURNS ROW) ne sont pas prises en charge.</li> <li>Il est recommandé d'installer au moins "IBM_DB2 9.7 Fix Pack 3a" pour éviter un problème de pilote JDBC confirmé lors de la lecture des erreurs/avertissements après l'exécution. Cela permet aussi de régler un problème avec le fournisseur ADO qui entraîne la perte d'une ligne de résultat.</li> </ul>
Firebird	Pris en charge dans MapForce: procédures stockées, fonctions table
Informix	Pris en charge dans MapForce: procédures stockées, fonctions table.
MariaDB	Pris en charge dans MapForce: procédures stockées, fonctions scalaires
MySQL	<ul> <li>Pris en charge dans MapForce: procédures stockées, fonctions scalaires</li> <li>MySQL contient une prise en charge complète pour des procédures stockées et des fonction en commençant avec la version 5.5. Si vous utilisez une version précédente, les fonctions dans MapForce sont limitées.</li> </ul>
Oracle	<ul> <li>Pris en charge dans MapForce: procédures stockées, fonctions scalaires, fonctions table. Cela contient des procédures stockées et des fonctions autonomes ainsi que celles définies dans un package Oracle.</li> <li>Il est recommandé d'utiliser un pilote Oracle natif au lieu de Microsoft OLE DB Provider for Oracle.</li> <li>Oracle a un moyen spécial de retourner les ensembles de résultat dans le client en utilisant des paramètres de sortie de type REF CURSOR. Cela est pris en charge par MapForce pour des for procédures stockées, mais pas pour des fonctions. Les noms et le nombre des enregistrements est donc toujours fixées pour des procédures stockées Oracle.</li> </ul>
PostgreSQL	<ul> <li>Pris en charge dans MapForce: fonctions scalaires, fonctions ligne, fonctions table.</li> <li>Dans PostgreSQL, tous les paramètres de sortie définis dans une fonction décrit les colonnes de l'ensemble de résultat. Cette information est utilisée automatiquement par MapForce - aucune détection par exécution ou entrée des enregistrements n'est nécessaire. Les paramètres de type récurseur ne sont pas pris en charge.</li> </ul>
Progress OpenEdge	Pris en charge dans MapForce: procédures stockées.
SQL Server	<ul> <li>Pris en charge dans MapForce: procédures stockées, fonctions scalaires, fonctions table.</li> <li>Il est recommandé d'utiliser le dernier pilote SQL Server Native Client au lieu de Microsoft OLE DB Provider for SQL Server.</li> <li>L'API d'ADO a une prise en charge limitée pour certains types de données introduits avec SQL Server 2008 (datetime2, datetimeoffset). Si vous rencontrez des problèmes de raccourcissement de données avec ces types</li> </ul>

Base de données	Notes de prise en charge
	<ul> <li>temporels lorsque vous utilisez ADO avec le SQL Server Native Client, vous pouvez définir l'argument de string de connexion DataTypeCompatibility=80 ou utiliser ODBC.</li> <li>Les SQL Server Procedures ont un paramètre de retour implicite de type int null, qui est disponible pour le mappage. Si la procédure omet une instruction RETURN, la valeur résultante est 0.</li> </ul>
SQLite	<ul> <li>SQLite n'utilise pas de procédures stockées.</li> </ul>
Teradata	<ul> <li>Pris en charge dans MapForce: procédures stockées, macros.</li> <li>Fonctions scalaires, fonctions d'agrégat et fonctions de table ne sont pas prises en charge</li> <li>Problème connu : Le pilote Teradata ODBC refuse de remplir les valeurs de paramètre de sortie après un appel de procédure.</li> </ul>

# 4.2.9.1 Ajouter des procédures stockées dans le mappage

Dans la zone de mappage, les procédures stockées sont affichées en tant que partie intégrante du composant de données auquel elles appartiennent. Pour que les procédures stockées soient visibles sur le composant de la base de données, vous devez les choisir explicitement lorsque vous ajoutez le composant de base de données au mappage, comme indiqué ci-dessous. Dans cet exemple, nous nous connectons à la base de données "AdventureWorks" exécutée sur SQL Server. Les instructions sont semblables pour les autres types de base de données.

Dans le cas de base de données Oracle, des procédures stockées ou des fonctions peuvent être autonomes ou faire partie de packages Oracle. Vous pouvez ajouter les deux catégories au mappage. Les procédures stockées ou les fonctions appartenant à un package apparaissent sous le nom du package respectif dans le dialogue "Insérer des objets de base de données" illustré ci-dessous.

### Pour ajouter des procédures stockées au mappage :

- 1. Choisir parmi les deux possibilités suivantes :
  - Dans le menu Insérer, cliquer sur Base de données.
  - Cliquer sur la touche the la barre d'outils Insérer base de données ( 📴 ).
- Suivre les instructions de l'assistant de la base de données jusqu'à ce que vous parveniez au dialogue "Insérer Objets base de données". Pour consulter des instructions plus détaillées applicables aux différents types de base de données, voir <u>Exemples de connexion à des bases de données</u><sup>(91)</sup>.

🖻 Insert Database Objects	×
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Please choose Tables, Views, Stored Procedures, or create your own SQL SELECT statements to insert into your mapping. SQL SELECT statements defined by the user are stored locally in the mfd file.	
Source	
💎 🗟 Add/Edit SELECT Statement Add/Edit Relations (0) Add/Edit Recordset Structures	
AdventureWorks2016   AdventureWorks2016   Schemas   The fight dbo   The fight dbo	
Show Preview >> Use object names relative to default schema	Cancel

3. Choisir les cases à cocher situées à côté des objets de base de données que vous souhaitez voir affichées sur le mappage et cliquer sur **OK**. Dans cet exemple, nous avons choisi toutes les tables, modes et procédures stockées disponibles dans le schéma "HumanResources".

#### Notes

- Vous pouvez modifier les objets sélectionnés ultérieurement à tout moment, en cliquant avec la touche de droite sur la barre de titre d'un composant de base de données, et en choisissant **Ajouter/Supprimer/Éditer des objets de base de données** depuis le menu contextuel.
- Votre compte d'utilisateur de base de données doit disposer des droits pour consulter et exécuter des procédures stockées dans la base de données.

Le composant de base de données est maintenant ajouté au mappage. Veuillez noter que les procédures stockées sont identifiées par l'icône 📰. De plus, les tables, modes et les procédures sont triées alphabétiquement dans le composant de base de données.



La touche **Afficher menu contextuel** située à côté de chaque procédure stockée vous permet de configurer la manière dont vous souhaitez appeler la procédure stockée, vous trouverez également d'autres paramètres liés à la procédure, comme indiqué ci-dessous :

Option	Utilisation
Afficher nœuds en tant que Source	Choisir cette option si vous souhaitez appeler une procédure stockée sans paramètres afin d'extraire des données depuis une base de données et la mapper vers un autre composant pris en charge par MapForce (XML, texte, EDI, etc.). Pour consulter un exemple, voir <u>Procédures stockées</u> <u>en tant que source de données</u> <sup>402</sup> .
Afficher nœuds en tant que cible	Choisir cette option si vous souhaitez appeler une procédure stockée pour modifier la base de données ou effectuer une autre action spécifique dans laquelle vous n'avez pas besoin de la sortie de la procédure stockée. Pour consulter un exemple, voir <u>Procédures stockées dans</u> <u>composants de cible</u> <sup>409</sup> .
Insérer appel avec Paramètres	Choisir cette option si vous souhaitez appeler une procédure stockée <i>avec des paramètres</i> et que vous souhaitez mapper les données retournées dans un autre composant pris en charge par MapForce. Pour consulter un exemple, voir <u>Procédures stockées avec paramètres d'entrée</u> <u>et de sortie</u>
Éditer Structures de Recordset	Applicable pour les procédures stockées qui retournent des recordsets. Choisir cette option pour exécuter la procédure stockée une fois, afin que MapForce puisse déterminer la structure du recordset retourné et l'afficher dans le mappage. En alternative, si vous ne souhaitez pas exécuter la procédure stockée au moment du design time, you can define the recordset structure manually.

Paramètres de procédure	Applicable uniquement pour les procédures stockées qui ont été configurées en tant que "target" (c'est à dire, celles qui mettent à jour la base de données). Choisir cette option pour configurer des paramètres supplémentaires liés à la procédure, comme exécuter un requête SQL personnalisée avant d'appeler la procédure, ou activer des transactions de base de données.
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# 4.2.9.2 Procédures stockées en tant que source de données

Cet exemple vous montre comment appeler une procédure qui ne prend pas de paramètres d'entrée et se limite à extraire des données de la base de données. Dans ce scénario, la procédure stockée se comporte comme un composant de source dans le mappage, et vous pouvez mapper des données extraites à tout autre composant cible pris en charge par MapForce. Si vous souhaitez appeler une procédure stockée avec des paramètres d'entrée, voir <u>Procédures stockées avec Entrée et Sortie</u><sup>405</sup>.

Créons tout d'abord la procédure stockée de démonstration dans la base de données "AdventureWorks". Pour ce faire, exécuter le script ci-dessous par rapport à la base de données. Vous pouvez ce faire à partir d'une fenêtre de requête de **Microsoft SQL Server Management Studio**, ou directement depuis l'onglet **Requête BD** de MapForce, voir <u>Parcourir et interroger des bases de données</u><sup>(32)</sup>. Dans tous les cas, s'assurer que votre compte d'utilisateur de base de données a la permission de créer des procédures stockées.

CREATE PROCEDURE HumanResources.uspGetAllEmployees AS SELECT LastName, FirstName, JobTitle, Department FROM HumanResources.vEmployeeDepartment

La procédure stockée ci-dessus retourne des informations d'employés depuis le mode **vEmployeeDepartment**. Les étapes suivantes vous montrent comment créer un mappage qui consomme des données retournées par cette procédure.

- Se connecter à la base de données "AdventureWorks" depuis MapForce, et ajouter la procédure stockée au mappage comme décrit dans la section <u>Ajouter des procédures stockées dans le</u> <u>mappage</u>
   Veuillez vous assurer que votre compte utilisateur de base de données détienne la permission de consulter et d'exécuter des procédures stockées.
- 2. Cliquer sur la touche Afficher menu contextuel is située à côté de la procédure stockée et choisir Afficher nœuds en tant que source.

AdventureWorks2016 [AdventureWorks2	016]	
HumanResources.uspGetAllEmployees	<b>-</b>	₽
		Show Nodes as Source
		Show Nodes as Target
		Insert Call with Parameters
		Edit Recordset Structures
		Procedure Settings

3. Cliquer à nouveau sur la touche Afficher menu contextuel 🖾, choisir Éditer les structures de Recordset. Le dialogue "Structures de Recordset" apparaît.

😢 Recordset Structures - [AdventureWorks2016].[HumanResources].[uspGetAllEmployees]	×
A stored procedure may - in addition to output parameters - return one or more recordsets, which can be use as data sources in MapForce. Since the structure of these recordsets depends on the procedure's implementation and cannot be read from the database catalog, you have the following options:	ed
<ul> <li>Retrieve recordsets automatically by calling the procedure with user-defined parameters.</li> </ul>	
Define input parameters and call procedure	
Always call procedure with these parameters to refresh structure when needed	
Define recordset structures manually by typing in column names and data types.	
Recordsets:	_
	$\leq$
Name Type Nullable	
Notice, stored procedures that do not return the same recordset structures with each call are not supported	
OK Cancel	
Notice, stored procedures that do not return the same recordset structures with each call are not supported OK Cancel	ffet:
procedure stockée au moment de la conception, ne pas cliquer sur <b>Exécuter</b> , comme déc les étapes suivantes. Au lieu, définir le recordset attendu dans le dialogue "Structures de recordset", en ajoutant des recordsets et leurs colonnes associées manuellement. Utiliser	crit r le:

4. Cliquer sur **Définir les paramètres d'entrée et appeler la procédure**. Le dialogue "Évaluer procédure stockée" apparaît.

touches Ajouter recordset ou Ajouter colonne dans le dialogue "Structures de recordset".

😢 Evaluate Stored Procedure -	[AdventureWorks2016].[Human	Resources].[uspGetAllEmpl 🗙
This procedure does not have any	input or in/out parameters.	
Click Execute to execute the proce	dure and retrieve recordset struct.	ires.
Execute the procedure in a	transaction that is rolled back imm	ediately
Warning: Executing a stored p on the procedure's implementa	rocedure can have side effects tha ation.	t cannot be rolled back depending
Execute		
Execution result status: Not yet executed. Output parameters:		
Name	Туре	Value
@uspGetAllEmployees	int	not yet retrieved
		OK Cancel

5. Cliquer sur **Exécuter**, puis sur **OK**. La structure de recordset ("RS1") est maintenant visible aussi bien dans le dialogue "Structures Recordset" que dans le mappage.



6. À ce niveau, vous pouvez ajouter un composant cible dans lequel les données extraites seront rédigées. Dans cet exemple, les données seront rédigées dans un fichier CSV. Dans le menu Insérer, cliquer sur Fichier texte, et ajouter un composant CSV au mappage. Pour plus d'informations, voir <u>Fichiers CSV et Texte</u><sup>446</sup>.



Vous pouvez maintenant consulter le mappage. Cliquer sur la touche **Sortie** et observez le résultat de mappage dans le volet **Sortie**, par exemple :



# 4.2.9.3 Procédures stockées avec Entrée et Sortie

Cet exemple vous montre comment appeler une procédure qui prend des paramètres d'entrée et extrait des données de sortie de la base de données. Dans ce scénario, la procédure stockée est appelée comme pour un service Web ou une fonction et vous pouvez mapper des données extraites à tout autre composant cible pris en charge par MapForce.

Créons tout d'abord la procédure stockée de démonstration dans la base de données "AdventureWorks". Pour ce faire, exécuter le script ci-dessous par rapport à la base de données. Vous pouvez ce faire à partir d'une fenêtre de requête de **Microsoft SQL Server Management Studio**, ou directement depuis l'onglet **Requête BD** de MapForce, voir <u>Parcourir et interroger des bases de données</u><sup>(362)</sup>. Dans tous les cas, s'assurer que votre compte d'utilisateur de base de données a la permission de créer des procédures stockées.

```
CREATE PROCEDURE Production.uspSearchProducts
  @SearchString nvarchar(50)
  ,@MaxPrice money
  ,@ComparePrice money OUTPUT
AS
BEGIN
  SET NOCOUNT ON
  SELECT pr.[Name], pr.ListPrice FROM [Production].[Product] pr
  WHERE pr.[Name] like @SearchString AND pr.ListPrice < @MaxPrice
  SET @ComparePrice = @MaxPrice</pre>
```

**RETURN** @ComparePrice **END** 

La procédure stockée ci-dessus extrait un recordset contenant des informations de produit. Elle prend deux paramètres en tant qu'entrée : un string avec le nom du produit (@SearchString) et le prix maximum du produit (@MaxPrice). Outre le recordset et le paramètre de retour par défaut, elle extrait aussi un paramètre de sortie (@ComparePrice).

Les étapes suivantes vous montrent comment créer un mappage qui consomme des données retournées par cette procédure.

- Se connecter à la base de données "AdventureWorks" depuis MapForce, et ajouter la procédure stockée au mappage comme décrit dans la section <u>Ajouter des procédures stockées dans le</u> <u>mappage</u><sup>609</sup>. Veuillez vous assurer que votre compte utilisateur de base de données détienne la permission de consulter et d'exécuter des procédures stockées.
- 2. Cliquer sur la touche Afficher menu contextuel Située à côté de la procédure stockée et choisir Insérer appel avec paramètres. La procédure stockée apparaît maintenant en tant que composant séparé dans le mappage, à l'endroit où le côté gauche recense les paramètres d'entrée, et le côté droit contient les paramètres de retour et de sortie.

O AdventureWorks2016		Production.uspSearchProduction.uspSearchProduction	ıcts
Production.uspSearchProducts	1	Pierocedure	ခု 🕪 result ပြ
	I	>>> @SearchString	📲 📲 @uspSearchProducts
	I	>>> @MaxPrice	

 Cliquer à nouveau sur la touche Afficher menu contextuel , choisir Éditer les structures de Recordset. Cela est nécessaire pour fournir à MapForce des informations concernant la structure du recordset retourné par la procédure. Le dialogue "Structures de Recordset" apparaît.

♥ Recordset Structures - [AdventureWorks2016].[Production].[uspSearchProducts] ×
A stored procedure may - in addition to output parameters - return one or more recordsets, which can be used as data sources in MapForce. Since the structure of these recordsets depends on the procedure's implementation and cannot be read from the database catalog, you have the following options:    Retrieve recordsets automatically by calling the procedure with user-defined parameters.
Perordeato:
Name Type Nullable
Specify new recordset name here.
,
Notice, stored procedures that do not return the same recordset structures with each call are not supported.
OK Cancel

secondaires (selon la mise en place de la procédure). Si vous ne souhaitez pas exécuter la procédure stockée au moment de la conception, ne pas cliquer sur **Exécuter**, comme décrit dans les étapes suivantes. Au lieu, définir le recordset attendu dans le dialogue "Structures de recordset", en ajoutant des recordsets et leurs colonne dans le dialogue "Structures de recordset".

4. Cliquer sur **Définir les paramètres d'entrée et appeler la procédure**. Le dialogue "Évaluer procédure stockée" apparaît.

lame	Туре	Value	
@SearchString	nvarchar(100)	%Frame"	
@MaxPrice	money	500	
ck Execute to execute the	procedure with the above input	t values and retrieve recordset s	tructures
IN EACOULE IN EACOULE THE			
Execute the procedur	e in a transaction that is rolled	back immediately	
Execute the procedur	e in a transaction that is rolled	back immediately ffacts that cannot be called back	doponding
Execute to execute the procedur Warning: Executing a sto on the procedure's impler	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately ffects that cannot be rolled back	dependin
Warning: Executing a sto on the procedure's impler	e in a transaction that is rolled red procedure can have side e mentation.	back immediately ffects that cannot be rolled back	dependir
Warning: Executing a sto on the procedure's impler	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately ffects that cannot be rolled back	: dependir
	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately ffects that cannot be rolled back	: dependir
	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately ffects that cannot be rolled back	: dependir
Warning: Execute the procedur Warning: Executing a sto on the procedure's impler Execute	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately	: dependir
Warning: Executing a sto on the procedure's impler Execute Execute	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately ffects that cannot be rolled back	: dependir
Warning: Executing a sto on the procedure's impler Execute ecution result status: t yet executed.	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately	: dependir
Warning: Executing a sto on the procedure's impler Execute Execute ecution result status: t yet executed.	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately ffects that cannot be rolled back	: dependir
Warning: Executing a sto on the procedure's impler Execute Execute ecution result status: t yet executed. etput parameters: ame OutpoSearchProducts	e in a transaction that is rolled ored procedure can have side e mentation.	back immediately iffects that cannot be rolled back Value	: dependir
	re in a transaction that is rolled red procedure can have side e mentation. Type int money	back immediately iffects that cannot be rolled back Value not yet retrieved not yet retrieved	: dependir

- 5. Remplir les valeurs de paramètre comme indiqué ci-dessus et cliquer sur Exécuter.
- 6. Cliquer sur **OK**. La structure de recordset ("RS1") est maintenant visible aussi bien dans le dialogue "Structures Recordset" que dans le mappage.
- 7. À ce niveau, vous pouvez ajouter un composant cible dans lequel les données extraites seront rédigées. Dans cet exemple, les données seront rédigées dans un fichier CSV. Dans le menu Insérer, cliquer sur Fichier texte, et ajouter un composant CSV au mappage, voir Fichiers CSV et Texte<sup>446</sup>. Ensuite tracer les connexions de mappage tel qu'illutré ci-dessous. Veuillez noter que les paramètres d'entrée de la procédure sont fournis par le biais de constante. Pour plus d'informations concernant les constantes, voir Ajouter une constante au mappage<sup>652</sup>.



Vous pouvez maintenant consulter le mappage. Cliquer sur la touche **Sortie** et observez le résultat de mappage dans le volet **Sortie**, par exemple :



# 4.2.9.4 Procédures stockées dans les Composants cibles

Cet exemple vous montre comment appeler une procédure qui prend des paramètres d'entrée et met à jour une base de données. Le fait d'appeler une procédure de cette manière permet d'activer des transactions et d'annuler l'action en cas d'erreur, ou bien d'ajouter une instruction SQL personnalisée à exécuter avant que la procédure soit appelée. Ce scénario implique que la procédure stockée agisse comme un un composant cible dans MapForce et que vous n'êtes pas intéressé par la sortie retournée par la procédure stockée. Pour consulter un exemple qui illustre comment passer des paramètres et mapper des données retournées par une procédure stockée, voir <u>Procédures stockées avec Entrée et Sortie</u>

Créons tout d'abord la procédure stockée de démonstration dans la base de données "AdventureWorks". Pour ce faire, exécuter le script ci-dessous par rapport à la base de données. Vous pouvez ce faire à partir d'une fenêtre de requête de **Microsoft SQL Server Management Studio**, ou directement depuis l'onglet **Requête BD** de MapForce, voir <u>Parcourir et interroger des bases de données</u><sup>(62)</sup>. Dans tous les cas, s'assurer que votre compte d'utilisateur de base de données a la permission de créer des procédures stockées.

END

```
,@Inst
,NEWID()
,GETDATE())
```

La procédure stockée ci-dessus prend deux paramètres (@ModelName, @Inst) en tant qu'entrée et insère les valeurs correspondantes dans la table ProductModel de la base de données AdventureWorks, avec quelques données générées par base de données.

Les étapes suivantes vous montrent comment créer un mappage qui consomme des données retournées par cette procédure.

- Se connecter à la base de données "AdventureWorks" depuis MapForce, et ajouter la procédure stockée au mappage comme décrit dans la section <u>Ajouter des procédures stockées dans le</u> <u>mappage</u><sup>339</sup>. Veuillez vous assurer que votre compte utilisateur de base de données détienne la permission de consulter et d'exécuter des procédures stockées.
- Cliquer sur la touche Afficher menu contextuel is située à côté de la procédure stockée et choisir Afficher nœuds en tant que cible. La procédure stockée apparaît maintenant en tant que composant cible dans le mappage, à l'endroit où le côté gauche recense les paramètres d'entrée.



- 3. Cliquer à nouveau sur la touche **Afficher menu contextuel**, choisir **Paramètres de procédure**. Cette étape optionnelle vous permet d'exécuter la procédure stockée dans une transaction qui peut être annulée. Vous pouvez aussi ajouter une instruction SQL personnalisée à exécuter avant que la procédure soit appelée.
- 4. Cocher la case Utiliser transactions.

😢 Database Procedure Set	tings - Production.uspAddProductModel	×
SQL statement to execute b	efore procedure call	
Database transaction handle Use <u>t</u> ransactions When an error occurs:	rollback top transaction and stop $\sim$	
Database trace Trace level:	Use component settings V Parameters	
	OK Cancel	

- **Note :** Dans cet exemple, le traçage de base de données est désactivé au niveau du composant de base de données et aucun traçage n'est défini pour être exécuté. Néanmoins, vous pouvez activer un traçage de base de données pour les procédures stockées, le cas échéant. Pour plus d'informations, voir <u>Traçage de base de données et journal d'erreur</u><sup>435</sup>.
  - Ajouter le composant source qui fournit les données à insérer dans la base de données. Dans cet exemple, les données de source sont fournies par des constantes ; néanmoins, tout autre composant de source pris en charge par MapForce peut agir en tant qu'entrée. Pour plus d'informations concernant les constantes, voir <u>Ajouter une constante au mappage</u><sup>552</sup>.



Puisque ce mappage met à jour une base de données, vous ne pouvez pas consulter sa sortie directement comme c'est le cas pour d'autres mappages. Au lieu, cliquer sur la touche **Sortie** pour afficher le pseudo-SQL contenant des indices concernant la manière dont la base de données sera modifiée. Si vous avez activé des transactions, celles-ci se produiront selon les indications par les commentaires.

7		
8 9	SET QUOTED_IDENTIFIER ON	
10	begin transaction	
11 12 13 14	<pre>EXECUTE NULL = [Production].[uspAddProductModel] 'Tractor part 4711', 'coupling' commit transaction</pre>	
1 22		
Mappi	ng DB Query   Output	
🔊 uspA	\ddProductModel.mfd 4	×

Le pseudo-SQL afficher dans le volet **Sortie** ne montre pas les commandes de transaction réelles, il ne montre que des indices (comme des commentaires). Les commandes SQL réelles sont néanmoins envoyées à l'API de base de données sous-jacente.

Pour exécuter le mappage par rapport à la base de données, suivre une des deux étapes :

- Dans le menu Sortie, cliquer sur Exécuter Script SQL.
- Cliquer sur la touche de la barre d'outils Exécuter Script SQL <sup>16</sup>

### Procédures stockées et entrées doubles

Si vous souhaitez mapper des données provenant de plusieurs sources sur le mappage dans la même procédure stockée de manière à ce qu'elle accepte plusieurs entrées. Pour ce faire, cliquer avec la touche de droite sur l'item de procédure stocké dans le composant et sélectionner **Ajouter une entrée double** depuis le menu contextuel, voir aussi <u>Doubler les entrées</u><sup>(80)</sup>. Lorsque le mappage est exécuté, ces procédures stockées doubles seront appelées une fois pour chaque entrée double.

Veuillez noter que la commande **Ajouter une entrée double** est désactivée pour les paramètres de procédure stockée, parce que chaque paramètre est une valeur atomique (et pourrait être "nullable").

## 4.2.9.5 Procédures stockées et Relations locales

Les relations locales sont des relations logiques entre les champs de base de données que vous pouvez créer dans MapForce, ce qui vous évite de modifier la base de données sous-jacente, voir aussi <u>Définir des relations</u> <u>locales</u>. Vous pouvez définir des relations locales non seulement pour les champs de base de données, mais aussi pour les procédures stockées, aussi bien pour les composants de source et de cible.

Dans les composants de source, les relations locales permettent de lire des données depuis des objets liés, par exemple, lire des ID depuis une table de base de donnée et appeler une procédure stockée avec chacune de ces ID pour extraire des informations connexes. Il est également possible d'appeler une procédure stockée avec des données extraite d'une autre procédure.

Dans les composants de cible, les relations locales permettent de définir un ordre hiérarchique dans lequel plusieurs procédures connexes doivent être appelées. Par exemple, vous pouvez d'abord appeler une procédure stockée qui crée une valeur ID, et une autre qui insère des informations connexes dans une table. Il est également possible de mélanger des procédures stockées et des tables dans des relations locales. Par exemple, vous pouvez effectuer l'insertion directement sur une table liée au lieu d'appeler une autre procédure, voir <u>Utiliser des procédures stockées pour générer des clés</u><sup>(419)</sup>.

### Pour créer une relation locale :

- Cliquer sur la barre de titre d'un composant de base de données et choisir Ajouter/Supprimer/Éditer des objets de base de données depuis un menu de contexte. Le dialogue "Ajouter/Supprimer/Éditer des objets de base de données" s'ouvre.
- 2. Cliquer sur Ajouter/Éditer relations.
- 3. Cliquer sur Ajouter relation et sélectionner les objets entre lesquels vous pouvez créer la relation.

<ul> <li>Add/Edit Kelations</li> <li>this dialog you can define relations in a toreig</li> <li>ation a primary key object and a foreig</li> </ul>	addition to those defined in the database. n key object and at least one column pai	Please select for each r with matching datatype
Primary/Unique Key Object [select object] [select object first]	Foreign Key Object         [select object]         [select object first]	OK Cancel
		Add Relation
		Add Column Delete Column

Comme illustré ci-dessus, une relation locale consiste en un objet de **clé primaire/unique** et un objet de **clé étrangère**. On pourrait le comparer à une relation parent-enfant. Dans le composant de mappage, l'objet (table, mode, procédure, etc.) dans lequel la clé primaire/unique se trouve apparaîtra en tant que parent alors que l'objet dans lequel la clé étrangère se trouve apparaîtra imbriquée en-dessous. Par exemple, dans le composant de base de données comme illustré ci-dessous, une relation locale a été définie entre une colonne recordset (**RS1.Department**) et une colonne de table (**Department.Name**). Par conséquent, la table **Department** apparaît en tant qu'enfant d'une procédure stockée sur le mappage. Cet exemple est discuté avec plus de détails dans <u>Relations locales dans les composants de source</u><sup>414</sup>.

AdventureWorks2016 [AdventureWorks2016]	
	onta
🕀 🧱 HumanResources.uspGetAllEmployees 🔣	¢
	¢
🖓 🥅 🖽 RS1	¢
LastName	¢
III FirstName	¢
JobTitle	¢
	¢
HumanResources.Department Name Lookup	tabl
DepartmentID Primary key for Depart	ment
Name Name of the department., RS1	.Dep[>
GroupName Name of the group to w	vhich
ModifiedDate Date and time the reco	ord v(>

La table suivante recense tous les champs possibles entre lesquels vous pouvez définir des relations locales. Des relations mixtes sont possibles (par exemple, mapper la sortie d'une procédure stockée dans une colonne de base de données). Les champs participant à la relation doivent avoir les mêmes types de données ou un type de données compatible.

Clé primaire/unique	Clé étrangère
<ul> <li>La colonne d'une table de base de données ou d'un mode</li> <li>Paramètre de sortie ou valeur de retour d'une procédure stockée, voir aussi <u>Procédures</u> <u>stockées</u></li> <li>La colonne d'un recordset retournée par une procédure stockée*</li> <li>La colonne d'une instruction SELECT définie par l'utilisateur, voir aussi <u>Instructions</u> <u>SELECT SQL en tant que tables</u> <u>virtuelles</u></li> </ul>	<ul> <li>La colonne d'une table de base de données ou d'un mode</li> <li>Paramètre d'entrée d'une procédure stockée</li> <li>Paramètre d'entrée d'une instruction SELECT définie par l'utilisateur</li> </ul>

\* Applicable si la procédure stockée est appelée soit en tant que source de données (sans paramètres) ou en tant que fonction (avec paramètres d'entrée et de sortie). Pour que le recordset devienne disponible pour la sélection, vous devez exécuter la procédure stockée une fois, pour extraire le recordset.

## 4.2.9.6 Relations locales dans les composants de source

L'exemple suivant vous montre comment combiner des données retournées par une procédure stockée avec des données provenant d'une table dans la même base de données, à l'aide des relations locales.

Si vous ne l'avez pas encore fait précédemment, exécuter le script suivant pour créer la procédure stockée de démonstration dans la base de données "AdventureWorks". Vous pouvez y accéder depuis une fenêtre de requête de **Microsoft SQL Server Management Studio**, ou directement depuis le volet **Requête BD** de

MapForce, voir <u>Parcourir et interroger des bases de données</u><sup>682</sup>. Dans tous les cas, veuillez vous assurer que votre compte d'utilisateur de base de données a la permission de créer des procédures stockées.

```
CREATE PROCEDURE HumanResources.uspGetAllEmployees
AS
SELECT LastName, FirstName, JobTitle, Department
FROM HumanResources.vEmployeeDepartment
```

La procédure stockée ci-dessus retourne l'information de l'employé depuis le mode **vEmployeeDepartment**. Les étapes suivantes vous montrent comment créer un mappage qui consomme les données retournées par cette procédure.

- Se connecter à la base de données "AdventureWorks" depuis MapForce, comme décrit dans <u>Ajouter</u> <u>des procédures stockées dans le mappage</u><sup>339</sup>. Assurez-vous que votre compte d'utilisateur de base de données détient la permission de consulter et d'exécuter les procédures stockées.
- 2. Lorsque vous êtes invité à choisir des objets de base de données, choisir la table, choisir la table **Department** et la procédure stockée **uspGetAllEmployees**.



- 3. Cliquer sur la touche Afficher menu contextuel 🔤 située à côté de la procédure stockée et choisir Afficher nœuds en tant que cible.
- 4. Cliquer à nouveau sur la touche Afficher menu contextuel 🔜, choisir Paramètres de procédure. Le dialogue "Structures Recordset" apparaît.
- 5. Cliquer sur **Définir les paramètres d'entrée et appeler la procédure**. Le dialogue "Évaluer la procédure stockée" apparaît.

😢 Evaluate Stored Procedure -	[AdventureWorks2016].[Human	Resources].[uspGetAllEmpl ×		
This procedure does not have any input or in/out parameters.				
Click Execute to execute the proce	dure and retrieve recordset struct.	ures.		
Execute the procedure in a	transaction that is rolled back imm	ediately		
Warning: Executing a stored p on the procedure's implementa	rocedure can have side effects tha ation.	at cannot be rolled back depending		
Execute	Execute			
Execution result status: Not yet executed. Output parameters:				
Name	Туре	Value		
wuspoetAllEmployees	jint.	OK Cancel		

6. Cliquer sur **Exécuter**, puis sur **OK**. La structure recordset ("RS1") est maintenant visible aussi bien dans le dialogue "Structures Recordset" que dans le mappage.

O AdventureWorks2016 [AdventureWorks2016]	
	con
🕀 🧱 HumanResources.uspGetAllEmployees 🔜	- ¢
	Ļ
🗄 🖽 🛱 🖽 RS1	þ
III LastName	Ļ
FirstName	þ
JobTitle	þ
Department	þ

## Définir les relations locales

À présent, il s'agit de définir une relation locale entre la colonne **Department** du recordset retourné et la colonne **Name** de la table **Department**.

- 1. Cliquer avec la touche de droite sur la barre de titre du composant de la base de données et choisir Ajouter/Supprimer/Éditer objets de base de données du menu contextuel.
- 2. Cliquer sur **Ajouter/Éditer relations**, puis cliquer sur **Ajouter relation**. Définir les relations comme indiqué ci-dessous.

Add/Edit Relations	×
In this dialog you can define relations in addition to those defined in the database. Please select relation a primary key object and a foreign key object and at least one column pair with matching	for each datatypes.
Primary/Unique Key Object       Foreign Key Object         Image: State of the sta	OK Cancel Add Relation Delete Relation
	Add Column Delete Column

3. Cliquer sur **OK** pour fermer le dialogue. Veuillez noter que la table **Department** devient un enfant du recordset **RS1**.



## Achever le mappage

Grâce à la relation qui vient d'être créée, il est maintenant possible de mapper des données depuis le recordset combiné avec des données provenant de la table. Pour illustrer notre exemple, nous écrivons des données vers un fichier cible, comme suit :

- 1. Dans le menu Insérer, cliquer sur Schéma XML/Fichier et choisir le fichier suivant : <Documents>\Altova\MapForce2023\MapForceExamples\EmployeesWithDetails.xsd.
- 2. Lorsque vous êtes invité à fournir un échantillon d'instance XML, cliquer sur Sauter.
- 3. Tracer les connexions de mappage comme indiqué ci-dessous.



Le mappage illustré ci-dessus écrit des données depuis la base de données vers un fichier XML cible. Les données de source sont constituées par une combinaison de données extraites par la procédure stockée avec des données extraites directement depuis une table. Le mappage utilise la fonction <u>concat</u> pour produire un string qui inclut le nom du département, suivi d'un tiret et suivi du nom du groupe.

Pour consulter le mappage, cliquer sur la touche **Sortie** et observer le résultat de mappage dans le volet **Sortie**, par exemple :



# 4.2.9.7 Utiliser des procédures stockées pour générer des clés

Cet exemple vous montre comment insérer des clés (ID) générées par une procédure stockée dans une autre table, avec l'aide des relations locales.

Créons tout d'abord la procédure stockée de démonstration dans la base de données "AdventureWorks". Pour ce faire, exécuter le script ci-dessous par rapport à la base de données. Vous pouvez ce faire à partir d'une fenêtre de requête de **Microsoft SQL Server Management Studio**, ou directement depuis l'onglet **Requête BD** de MapForce, voir <u>Parcourir et interroger des bases de données</u><sup>(32)</sup>. Dans tous les cas, s'assurer que votre compte d'utilisateur de base de données a la permission de créer des procédures stockées.

```
CREATE PROCEDURE Production.uspAddProductModelEx
  @ModelName nvarchar(50)
  ,@Inst xml
  ,@ProductModelID int OUTPUT
AS
BEGIN
INSERT INTO [Production].[ProductModel]
           ([Name]
           ,[Instructions]
           ,[rowguid]
           ,[ModifiedDate])
     VALUES
           (@ModelName
           ,@Inst
           ,NEWID()
           ,GETDATE())
   SELECT @ProductModelID = SCOPE_IDENTITY()
END
```

La procédure stockée ci-dessus prend deux paramètres (@ModelName, @Inst) en tant qu'entrée et effectue une opération INSERT dans la table **ProductModel**. Elle retourne ensuite le @ProductModelID généré en tant que paramètre de sortie. L'exigence est d'insérer le @ProductModelID retourné par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la table **ProductModelIII retourné** par la procédure stockée dans la par la procédure stockée dans la par la

Les étapes suivantes vous montrent comment créer un mappage qui satisfait aux exigences ci-dessus.

- Se connecter à la base de données "AdventureWorks" depuis MapForce, comme décrit dans la section <u>Ajouter des procédures stockées dans le mappage</u><sup>333</sup>. Veuillez vous assurer que votre compte utilisateur de base de données détienne la permission de consulter et d'exécuter des procédures stockées.
- Lorsque vous êtes invité à choisir des objets de base de données, choisir la table ProductModelIIIustration et la procédure stockée uspAddProductModelEx.



 Cliquer sur la touche Afficher menu contextuel is située à côté de la procédure stockée et choisir Afficher nœuds en tant que cible. La procédure stockée apparaît maintenant en tant que composant de cible dans le mappage, et le côté gauche recense les paramètres d'entrée.



- 4. En option, si vous souhaitez exécuter la procédure stockée dans une transaction, cliquer à nouveau sur la touche Afficher menu contextuel , choisir Paramètres de procedure, et cocher la case Utiliser des transactions. Le fait de définir la transaction pour la procédure stockée permet d'assurer que l'extraction de la clé et l'insertion de l'enregistrement se produise pendant la même transaction.
- 5. Cliquer avec la touche de droite sur la barre de titre du composant de base de données et choisir Ajouter/Supprimer/Éditer objets de base de données depuis le menu contextuel.
- 6. Cliquer sur **Ajouter/Éditer des relations**, puis cliquer sur **Ajouter relation**. Définir les relations comme indiqué ci-dessous.

Add/Edit Relations	×
In this dialog you can define relations in addition to those defined in the database. relation a primary key object and a foreign key object and at least one column pair	Please select for each with matching datatypes.
Primary/Unique Key Object Foreign Key Object	ОК
📮 🧱 Production.uspAddProductModelEx 💌 🔟 Production.ProductMod	elllustration 💌 🔶
🔄 🔝 @ProductModellD (int) 📰 🔃 ProductModellD (int)	
	Add Relation Delete Relation
	Add Column Delete Column
	-

7. Cliquer sur OK pour fermer le dialogue. Veuillez noter que la table ProductModeIIIIustration apparaît maintenant en tant qu'enfant de la procédure stockée. Le paramètre de sortie de la procédure stockée (@ProductModelID) est affiché en tant qu'un indicateur qu'il sera utilisé dans la relation locale, mais il n'a pas de connecteurs d'entrée ou de sortie.

🖯 Adventure	Works2016 [AdventureWorks2016]
⇒ 🖽 Product	ion.ProductModellllustration Cross-reference table (
🖓 🖾 🛱 Product	ion.uspAddProductModelEx 🔠
🖕 🛶 🖬 @Moo	delName
👌 🔤 🕨 🖉 🖓	
₩ @Proc	ductModelID
	luction.ProductModellllustration ProductModelID Cross-reference tal
🗊 Pro	oductModelID Primary key. Foreign key to ProductMe
բ 🔐 🔐	IstrationID Primary key. Foreign key to Illustration.Illu
¢ <sup>i</sup> <b>⊡ M</b> ∢	odifiedDate Date and time the record was last updat

8. Dans cet exemple, le paramètre @Inst est de type XML. Cliquer avec la touche de droite sur le paramètre @Inst dans le composant et choisir Attribuer Schéma XML au champ depuis le menu contextuel. Ensuite, choisir le schéma Production.ManuInstructionsSchemaCollection depuis la base de données. Lorsque vous serez invité à choisir un élément racine, laisser la valeur par défaut telle quelle et cliquer sur OK. Pour plus d'informations concernant le mappage de données vers des champs XML de base de données, voir Mapper des données XML de / vers des champs de base de données

😢 Assign XML Schema to Field		$\times$		
XML Schema © <u>D</u> atabase: O <u>F</u> ile:	Production.ManuInstructionsSchemaCollecti v	Bro <u>w</u> se	Edi <u>t</u>	

 Ajouter les composants de source qui fournissent les données à insérer dans la base de données. Dans cet exemple, les données de source sont fournies par des constantes ; néanmoins, tout autre composant de source pris en charge par MapForce pourrait agir en tant qu'entrée. Pour plus d'informations concernant les constantes, voir <u>Ajouter une constante au mappage</u><sup>652</sup>.



Puisque ce mappage met à jour une base de données, vous ne pouvez pas consulter sa sortie directement comme c'est le cas pour d'autres mappages. Au lieu, cliquer sur la touche **Sortie** pour afficher le pseudo-SQL contenant des indices concernant la manière dont la base de données sera modifiée. Si vous avez activé des transactions, celles-ci se produiront selon les indications par les commentaires.

10	begin transaction		
11			
12	<pre>NULL = {{{[Production].[uspAddProductModelEx]}}} ( 'Tractor part 4711','<r xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;oot&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;pre&gt;ProductModelManuInstructions"><location locationid="17"><step>coupling</step></location>',NULL )</r></pre>		
13	>>> %@uspAddProductModelEx1%		
14	>>> %@ProductModelID1%		
15			
16	<pre>INSERT INTO [Production].[ProductModelIllustration] ([ProductModelID],</pre>		
	[IllustrationID]) VALUES ('%@ProductModelID1%', 11)		
17			
18	commit transaction		
19			-
Manni	Ing DB Query Quitout		
[ mappi	ing boddery   output		
🗞 uspA	ddProductModelEx	⊲ ⊳	×

Le pseudo-SQL afficher dans le volet **Sortie** ne montre pas les commandes de transaction réelles, il ne montre que des indices (comme des commentaires). Les commandes SQL réelles sont néanmoins envoyées à l'API de base de données sous-jacente.

Pour exécuter le mappage par rapport à la base de données, suivre une des deux étapes :

- Dans le menu Sortie, cliquer sur Exécuter Script SQL.
- Cliquer sur la touche de la barre d'outils Exécuter Script SQL <sup>1</sup>/<sub>2</sub>.

# 4.2.10 Gérer des exceptions de base de données

Lors de l'exécution de mappages qui modifient une base de données, MapForce (ou MapForce Server, ou un programme généré par MapForce) peut éventuellement rencontrer plusieurs erreurs reliées à la base de données. Par exemple, le compte de base de données peut ne pas avoir suffisamment de privilèges pour effectuer une action de base de données spécifique, ou bien certaines contraintes de base de données peuvent exister qui empêchent le mappage d'insérer ou de mettre à jour des données invalides. Ce dernier cas de figure peut se produire, par exemple, lorsqu'une colonne de table est obligatoire mais que le mappage ne fournit pas de valeur pour cette colonne.

Afin de permettre d'annuler les données en cas d'apparition d'exceptions de base de données, vous pouvez configurer le mappage pour utiliser des transactions de base de données. Comme décrit plus vas, vous pouvez activer des transactions au niveau du composant de base de données et au niveau d'action de table.

Les paramètres de rollback transaction que vous configurez dans MapForce sont préservés lorsque vous exécutez le mappage soit avec MapForce Server, soit avec un programme généré par MapForce.

### Transactions au niveau de composant de base de données

Pour activer les transactions au niveau du composant de base de données, procédez comme suit :

1. Cliquer avec la touche de droite sur le composant de base de données, et choisissez **Propriétés** depuis le menu contextuel (en alternative, double-cliquer sur la barre de titre du composant de base de données). Le dialogue Paramètres de composant de la base de données s'ouvre.

Gestion de transaction de base de données					
Utiliser les transactions					
Lorsqu'une erreur se produit:	annuler la transaction supérieure et arrêter	$\sim$			

2. Cocher la case Utiliser les transactions.

Tous les changements effectués par le composant de base de données sont maintenant contenus dans une transaction qui sera annulée en cas d'erreur. Lorsqu'une erreur se produit pendant l'exécution de mappage, le résultat dépend de l'option que vous avez sélectionnée depuis la liste déroulante :

- **Annuler transaction supérieure et arrêter** La transaction qui contient tous les changements effectués par le composant de base de données est annulée et l'exécution du mappage s'interrompt.
- Annuler transaction supérieure et continuer Comme ci-dessus, mais l'exécution du mappage se poursuit après le rollback (par exemple, pour traiter un deuxième composant cible, le cas échéant).

Certains mappages peuvent contenir plusieurs composants de base de données. Ceux-ci peuvent utiliser la même connexion de base de données, ou bien, des connexions vers des base de données complètement différentes. Le résultat de ces mappages en cas d'erreur de base de données dépend du moteur d'exécution :

- Si le mappage est effectué avec MapForce, seul un composant de cible peut être exécuté lorsque le mappage est exécuté. Il s'agit du composant dans lequel la touche **Aperçu** est activée. Si une erreur de base de donnée se produit dans ce composant, et la coche **Utiliser les transactions** est activée, tous les changements effectués par le composant seront annulés.
- Si le mappage est effectué avec MapForce Server ou un programme généré par MapForce, tous les composants de cible sont exécutés, séquentiellement. Dans ce cas, lorsqu'une erreur de base de données se produit, l'annulation se produira pour le composant de base de données à l'endroit où l'erreur s'est produite. Le mappage s'arrêtera ou continuera à exécuter le composant de cible suivant selon la valeur que vous avez sélectionnée depuis la liste déroulante Lorsqu'une erreur se produit, comme mentionné ci-dessus.

Pour plus de références à tous les paramètres disponibles dans le dialogue Paramètres de composant de base de données, voir <u>Paramètres de composant de base de données</u><sup>281</sup>.

## Transactions au niveau de l'action de table

Les transactions au niveau de l'action de table contient chaque action de table individuelle (insérer, mettre à jour, etc.) dans une transaction. Avec cette configuration, si des actions d'insertion ou de mise à jour échoue dans un enregistrement, il est possible d'annuler uniquement l'action échouée actuelle et continuer le traitement d'autres enregistrements. Pour activer des transactions au niveau de l'action de table, sélectionnez la case à cocher **Utiliser les transactions** depuis les dialogue Actions de table de base de données (voir aussi Paramètres Actions de table de base de données <sup>(30)</sup>).

11 int		NULL egale	Insérer tout	-	^	Apposer action
			valeur mappée	-		Insérer action
🕅 user_id			valeur mappée	-		
created_at			valeur mappée	•	~	
oprimer les données d	ans les tables enfant				^	
orer les données d'en	trée enfant					
III orderedproducts						
					$\sim$	Supprimer action
SI UNE EFFEUR SULULT.	annuler la transaction si	uperieure et arr	eter			~
	annuler la transaction su	upérieure et arr	êter			
açage de base de donr	annuler la transaction su annuler la transaction su annuler la transaction a	upérieure et arr upérieure et cor ctuelle et arrête	êter tinuer r			
açage de base de donr Viveau de traçage:	annuler la transaction su annuler la transaction su annuler la transaction au annuler la transaction au annuler la transaction au	upérieure et arr upérieure et cor ctuelle et arrête ctuelle et contin	êter tinuer r uer			снатрь
açage de base de donr Niveau de traçage:	annuler la transaction su annuler la transaction su annuler la transaction a annuler la transaction a ounser parametres de c	upérieure et arr upérieure et cor ctuelle et arrête ctuelle et contin composant	êter tinuer r uer		_	champs
açage de base de donr Niveau de traçage:	annuler la transaction su annuler la transaction su annuler la transaction au annuler la transaction au ounser parametres de c	upérieure et arr upérieure et cor ctuelle et arrête ctuelle et contin omposant	êter tinuer r Jer		_	enamps
açage de base de donr Niveau de traçage: Jtiliser le transfert Bulk	annuler la transaction su annuler la transaction su annuler la transaction au annuler la transaction au ourser parametres de c (MapEorce Server uniqu	upérieure et arr upérieure et cor ctuelle et arrête ctuelle et contin omposant ement)	Eter tinuer r Jer	1000	enregistrements	спатрэ

Les options disponibles sont les suivantes ::

Annuler la transaction supérieure et arrêter — Si le mappage va mettre à jour plusieurs tables possédant une relation parent-enfant, il est possible d'activer des transactions non seulement pour la table de parent, mais aussi pour la table enfant. Puisque les transactions puissent être activées sur plusieurs niveaux, "haut" se réfère à la transaction se plaçant au niveau supérieur dans la hiérarchie. C'est la transaction provenant du niveau de composant de base de données, si vous avez activé les transactions à ce niveau. Sinon, il s'agit de la transaction définie dans l'action de table supérieure, en partant du principe que le composant contient plusieurs actions de table imbriquées. Pour comprendre cette option, rappelez-vous qu'il est possible qu'un composant puisse mettre à jour plusieurs tables hiérarchiquement, comme décrit dans Insérer des données dans plusieurs tables liées<sup>220</sup>. Par exemple, un composant peut avec une action "Insert" parent qui insère des enregistrements de personne et une action "Insert" imbriquée qui insère une ou plusieurs adresses pour chaque enregistrement de personne dans une table séparée. Puisque vous pouvez activer des transactions aussi bien pour les actions d'insertion, "haut" se réfère au niveau le plus élevé dans la hiérarchie (dans cet exemple, l'action d'insertion de personnes). C'est pourquoi, lorsqu'une exception est rencontrée lors de l'insertion d'une adresse, cela annule la transaction supérieure (qui se trouve au niveau de table de personne, dans cet exemple) et l'exécution s'arrête. En d'autres termes, aucun enregistrement de personnes ne sera inséré s'il n'était pas possible d'insérer l'adresse de la personne en premier lieu.

- Annuler la transaction supérieure et continuer Comme ci-dessus, mais le mappage continue d'être exécuté après l'annulation (par exemple, pour traiter un autre composant de cible, le cas échéant).
- Annuler la transaction actuelle et arrêter Lorsqu'une exception survient, cela annule uniquement les changements contenus dans la transaction actuelle ; les changements effectués précédemment en dehors de la transaction actuelle seront archivés.
- Annuler la transaction actuelle et continuer Comme ci-dessus, mais le mappage continue d'être exécuté après l'annulation.

## Transactions au niveau de la procédure stockée

Vous pouvez aussi activer des transactions de base de données pour les procédures stockées qui sont appelées en tant que cible de données. Pour un exemple, voir <u>Procédures stockées dans les Composants</u> cibles appelées en tant que source de données (sans les paramètres d'entrée), ou celles qui sont appelées en tant qu'une fonction (avec l'entrée et la sortie).

## Annuler

Pour voir si les transactions vont se produire au moment de l'exécution du mappage, cliquez sur l'onglet **Sortie** et observez le code pseudo-SQL généré par MapForce. L'image ci-dessous illustre l'aperçu d'un mappage dans lequel des transactions sont activées au niveau de composant et au niveau de l'action de table. Comme indiqué ci-dessous, une transaction principale contient toute l'opération (identifié par les commentaires --begin transaction et --commit transaction). De même, plusieurs petites transactions se produisent pour chaque opération d'insertion (celles-ci sont identifiées par la commande SAVEPOINT).

7	
8	begin transaction
9	
10	PRAGMA foreign_keys = ON;
11	
12	begin transaction
13	
14	INSERT INTO "products" ("name") VALUES ('Product A')
15	
16	begin transaction
17	
18	INSERT INTO "products" ("name") VALUES ('Product B')
19	
20	begin transaction
21	
22	INSERT INTO "products" ("name") VALUES ('Product C')
23	
Mappa	ge Requête BD Sortie

Lorsque le mappage est exécuté dans MapForce Server, ou depuis un programme généré par MapForce, l'annulation se produira sur la base des options que vous avez configurées tout en concevant le mappage, comme décrit ci-dessus. Par exemple, si vous avez choisi l'option **Annuler la transaction supérieure et arrêter**, une opération d'insertion échouée entraînera l'annulation de tous les changements, même si certaines insertions peuvent avoir réussi. Néanmoins, si vous avez sélectionné **Annuler la transaction actuelle et**  **continuer**, seule l'insertion échouée sera annulée jusqu'au point d'enregistrement le plus proche, et le mappage continuera pour tenter l'insertion d'autres enregistrements.

Si vous exécutez un mappage activé par la transaction directement dans MapForce avec la commande de menu **Sortie | Exécuter Script SQL**, un dialogue vous informe lorsque l'exception de base de données est rencontrée, par exemple :

𝘢 MapForce: Database Transaction Exception	$\times$
Error in Transaction: PRODUCTS	
Reason for the error:	
NOT NULL constraint failed: products.quantity	$\hat{}$
SQL statement:	
INSERT INTO "products" ("name") VALUES (?)	
Choose how to continue	
Rollback this transaction and stop	
Rollback this transaction and continue	
Rollback/continue for all further errors	
Rollback top and stop	
O Rollback top and continue	
OK	

Vous pouvez ensuite changer le comportement suivant comme suit :

### Annuler cette transaction et arrêter

- La transaction actuelle est déjà annulée au moment où le dialogue survient
- Une erreur s'affiche en haut de la fenêtre d'aperçu et dans la fenêtre Messages

C Tra Err Re	Transformation-Error in Database SQL-execution: Error in: INSERT INTO "products" ("name") VALUES (?) Reason: NOT NULL constraint failed: products.quantity.					
1	/* The following SQL statements were executed during "Generate	<b></b>				
2	output" function.	-				
Mapp	ing DB Query Output					

- Toute transaction de parent est archivée
- L'exécution de mappage s'arrête avec un code d'erreur

#### Annuler cette transaction et continuer

- La transaction actuelle est déjà annulée au moment où le dialogue survient
- L'erreur est traitée en tant qu'avertissement, et sera affichée dans la fenêtre Messages lorsque l'exécution de mappage se termine.

Messages	×
A DOB 999 X	
🔁 🖓 SQLiteException.mfd: Execution successful - 0 error(s), 0 warning(s)	
Transformation-Error in Database SQL-execution:	
The second secon	
Reason: NOT NULL constraint failed: products.quantity.	
	-

- Toute transaction de parent n'est pas touchée
- L'exécution de mappage continue avec l'enregistrement suivant (si cette transaction ne se trouve pas au niveau du composant) ou avec le composant suivant (MapForce Server uniquement)

### Annuler le haut et arrêter

- La transaction actuelle est déjà annulée au moment où le dialogue survient
- Une erreur s'affiche en haut de la fenêtre d'aperçu et dans la fenêtre Messages
- Toute transaction de parent est annulée
- L'exécution de mappage s'arrête avec un code d'erreur

Note :l'option Annuler le haut et arrêter est disponible dans le dialogue uniquement lorsqu'il y a des transactions imbriquées (points d'enregistrement).

### Annuler le haut et continuer

- La transaction actuelle est déjà annulée au moment où le dialogue survient
- L'erreur est traitée en tant qu'avertissement, et sera affichée dans la fenêtre Messages lorsque l'exécution de mappage se termine.
- Toute transaction de parent est annulée
- L'exécution de mappage continue avec l'enregistrement suivant (si cette transaction ne se trouve pas au niveau du composant) ou avec le composant suivant (MapForce Server uniquement)

Note :l'option Annuler le haut et continuer est disponible dans le dialogue uniquement lorsqu'il y a des transactions imbriquées (points d'enregistrement).

# 4.2.10.1 Exemple : Rollback de Transaction

Cet exemple illustre les moyens se trouvant à votre disposition pour gérer les exceptions de base de données lors de la gestion d'un mappage qui modifie une base de données. La base de données utilisée dans cet exemple stocke des enregistrements d'utilisateur (users table) et les adresses (addresses table). Chaque utilisateur peut avoir zéro, un ou deux adresses (par exemple, une adresse personnelle et une adresse professionnelle). Concrètement, chaque adresse dans la table d'adresse contient un champ user\_id qui pointes vers le champ id dans la table users. La table users est donc une table "parent" pour addresses. L'extrait de code suivant illustre les scripts de création SQL pour les deux tables (noter que la syntaxe est applicable à une base de données SQLite) :

CREATE	TABLE
	users (id INTEGER NOT NULL PRIMARY KEY,
	first_name TEXT NOT NULL,
	last_name TEXT <b>NOT NULL</b> ,
	email TEXT UNIQUE NOT NULL);
CREATE	TABLE
	addresses (id INTEGER NOT NULL PRIMARY KEY,
	user_id INTEGER NOT NULL,
	is_shipping INTEGER,
	is_billing INTEGER,
	type TEXT NOT NULL,
	city TEXT NOT NULL,
	street TEXT NOT NULL,
	number INTEGER NOT NULL,
	FOREIGN KEY (user_id) REFERENCES users (id) ) ;

#### Table de source

L'exigence recherchée et de copier toutes les données provenant des tables users et addresses dans de nouvelles tables, concrètement, les tables new\_users et new\_addresses. Celles-ci sont presque identiques aux tables users et addresses. La seule différence est que la table new\_addresses ne permet pas des valeurs null pour les colonnes is\_shipping and is\_billing, comme marqué en jaune dans l'extrait de code cidessous. Cela est important parce que cela signifie que des exceptions se produiront si des valeurs null sont rencontrées.

```
CREATE TABLE

new_users (id INTEGER NOT NULL PRIMARY KEY,

first_name TEXT NOT NULL,

last_name TEXT NOT NULL,

email TEXT UNIQUE NOT NULL);

CREATE TABLE

new_addresses (id INTEGER NOT NULL PRIMARY KEY,

user_id INTEGER NOT NULL,

is_shipping INTEGER NOT NULL,
```

```
is_billing INTEGER NOT NULL,
type TEXT NOT NULL,
city TEXT NOT NULL,
street TEXT NOT NULL,
number INTEGER NOT NULL,
FOREIGN KEY (user_id) REFERENCES new_users (id) );
```

Table de cible

Le mappage qui copie toutes les données des anciennes tables vers les tables récentes est illustré cidessous. Vous pouvez trouver ce mappage sous le chemin suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\DatabaseExceptions.mfd.



Comme illustré ci-dessus, ce mappage est configuré pour copier des données verbatim depuis un composant de base de données de source vers un composant de base de données de cible. Aucune autre transformation ne se produit, pour des raisons de simplicité. Dans cet exemple, aussi bien le composant de source et de cible appartient à la même base de données, seules les tables de source et de cible sont différentes. Cliquer sur la touche <u>DEL, A:In</u> et noter que tous les enregistrements sont configurés pour être supprimés avant chaque action d'insertion. Cela permet de garantir que les tables de cible sont toujours vides avant chaque insertion.



Comme indiqué précédemment, des exceptions se produiront si la table addresses de source contient des valeurs null. Vous pouvez contrôler que la table de source contient des valeurs null comme suit :

- 1. Cliquer sur l'onglet **Requête DB**.
- 2. Cliquer avec la touche de droite sur la table addresses, et choisir Afficher dans SQL Éditeur SQL | SELECT.
- 3. Cliquer sur **Exécuter requête**

🖳 ExceptionsDemo 🔹 main 🔹 🗖 🖬 🕨 🖓 🛱 🙆								
		tonsDemo ain Tables User Table address address mew_add mew_add mew_use	SQLite s es dresses ers		2	SELECT "id" "is_shippin "city", "st "main"."add	, "user_id", g", "is_billi reet", "numbe resses";	ng", "type", r" FROM
	6							
	id 🔹	user_id •	is_shipping •	is_billing 🔹	type 🔹	city •	street •	number •
1	1	1	1	0	work	Bridgedell	Maple Lane	1
2	2	1	[NULL]	1	home	Bridgedell	Olive Street	6
3	3	3	1	[NULL]	home	Roseford	Evergreen Lane	34
4	4	4	1	1	work	Beardale	Route 44	9
5	5	5	[NULL]	1	home	Johnson City	Franklin Avenue	11
Finished Retrieval     Rows: 5, Cols: 8     0.047 sec     11:36:08       Results     Messages       Mapping     DB Query     Output								
🔁 DatabaseExceptions.mfd 🛛 🗠 🗙								

Comme illustré ci-dessus, il existe plusieurs champs NULL dans la table addresses de source. Vous disposez donc de plusieurs options pour gérer des exceptions pendant ce mappage, par exemple :

- A) Si une exception quelconque est rencontrée, annuler tous les changements. En d'autres termes, ne pas insérer d'enregistrements si les données de source contient des enregistrement invalides.
- B) Si une exception quelconque est rencontrée, sauter les enregistrements dans lesquels l'exception se produit mais garde les enregistrements d'insertion qui sont valides.

## Scénario A : Annuler les changements sur l'exception

Exigence : Je dois annuler tous les changements dans la base de données si une exception se produit. Pour configurer le mappage :

 Cliquer avec la touche de droite sur la barre de titre du composant de base de données cible et sélectionner **Propriétés** depuis le menu contextuel. En alternative, double-cliquer sur la barre de titre. Le dialogue Paramètres de composant de base de données s'ouvre.

### 2. Cocher la case Utiliser les transactions, et choisir annuler la transaction supérieure et arrêter.

Lorsque le mappage est configuré comme dans l'exemple ci-dessus, toute exception rencontrée entraînera l'annulation de la transaction de niveau supérieur au niveau du composant de base de données. Pour exécuter le mappage avec ces paramètres :

- 1. Cliquer sur l'onglet **Sortie**.
- Dans le menu Sortie, cliquer Exécuter Script SQL. À ce niveau, le mappage rencontre une exception en raison d'une valeur null dans la table des adresses de source. Le dialogue Exception de transaction de base de données s'ouvre.

😢 MapForce: Database Transaction Exception	$\times$
Error in Transaction:	
Reason for the error:	
NOT NULL constraint failed: new_addresses.is_shipping	0
SQL statement:	
INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "type", "city", "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?)	
Choose how to continue	
Rollback this transaction and stop	
O Rollback this transaction and continue	
Rollback/continue for all further errors	
OK	

3. Ne pas modifier l'option présélection par défaut et cliquer sur **OK**.

Le résultat est le suivant :

- Tous les changements sont annulés.
- Aucun enregistrement n'est inséré dans la table new\_users
- Aucun enregistrement n'est inséré dans la table new\_addresses

### Scénario B : Annuler la transaction actuelle et continuer l'exécution

Exigence : Je souhaite sauter des enregistrements qui génèrent une exception mais qui continue d'insérer des enregistrement valides. Pour configurer le mappage afin d'accomplir ceci :

- 1. Cliquer sur la touche <u>DEL, A:In</u> située à côté de la table <u>new\_users</u> table, cocher la case **Utiliser** transactions et choisir annuler la transaction actuelle et continuer
- 2. Cliquer sur la touche Ain située à côté de la table new\_addresses table, cocher la case Utiliser transactions et choisir annuler la transaction actuelle et continuer
Lorsque vous exécutez le mappage avec ces paramètres, le dialogue Exception de transaction de base de données. Sélectionner les options comme indiqué ci-dessous pour n'annuler que la transaction actuelle et continue d'exécuter le mappage :

Fror in Transaction: NEWADDRESSESUSERID	
Reason for the error:	
NOT NULL constraint failed: new_addresses.is_shipping	
SQL statement: INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?, ?)	e", "city",
SQL statement: INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?)	ve", "city",
SQL statement: INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?) - Choose how to continue	e", "city",
SQL statement: INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?) Choose how to continue O Rollback this transaction and stop	e", "city",
SQL statement: INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?) Choose how to continue Rollback this transaction and stop Rollback this transaction and continue	e", "city",
SQL statement:  INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?)  Choose how to continue  Rollback this transaction and stop  Rollback this transaction and continue Rollback this transaction and continue Rollback/continue for all further errors	ve", "city",
SQL statement:  INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "typ "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?)  Choose how to continue  Rollback this transaction and stop  Rollback this transaction and continue Rollback top and stop  Rollback top and stop	e", "city",

Le résultat est le suivant :

- Toutes les transactions erronées sont annulées.
- 5 utilisateurs de 5 sont insérés (parce qu'aucun enregistrement d'utilisateur n'a généré d'exception de base de données)
- 2 adresses de 5 sont insérés (parce que 3 adresses contiennent des valeurs null et des exceptions générées)

#### Scénario C : Annuler la transaction supérieur et continuer l'exécution

Exigence : si une adresse n'est pas insérée, l'enregistrement d'utilisateur parent ne doit pas non plus être inséré. Pour configurer le mappage pour accomplir cela :

- 1. Double-cliquer sur la barre de titre du composant de base de données et décocher la case **Utiliser transactions**.
- 2. Cliquer sur la touche <u>DEL, A:In</u> située à côté de la table <u>new\_users</u>, cocher la case **Utiliser** transactions et choisir annuler la transaction actuelle et continuer
- 3. Cliquer sur la touche <u>A:In</u> située à côté de la table <u>new\_addresses</u>, cocher la case **Utiliser** transactions et choisir annuler la transaction du haut et continuer

Lorsque vous exécutez le mappage avec ces paramètres, le dialogue Exception de transaction de base de données est affiché. Sélectionnez les options comme indiqué ci-dessous, et répétez le choix autant de fois que nécessaire :

😢 MapForce: Database Transaction Exception	$\times$
Error in Transaction: NEWADDRESSESUSERID	
Reason for the error:	
NOT NULL constraint failed: new_addresses.is_shipping	^
	~
SQL statement:	
INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "type", "crty", "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?)	
Choose how to continue	
O Rollback this transaction and stop	
O Rollback this transaction and continue	
Rollback/continue for all further errors	
O Rollback top and stop	
Rollback top and continue	
ОК	

Le résultat est le suivant :

- 2 utilisateurs sont insérés (Sharda Junker, Tobie Hughey)
- 1 adresse est insérée

Explication : L'utilisateur "Sharda Junker" est le seul utilisateur qui ne possède pas d'adresse. Aucune exception ne s'est produite pour cet enregistrement, donc il a été inséré. Le second utilisateur, "Tobie Hughey", a été inséré parce qu'il s'agit du seul utilisateur dans lequel aucune exception ne s'est produite au niveau de l'adresse. Tous les autres utilisateurs n'ont pas été insérés parce qu'ils avaient au moins une adresse lorsqu'une exception s'est produite et les transactions ont été annulées.

Il y a deux adresses qui n'ont pas de nulls et qui ne lancent donc pas d'exceptions. Ce sont les adresses avec id=1 et id=4. La The insert transaction for the first address, however, was rolled back because the parent user transaction had to be rolled back. Therefore, only the address with id=4 was inserted.

Veuillez noter que le même résultat peut aussi s'effectuer comme suit :

- 1. Cliquer sur la touche <u>DEL, A:In</u> située à côté de la table <u>new\_users</u>, cocher la case **Utiliser** transactions et choisir annuler la transaction actuelle et continuer
- 2. Cliquer sur la touche A:In située à côté de la table new\_addresses, et décocher la case Utiliser transactions.

### Exécuter le mappage avec MapForce Server

Si vous avez mis MapForce Server sous licence, vous pouvez aussi exécuter le mappage au niveau de la ligne de commande, sur un appareil Linux, macOS ou Windows, comme suit :

- 1. Compiler le mappage dans un fichier d'exécution MapForce Server (.mfx) avec la commande de menu **Fichier | Compiler sur le fichier d'exécution MapForce Server**, voir aussi <u>Compiler des mappages</u> <u>dans les fichiers d'exécution MapForce Server</u><sup>(937)</sup>.
- Copier le fichier .mfx et la base de données ExceptionsDemo.sqlite dans un répertoire sur le serveur. Nous l'appellerons "répertoire de travail".
- 3. Changer l'invite de commande dans le répertoire de travail et exécuter MapForce Server avec la commande ci-dessous :

mapforceserver run DatabaseExceptions.mfx

#### Notes :

 mapforceserver est le chemin vers le programme d'exécution de MapForce Server tel qu'il est applicable pour votre système d'exploitation. Changer le chemin vers le fichier .mfx tel qu'applicable. Par exemple, sur un appareil Linux, le chemin est

:/opt/Altova/MapForceServer2023/mapforceserver.

Dans l'exécution de serveur, vous pouvez aussi exécuter des mappages en tant qu'un appel API, ou en tant que tâches FlowForce Server, soit sur demande soit de manière récurrente. Pour plus d'informations, voir <u>Automatisation avec MapForce Server</u><sup>(950)</sup>.

# 4.2.11 Traçage de base de données et journalisation d'erreur

Lorsqu'un mappage écrit des données dans une base de données, vous pouvez choisir d'activer le traçage et base de données et la journalisation d'erreur. Le traçage est utile si vous souhaitez suivre tous les changements effectués par le mappage dans la base de données lorsque le mappage est en cours d'exécution. Lorsque le traçage est activé, des événements comme des actions d'insertion ou de mise à jour de la base de données ou des erreurs sont journalisées dans un fichier XML que vous pouvez ensuite analyser ou éventuellement traiter ultérieurement de manière automatisée. Si vous préférez que le fichier de journal apparaisse dans un format différent de XML, optionnellement, vous pouvez mapper des données vers d'autres types de composant pris en charge par MapForce (par exemple, un fichier texte ou même une autre base de données).

**Note :** Pour qu'un mappage puisse prendre en charge le traçage et la journalisation d'erreur, le langage de mappage doit être réglé sur **Built-in**. La journalisation et le traçage de base de données n'est pas pris en charge dans C#, C++ ou Java.

Le traçage de base de données est configurable : vous pouvez choisir de tracer tous les messages, uniquement des erreurs ou bien, vous pouvez désactiver complètement le traçage dans les niveaux suivants :

 Niveau de composant de base de données. À ce niveau, vous pouvez activer ou désactiver le traçage pour ce composant de base de données. Cela est utile pour les mappages qui possèdent plusieurs composants de base de données cible, et vous ne nécessitez le traçage que pour quelques uns d'entre eux.

- Niveau de table ou de procédure stockée. Dans un composant de base de données, vous pouvez décider si vous souhaitez que le traçage se produise pour chaque table ou procédure stockée. S'il est activé le traçage à ce niveau contient des événements touchant à des actions de table comme "Insert All", "Update If", "Ignore If", "Delete If". Dans le cas de procédures stockées, les événements liés à l'appel de procédure stockée sont tracés.
- Niveau de champ de base de données. Par défaut, tous les champs sont tracés, mais vous pouvez exclure certains champs du traçage, ou bien, vous pouvez spécifier qu'ils doivent être inclus dans le fichier de traçage uniquement en cas d'erreur.

Chose importante, les trois niveaux au-dessus sont hiérarchiques et les paramètres de traçage sont ventilés de haut en bas. Ainsi, si vous désactivez le traçage au niveau du composant de base de données, il ne sera pas possible de l'activer au niveau de table ou de procédure stockée. De même, si vous désactivez le traçage au niveau de la table, il ne sera pas possible de le configurer au niveau du champ de base de données (colonne). La même chose vaut lorsque vous réduisez le niveau de traçage. Par exemple, si vous limitez le traçage uniquement à des erreurs au niveau du composant de base de données, il ne sera pas possible d'utiliser le traçage complet au niveau de la table ou de la procédure stockée.

### Traçage au niveau de composant de base de données

Pour activer le traçage au niveau du composant de base de données :

 Cliquer avec la touche de droite sur le composant de base de données, et choisir Propriétés depuis le menu contextuel (en alternative, double-cliquer sur la barre de titre du composant de base de données). Cela ouvre le dialogue Paramètres de composant de base de données.

Traces		
Trac <u>e</u> level:	Always $\checkmark$	
Trace <u>fi</u> le:	Log.xml	Browse

- Choisir le niveau de traçage désiré (Toujours, Erreurs, Désactivé). En sélectionnant Erreurs, vous créez un fichier de traçage uniquement si des erreurs de base de donnéesse produisent lorsque le mappage est en cours d'exécution.
- À côté de Fichier de traçage, saisir le nom ou le chemin du fichier XML dans lequel tous les détails de traçage seront rédigés (par exemple, Log.xml). Si vous saisissez un nom de fichier au lieu d'un chemin complet, le fichier de traçage sera créé dans le même répertoire que le mappage sera exécuté.

Une fois avoir activé le traçage, le composant de base de données change sa structure dans le mappage. Concrètement, il affiche, outre la structure de la base de données, une nouvelle hiérarchie de nœud des items lorsque l'item supérieur dans la hiérarchie est le fichier de traçage. Pour plus d'informations, voir <u>Structure du</u> <u>fichier de traçage</u>

L'activation du traçage au niveau du composant de base de données l'active automatiquement pour toutes les tables ou les procédures stockées sur le composant pour lequel une connexion de mappage existe. Le cas échéant, vous pouvez modifier encore le traçage au niveau de table ou de procédure stockée, comme décrit cidessous. Veuillez noter que les tables ou les procédures stockées qui sont affichées dans le composant mais qui ne participent pas au mappage (cela signifie qu'aucune connexion de mappage n'existe pour lui) ne seront pas tracées.

### Traçage au niveau de la table ou de la procédure stockée

Pour activer le traçage au niveau de la table ou de la procédure stockée :

- 1. Assurez-vous que le niveau de traçage contenu dans le niveau de composant de base de données est configuré soit sur **Toujours** soit sur **Erreur** (voir ci-dessus).
- 2. Choisir une des étapes suivantes :
  - Cliquer sur la touche d'action de table (par exemple, A:In) situé à côté de la table dans laquelle vous souhaitez activer le traçage, ou
  - S'il s'agit d'une procédure stockée, cliquer sur la touche Afficher le menu contextuel 📖, et choisir Paramètres de procédure depuis le menu contextuel.

Traces			
Trac <u>e</u> level:	Use component settings	~	Eields

3. Choisir le niveau de traçage. L'option Utiliser les paramètres de composant hérite des mêmes paramètres qui ont été définis au niveau du composant. L'option Limiter aux erreurs limite le traçage uniquement aux événements d'erreur. Toujours désactivé signifie qu'aucun traçage ne se produira pour cette table ou procédure stockée.

### Traçage au niveau du champ de base de données

Lorsque vous activez le traçage au niveau de composant de base de données et au niveau de la table ou de la procédure stockée, tous les champs (colonnes de base de données) sont inclus par défaut dans le rapport de traçage. En option, vous pouvez limiter le traçage uniquement à des champs de base de données spécifiques ou à des paramètres de procédure stockée particulières. Pour ce faire, cliquer sur la touch **Champs** dans le menu Actions de table de base de données ou Paramètres de procédure de base de données, voir ci-dessus.

	O Hide	O Include on error	O Include
id	0	0	۲
name	0	۲	0
quantity	0	۰	0
description	۲	0	0
created_at	۲	0	0
updated_at	۲	0	0

Les sélections effectuées dans ce dialogue affectent la structure du fichier de traçage. Par exemple, tous les champs que vous avez décidé de dissimuler n'apparaîtront plus dans le fichier de traçage. Vous pouvez utiliser cette technique par exemple pour réduire la taille du fichier de journal.

Pour dissimuler ou inclure plusieurs champs simultanément, cliquer sur l'option respective dans l'en-tête.

## 4.2.11.1 Structure du fichier de traçage

Lorsque le traçage est activé pour un composant de base de données, une structure de traçage devient disponible dans la partie inférieure du composant, par exemple :



Comme illustré ci-dessus, le nœud supérieur dans la structure de traçage indique le nom du fichier de traçage qui sera rédigé lorsque le mappage est exécuté. Le reste de la structure de traçage est modelé après la structure des tables de base de données ou des procédures stockées qui participent au mappage. Dans cet exemple, le nom de la base de données est "TargetDatabase", donc un item semblable existe dans la structure de traçage.

L'item enfant de "TargetDatabase" est **users**, qui correspond à la table de base de données "users". Dans cet exemple, le composant de base de données ne contient qu'une table, "users". Néanmoins, dans un composant avec de nombreuses tables, la structure de traçage contient chaque table présente dans le composant. Si vous ajoutez ou supprimez des tables ou des procédures stockées depuis le composant de base de données, la structure de traçage est mise à jour. Pour consulter des instructions, voir <u>Ajouter, éditer et supprimer des objets de base de données</u><sup>254</sup>.

Dans cet exemple, une structure **init:users** apparaît aussi. Cela se produit lorsque vous avez configuré le mappage pour qu'il effectue une action initiale avant l'insertion, la mise à jour ou principale ou toute autre action. Par exemple, si vous avez choisi de supprimer tous les enregistrements existants de la table avant d'en insérer de nouveaux, la structure **init** est présente. Elle rassemble des informations de traçage concernant l'action initiale (comme une suppression ou une requête SQL personnalisée) et toute erreur qui pourrait se produire à ce niveau.

La structure **trace:values** affiche toutes les colonnes de la table de la base de données (dans ce cas, les colonnes de la table "users"). En ce qui concerne les procédures stockées, cette structure affiche les paramètres de la procédure stockée. Par défaut, toutes les colonnes sont définies pour être tracées, mais vous pouvez configurer les colonnes à tracer, comme décrit plus haut.

Les items **trace:actions** indiquent quelles actions sont actuellement prévues dans la table de la base de données. Dans cet exemple, une action de mise à jour de base de données A:Up est configurée pour se produire, donc l'item **trace:update** est disponible. Si vous changez le composant pour effectuer une action d'insertion de base de données A:In, alors la structure de traçage changera en **trace:insert**, etc. pour les autres types d'action. Dans le cas des procédures stockées, une action **trace:execute** sera visible à la place.

Chaque action de traçage possède un attribut **rows-affected** qui spécifie combien de lignes seront touchées par l'action de base de données respective.

L'item **trace:error** est rempli uniquement si une erreur de base de données se produit pendant l'exécution de mappage. Il possède deux attributs, **code** et **state**. Le texte de l'erreur et les valeurs d'attribut sont fournis par le pilote de base de données et seront donc différents pour chaque type de base de données.

L'item **trace:summary** contient un attribut **errors** qui rapporte le nombre d'erreurs rencontrées. La valeur d'attribut est **0** si aucune erreur n'est rencontrée.

### Consulter les fichiers de traçage

Lorsque le traçage est activé pour un composant de base de données cible, le fichier de traçage est aussi inclus dans l'aperçu lorsque vous cliquez sur l'onglet **Sortie** pour consulter le résultat de mappage. Cela est utile si vous souhaitez voir à quoi le fichier ressemblera. Veuillez noter que le fichier de trace généré aura une apparence différente sur la base du résultat d'exécution de mappage.

## 4.2.11.2 Exemple: Journalisation des erreurs de base de données

Cet exemple vous montre comment configurer le traçage de base de donnée et la journalisation d'erreurs pour un mappage qui insère des données dans une base de données SQLite. Les instructions sont semblables à tous les types de base de données pris en charge par MapForce, ainsi que pour d'autres types d'action de base de données (comme "update", "ignore", "delete", etc.).

Vous pouvez trouver le mappage de démonstration référencé par cet exemple sous le chemin suivant : **>Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\DatabaseExceptions.mfd**. Ce mappage insert les données depuis deux tables d'une base de données en une nouvelle paire de tables de la même base de données, il est décrit plus en détail dans l'<u>exemple : Rollback de transaction</u> <sup>(22)</sup>. Veuillez noter que certains des enregistrements contenus dans la base de données sont délibérément null afin de générer des exceptions de base de données.



DatabaseExceptions.mfd

Dans cet exemple, l'exigence est de permettre le traçage pour le composant cible de la base de données afin de pouvoir déterminer combien d'erreurs ont été générées, et où exactement les erreurs se sont produites. Les étapes ci-dessous illustrent comment atteindre cet objectif.

## Étape 1 : Activer le rollback de transaction

Ce mappage de démonstration est configuré de manière à utiliser le rollback de transaction au niveau de composant de base de données. Ceci signifie que lorsque la première exception est rencontrée, le mappage arrêtera l'exécution et activera le rollback de toutes les données vers leur état initial. Modifions d'abord les options de rollback de transaction de sorte à ce que l'exécution de mappage continue, même si des exceptions sont rencontrées pour certains des enregistrements :

1. Cliquer sur la touche <u>DEL, A:In</u> à côté de la table new\_users, cochez la case **Utiliser les transactions** et choisir **annuler cette transaction et continuer**.

Database transaction har	ndling	
Use transactions		
When an error occurs:	rollback current transaction and continue	~

2. Cliquer sur la touche <u>DEL, A:In</u> à côté de la table <u>new\_addresses</u>, cocher la case **Utiliser les** transactions et choisir annuler cette transaction et continuer.

Vous pouvez également désactiver entièrement le rollback de transaction ; néanmoins, cela interrompra l'exécution de mappage à la première erreur rencontrée et n'annulera *pas* les changements, cela n'est donc pas recommandé. Dans ce cas particulier, un traçage et une journalisation ne reflèteront une exécution que jusqu'à la première exception rencontrée. Pour plus d'informations concernant le rollback de transaction, voir <u>Gérer les</u> exceptions de base de données<sup>423</sup>.

## Étape 2 : Configurer la journalisation d'erreur

Pour activer le traçage, procéder comme suit :

- 1. Cliquer avec la touche de droite sur le composant de base de données, et choisissez **Propriétés** depuis le menu contextuel (en alternative, double-cliquer sur la barre de titre du composant de base de données). Le dialogue Paramètres de composant de la base de données s'ouvre.
- 2. Sélectionner Erreurs en tant que niveau de traçage.
- À côté de Tracer le fichier, tapez le nom du fichier de traçage (dans cet exemple, Errors.xml). Ne pas saisir un chemin absolu si vous avez l'intention d'exécuter le mappage avec MapForce Server sur d'autres appareils ou sur d'autres systèmes d'exploitation.

Traces		
Trac <u>e</u> level:	Errors ~	
Trace <u>fi</u> le:	Errors.xml	Browse

Le composant de base de données cible a maintenant changé sa structure et affiche un compartiment supplémentaire avec la structure de fichier de traçage :

is_shipping	⊳
is_billing	Þ
TII type	¢
City	¢
III street	¢
number	¢
🖃 🗂 Trace file: Errors.xml	⊳
ExceptionsDemo	¢
⊕ () new_addresses	
🕀 🜔 init:new_users	
🕀 🕻 new_users	
🕀 🕻 trace:summary	

Jusqu'à présent, vous avez activé le traçage au niveau du composant de la base de données. Par conséquent, les paramètres de traçage seront hérités par toutes les actions de table sur ce composant de la base de données. Dans cet exemple, ils sont hérités par les actions "Insert" des deux tables new\_users et new\_addresses. Si vous cliquez sur le bouton d'action <u>A:In</u> pour chacune de ces tables, les paramètres illustrent ce fait :

Traces			
Trac <u>e</u> level:	Use component settings	$\sim$	Eields

En option, cliquer sur la touche **Fields** pour configurer quels champs doivent être inclus ou dissimulés dans le fichier de traçage. Dans cet exemple, tous les champs sont inclus.

## Étape 3 : Configurer la sortie de mappage

Cette étape vous montre comment mapper les valeurs provenant du fichier de traçage vers un autre type de composant (dans ce cas, un simple composant de sortie). Veuillez noter que cette étape est optionnelle ; un fichier de journal d'erreur est produit même si vous ne mappez pas le fichier de traçage vers un autre composant. Dans cet exemple, néanmoins, nous aimerions configurer explicitement le mappage pour qu'il retourne le nombre total d'erreurs rencontrées dans la sortie. Pour ce faire, suivre les étapes suivantes :

- Ajouter un simple composant de sortie en sélectionnant la commande de menu Insérer | Insérer sortie. Pour plus d'informations concernant les composants d'entrée simples, voir <u>Retourner des</u> valeurs de string depuis un mappage<sup>(433)</sup>.
- 2. Ajouter une constante, en sélectionnant la commande de menu **Insérer | Constante**. La constante apportera le texte statique "Total erreurs : " pour rendre le texte de sortie de mappage plus clair.
- 3. Glisser la fonction concat depuis la fenêtre Bibliothèques dans le mappage. Pour plus d'informations concernant les fonctions, voir <u>Aiouter une fonction au mappage</u><sup>552</sup>.

Comme illustré ci-dessous, la fonction **concat** concatène le texte de la constante avec la valeur **errors** fournie par le fichier de traçage.



Avec la configuration ci-dessus, lorsque le mappage est exécuté avec MapForce Server, la sortie affichera le texte : "Total erreurs : {value}", où value représente le nombre d'erreurs rencontrées au moment de l'exécution.

À ce niveau, l'exécution du mappage dans MapForce en maintenant la touche **Aperçu** e appuyée dans le composant "result" n'est pas possible et résultera en une erreur. Un fichier de traçage doit exister tout d'abord, parce que la sortie simple lit des données depuis le fichier de traçage. L'étape suivante vous montre comment produire le fichier de traçage.

## Étape 4 : Exécuter le mappage

Pour exécuter ce mappage avec MapForce :

- 1. Cliquer sur la touche Aperçu 🧖 dans le composant de base de données.
- 2. Cliquer l'onglet Sortie.
- Dans le menu Sortie, cliquer sur Exécuter Script SQL. À ce niveau, le mappage rencontre la première exception ; cela est tout à fait intentionnel. Le dialogue Exception de transaction de base de données apparaît.

rror in Transaction: NEWADDRESSESUSERID eason for the error: NOT NULL constraint failed: new_addresses.is_shipping
eason for the error: NOT NULL constraint failed: new_addresses.is_shipping
NOT NULL constraint failed: new_addresses.is_shipping
QL statement:
NSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "type", "city",
street, number) VALUES $(\ell, \ell, \ell, \ell, \ell, \ell, \ell, \ell)$
Choose how to continue
Choose how to continue O Rollback this transaction and stop
Choose how to continue O Rollback this transaction and stop Rollback this transaction and continue
Choose how to continue   Rollback this transaction and stop  Rollback this transaction and continue  Rollback/continue for all further errors
Choose how to continue O Rollback this transaction and stop Rollback this transaction and continue Rollback/continue for all further errors O Rollback top and stop

- 4. Sélectionner les options de rollback comme illustré ci-dessus, et cliquer sur OK.
- **Note :** Si un mappage est exécuté par MapForce Server, donc aucun dialogue n'apparaît et les transactions erronées sont annulées automatiquement, conformément aux paramètres de rollback configurés dans MapForce.

Dans cet exemple, une fois que le mappage termine l'exécution, toutes les transactions échouées sont annulées, et seules les transactions réussies sont archivées dans la base de données. Les erreurs rencontrées sont rédigées dans le fichier **Errors.xml**, un petit fragment est illustré ci-dessous :



Une fois que le fichier de journal a été produit, vous pouvez apercevoir la sortie qui retourne le nombre total d'erreurs dans l'exécution de MapForce, comme suit :

- 1. De retour dans le mappage, cliquer sur la touche Aperçu 🚾 dans le composant "result".
- 2. Cliquer sur l'onglet Sortie et observer le résultat.

1 Total errors: 3	
	-
Mapping DB Query Output	
😵 DatabaseExceptions.mfd	4 Þ ×

Cette partie du mappage lit des données depuis le fichier de journal produit précédemment. Si le fichier journal n'existe pas, une erreur sera générée.

# 4.3 Fichiers CSV et Texte

MapForce propose une prise en charge pour le mappage de données depuis ou vers des formats de fichier basés sur texte comme des fichiers de textes CSV (valeurs séparées par des virgules) et FLF (champ à longueur fixe). Par exemple, vous pouvez créer des transformations de données comme :

- schéma XML de/vers des formats de fichier plat
- base de données de/vers des formats de fichier plat

Veuillez noter que, en ce qui concerne les CSV, vos fichiers peuvent contenir non seulement des virgules en tant que délimiteurs, mais aussi des tabulateurs, des point-virgules, des espaces ou toute autre valeur personnalisée.

Outre les fichiers CSV et FLF, le mappage de et vers des fichiers de texte contenant des structures plus complexes ou personnalisées est possible grâce à MapForce FlexText (ce module est disponible dans l'édition MapForce Enterprise Edition). À la base, FlexText vous permet de définir la structure de vos données de texte personnalisé (en utilisant un soit-disant "modèle FlexText"), pour le mapper vers d'autres formats.

Le mappage de données de ou vers des fichiers de texte est pris en charge dans un des langages suivants : Java, C#, C++, ou BUILT-IN.

Il existe deux moyens de générer des données avec un fichier plat mappé :

- En cliquant sur l'onglet Sortie qui génère un aperçu en utilisant le moteur d'exécution Built-in. Vous pouvez aussi enregistrer les résultats de mappage en sélectionnant l'option de menu Sortie |
   Enregistrer le fichier de sortie, ou en cliquant sur l'icône .
- En sélectionnant Fichier | Générer du code dans | Java, C#, ou C++, et en compilant et en exécutant le code généré.

# 4.3.1 Exemple : Mapper des fichiers CSV vers XML

L'objectif de cet exemple est de créer un mappage qui lit les données depuis un simple fichier CSV et l'écrit dans un fichier XML. Les fichiers utilisés dans l'exemple sont disponibles dans le dossier

- 1. Choisir un des langages de transformation : Java, C#, C++ ou BUILT-IN.
- 2. Ajouter un composant de fichier de texte dans la surface de mappage (dans le menu Insérer, cliquer

sur Fichier texte, ou cliquer sur la touche de la barre d'outils Insérer fichier texte( $\square$ ).

 Dans le dialogue Paramètres de composant, cliquer sur Fichier d'entrée et chercher le fichier Altova\_csv.csv. Les contenus de fichier sont à présent visibles dans la partie inférieure du dialogue. Veuillez noter que seules les 20 premières lignes du fichier de texte sont affichées lorsque vous vous trouvez en mode d'aperçu.

😢 Paramètres de co	mposant								×
Nom de composant:	ant: Fichier texte							OK Annuler	
Fichier Entrée / Sortie									
C:\Users\Document	s\Altova\MapForce2	020\Ma	pForceExample	es\Tutorial\	Altov	a_csv.csv	Fichier E	Intrée	
	Eichier Sortie								
Fichier Sortie									
Enregistrer tous les chemins d'accès de requête relatifs au fichier MFD									
Encodage Entree / So	rtie								
Nom d'encodage:	Unicode UTF-8							$\sim$	
Ordre des octets:	Ordre des octets: Little Endian V Inclure indicateur								
Paramètres CSV Délimiteur de champ O Onglet O Point La première ligne	Paramètres CSV         Délimiteur de champ         Onglet O Point-virgule O Virgule Espace Personnaliser:         La première ligne contient des noms de champ         Traiter les champs vides comme absents							s	O CSV     O Fixe
Champ1	Champ2		Champ3		Cha	mp4		Champ5	^
string	string	•	string	-	strir	ng	-	string	
Vernon	Callaby		582		v.ca	llaby@nanon	ull.com	Office M	anager
Frank	Further		471		f.fu	rther@nanon	ull.com	Account	s Receivable
Loby	Matise		963		l.ma	tise@nanonu	II.com	Account	ing Managei
< <	Firstbread		621		li fire	thread@nanc	mull com	Marketir	Manager *
Ajouter champ Insér	er champ Supprim	er cham	p << >>	]					

4. Cliquer dans l'en-tête Champ1 et changer le texte en First-name. Procéder de même pour les autres champs, en suivant notre exemple : Champ 2 => Last-name, Champ 3 =>Tel-extension, Champ 4 => Email, Champ 5 => Position.

ASTUCE : Appuyer sur la touche **Tab** pour feuilleter rapidement dans les champs : header1, header2 etc.

First-name		Last-name		Tel-extensi	on	Email		Position	
string	-	string	-	string	-	string	•	string	
Vernon		Callaby		582		v.callaby@n	anonull.com	Office Manager	
Frank		Further		471		f.further@na	nonull.com	Accounts Receivable	
Loby		Matise		963		l.matise@nai	nonull.com	Accounting Manager	+
•		·				· - ·· · -		<u>-</u>	Þ

- 5. Cliquer sur OK.
- 6. Lorsque vous êtes invité à changer le nom de composant, cliquer sur "Changer le nom de composant". Le composant CSV est maintenant visible dans le mappage.
- 7. Ajouter **MFCompany.xsd** en tant que le composant XML cible du mappage (dans le menu **Insérer**, cliquer sur **Fichier /Schema XML**).

- 8. Cliquer sur **Sauter** lorsque vous êtes invité à fournir un fichier XML échantillon et choisir Company en tant que l'élément racine.
- 9. Mapper les items correspondants des deux composants, en s'assurant de mapper l'item **Rows** à l'item **Person** dans la cible de schéma.

Le connecteur provenant de l'item Rows dans le composant CSV vers l'item Person dans le schéma est essentiel, étant donné qu'il définit quels éléments seront itérés. Autrement dit, pour chaque ligne dans le fichier CSV, un nouvel élément Person sera créé dans le fichier de sortie XML.



10. Cliquer sur l'onglet Sortie pour voir le résultat.



Les données provenant du fichier CSV sont maintenant mappées avec succès à un fichier XML.

# 4.3.2 Exemple : Itérer à travers des items

Cet exemple illustre comment créer des itérations (plusieurs lignes) dans un fichier CSV cible. Le fichier de design de mappage qui accompagne cet exemple est disponible sous le chemin suivant : <br/>
<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Tut-xml2csv.mfd.



#### Tut-xml2csv.mfd

Ce mappage est resté inachevé intentionnellement. Si vous tentez de valider le fichier d'exemple en utilisant la commande de menu **Fichier | Valider le mappage**, vous remarquerez que des avertissements de validation se produiront. De même, si vous prévisualisez la sortie de mappage, une ligne individuelle est produite, ce qui peut constituer votre objectif (ou pas).

Partons du principe que votre objectif est de créer plusieurs lignes dans le fichier CSV depuis une séquence d'items dans le fichier XML. Vous pouvez obtenir ce résultat en tirant une connexion dans l'item Rows du fichier CSV cible.

Par exemple, pour itérer dans tous les bureaux et obtenir une sortie qui apparaisse dans le fichier CSV, il est nécessaire de connecter Office à Rows. De fait, vous indiquez à MapForce que : pour chaque item Office du XML de source, il faut créer une ligne dans le fichier CSV cible.



Les items Rows dans le composant CSV agissent comme un itérateur pour la séquence des items qui y sont connectés. Ainsi, si vous connectez l'item Office, la sortie crée une ligne pour chaque bureau trouvé dans le XML de source.

```
1 "Microtech, Inc.",Level 1 support,Major Ave 1,Vancouver,558833
2 "Microtech Partners, Inc.",Level 2 support,Perro Bvd 1324,Otto
3
```

De manière semblable, si vous connectez **Department** à l'item **Rows**, une ligne sera produite pour chaque département trouvé dans le XML de source.



La sortie ressemblerait à l'exemple suivant :

```
"Microtech, Inc.", Admin, Major Ave 1, Vancouver, 5588339, Clive, Clo
"Microtech, Inc.", Sales and Marketing, Major Ave 1, Vancouver, 558
"Microtech, Inc.", Manufacturing, Major Ave 1, Vancouver, 5588339, K
"Microtech, Inc.", Level 1 support, Major Ave 1, Vancouver, 5588339
"Microtech Partners, Inc.", Admin, Perro Bvd 1324, Ottowa, 3549202,
"Microtech Partners, Inc.", Sales and Marketing, Perro Bvd 1324, Ottow
"Microtech Partners, Inc.", Level 2 support, Perro Bvd 1324, Ottow
```

Enfin, le mappage de Person à l'item Rows résulte dans une sortie de toutes les Persons. Dans ce cas, MapForce itérera dans tous les enregistrements comme suit : chaque Person dans chaque Department, dans chaque Office.

# 4.3.3 Exemple : Créer des hiérarchies depuis des fichiers texte CSV et de longueur fixe

Cet exemple est disponible sous le chemin suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Tut-headerDetail.mfd. L'exemple utilise un fichier CSV (Orders.csv) qui dispose du format suivant :

- Champ 1 : H définit un enregistrement d'en-tête et D un enregistrement de détail.
- Champ 2 : Une clé commune pour les enregistrements d'en-tête et de détail.
- Chaque enregistrement Header ou Detail sur une ligne séparée.

Les contenus du fichier Orders.csv sont indiqués ci-dessous.

```
H,111,332.1,22537.7,,Container ship,,,
D,111,A-1579-227,10,3,400,Microtome,,
D,111,B-152-427,7,6,1200,Miscellaneous,,
H,222,978.4,7563.1,,Air freight,,,
D,222,ZZ-AW56-1,10,5,10000,Gas Chromatograph,,
```

L'objectif du mappage est le suivant :

- Mapper le fichier plat CSV à un fichier XML hiérarchique
- Filtrer les enregistrements Header, conçus avec un H
- Associer les enregistrements de détail respectifs, conçus avec un D, avec chacun des enregistrements d'en-tête.





Afin d'y parvenir, les enregistrements d'en-tête et de détail doivent avoir un champ en commun. Dans ce cas, le champ commun, ou la clé, est le deuxième champ du fichier CSV, c'est à dire OrderNo. Dans le fichier CSV, l'enregistrement du premier en-tête et les deux enregistrements de détail suivants contiennent la valeur commune 111.

Le fichier Orders.csv a été inséré deux fois pour rendre le mappage plus intuitif.

Le fichier de schéma **Tut-headerDetail.xsd** possède une structure hiérarchique : "Order" est l'élément racine, avec "Header" en tant que son élément enfant, et "Detail" étant un élément enfant de "Header".

Le premier fichier Orders.csv fournit les enregistrements **Header** (et tous les champs mappés) vers l'item "Header" dans le fichier cible de schéma. Le composant de filtre est utilisé pour filtrer tous les enregistrements ne commençant pas avec H. L'item **Rows** fournit ces enregistrements filtrés à l'item "Header" dans le fichier de schéma.

Le second fichier Orders.csv fournit les enregistrements **Detail** (et tous les champs mappés) en filtrant les enregistrements "Detail" qui correspondent à la clé OrderNo de l'enregistrement "Header". Cela est obtenu en :

- Comparant le champ OrderNo de l'enregistrement "Header" avec le même champ des enregistrements "Detail", en utilisant la fonction equal (le <u>contexte de priorité</u> est défini sur le paramètre a pour une meilleure performance).
- Utilisant la fonction **Logical-and** non seulement pour fournir ces enregistrements "Detail" contenant le même champ OrderNo, en tant qu'enregistrement "Header".

L'item **Rows** fournit ces enregistrements filtrés aux items "Header" et "Detail" dans le fichier de schéma, par le biais du paramètre on-true du composant de filtre.

Cliquer sur l'onglet Sortie pour produire le fichier XML affiché ci-dessous. Chaque enregistrement "Header" contient ses données et tous les enregistrements "Detail" associés qui ont le même "OrderNo".

1		xml version="1.0" encoding="UTF-8"?					
2	Corder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation						
3	ė	⇔ <header></header>					
4		<recordtype>H</recordtype>					
-5		<orderno>111</orderno>					
6		<total veight="">332.1</total>					
- 7		<totalunitcost>22537.7</totalunitcost>					
8		<currency></currency>					
9		<shipping-details>Container ship</shipping-details>					
10	ę	) <detail></detail>					
11		<recordtype>D</recordtype>					
12		<orderno>111</orderno>					
13		<productno>A-1579-227</productno>					
14		<unit veight="">10</unit>					
15		<unitno>3</unitno>					
16		<unitcost>400</unitcost>					
17		<unit-description>Microtome</unit-description>					
18	ŀ	-					
19	ę	) <detail></detail>					
20		<recordtype>D</recordtype>					
21		<orderno>111</orderno>					
22		<productno>B-152-427</productno>					
23		<unit veight="">7</unit>					
24		<unitno>6</unitno>					
25		<unitcost>1200</unitcost>					
26		<unit-description>Miscellaneous</unit-description>					
27	ŀ						
28							

Examinons à présent un autre exemple, qui utilise un fichier CSV légèrement différent et qui est disponible dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\** en tant que **Head-detail-inline.mfd**. La différence est que :

- Aucun désignateur d'enregistrement (H ou D) n'est disponible
- Un champ de clé commun, le premier champ du fichier CSV, existe toujours aussi bien pour les enregistrements header que detail (Head-key, Detail-key...). Le champ est mappé sur OrderNo dans la cible de schéma
- Le champ "Header" et tous les champs "Detail" respectifs se trouvent sur la même ligne.

111,332	.1,22537	7.7.,Container	ship,,,111,	A-1579-227	,10,3,400,	Microtome,	,111,B-15
222,978	.4,7563.	1. Air freight	.,,222,ZZ-A	W56-1,10,5	,10000,Gas	s Chromatog	raph,

Le mappage a été conçu comme suit :

- Les champs clé sont mappés vers les items OrderNo respectifs dans la cible de schéma.
- L'item "Detail" dans le fichier cible de schéma a été dupliqué et est affiché en tant que **Detail (2)**. Cela vous permet de mapper le deuxième ensemble d'enregistrements de détail vers l'item correct.
- Le résultat de ce mappage est en principe le même fichier XML qui a été produit dans le premier exemple.



Head-detail-inline.mfd

# 4.3.4 Configurer les options CSV

Une fois avoir ajouté un composant de texte dans la surface de mappage, vous pouvez configurer les paramètres applicables depuis le dialogue Paramètres de composant. Vous pouvez ouvrir ce dialogue d'une des trois manières suivantes :

- Choisir le composant et, dans le menu Composant, cliquer sur Propriétés.
- Double-cliquer sur l'en-tête de composant.
- Cliquer avec la touche de droite sur l'en-tête de composant, puis cliquer sur Propriétés.

😢 Paramètres de co	mposant						×
Nom de composant:	People						OK Annuler
Fichier Entrée / Sortie							
People.txt					Fichier Er	ntrée	
					Fichier S	ortie	
Enregistrer tous l	es chemins d'accès de re	quête relatifs a	au fichier MFD				
Encodage Entrée / So	rtie						
Nom d'encodage:	Unicode LITE-9					~	
Nom a encouage.	Unicode UTF-0		_			~	
Ordre des octets:	Little Endian		$\sim$	Inclure indicat	teur		
Paramètres CSV Délimiteur de cham O Onglet O Point La première ligne	p -virgule	ispace () Per amp	sonnaliser:	Caractère de g Aucun ( Ajouter si n	guillemets )' iécessaire ijours	•	© CSV     ○ Fixe
					-		
Company	Department	First	Last	Title		EMail	^
string 💌	string 💌	string 💌	string 💌	string	-	string	
Nanonull, Inc.	Administration	Vernon	Callaby	Office Manager		v.callaby	@nanonull.
Nanonull, Inc.	Administration	Frank	Further	Accounts Receivab	le	f.furthe	@nanonull.
Nanonull, Inc.	Administration	Loby	Matise	Accounting Manag	er Even	I.matise	@nanonull.(
	Marketing	106	Martin	Marketing Manage	FUILO	martin	> >
Ajouter champ Insér	er champ Supprimer ch	namp <<	>>				

Dialogue Paramètre de composant de texte (en mode CSV)

Les paramètres disponibles sont les suivants :

Nom du composant

Le nom de composant est généré automatiquement lorsque vous

	<ul> <li>créez un composant. Néanmoins, vous pouvez changer le nom à tout moment. Le nom de composant peut contenir des espaces et les caractères de point final. Le nom de composant ne doit pas contenir barres obliques, des barres obliques inversées, des points-virgule, des guillemets doubles, des espaces de début et de fin. Si vous souhaitez changer le nom du composant, veuillez noter :</li> <li>Si vous souhaitez déployer le mappage vers FlowForce Server, le nom du composant doit être unique.</li> <li>Il est recommandé de n'utiliser que des caractères qui peuvent être saisis dans la ligne de commande. Les caractères de types nationaux peuvent présenter des encodages différents dans Windows et dans la ligne de commande.</li> </ul>
Fichier d'entrée	Spécifie le fichier à partir duquel MapForce lira les données. Ce champ est pertinent pour un composant de source et est rempli lorsque vous créez tout d'abord le composant puis que vous l'attribuez à un fichier de texte. Le champ peut rester vide si vous utilisez le composant de fichier de texte en tant que cible pour votre mappage. Dans un composant de source, MapForce utilise la valeur de ce champ pour lire des noms de colonne et pour consulter les contenus du fichier de texte d'instance. Pour choisir un nouveau fichier, cliquer sur <b>Fichier d'entrée</b> .
Fichier de sortie	Spécifie le fichier vers lequel MapForce écrira des données. Ce champ est pertinent pour un composant de cible
	Pour choisir un nouveau fichier, cliquer sur Fichier de sortie.
Enregistrer tous les chemins de fichier relatifs au fichier MFD	Lorsque cette option est activée, MapForce enregistre les chemins de fichier affichés dans le dialogue des Paramètres de composant relatifs à l'emplacement du fichier MapForce Design (.mfd). Ce paramètre touche les fichiers d'entrée et de sortie utilisés par le composant du texte. Voir aussi <u>Utiliser des chemins relatifs sur un composant</u> <sup>83</sup> .
Encodage Entrée / Sortie	<ul> <li>Vous permet de spécifier les paramètres suivants du fichier d'instance de sortie :</li> <li>Nom d'encodage</li> <li>Ordre d'octet</li> <li>Si le caractère "byte order mark" (BOM) doit être inclus.</li> <li>Par défaut, tout nouveau composant doit avoir un encodage défini dans l'option Encodage par défaut pour les nouveaux composants. Vous pouvez accéder à cette option depuis Outils   Options, onglet Généralité.</li> </ul>
Délimiteur de champ	Les fichiers CSV sont délimités par des virgules "," par défaut. Cette option vous permet de sélectionner les caractères <b>Tabulateur</b> , <b>Point-</b>

	<b>virgule</b> , ou <b>Espace</b> en tant que délimiteurs. Vous pouvez aussi saisir un délimiteur personnalisé dans le champ <b>Personnaliser</b> .					
Première ligne contient des noms de champ	Choisir cette option pour instruire MapForce à traiter les valeurs dans le premier enregistrement du fichier de texte en tant qu'en-têtes de colonne. Les en-têtes de colonne apparaissent ensuite en tant que noms d'item dans le mappage.					
Traiter les champs vides en tant qu'absent	Lorsque cette option est activée, les champs vides dans le fichier de source ne produiront pas un item vide correspondant (élément ou attribut) dans le fichier cible.					
	Par exemple, l'enregistrement CSV "General outgassing pollutants,,,," consiste en quatre champs, dont les trois derniers sont vides.					
	Si on part du principe que la sortie est un fichier XML, lorsque cette option est désactivée, les champs vides seront créés dans la sortie avec une valeur vide (dans cet exemple, les éléments Last, Title et Email):					
	33       Image: symbol         34 <first>General outgassing pollutants</first> 35 <last></last> 36 <title></title> 37 <email></email> 38					
	Lorsque cette option est activée, les champs vides ne seront pas créés dans la sortie :					
	38 <person>         39           40</person>					
Caractère de guillemets	Si votre sortie contient des guillemets autour des valeurs de champ, choisir le caractère de guillemet qui existe dans le fichier de source. Le même paramètre sera aussi utilisé pour les fichiers de sortie.					
	Quote character None ' • • • • • • • • • • • • • • • • • •					
	En ce qui concerne les fichiers de sortie, vous pouvez spécifier des paramètres supplémentaires :					

	Ajouter si nécessaire	Ajoute le caractère de guillemet sélectionné uniquement aux champs dans lesquels le texte contient le délimiteur de champ, ou des sauts ligne.	
	Ajouter toujours	Ajoute le caractère de guillemet sélectionné à les champs du fichier CSV généré.	
CSV / Fixe	Change le type de com longueur fixe).	posant soit sur CSV ou FLF (champ de	
Zone d'aperçu	La partie inférieure du dialogue affiche un aperçu de jusqu'à 20 lignes du fichier sélectionné en tant qu'entrée ou sortie.		
	Si nécessaire, vous pou structure du fichier exis	uvez créer la structure du fichier (ou changer la tant), comme suit.	
	Ajouter champ	Crée un nouveau champ après le dernier enregistrement CSV.	
	Insérer champ	Crée un nouveau champ juste avant enregistrement CSV sélectionné actuellement.	
	Supprimer champ	Supprime le champ sélectionné actuellement.	
	<<	Déplace le champ sélectionné actuellement d'u position vers la gauche.	
	>>	Déplace le champ sélectionné actuellement d'u position vers la droite.	
	Pour changer le nom d' Field1), et saisir la nou champ ne sont pas édit contient des noms de	un champ, cliquer sur l'en-tête (par exemple, ivelle valeur. Veuillez noter que les noms de tables lorsque l'option <b>Première ligne</b> o <b>champ</b> est activée.	
	Field1		
	string		
		a dennésa d'un champ, chaisir la voleur requise	
	depuis la liste déroulan ainsi, si les données d'a pas, les données sont i	te. MapForce contrôle le type de données, entrée et le format de champ ne concordent marguées en rouge.	

Company	
decimal 🔹	
Nanonull, Inc.	
Nanonull, Inc.	
Nanonull, Inc.	
Les types de cha XML par défaut. F YYYY-MM-DD.	mp sont basés sur les types de données de schéma Par exemple, le type Date se présente sous la forme

# 4.3.5 FLF à base de données

Cet exemple illustre une opération de mappage de données entre un fichier de texte de longueur fixe (FLF) et une base de données Microsoft Access. Les fichiers utilisés dans l'exemple sont disponibles dans le dossier Tutorial. Le fichier de texte de source et la base de données cible stockent une liste d'employés. Dans le fichier de source, les enregistrements sont délimités implicitement par leur taille, comme suit :

Position et nom du champ	Taille (en caractères)
Champ 1 (Prénom)	8
Champ 2 (Nom de famille)	10
Champ 3 (Numéro de poste)	3
Champ 4 (E-mail)	25
Champ 5 (Position)	25

Le but du mappage est de mapper les données FLF au composant de base de données. Nous voulons également mapper les extensions de téléphone avec un nouveau préfixe. Pour ce faire, suivre les étapes suivantes :

### Étape 1: Insérer et configurer le composant de texte

- Choisir l'option de menu Insérer | Fichier texte, ou cliquer sur l'icône Insérer fichier texte ((Insérer fichier texte) .
- 2. Cliquez sur fichier d'entrée dans la boîte de dialogue **Paramètres de composant** (*voir ci-dessous*) et sélectionnez Altova-FLF.txt.

😢 Paramètres de co	mposant		×
Nom de composant:	Fichier texte		OK Appuler
Fichier Entrée / Sortie	1		Annaici
C:\Users\n.kourims	ky\Documents\Altova\MapForce2020\MapForceExamples\T	Fichier Entrée	
		Fichier Sortie	
Enregistrer tous l	es chemins d'accès de requête relatifs au fichier MFD		
Encodage Entrée / Sc	rtie		
Nom d'encodage:	Unicode UTF-8	~	
Ordre des octets:	Little Endian $\sim$	Inclure indicateur	
Paramètres CSV Délimiteur de cham O Onglet O Point	p :-virgule	Caractère de guillemets	() CSV
Traiter les champs	contrent des noms de champ s vides comme absents	<ul> <li>Ajouter si nécessaire</li> <li>Ajouter toujours</li> </ul>	○ Fixe
Champ1 string Vernon##Callaby###	582v.callaby@nanonull.com###Office Manager#####	######Frank###Further###47	1f.further@nan
Ajouter champ Insé	er champ Supprimer champ << >>		

- 3. Choisir Fixe.
- 4. Décocher la case Assumer la présence des délimiteurs d'enregistrement.
- Les trois lignes marquées en jaune sont éditables, et vous permettent de spécifier i) le nom du champ ii) le type de données et iii) la taille du champ. Saisir 8 en guise de la nouvelle taille du champ et appuyer sur Entrée. Vous verrez apparaître plus de données dans la première colonne qui est maintenant définie avec une largeur de 8 caractères.
- 6. Cliquer sur **Ajouter champ** pour ajouter un nouveau champ, et définir la longueur du deuxième champ à 10 caractères.
- 7. Utiliser la même méthode pour créer trois autres champs avec les longueurs suivantes : 3, 25, et 25 caractères. Puis changer les en-tête de champ,tel qu'affiché ci-dessous.

First	Last	TelExt	Email	Title	^
string 💌	string 💌	string 💌	string 💌	string	
8	10	3	25	25	
Susi####	Sanna####	753	s.sanna@nanonull.com#####	Art Director####	
Fred#### Landis####		951	f.landis@nanonull.com####	Program Manage	
Michelle	Butler####	654	m.landis@nanonull.com#### Software F		v
<				>	
Ajouter champ Ins <u>é</u> rer o	champ Supprimer champ	>>			

- Dans le groupe Paramètres de champ à longueur fixe, choisissez Personnaliser et saisissez le caractère dièse (#). Cela indique à MapForce de traiter le caractère # en tant que caractère de remplissage.
- 9. Cliquer sur OK.
- 10. MapForce vous demandera si vous souhaitez changer le nom du composant afin qu'il corresponde aux fichiers d'instance. Cliquer sur **Changer le nom du composant**. Le composant Altova-FLF apparaît dans la fenêtre de mappage.

## Étape 2: Insérer le composant de base de données

- 1. Choisir la commande de menu Insérer | Base de données, choisir Microsoft Access, puis cliquer sur Suivant.
- 2. Choisir la base de données altova.mdb et cliquer sur Se connecter.
- 3. Choisir la table **Person** (*voir ci-dessous*) et cliquer sur **OK**.

💷 Insérer objets de base de données
Veuillez choisir les Tables, Aperçus, Procédures stockées ou créez vos propres insl SELECT définies par l'utilisateur sont stockées localement dans le fichier mfd.
Source
💎 🔽 💠 🛛 Ajouter/Modifier l'instruction SELECT 🛛 Ajouter/Modifier
altova     altova
Afficher aperçu >> Utiliser des noms d'objet par rapport au schéma par défa

## Étape 3: Concevoir le mappage

Dans la prochaine étape, nous allons créer un mappage :

- 1. Glisser la fonction <u>core | concat</u><sup>709</sup> depuis la fenêtre Bibliothèques dans le mappage.
- 2. Choisir la commande de menu **Insérer | Constante**, choisir *Nombre* en tant que type et saisir une valeur de 100 dans le champ de texte. Cette constante stocke le nouveau préfixe de poste téléphonique.
- 3. Créer le mappage comme indiqué ci-dessous.



- 4. Dans le composant de base de données, cliquer sur la touche **Action de table**. Ceci ouvre la boîte de dialogue **Actions de table de la base de données** (*voir la capture d'écran ci-dessous*).
- 5. À côté de Action sur données d'entrée, choisir Mettre à jour si. Définissez les champs First et Last sur l'action equal. Cliquez sur OK. MapForce sera invité à mettre à jour la table Person uniquement si les prénom et nom de famille dans le fichier source sont égaux champs de base de données correspondants. Si cette condition est vraie, numéro de poste est préfixé avec 100, et sera copié dans le champ PhoneExt de la table Person.

Actions à exécuter pour chaque enregistrement					
Les données d'entrée sont comparé Cocher la case "NULL equal" pour tr Si toutes les comparaisons sont vra	ées aux donné aiter les valeu ies, l'action sp	es de la table d urs NULL en tant pécifique est exé	e BD, a qu'ég cutée	avec les opérateurs définis ici. ales.	
Action sur enregistrement	NULL égale	Insérer tout	<b>_</b>		$\sim$
💵 PrimaryKey			•		
ForeignKey			-		
D EMail			-		
I First		equal	-		
Last		equal	-		
PhoneExt			-		
1 Title			-		
					~

Pour générer l'instruction SQL (pour un aperçu dans MapForce), cliquer sur l'onglet Sortie. Pour exécuter les instructions SQL par rapport à la base de données, cliquer sur la touche Exécuter script SQL

# 4.3.6 Configurer les options FLF

Une fois avoir ajouté un composant de texte dans la surface de mappage, vous pouvez configurer les paramètres applicables depuis le dialogue Paramètres de composant. Vous pouvez ouvrir ce dialogue d'une des trois manières suivantes :

• Choisir le composant et, dans le menu Composant, cliquer sur Propriétés.

- Double-cliquer sur l'en-tête de composant.
- Cliquer avec la touche de droite sur l'en-tête de composant, puis cliquer sur Propriétés.

Paramètres de cor	nposant				>	
Nom de composant:	Quotations			]	OK Annuler	
Fichier Entrée / Sortie						
Quotations.dat				Fichier Entrée		
Quetations output	dat			Fichier Sertie		
Quotations-output.	ual			Fichier Sorue		
Enregistrer tous l	es chemins d	'accès de requête relatifs au fichie	r MFD			
Encodage Entrée / So	rtie					
Nom d'encodage:	Unicode I ITI	8		~		
- to the state	onicode on	•				
Ordre des octets: Little Endian V Indure indicateur						
Caractère de rempl Espace O Assumer la préser	issage Période nce des cara s vides comm	Personnaliser:	nent		⊖ CSV (● Fixe	
Champ1		Champ2	Champ3	Champ4		
tring	•	string 🗾	string 💌	string		
i		8	4	120	120	
lbert		Einstein	1879	Only two things are infinite, the		
Voody		Allen	1935	When I was kidnapped, my parer		
<						

Dialogue Paramètre de composant de texte (en mode champ de longueur fixe)

Les paramètres disponibles sont les suivants :

Nom du composant	Le nom de composant est généré automatiquement lorsque vous créez un composant. Néanmoins, vous pouvez changer le nom à tout moment. Le nom de composant peut contenir des espaces et les caractères de point final. Le nom de composant ne doit pas contenir barres obliques, des barres obliques inversées, des points-virgule, des guillemets doubles, des espaces de début et de fin. Si vous souhaitez changer le nom du composant, veuillez noter :
	Si vous souhaitez déployer le mappage vers FlowForce

	<ul> <li>Server, le nom du composant doit être unique.</li> <li>Il est recommandé de n'utiliser que des caractères qui peuvent être saisis dans la ligne de commande. Les caractères de types nationaux peuvent présenter des encodages différents dans Windows et dans la ligne de commande.</li> </ul>			
Fichier d'entrée	Spécifie le fichier à partir duquel MapForce lira les données. Ce champ est pertinent pour un composant de source et est rempli lorsque vous créez tout d'abord le composant puis que vous l'attribuez à un fichier de texte. Le champ peut rester vide si vous utilisez le composant de fichier de texte en tant que cible pour votre mappage.			
	Dans un composant de source, MapForce utilise la valeur de ce champ pour lire des noms de colonne et pour consulter les contenus du fichier de texte d'instance.			
	Pour choisir un nouveau fichier, cliquer sur Fichier d'entrée.			
Fichier de sortie	Spécifie le fichier vers lequel MapForce écrira des données. Ce champ est pertinent pour un composant de cible.			
	Pour choisir un nouveau fichier, cliquer sur Fichier de sortie.			
Enregistrer tous les chemins de	Lorsque cette option est activée, MapForce enregistre les chemins de			
fichier relatifs au fichier MFD	fichier affichés dans le dialogue des Paramètres de composant relatifs			
	à l'emplacement du fichier MapForce Design (.mfd). Ce paramètre			
	touche les fichiers d'entrée et de sortie utilisés par le composant du			
	texte. Voir aussi <u>Utiliser des chemins relatifs sur un composant</u> <sup>3</sup> .			
Encodage Entrée / Sortie	Vous permet de spécifier les paramètres suivants du fichier d'instance			
	de sortie :			
	<ul> <li>Nom d'encodage</li> <li>Ordre d'octet</li> <li>Si le caractère "byte order mark" (BOM) doit être inclus.</li> <li>Par défaut, tout nouveau composant doit avoir un encodage défini dans l'option Encodage par défaut pour les nouveaux composants. Vous pouvez accéder à cette option depuis Outils   Options, onglet Généralité.</li> </ul>			

Caractère de remplissage	Cette option vous permet de définir les caractères qui doivent être utilisées pour terminer ou remplir le reste du champ (fixe) lorsque les données entrantes sont plus courtes que les définitions de champ respectives. Le champ personnalisé vous permet de définir votre propre caractère de remplissage dans le champ Custom. Si les données entrantes contiennent déjà des caractères de remplissage spécifiques et que vous saisissez le même caractère de remplissage dans le champ Custom, alors les données entrantes seront débarrassés de ces caractères de remplissage !			
Assumer la présence des délimiteurs d'enregistrement	Cette option est utile lorsque vous souhaitez lire des données provenant d'un fichier plat de source qui ne contient pas de délimiteurs d'enregistrement comme CR/LF, ou lorsque vous souhaitez produire un fichier FLF plat cible sans délimiteur d'enregistrement.			
	délimiteurs d'enregistrement <sup>463</sup> ci-dessous.			
Traiter les champs vides comme absents	Lorsque cette option est activée, les champs vides dans le fichier de source ne produiront pas un item vide correspondant (élément ou attribut) dans le fichier cible.			
	En partant du principe que la sortie est un fichier XML, lorsque cette option est désactivée, les champs vides seront créés dans la sortie avec une valeur vide (dans cet exemple, les éléments Last, Title, et Email):			
	33 <person>         34       <first>General outgassing pollutants</first>         35       <last></last>         36       <title></title>         37       <email></email>         38</person>			
	Lorsque cette option est activée, les champs vides ne seront pas créés dans la sortie :			
	38           39			
CSV / Fixe	Change le type de composant soit sur CSV ou FLF (champs de longueur fixe).			

Zone d'aperçu	La partie inférieure du dialogue affiche un aperçu de jusqu'à 20 lignes choisies en tant qu'entrée ou sortie. Si nécessaire, vous pouvez créer la structure du fichier (ou changer la structure de celle existante), comme suit :			
	Ajouter champ	Crée un nouveau champ après le dernier enregistrement .		
	Insérer champ	Crée un nouveau champ juste avant l'enregistrement sélectionné actuellement.		
	Supprimer champ	Supprime le champ sélectionné actuellement.		
	<<	Déplace le champ sélectionné actuellement d'u position vers la gauche.		
	>>	Déplace le champ sélectionné actuellement d'u position vers la droite.		
	Pour changer le nom d'un champ, cliquer sur l'en-tête (par exemple,			
	Champ1), et saisir la nouvelle valeur.			
	Field4			
	etring			
	Vernon##			
	Frank###			
	Loby####			
	Joe#####			
	Susi####			
	Fred####			
	Pour changer le type de	e données d'un champ, choisir la valeur requise		
	depuis la liste déroulante. MapForce contrôle le type de données,			
	ainsi, si les données d'entrée et le format de champ ne concordent			
	pas, les données sont r	marquées en rouge.		

Field1
decimal 🔹
8
Vernon##
Frank###
Loby####
Joe#####
Susi####
Fred####
Pour définir la taille du champ dans les caractères, saisir la taille de champ dans la troisième ligne depuis le haut.

### Comprendre l'option "Assumer la présence des délimiteurs d'enregistrement"

Afin de mieux comprendre cette option, ouvrir le fichier **Altova-FLF.txt** disponible dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\**. Noter que le fichier consiste en des enregistrements longs de 71 caractères, sans délimiteurs comme CR/LF. Si vous souhaitez lire des données depuis ce fichier spécifiquement, vous devez partager ce fichier dans des enregistrements. C'est à dire, créer plusieurs champs dont la taille totalise jusqu'à 71 caractères (comme indiqué ci-dessous), et puis désactiver **Assumer la présence des délimiteurs d'enregistrement**. Pour un exemple étape par étape, voir <u>Exemple :</u> Mapper des fichiers de texte de longueur fixe dans des bases de données <sup>459</sup>.

Paramètres de champ à l Caractère de remplissa	longueur fixe age iriode OPersonnaliser:			0.00	
Assumer la présence	des caractères de délimitatio des comme absents	n d'enregistrement		<ul> <li>Esv</li> <li>Fixe</li> </ul>	
Champ1	Champ2	Champ3	Champ4	Champ5	~
string 🔹	string 🔹	string 💌	string 💌	string	
8	10	3	25	25	
Vernon##	Callaby###	582	v.callaby@nanonull.com###	Office Manager####	
Frank###	Further###	471	f.further@nanonull.com###	Accounts Receivable#	
Loby####	Matise####	963	I.matise@nanonull.com####	Accounting Manager	
Joe#####	Firstbread	621	j.firstbread@nanonull.com	Marketing Manager I	
Susi####	Sanna####	753	s.sanna@nanonull.com#####	Art Director######	
Fred####	Landis####	951	f.landis@nanonull.com####	Program Manager##	<b>v</b> .
< >					
Ajouter champ Insérer of	champ Supprimer champ	« »			
Si vous souhaitez écrire des données depuis ce fichier dans un fichier de destination qui utilise la même structure, puis activer **Assumer la présence des délimiteurs d'enregistrement** crée un nouvel enregistrement tous les 71 caractères.

1	Vernon##Callaby###582v.callaby@nanonull.com###Office Manager######################
2	Frank###Further###471f.further@nanonull.com###Accounts Receivable######
3	Loby####Matise####9631.matise@nanonull.com#####Accounting Manager#######
4	Joe#####Firstbread621j.firstbread@nanonull.comMarketing Manager Europe#
5	Susi####Sanna#####753s.sanna@nanonull.com#####Art Director###############
6	Fred####Landis####951f.landis@nanonull.com####Program Manager############
7	MichelleButler####654m.landis@nanonull.com#####Software Engineer########
8	Ted#####Little#####852t.little@nanonull.com####\$Software Engineer########
9	Ann#####Way######951a.way@nanonull.com#######Technical Writer####################
10	Liz#####Gardner###7531.gardner@nanonull.com###Software Engineer########
11	Paul####Smith#####334p.smith@nanonull.com#####Software Engineer########
12	Alex####Martin#####778a.martin@nanonull.com#####IT Manager#######################
13	George##Hammer####223g.hammer@nanonull.com####Web Developer####################################
14	Jessica#Bander####241j.band@nanonull.com######Support Engineer#########
15	Lui#####King######3451.king@nanonull.com######Support Engineer#########
16	Steve###Meier#####114s.meier@nanonull.com#####Office Manager######################
17	Theo####Bone######331t.bone@nanonull.com######Accounts Receivable######
18	Max#####Nafta#####122m.nafta@nanonull.com#####PR & Marketing Manager US
19	ValentinBass######716v.bass@nanonull.com######IT Manager##########################
20	Carl####Franken###147c.franken@nanonull.com###Support Engineer###################################
21	Mark####Redgreen##152m.redgreen@nanonull.com##Support Engineer###################################
22	

Le résultat de mappage lorsque "Assumer la présence des délimiteurs d'enregistrement" est activé

Si **Assumer la présence des délimiteurs d'enregistrement** est désactivé, le résultat de mappage apparaît sous la forme d'un long string.



Le résultat de mappage lorsque "Assumer la présence des délimiteurs d'enregistrement" est désactivé

# 4.4 Binary Files

MapForce permet de lire des données BLOB (binary large object) depuis des fichiers binaires dans un mappage, puis de les consommer sans changer la structure interne des données binaires (raw). Par exemple, vous pouvez enregistrer des données binaires dans un champ de base de données BLOB, vers un champ de type xs:base64Binary dans un fichier XML, ou l'envoyer vers un service Web\*.

\* Les appels de service Web sont uniquement pris en charge dans MapForce Enterprise Edition.

Vous pouvez aussi créer des mappages qui lisent des données binaires depuis des sources (comme un champ BLOB dans une base de données, un champ de type xs:base64Binary dans un fichier XML, ou un service Web) puis écrire des fichiers binaires dans le disque.

Les scénarios suivants sont juste des exemples qui profitent de la capacité de lire ou d'écrire des fichiers binaires :

- Extraire du contenu binaire encodé en tant que données base-64 depuis un fichier XML et l'enregistrer sur le disque (par exemple, en tant que fichier PDF)
- Traiter les fichiers d'image stockés sur le disque puis les envoyer en tant que contenu binaire encodé en base-64 dans un service Web
- Extraire du contenu BLOB depuis une table de base de données et l'enregistrer en tant que fichiers d'image sur le disque (un fichier d'image pour chaque ligne dans la table de base de données)
- Lire des fichiers d'image depuis le disque et les enregistrer dans une table de base de données en tant que champs de données BLOB.
- **Note :** Le mappage de données de ou vers des fichiers binaires nécessite <u>BUILT-IN</u><sup>(23)</sup> en tant que langage de transformation. Vous pouvez consulter le mappage dans MapForce (et enregistrer les fichiers de sortie, le cas échéant) ou choisir de l'exécuter avec MapForce Server (licencé séparément) sur un autre ordinateur ou une autre plateforme. Il ne peut pas générer un exécutable C#, C++, ou Java depuis des mappages qui lisent ou écrivent des fichiers binaires.

## Ajouter des fichiers binaires dans le mappage

Il n'y a pas de type de composant en tant que tel associé avec les fichiers binaires dans MapForce, comme c'est le cas par exemple avec XML, texte, ou les fichiers JSON. Au lieu, pour vous aider à accomplir vos objectifs, comme par exemple celui ci-dessus, les fonctions built-in MapForce suivantes sont disponibles :

- <u>read-binary-file</u><sup>745</sup>
- write-binary-file 746

Vous trouverez ces fonctions dans la fenêtre Bibliothèques (en particulier dans la bibliothèque "lang"). Pour utiliser une des fonctions dans un mappage, les glisser depuis la fenêtre Bibliothèques dans le mappage, ou double-cliquer dans une zone vide du mappage et commencer à saisir le nom de la fonction pour la choisir. Pour plus d'informations, voir <u>Ajouter des fonctions au mappage</u><sup>652</sup>.

## read-binary-file

Retourne le contenu du fichier spécifié en tant qu'un BLOB (binary large object) de type xs:base64Binary. Veuillez noter que bien que le type de données est appelé "base64Binary", la représentation interne est juste un BLOB. Ce n'est que lorsque vous mappez le résultat de la fonction dans un nœud XML de type xs:base64Binary qu'il sera encodé en base64. Vous pourriez aussi mapper le résultat de la fonction dans xs:hexBinary, dans un blob de base de données ou dans un champ binaire dans une structure Protocol Buffers.

$f_{\!8}$ read-binary-file		
> filepath	contont C	
must-exist	content	

Pour lire un fichier binaire dans un mappage, fournir son chemin en tant qu'entrée dans l'argument **filepath**. Si le **filepath** est relatif, MapForce cherchera le fichier dans le même répertoire que le mappage. L'argument **must-exist** est optionnel ; si le fichier ne peut pas être ouvert et que ce paramètre est **vrai**, le mappage lance une erreur. Si le fichier ne peut pas être ouvert et que ce paramètre est **faux**, un binaire vide est retourné.

#### write-binary-file

Cette fonction écrit du contenu binaire dans le chemin de fichier spécifié et retourne le chemin du fichier écrit. Si un fichier binaire est la seule sortie désirée, connecter le résultat de la fonction à un composant <u>sortie</u> <u>simple</u><sup>469</sup>. Étant donné que cette fonction écrit un fichier à chaque fois qu'une sortie est utilisée dans le mappage, il est recommandé de connecter le résultat de la fonction directement dans un composant cible, sans utiliser un autre traitement entre temps.

f <sub>8</sub> write-t	pinary-file
> filepath	files at h
¢ content	Thepath C

Pour écrire des fichiers binaires, fournir leurs chemins en tant qu'entrée dans l'argument **filepath**. Si **filepath** est relatif, MapForce générera le fichier dans le même répertoire que le mappage. L'argument **content** doit être connecté au contenu binaire (par exemple, un champ BLOB dans une base de données).

Lorsque vous consultez le mappage dans MapForce, la fonction génère des fichiers temporaires par défaut, au lieu d'écrire des fichiers directement sur le disque. Pour enregistrer les fichiers temporaires sur le disque, cliquer tout d'abord sur l'onglet **Sortie**, puis cliquer sur la touche de la barre d'outils **Enregistrer sortie générée u** ou **Enregistrer toutes les sorties générées**, selon les cas.

Pour configurer MapForce pour écrire une sortie directement dans des fichiers finaux au lieu des fichiers temporaires, sélectionner la commande de menu **Outils | Options**, cliquer sur **Général**, puis sélectionner l'option **Écrire directement dans les fichiers de sortie finaux**. Veuillez noter que cette option écrase tous les fichiers existants avec le même nom.

La fonction retourne le nom du fichier final (pas temporaire), même si le fichier final n'est pas encore enregistré sur le disque (c'est le cas lorsque vous consultez le mappage et que l'option Écrire directement dans les fichiers de sortie finaux est désactivée).

Veuillez noter qu'il n'est pas possible pour un mappage de relire son propre fichier de sortie.

#### Exemples

Voir <u>Exemple: Lire des fichiers binaires</u><sup>473</sup> et <u>Exemple: Écrire des fichiers binaires</u><sup>474</sup>.

# 4.4.1 Exemple: Lire des fichiers binaires

Le mappage illustré ci-dessous lit des données depuis un fichier d'image et l'écrit dans une table de base de données. La base de données de cible est SQLite. Veuillez noter que le type de données du champ de base de données **picture** est BLOB.



Pour extraire le contenu binaire du fichier, la fonction read-binary-file a été utilisée. Dans cet exemple, le premier argument, **filepath** est fourni par une constante Étant donné que le chemin est relatif, MapForce cherchera le fichier d'image dans le même répertoire que le mappage.

Dans la base de données de cible, les champs suivants sont configurés en tant que NOT NULL, le mappage doit donc les fournir :

 id - Dans cet exemple, le composant de base de données est configuré de manière à ce que id est généré par base de données au lieu d'être fourni par le mappage. Pour plus d'informations, voir <u>Insérer</u> des données dans une table<sup>286</sup>.

😢 Database Table Actions - products				
SQL statement to execute before first record         Image: SQL statement to execute before first records         Image: SQL statement to execu			ables	
Actions to execute for each record All input data are compared to the DB table data, using the operators of Select the "NULL equal" check box to treat NULL values as equal. If all comparisons are true, then the specific action is executed.				
Action on record NULL equal Insert All				
💵 id		DB-generated	-	
III title		mapped value	-	
description		mapped value	-	
picture		mapped value	-	

 title - cette valeur est fournie par un simple composant d'entrée portant le même nom. Veuillez noter qu'une valeur d'exécution design-time est définie ("product1") afin de permettre le prévisualiser le mappage. Pour plus d'informations, voir <u>Fournir des paramètres dans le mappage</u><sup>477</sup>.

🙁 Create	Input X
<u>N</u> ame:	title
<u>D</u> atatype:	string ~
	✓ Input is required
Design-tir	ne Execution
Speci	fy value
<u>V</u> alue:	product1
	OK Cancel

• **picture** - ce champ reçoit la sortie directe de la fonction **read-binary-file**.

Étant donné que le composant de cible est une base de données, la prévisualisation du mappage génère un script pseudo-SQL que vous pouvez consulter mais n'envoie pas de modifications à la base de données. Pour exécuter le script par rapport à la base de données, choisir la commande de menu **Sortie | Exécuter SQL-Script**.

# 4.4.2 Exemple: Écrire des fichiers binaires

Le mappage illustré ci-dessous lit des données depuis une base de données SQLite. La base de données a une table appelée **products**, qui présente les colonnes suivantes (champs) :

- id (entier, le seul nombre de série permanent de l'enregistrement)
- title (texte, le titre du produit)
- **description** (*texte*, la description du produit)
- picture (blob, l'image de produit)

Pour le effectuer cet exemple, seuls les champs id et picture sont pertinents.



L'objectif du mappage est d'extraire toutes les images provenant de la table de **products** et de les écrire en tant que fichiers dans le disque. Comme illustré ci-dessus, la fonction **write-binary-file** est utilisée à cet escient. Le premier argument, **filepath**, reçoit le chemin de fichier pour chaque image. Le chemin doit être unique pour éviter de contourner des fichiers, il est donc généré en concaténant la base de données unique **id** de chaque enregistrement avec le mot "image" puis en ajoutant l'extension de fichier ".png" (on part du principe que toutes les images sont en format PNG). Veuillez noter que le chemin est relatif, les fichiers sont donc écrits dans le même répertoires que le mappage.

Le second argument, **content**, reçoit le contenu binaire stocké dans la base de données.

La fonction count retourne un décompte de tous les fichiers générés et combine ce nombre avec le string "file(s) written". Un rapport est produit indiquant combien de fichiers ont été générés par le mappage. Dans cet exemple, la base de données contient uniquement deux enregistrements de produit, la sortie de mappage réfléchira cela :

•	•	Preview 1 of 3	Preview of simple output: result	$\sim$
1	2	file(s) writt	en.	*
				-
Мар	ping	DB Query	Output	

Comme évoqué précédemment, cette fonction génère des fichiers temporaires lorsque le mappage est exécuté dans l'exécution de prévisualisation. Pour consulter chaque fichier individuel, utiliser les touches flèchées pour naviguer dans l'enregistrement qui vous intéresse, cliquer sur la touche **Ouvrir avec**, et choisir un éditeur d'image.

• •	Content 2 of 3		911/4/11/11/19/19	ExampleFiles\EN\MapForce\Functions\write-binary-file\image1.png	~
File size: 3	3727 bytes				
Open	With				
Mapping	) DB Query	Output			

Pour enregistrer tous les fichiers, cliquer sur la touche de barre d'outils Enregistrer la sortie générée la ou Enregistrer toutes les sorties générées , le cas échéant.

# **5** Composants de transformation

Cette section décrit les composants de transformation qui peuvent être utilisés pour transformer les données ou les stocker temporairement pour traitement ultérieur. La liste des composants de transformation est recensée ci-dessous :

- <u>Entrée simple</u><sup>477</sup>
- Sortie simple
   <sup>488</sup>
- Variables 492
- <u>Composants Join</u>
   <sup>505</sup>
- <u>Composant de tri</u>
   <sup>517</sup>
- Filtres et conditions 523
- <u>Value-Map</u><sup>530</sup>
- Fonctions de regroupement<sup>540</sup>
- Exceptions 547

Notez que les fonctions appartiennent également aux composants de transformation. Toutefois, les <u>fonctions</u> sont organisées en section autonome.

# 5.1 Entrée simple

Si vous devez créer un mappage qui prend des paramètres en tant qu'entrée, vous pouvez le faire en ajoutant un type de composant spécial appelé "composant d'entrée simple". Les composants d'entrée simple possèdent toujours un type de données simple (par exemple, string, entier, etc.) au lieu d'une structure d'items et de séquences. Par exemple, dans le mappage illustré ci-dessous, il existe un composant d'entrée simple **count**. Son rôle est de fournir en tant que paramètre le nombre maximum de lignes qui doit être extrait depuis le fichier XML de source (avec la valeur **10** en tant que défaut). Chose importante, les nœuds fournis en tant qu'entrée de la fonction <u>first-items</u> sont triés avec l'aide d'un composant sort, afin que le mappage sorte uniquement les températures *N* les plus élevées, où *N* est la valeur du paramètre.



FindHighestTemperatures.mfd

Une autre utilisation assez commune des composants d'entrée simple est de fournir un nom de fichier au mappage. Cela est utile dans des mappages qui lisent des fichiers d'entrée ou qui écrivent des fichiers de sortie dynamiquement, voir <u>Traiter plusieurs fichiers d'entrée ou de sortie dynamiquement</u>.<sup>(833)</sup>

Vous pouvez utiliser des composants d'entré simples dans un des langages de transformation MapForce suivants :

- BUILT-IN (lorsque vous consultez la transformation de mappage directement dans MapForce, depuis l'onglet **Aperçu**)
- BUILT-IN (lorsque vous exécutez un fichier d'exécution de MapForce Server compilé)
- XSLT 1.0, XSLT 2.0, XSLT 3.0
- XQuery
- C++
- C#
- Java

Dans le cas des mappages exécutés avec MapForce Server ou par le biais du code généré, des composants d'entrée simple deviennent des paramètres de ligne de commande. Dans les cas de mappages générés comme transformations XSLT, les composants d'entrée simple correspondent aux paramètres de feuille de style dans le fichier XSLT généré.

Vous pouvez créer chaque composant d'entrée simple (ou paramètre) comme optionnel ou obligatoire (voir <u>Ajouter des composants d'entrée simple</u><sup>478</sup>). Le cas échéant, vous pouvez aussi créer des valeurs par défaut pour les paramètres d'entrée de mappage voir <u>Créer une valeur d'entrée par défaut</u><sup>481</sup>. Ceci vous permet

d'exécuter le mappage en toute sécurité, même si vous ne fournissez pas explicitement une valeur de mappage au moment de l'exécution. Pour consulter un exemple, voir <u>Exemple : Utiliser des noms de fichier en tant que Paramètres de mappage</u><sup>482</sup>.

Les paramètres d'entrée ajoutés dans la zone de mappage principale ne doivent pas être confondus avec les paramètres d'entrée dans les <u>fonctions définies par l'utilisateur</u><sup>576</sup>. Il existe des similarités et des différences entre les deux comme suit.

Paramètres d'entrée sur le mappage	Paramètres d'entrée des fonctions définies par l'utilisateur
Ajouté depuis le menu Fonction   Insérer entrée.	Ajouté depuis le menu Fonction   Insérer entrée.
Peut avoir des types de données simples (string, entier, etc.).	Peut avoir des types de données simples et complexes.
Applicable à tout le mappage.	Applicable uniquement dans le contexte de la fonction dans laquelle ils ont été définis.

Lorsque vous avez créez un mappage inversé (utiliser la commande de menu **Outils | Créer le mappage inversé**), un composant d'entrée simple devient un composant de sortie simple.

# 5.1.1 Ajouter des composants d'entrée simple

#### Pour ajouter une entrée simple dans le mappage :

- 1. Veuillez vous assurer que la fenêtre de mappage affiche le mappage principal (pas une fonction définie par l'utilisateur).
- 2. Choisir une des options suivantes :
  - Dans le menu Fonction, cliquer sur Insérer entrée.
  - Dans le menu Insertion, cliquer sur Insérer entrée.
  - Cliquer sur la touche de la barre d'outils Insérer entrée \*1.

🙁 Créer	Entrée X
Nom :	inputFileName
Type de données:	string ~
	Entrée requise
Exécutio	n Design-time
Spéc	ifier la valeur
Valeur :	
	OK Annuler

- Saisir un nom et choisir le type de données requis pour cette entrée. Si l'entrée doit être traitée comme un paramètre de mappage obligatoire, cocher la case Entrée requise. Pour une liste complète de paramètres, voir <u>Paramètres de composant d'entrée simple</u><sup>479</sup>.
- **Note :** Le nom du paramètre peut contenir uniquement des lettres, des chiffres et des tirets bas ; aucun autre caractère n'est autorisé. Cela permet à un mappage de fonctionner sur tous les langages de génération de code.
  - 4. Cliquer sur **OK**.

Vous pouvez changer ultérieurement tous les paramètres définis ici (voir <u>Paramètres de composant d'entrée</u> simple<sup>479</sup>).

## 5.1.2 Paramètres de composant d'entrée simple

Vous pouvez définir les paramètres applicables à un composant d'entrée simple en l'ajoutant à la zone de mappage. Vous pouvez aussi changer les paramètres ultérieurement, depuis le dialogue Éditer entrée.

#### Pour ouvrir le dialogue Éditer entrée, choisir une des options suivantes :

- Sélectionner le composant et, dans le menu Composant, cliquer sur Propriétés.
- Double-cliquer sur le composant.
- Cliquer avec la touche de droite sur le composant puis cliquer sur Propriétés.

😢 Créer I	Entrée X
Nom :	input
Type de données:	string ~
	Entrée requise
Exécutio	n Design-time
Spéc	fier la valeur
Valeur :	
	OK Annuler

Dialogue Éditer entrée

#### Pour ouvrir le dialogue Éditer entrée, suivre une des étapes suivantes :

- Choisir le composant et, dans le menu Composant, cliquer sur Propriétés.
- Double-cliquer sur le composant.
- Cliquer avec la touche de droite sur le composant, et puis cliquer sur Propriétés.

Les paramètres disponibles sont les suivants.

Nom	Saisir un nom descriptif pour le paramètre d'entrée correspondant à ce composant. Au moment du temps d'exécution de mappage, la valeur saisie dans cette fenêtre de texte devient le nom du paramètre fourni au mappage ; ainsi aucune espace ni aucun caractère spécial n'est autorisé.
Type de données	Par défaut, tout paramètre d'entrée est traité en tant que type de données de string. Si le paramètre doit avoir un type de données différent, choisir la valeur respective depuis la liste. Lorsque le mappage est exécuté, MapForce envoie le paramètre d'entrée vers le type de donnée sélectionné ici.
Entrée est requise	Si activé, le paramètre rend le paramètre d'entrée obligatoire (c'est à dire, le mappage ne peut pas être exécuté à moins que vous fournissiez une valeur de paramètre). Désactiver cette case à cocher si vous spécifiez une valeur par défaut pour le paramètre d'entrée (voir <u>Créer une valeur d'entrée par défaut</u> ).
Spécifier valeur	Ce paramètre est applicable uniquement si vous exécutez le mappage pendant la période de conception du design, en cliquant sur l'onglet <b>Aperçu</b> . Il vous permet de saisir directement dans le composant la valeur à utiliser en tant que l'entrée de mappage.

Valeur	aramètre est uniquement applicable si vous exécutez le mappage pendant la ode de conception du design, en cliquant sur sur l'onglet <b>Aperçu</b> . Pour saisir une ur à utiliser par MapForce en tant qu'entrée de mappage, sélectionner la case à ner Spécifier valeur, puis saisir la valeur requise.			
	Note : Si vous cochez la case <b>Spécifier valeur</b> et saisissez une valeur dans la case adjacente, la valeur saisie prend précédence sur la valeur par défaut lorsque vous consultez le mappage (c'est à dire au moment de l'exécution au moment du design). Néanmoins, la valeur au moment du design n'a pas d'effet dans le XSLT, XQuery, ou de code de programme généré, dans exécution par MapForce Server, ou le déploiement sur FlowForce Server.			

# 5.1.3 Créer une valeur d'entrée par défaut

Une fois avoir ajouté un composant d'entrée dans la zone de mappage, veuillez noter l'item de **défaut** situé à gauche du composant.



#### Composant d'entrée simple

L'item par défaut vous permet de vous connecter à une valeur par défaut optionnelle à ce composant d'entrée, comme suit :

1. Ajouter un composant de constante (dans le menu **Insérer**, cliquer sur **Constant**), puis le connecter dans l'item **default** du composant d'entrée.



2. Double-cliquez le composant d'entrée et désactivez la case à cocher **Entrée requise**. Lorsque vous créez une valeur d'entrée par défaut, ce paramètre n'est pas significatif et entraîne des avertissements de validation de mappage.

😤 Créer Entrée 🛛 🗙				
Nom :	input			
Type de données:	string	~		
	Entrée requise			
Exécution	n Design-time			
Spéci	fier la valeur			
Valeur :				
	OK	Annuler		

- 3. Cliquer sur OK.
- **Note :** Si vous cochez la case **Spécifier valeur** et saisissez une valeur dans la case adjacente, la valeur saisie prend précédence sur la valeur par défaut lorsque vous consultez le mappage (c'est à dire au moment de l'exécution au moment du design). Néanmoins, la valeur au moment du design n'a pas d'effet dans le XSLT, XQuery, ou de code de programme généré, dans exécution par MapForce Server, ou le déploiement sur FlowForce Server.

## 5.1.4 Exemple : Utiliser les noms de fichier en tant que paramètres de mappage

Cet exemple vous guide dans les étapes nécessaires pour exécuter un mappage qui prend les paramètres d'entrée lors du runtime. Le fichier de conception de mappage utilisé dans cet exemple est disponible dans le chemin suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\FileNamesAsParameters.mfd.

Ce mappage lit des données depuis un fichier XML de source et l'écrit dans un fichier XML cible. Les données sont écrites dans le fichier cible de manière presque inchangée ; seuls les attributs **PrimaryKey** sont **Name** remplis avec des valeurs constantes depuis le mappage. L'objectif principal du mappage est de permettre à l'appelant de spécifier le nom du fichier d'entrée, en tant que paramètres de mappage, au moment du runtime du mappage.

Pour ce faire, le mappage a deux composants d'entrée : **InputFileName** et **OutputFileName**. Ceux-ci apportent le nom d'entrée du fichier (et le nom du fichier de sortie respectivement) du fichier XML de source et de cible. Pour cette raison, ils sont connectés à l'item **Fichier : <dynamic>**. Vous pouvez passer un composant dans ce mode en cliquant sur la touche **File/String** (<u>File/String</u>), et en sélectionnant **Utiliser Noms de fichier dynamique fournis par le mappage**.



FileNamesAsParameters.mfd (MapForce Enterprise Edition)

Si vous double-cliquez sur la barre de titre du composant **InputFileName** ou **OutputFileName**, vous pouvez consulter ou éditer leurs propriétés. Par exemple, vous pouvez spécifier le type de données du paramètres d'entrée ou changer le nom de paramètre d'entrée, comme décrit dans <u>Paramètres de composant d'entrée</u> <u>simple</u><sup>479</sup>. Dans cet exemple, les paramètres d'entrée et de sortie sont configurés comme suit :

- Le paramètre **InputFileName** est de type "string" et il a une valeur par défaut fourni par une constante définie dans le même mappage. La constante est de type "string" et sa valeur est "Altova\_Hierarchical.xml". Ainsi, lorsque ce mappage est exécuté, il tentera de lire des données depuis un fichier nommé "Altova\_Hierarchical.xml", en partant du principe que vous ne fournissez pas d'autres valeurs en tant que paramètre.
- Le paramètre OutputFileName est de type "string" et il a aussi une valeur par défaut fournie par une constante définie dans le même mappage. La constante est de type "string" et sa valeur est "Altova\_Hierarchical\_output.xml". C'est pourquoi le mappage créera un fichier de sortie XML appelé "Altova\_Hierarchical\_output.xml" lorsqu'il est exécuté, en partant du principe que vous ne fournissez pas d'autres valeurs en tant que paramètres.

Les sections suivantes illustrent comment exécuter le mappage et fournir des paramètres dans les langages de transformation suivants :

- <u>XSLT 2.0</u><sup>484</sup>, utiliser RaptorXML Server
- Built-in (fichier d'exécution MapForce Server)
   484, avec MapForce Server
- <u>Java</u> 485
- <u>C#</u>486
- <u>C++</u>486

## XSLT 2.0

Si vous générez du code dans XSLT 1.0, XSLT 2.0 ou XSLT 3.0, un fichier batch **DoTransform.bat** est généré dans le répertoire de cible choisi, en plus du fichier XSLT. Le fichier **DoTransform.bat** vous permet d'exécuter le mappage avec RaptorXML Server, voir <u>Automatisation avec RaptorXML Server</u><sup>(949)</sup>.

Pour utiliser un fichier d'entrée (ou de sortie) différent, éditer le fichier **DoTransform.bat** pour inclure les paramètres requis, comme suit :

- 1. Tout d'abord, générer le code XSLT. Par exemple, pour générer XSLT 2.0, sélectionner la commande de menu Fichier | Générer du Code dans | XSLT 2.0.
- Copier le fichier Altova\_Hierarchical.xml depuis
   <Documents>\Altova\MapForce2023\MapForceExamples\ dans le répertoire où vous avez généré
   le code XSLT 2.0 (dans cet exemple, c:\codegen\examples\xslt2). Comme indiqué précédemment,
   le mappage tentera de lire ce fichier si vous ne fournissez pas de valeur personnalisée dans le
   paramètre InputFileName.
- Éditer DoTransform.bat pour inclure le paramètre d'entrée personnalisé soit avant soit après \*\*. Veuillez noter que la valeur de paramètre est contenue entre des guillemets simples. Les paramètres d'entrée disponibles sont recensés dans la section rem (Remarque). Supposons que vous souhaitez générer un fichier de sortie appelé output.xml. Pour ce faire, changer le fichier DoTransform.bat comme suit :

```
@echo off
RaptorXML xslt --xslt-version=2
    --input="MappingMapToAltova_Hierarchical.xslt"
    --param=OutputFileName:'output.xml' %* "MappingMapToAltova_Hierarchical.xslt"
rem --param=InputFileName:
rem --param=OutputFileName:
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
```

Lorsque vous exécutez le fichier DoTransform.bat, RaptorXML Server termine la transformation en utilisant **Altova\_Hierarchical.xml** en tant que paramètres d'entrée. Si vous avez suivi les étapes ci-dessus, le nom du fichier de sortie généré sera **output.xml**.

Command Prompt	_		×
c:\codegen\examples\xslt2>DoTransform.bat			^
file:///c:/codegen/examples/xslt2/MappingMapToAltova_Hierarchical.xslt: result="OK" xslt.main_output_files="" xslt.additional_output_files="file examples/xslt2/output.xml"	runtim :///c:	e="31ms /codege	s" en/
c:\codegen\examples\xslt2>			~

### Fichier d'exécution MapForce Server

Pour fournir des paramètres d'entrée personnalisés vers un fichier d'exécution MapForce Server :

1. Si vous ne l'avez pas encore fait, ouvrir l'exemple FileNamesAsParameters.mfd depuis le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\.

- Dans le menu Fichier, cliquer sur Compiler sur le fichier d'exécution MapForce, voir aussi Compiler des Mappages dans les fichiers d'exécution MapForce Server<sup>957</sup>. Sur invitation, enregistrer le fichier d'exécution .mfx dans un répertoire sur votre ordinateur (dans cet exemple, c: \codegen\examples\mfx ).
- Copier le fichier Altova\_Hierarchical.xml depuis le répertoire
   <Documents>\Altova\MapForce2023\MapForceExamples\ vers le répertoire dans lequel vous avez enregistré le fichier .mfx.
- 4. Exécuter MapForce Server avec la commande suivante :

```
MapForceServer.exe run "C:\codegen\examples\mfx\FileNamesAsParameters.mfx" -
p=InputFileName:"C:\codegen\examples\mfx\Altova_Hierarchical.xml" -
p=OutputFileName:"C:\codegen\examples\mfx\OutputFile.xml"
```

Dans la commande MapForce Server ci-dessus, -p=InputFileName et -p=OutputFileName sont les paramètres d'entrée dans le mappage. Vous pouvez utiliser tout nom de fichier en tant que la valeur de - **OutputFileName**. Néanmoins le nom de fichier fourni dans le paramètre dans -**InputFileName** doit exister en tant que fichier physique ; sinon le mappage échouera.

Note : Si vous voyez le message "MapForceServer.exe n'est pas reconnu en tant que commande interne ou externe, programme opérable, ou le fichier batch", changer le répertoire actuel sur celui dans lequel l'exécutable de MapForce Server est installé. Pour éviter de changer de chemin à chaque fois que vous exécutez un mappage, ajouter à la variable d'environnement PATH du système d'exploitation le chemin du répertoire dans lequel l'exécutable MapForce Server est installé (par exemple, C:\Program Files (x86)\Altova\MapForceServer2023\bin).

Avec MapForce Server, l'exécution d'un mappage est aussi possible en appelant l'API de MapForce Server (qui est invocable depuis des langages comme C++, C#, ou Java). Pour plus d'informations concernant ce scénario, se référer à la documentation MapForce Server (<u>https://www.altova.com/fr/documentation</u>).

#### Java

Pour fournir un paramètre d'entrée personnalisé dans une application Java .jar :

- 1. Si vous ne l'avez pas encore fait, ouvrir l'exemple FileNamesAsParameters.mfd depuis le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\.
- 2. Dans le menu Fichier, cliquer sur Générer du Code dans | Java.
- Compiler code Java dans un fichier JAR exécutable. Pour un exemple de la manière de procéder dans Eclipse, voir <u>Exemple : Générer et exécuter du code Java</u><sup>1097</sup>.
- Copier le fichier Altova\_Hierarchical.xml depuis
   <-Documents>\Altova\MapForce2023\MapForceExamples\ vers le répertoire dans lequel le fichier
   .jar se trouve. Comme indiqué précédemment, le mappage tentera de lire ce fichier si vous ne fournissez pas de valeur personnalisée dans le paramètre InputFileName.
- 5. Exécuter l'application Java avec la commande suivante :

java -jar Mapping.jar /OutputFileName "output.xml"

Dans la commande ci-dessus, le paramètre d'entrée /OutputFileName fournit le nom du fichier de sortie à générer.

**Note :** Si vous utilisez des caractères génériques lors de la transmission de paramètres vers des fichiers .jar, il faut contenir les paramètres de caractères génériques entre guillemets, par exemple :

java -jar Mapping.jar /InputFileName "altova-\*.xml"

## C#

Pour fournir un paramètre d'entrée personnalisé dans une application de ligne de commande C# générée par MapForce :

- 1. Si vous ne l'avez pas encore fait, ouvrir l'exemple FileNamesAsParameters.mfd depuis le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\.
- 2. Dans le menu Fichier, cliquer sur Générer du Code dans | C#, et sélectionner un répertoire cible (C: \codegen\examples\cs, dans cet exemple).
- 3. Ouvrir la solution dans Visual Studio et la construire (Ctrl + Shift + B).
- 4. Copier le fichier Altova\_Hierarchical.xml depuis <Documents>\Altova\MapForce2023\MapForceExamples\ dans le répertoire où Mapping.exe a été généré (dans cet exemple, C:\codegen\examples\cs\Mapping\bin\Debug). Comme indiqué précédemment, le mappage tentera de lire ce fichier si vous ne fournissez pas de valeur personnalisée dans le paramètre InputFileName.
- 5. Ouvrir une fenêtre d'Invite de commande et passer au répertoire dans lequel se trouve **Mapping.exe**.

cd C:\codegen\examples\cs\Mapping\bin\Debug

6. Exécuter l'application avec la commande suivante :

Mapping.exe /OutputFileName output.xml

Dans la commande ci-dessus, le paramètre d'entrée /OutputFileName fournit le nom du fichier de sortie à générer.

#### C++

Pour fournir un paramètre d'entrée personnalisé dans une application de ligne de commande C++ générée par MapForce :

- 1. Si vous ne l'avez pas encore fait, ouvrir l'exemple FileNamesAsParameters.mfd depuis le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\.
- Dans le menu Fichier, cliquer sur Générer du Code dans | C++, et sélectionner un répertoire cible (C:\codegen\examples\cpp, dans cet exemple).
- 2. Ouvrir la solution dans Visual Studio et la construire (Ctrl + Shift + B).
- Copier le fichier Altova\_Hierarchical.xml depuis
   <Documents>\Altova\MapForce2023\MapForceExamples\ dans le répertoire où Mapping.exe a
   été généré (dans cet exemple, C:\codegen\examples\cpp\Mapping\Debug). Comme indiqué
   précédemment, le mappage tentera de lire ce fichier si vous ne fournissez pas de valeur personnalisée
   dans le paramètre InputFileName.
- 4. Ouvrir une fenêtre d'Invite de commande et passer au répertoire dans lequel se trouve Mapping.exe.

cd C:\codegen\examples\cpp\Mapping\Debug

5. Exécuter l'application avec la commande suivante :

Mapping.exe /OutputFileName output.xml

Dans la commande ci-dessus, le paramètre d'entrée /OutputFileName fournit le nom du fichier de sortie à générer.

# 5.2 Sortie simple

Un composant de sortie (ou "sortie simple") est un composant MapForce qui vous permet de retourner une valeur string provenant du mappage. Les composants de sortie représentent uniquement un type possible type de <u>composants de cible</u> and <u>mappage</u>. Les composants de sortie représentent uniquement un type possible type de <u>sortie simple lorsque vous souhaitez retourner une valeur de string depuis le mappage</u>. Dans la zone de mappage, les composants de sortie simple jouent un rôle simple d'un composant cible qui a un type de données string à la place d'une structure d'items et de séquences. Par conséquent, vous pouvez créer un composant de sortie simple au lieu (ou en plus de ) d'un composant sur base de fichier. Par exemple, vous pouvez utiliser un composant de sortie simple pour tester et consulter rapidement la sortie d'une fonction (voir <u>Exemple : Consulter la sortie de fonction</u>). L'objectif principal d'un composant de sortie simple est, néanmoins, de retourner un string lors de l'appel de MapForce Server API, sans écrire de fichiers.

Les composants de sortie simple ne devraient pas être confondus avec les paramètres de sortie de fonctions définies par l'utilisateur (voir <u>Fonctions définies par l'utilisateur</u>). Il existe des similarités et des différences entre les deux comme suit.

Composants de sortie	Paramètres de sortie des fonctions définies par l'utilisateur	
Ajouté depuis le menu Fonction   Insérer sortie.	Ajouté depuis le menu Fonction   Insérer sortie.	
A "string" en tant que type de données.	Peut avoir des types de données simples et complexes.	
Applicable à tout le mappage.	Applicable uniquement dans le contexte de la fonction dans laquelle ils ont été définis.	

Le cas échéant, vous pouvez ajouter plusieurs composants de sortie simples à un mappage. Vous pouvez aussi utiliser des composants de sortie simples en combinaison avec des composants cibles basés sur fichier et de base de données. Si votre mappage contient plusieurs composants cibles, vous pouvez consulter les données retournées par un composant particulier en cliquant sur la touche **Aperçu** ( ) dans la barre de titre du composant, puis en cliquant sur l'onglet **Sortie** dans la fenêtre Mappage.

Vous pouvez utiliser les composants de sortie simples comme suit dans les langages de transformation MapForce :

Langage	Comment cela fonctionne ?
BUILT-IN (lors de la consultation de la transformation de mappage)	Vous pouvez consulter les composants de Sortie de la même manière que vous consulteriez une sortie de mappage basée sur fichier, en cliquant sur l'onglet <b>Sortie</b> dans la fenêtre Mappage.
BUILT-IN (lors de l'exécution du fichier d'exécution MapForce Server)	Lorsque vous exécutez un fichier d'exécution compilé de MapForce Server (voir <u>Compiler un mappage MapForce</u> ), la sortie de mappage est renvoyée dans le stream de sortie standard (stdout) pour que vous puissiez le voir ou le rediriger vers un fichier. Par exemple, en partant du principe que le nom du fichier d'exécution MapForce est <b>MyMapping.mfx</b> , utilisez la syntaxe suivante pour rediriger la sortie de mappage vers le fichier <b>output.txt</b> et toute erreur vers le fichier <b>log.txt</b> :

Langage	Comment cela fonctionne ?
	MapForceServer.exe run MyMapping.mfx >output.txt 2>log.txt
XSLT 1.0, XSLT 2.0, XSLT 3.0	Dans les fichiers XSLT générés, un composant de sortie simple défini dans le mappage devient la sortie de la transformation XSLT. Si vous utilisez RaptorXML Server, vous pouvez instruire RaptorXML Server pour écrire la sortie de mappage sur le fichier transmis en tant que valeur dans le paramètre
	Pour écrire la sortie dans un fichier, ajouter ou éditer dans le paramètre output dans le fichier <b>DoTransform.bat</b> . Par exemple, le fichier <b>DoTransform.bat</b> suivant a été édité pour écrire la sortie de mappage dans le fichier <b>Output.txt</b> (voir texte marqué).
	RaptorXML xsltxslt-version=2 input="MappingMapToResult1.xslt"output="Output.txt" %* "MappingMapToResult1.xslt"
	Si un paramètreoutput n'est pas défini, la sortie de mappage sera écrite dans le flux de sortie standard (stdout) lorsque le mappage est exécuté.
C++, C#, Java	Dans le code C++, C# et Java généré, la sortie de mappage est écrite dans la sortie standard de l'application générée.
	Si le mappage contient plusieurs composants cibles, l'application générée concatène la sortie standard de chaque composant cible et le retourne en tant qu'une sortie standard unifiée

Si vous créez un mappage inversé (en utilisant les commandes de menu **Outils | Créer mappage inverse**), le composant de sortie simple devient un composant d'entrée simple.

# 5.2.1 Ajouter des composants de sortie simples

#### Pour ajouter un composant de sortie vers la zone de mappage :

- 1. Veuillez vous assurer que la fenêtre de mappage affiche le mappage principal (n'est pas une fonction définie par l'utilisateur).
- 2. Choisir une des étapes suivantes :
  - a. Dans le menu Fonction, cliquer sur Insérer sortie.
  - b. Cliquer sur la touche de la barre d'outils Insérer sortie

- 3. Saisir un nom pour le composant.
- 4. Cliquer sur **OK**.

😢 Créer	Sortie	×
Nom:	result	
Type de données:	string	~
donnees.		
	ОК	Annuler

Dialogue Créer sortie

Vous pouvez changer le nom du composant à tout moment ultérieurement, d'une des manières suivantes :

- Sélectionner le composant et, dans le menu Composant, cliquer sur Propriétés.
- Double-cliquer sur l'en-tête de composant.
- Cliquer avec la touche de droite sur l'en-tête de composant, puis cliquer sur Propriétés.

## 5.2.2 Exemple : Consulter la sortie de fonction

Cet exemple illustre comment consulter la sortie retournée par les fonctions MapForce à l'aide de composants de sortie simples. Vous profiterez au maximum de cet exemple si vous possédez des connaissances de base des fonctions en général et des fonctions MapForce en particulier. Si vous ne connaissez pas les fonctions de MapForce, vous pouvez consulter <u>Utiliser les fonctions</u> avant de continuer.

Notre objectif est d'ajouter un nombre de fonctions dans la zone de mappage et d'apprendre comment consulter leur sortie à l'aide de composants de sortie simples. En particulier, l'exemple utilise quelques fonctions simples disponibles dans la bibliothèque centrale. Voici un sommaire de leur utilisation :

string-length <sup>712</sup>	Retourne le nombre de caractères contenus dans le string fourni en tant qu'argument. Par exemple, si vous transmettez à cette fonction la valeur "Lorem ipsum", le résultat est "11", étant donné qu'il s'agit du nombre de caractères contenus dans le texte "Lorem ipsum".
substring- after <sup>713</sup>	Retourne la partie du string qui se produit après le séparateur fourni en tant qu'argument. Par exemple, si vous passez dans cette fonction la valeur "Lorem ipsum" et le caractère d'espace (" "), le résultat sera "ipsum".
substring- before <sup>714</sup>	Retourne la partie du string qui se produit avant le séparateur fourni en tant qu'argument. Par exemple, si vous passez dans cette fonction la valeur "Lorem ipsum" et le caractère d'espace (" "), le résultat sera "Lorem".

Pour tester chacune de ces fonctions par rapport à une valeur de texte personnalisée ("Lorem ipsum", dans cet exemple), suivre les étapes ci-dessous :

- Ajoutez une constante avec la valeur "Lorem ipsum" à la zone de mappage (utilisez la commande de menu Insérer | Constante). La constante sera le paramètre d'entrée pour chacune des fonctions à tester.
- 2. Ajoutez les fonctions string-length, substring-after et substring-before à la zone de mappage, en les glissant vers la zone de mappage depuis la bibliothèque core, la section string functions.
- 3. Ajouter une constante avec une espace vide (" ") en tant que valeur. Cela sera le paramètre de séparateur requis par les fonctions substring-after et substring-before.
- 4. Ajoutez trois composants de sortie simple (utiliser la commande de menu **Fonction | Insérer sortie**). Dans cet exemple, ils ont été nommés *Result1*, *Result2* et *Result3*, bien que vous puissiez leur donner un autre titre.



5. Connecter les composants comme indiqué ci-dessous.

Tester la sortie de fonction avec des composants de sortie simples

Comme indiqué dans l'échantillon ci-dessus, le string "Lorem ipsum" agit en tant que paramètre d'entrée pour chaque fonction string-length, substring-after et substring-before. De plus, les fonctions substring-after et substring-before prennent une valeur d'espace dans leur deuxième paramètre d'entrée. Les composants **Result1**, **Result2** et **Result3** peuvent être utilisés pour consulter le résultat de chaque fonction.

#### Pour consulter la sortie de toute fonction :

Cliquez sur la touche Aperçu ( ) dans la barre de titre du composant, puis cliquez sur l'onglet
 Sortie dans la fenêtre de mappage.

# 5.3 Les variables

Une variable est un type de composant spécial utilisé pour stocker un résultat de mappage intermédiaire pour un traitement ultérieur. Les variables peuvent être de type simple (par exemple, string, entier, booléennes, etc) ou de type complexe (une structure arborescente). Voir les exemples des deux types dans les sous-rubriques ci-dessous.

Un des aspects les plus importantes concernant les variables est qu'elles sont des séquences, et qu'elles peuvent être utilisées pour créer des séquences. Le terme *séquence* signifie une liste d'items zéro ou plus. Cela permet à une variable de traiter plusieurs items pour la durée du cycle de vie du mappage. Pour plus d'information, voir aussi <u>Règles et stratégies de mappage</u>. Toutefois, il est également possible d'assigner la valeur à une variable une fois et préserver cette valeur pour le reste du mappage. Pour des détails, voir <u>Changer</u> le contexte et l'étendue des variables.

## Variables simples

Une simple variable est créée pour représenter des types atomiques tels que les strings, nombres booléennes (voir la capture d'écran ci-dessous).

VAB	Ī
Compute-when	1
[≥ <b>₩</b> ] value	þ

## Variables complexes

Une variable complexe a une structure d'arborescence. Les structures sur lesquelles une variable complexe peut être basée sont résumées dans la liste ci-dessous.

<u>Édition de base de MapForce :</u>

Structures de schéma XML

<u>Édition professionnelle de MapForce :</u>

- Structure de schéma XML
- Structure de base de données

<u>Édition MapForce Enterprise :</u>

- Structure de schéma XML
- Structure de base de données
- Structure EDI
- Structure FlexText
- Structure de schéma JSON

#### Exemple 1 : Variable basée sur Schéma XML

Vous pouvez créer une variable de type complexe en fournissant un schéma XML qui définit la structure de la variable (*voir la capture d'écran ci-dessous*). Si le schéma définit des éléments globalement, vous pouvez choisir lequel doit devenir le nœud de racine de la structure variable. Veuillez noter qu'une variable n'a pas de fichier XML d'instance associé. Les données de la variable sont calculées durant l'exécution du mappage.



#### Exemple 2 : Variable basée sur la base de données (éditions MapForce Professional et Enterprise)

Si vous choisissez une structure de base de données pour votre variable(*voir la capture d'écran ci-dessous*), vous pouvez choisir une table de base de données spécifique en tant qu'item de racine pour la structure de variable. MapForce vous permet de créer des variable basés sur BD avec une arborescence de tables associées. L'arborescence de tables associées représente une structure in-memory qui n'a pas de connexion vers la base de données lors de l'exécution. Ceci signifie également qu'il n'y a pas de gestion automatique de la clé étrangère et pas d'actions de table dans les paramètres ou variables.

V.	AB)	
5	compute-when	
ĻΕ	Applications	Ŗ
φ	📲 AppID	¢
Þ	AppName	Ŕ
¢	Description	ķ
P	Category	¢
¢	URL	P
L		

#### Compute-when

Dans les deux exemples ci-dessus, chaque variable est dotée d'un élément dénommé compute-when. Connecter cet élément est optionnel : Ceci vous permet de contrôler comment la valeur de variable doit être calculée dans le mappage. Pour plus d'informations, voir <u>Changer le contexte et l'étendue des variables</u><sup>493</sup>.

## Variables avec des entrées dupliquées

Si nécessaire, les items d'une structure de variable peuvent être dupliqués pour accepter les données de plus d'une connexion source. Ceci est semblable aux <u>entrées dupliquées</u> dans les composants standard. Néanmoins, cela ne s'applique pas aux variables créées depuis les tables de base de données. La capture d'écran ci-dessous illustre une variable simple avec des entrées dupliquées.

VAR	
Compute-when	_
> <b>₩</b> i value	ķ
>₩İ value (2)	
> <u>₩</u> value (3)	┛

### Mappages en chaîne par rapport aux variables

Les variables peuvent être comparées aux composants intermédiaires d'un <u>mappage en chaîne</u><sup>841</sup>. Néanmoins, les variables sont plus flexibles et pratiques si vous ne devez pas produire de fichiers intermédiaires à chaque étape du mappage. La table suivante souligne les différences entre les variables et les mappages en chaîne.

Mappages enchaînés	Variables
Les mappages en chaîne impliquent deux étapes indépendantes. Par exemple, un mappage a trois composants, notamment A, B et C. Étape 1 : mapper des données de A à B. Étape 2 : mapper des données de B à C.	Vous pouvez contrôler quand et combien de fois la valeur variable est calculée quand le mappage est exécuté. Pour les détails, voir <u>Changer le contexte et</u> <u>l'étendue des variables</u> <sup>498</sup> .
Lorsque le mappage est exécuté, les résultats intermédiaires seront stockés extérieurement dans des fichiers.	Lorsque le mappage est exécuté, des résultats intermédiaires sont stockés intérieurement. Aucun fichier externe contenant des résultats d'une variable n'est produit.
Le résultat intermédiaire peut être consulté en utilisant la touche d'aperçu.	Un résultat de variable ne peut pas être consulté, puisqu'il est calculé lors de l'exécution du mappage.

Note : les variables ne sont pas prises en charge si le langage de transformation de mappage est réglé sur XSLT 1.0.

# 5.3.1 Ajouter des variables

Cette rubrique explique comment ajouter une variable au mappage. La première option est d'ajouter une variable par la commande de menu ou de la barre d'outils. La deuxième option vous permet d'ajouter une variable par le menu contextuel.

## Option 1 : par la commande de menu ou de la barre d'outils

Cette option vous permet d'ajouter une variable par la commande de menu ou une barre d'outils. Suivez les étapes ci-dessous :

1. Allez au menu **Insérer** and cliquez sur **Variable**. En alternative, cliquez sur **bouton** de la barre d'outils (**Variable**).

😢 Créer Variable				×
Type O Type simple type (e	ntier, chaîne, etc.)			
Type de données :	string	~	•	
Type complexe (str	ucture arborescent	e)		
Structure :			Choisir	Éditer
Racine :			Choisir	
	Enregistrer le chemi	in d'accès de structure	relatif au fi	chier MFD
			OK	Annuler

2. Sélectionnez le type de variable que vous souhaitez insérer (type simple ou complexe).

Si vous sélectionnez type complexe, vous devez suivre quelques étapes supplémentaires :

Cliquez sur Choisir pour sélectionner la source qui devrait fournir la <u>structure de la variable</u>
 Les structures illustrées dans la capture d'écran s'appliquent uniquement à MapForce Enterprise Edition. Voir la liste de structures pertinentes à d'autres éditions de MapForce dans la <u>rubrique précédente</u>

.

😤 Create Variable		×
Choose an existing structure or a new structure type	e to insert:	
○ Insert an existing structure as a parameter:		
Filename / Connection	Path / Description	
1		
Insert a new structure of the one of the following	g types:	
🚭 XML Schema Structure		
Database Structure		
EDI Structure		
FlexText Structure		
JSON Schema Structure		_
1		
	OK Cancel	

4. Lorsque vous serez invité à le faire, spécifiez un item de racine de la structure de variable. Par exemple, dans les schémas XML, vous pouvez sélectionner tout élément ou type de la source sélectionnée (*voir la capture d'écran ci-dessous*).

😢 Choose Root Item		×
Please choose a root item for the parameter:		
<ul> <li></li></ul>		^
<		>
, Show <u>a</u> nnotations Show <u>types</u>	ОК	Cancel

## Option 2 : par le menu contextuel

La deuxième option vous permet de créer une variable utilisant le menu contextuel. Les options possibles sont recensées ci-dessous.

#### Variable depuis un nœud source

Pour créer une variable depuis un nœud source, cliquez de la touche droite sur le connecteur de sortie d'un composant (dans cet exemple, le connecteur de sortie de l'élément <Article>) et sélectionnez **Créer une** variable depuis le nœud source (voir la capture d'écran ci-dessous).



Ceci crée une variable complexe avec le schéma source du composant Articles. Tous les éléments sont automatiquement connectés avec une <u>connexion copy-all</u><sup>(95)</sup> (*voir la capture d'écran ci-dessous*).



#### Variable depuis un nœud cible

Pour créer une variable depuis un nœud cible, cliquez de la touche droite sur le connecteur d'entrée d'un composant cible et sélectionnez **Créer une variable depuis un nœud cible**. Ceci crée une variable complexe avec le même schéma que pour la cible. Tous les items sont automatiquement connectés avec une connexion copy-all.

#### Variable depuis un filtre :

Pour créer une variable utilisant un filtre, cliquez de la touche droite sur le connecteur de sortie d'un composant de filtre (on-true/on-false) et sélectionnez **Créer une variable depuis le nœud source**. Cela crée un composant complexe avec le schéma de source, et utilise automatiquement l'item lié à l'entrée de filtre en tant que l'élément racine du composant intermédiaire.

## 5.3.2 Changer le contexte et l'étendue des variables

Chaque variable a un élément d'entrée compute-when (voir la capture d'écran ci-dessous), qui vous permet de contrôler l'étendue de la variable. Ceci signifie que vous pouvez contrôler quand et combien de fois la valeur de variable est calculée quand le mappage est exécuté. Vous ne devez pas connecter cette entrée dans de nombreux cas, mais il peut ête essentiel d'écraser le contexte par défaut ou d'optimiser la performance de mappage.

VAB	I
Compute-when	1
≥ <mark>⊭ivalue</mark>	Ь

Les termes suivants sont pertinents pour la discussion sur l'étendue et le contexte des variables : *subtree* et *variable value*. Une sous-arborescence est un ensemble d'items/nœuds dans un composant cible et tous ses descendants : par exemple, un élément <Person> avec ses éléments enfant <FirstName> et <LastName>.

Une valeur variable signifie les données qui sont disponibles du côté sortie du composant variable.

- Pour des variables simples, il s'agit d'une séquence de valeurs atomiques dont le type de données est spécifié dans les propriétés de composant.
- Pour les variables complexes, il s'agit d'une séquence de nœuds de racine (du type spécifié dans les propriétés de composant), chacune incluant tous ses nœuds descendants.

La séquence des valeurs atomiques (ou nœuds) peut contenir un ou plusieurs éléments zéro. Cela dépend de ce qui est connecté du côté entrée de la variable, et des items de parents dans les composants source et cible.

### « Compute-when » n'est pas connecté (défaut)

Si l'item d'entrée compute-when n'est pas connecté vers un nœud de sortie d'un composant de source, la valeur de variable est calculée à chaque fois qu'il est utilisé la première fois dans une sous-arborescence de cible soit directement par le biais d'un connecteur depuis le composant de variable vers un nœud dans le composant de cible, ou indirectement par le biais des fonctions. La même valeur de variable est aussi utilisée pour tous les nœuds enfants de cible dans la sous-arborescence.

La valeur de variable réelle dépend des connexions entre les items de parent des composants de source et de cible. Ce comportement par défaut est le même que celui des sorties complexes des <u>fonctions régulières</u> <u>définies par l'utilisateur</u> et les appels de fonction de service Web. Si la sortie de variable est connectée à plusieurs nœuds de cible non-liés, la valeur de variable est calculée *séparément pour chacun d'entre eux*. Cela peut produire des résultats différents dans chaque cas, parce que plusieurs connexions de parent influent sur le contexte dans lequel la valeur de la variable est évaluée.

## « Compute-when » est connecté

En connectant un connecteur de sortie d'un composant source vers compute-when, la variable est calculée à chaque fois que cet item source est utilisé la première fois dans une sous-arborescence cible.

La variable agit comme si elle était un item enfant de l'item connecté à compute-when. Cela permet de lier la variable à un item de source spécifique. Concrètement, lors du runtime, la variable est réévaluée lorsqu'un nouvel item est lu depuis la séquence dans le composant de source. Cela est lié aux règles générales qui gouvernent les connexions dans MapForce : Pour chaque item source, un item cible est créé. Dans ce cas, compute-when donne l'instruction à MapForce de calculer la valeur variable pour chaque item source. Pour plus d'information, voir Règles et stratégies de mappage<sup>372</sup>.

#### « Compute-once »

Si nécessaire, vous pouvez choisir de calculer la valeur de variable *une fois avant chaque composants de cible*, rendant la variable essentiellement une constante globale pour le reste du mappage. Pour ce faire, cliquer avec la touche de droite sur l'item compute-when et sélectionner **Calculer une fois** depuis le menu contextuel :

VAR	-	
<b>⊳</b> ₩i value	Calculer une <u>f</u> ois	
	Supprimer les co <u>n</u> nexions Créer composant varia <u>b</u> le Fonctions n <u>œ</u> ud et défauts	* * *
	Commente	

Lorsque vous changez l'étendue d'une variable vers compute-when=once, le connecteur d'entrée est supprimé de l'item compute-when, puisqu'une telle variable est uniquement évaluée une fois. Dans une fonction définie

par l'utilisateur, la variable compute-when=once est évaluée à chaque fois que la fonction est appelée avant que le résultat de fonction réel soit évalué.

#### Parent-context

Ajouter un item parent-context peut être nécessaire, par exemple si votre mappage utilise plusieurs filtres et que vous nécessitez un nœud de parent supplémentaire pour les itérations. Pour plus de détails, voir <u>Exemple</u> <u>: Changer le contexte Parent</u> . Pour ajouter un parent-context, cliquez avec la touche de droite sur le nœud racine (dans cet exemple, PersonList) et sélectionnez **Ajouter parent-context** depuis le menu contextuel. Cela ajoute un nouveau nœud, parent-context, à la hiérarchie existante.



Le contexte parent ajoute un nœud parent virtuel à la hiérarchie dans le composant. Cela vous permet d'itérer sur un nœud supplémentaire dans le même composant de source ou dans un composant de source différent.

## 5.3.3 Exemple : Compter les lignes de table de base de données

Le mappage illustré dans cet exemple est disponible en tant que **DB\_UserList.mfd** dans le dossier **>Documents>\Altova\MapForce2023\MapForceExamples\**. Ce mappage extrait des enregistrements d'utilisateur depuis une table de base de données appelée "Users" et les écrit dans un fichier XML. La colonne de base de données "Username" contient aussi bien le prénom que le nom d'une personne (par exemple, "Vernon Callaby"). Ce mappage présente les objectifs suivants :

- 1. Pour chaque enregistrement dans la table "Users", créer un nouvel élément Person dans le fichier XML.
- 2. Partager la valeur extraite depuis le champ de base de données "Username" en deux champs séparés dans le fichier XML ("First" et "Last").
- Pour chaque enregistrement, trouver le numéro séquentiel comparé au nombre des enregistrements totaux présents dans la base de données (par exemple, "Record 1 of 4") et écrire cette information dans l'élément Details.



#### DB\_UserList.mfd

Comme illustré ci-dessus, afin d'atteindre le premier objectif, une connexion est établie entre la table de source "Users" et l'élément Person du fichier XML cible. Cela permet d'assurer que, pour chaque enregistrement dans la table de source, un nouvel élément Person sera créé dans la cible.

La valeur du champ "Username" est fournie dans les fonctions <u>substring-before</u><sup>(14)</sup> et <u>substring-after</u><sup>(13)</sup>. Ces deux fonctions extraient le texte avant et après le caractère d'espace (" "), respectivement, ce qui permet de finaliser le deuxième objectif de mappage.

Enfin, pour atteindre le troisième objectif, le mappage utilise la fonction count. Le résultat de la fonction count est passé dans la variable. La variable permet d'assurer que ce résultat est stocké dans le mappage et est disponible lors de l'écriture de l'élément "Details" de chaque personne dans le XML cible. Veuillez noter que, pour des raisons d'efficacité, les enregistrements de base de données ne doivent être comptés qu'une seule fois, l'étendue de la variable est donc définie à compute-when=once (voir <u>Changer le contexte et l'étendue des</u> variables<sup>(43)</sup>)

# 5.3.4 Exemple : Filtre et numéroter les nœuds

Le mappage illustré dans cet exemple est disponible sous **PositionInFilteredSequence.mfd** dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\**.

Ce mappage lit un fichier XML qui contient les coordonnées de plusieurs personnes, les filtre et les écrit dans un fichier XML cible. L'objectif du mappage est de filtrer depuis le fichier XML de source uniquement les personnes dont le nom de famille commence avec la lettre "M" ou une des lettres suivantes de l'alphabet. Ensuite, les contacts extraits doivent être numérotés. Le nombre va servir d'identifiant unique de chaque contact dans le fichier XML cible.



PositionInFilteredSequence.mfd

Pour atteindre l'objectif ci-dessus, les types de composant suivants ont été ajoutés au mappage :

- Un filtre (voir <u>Filtres et conditions</u><sup>623</sup>)
- Une variable complexe (voir <u>Ajouter des variables</u><sup>494</sup>)
- Les fonctions greater 665 et position 700 (voir <u>Ajouter une fonction au mappage</u> 552)
- Une constante (Pour ajouter une constante, sélectionner la commande de menu Insérer | Constante).

La variable utilise le même schéma que le composant de source. Si vous cliquez de la touche droite sur la variable et sélectionnez **Propriétés** depuis le menu contextuel, notez que le nœud **BranchOffices/Office/Contact** est sélectionné comme nœud racine pour cette structure de variable.

Tout d'abord, les données du composant de source sont passées dans le filtre. Le filtre passe ensuite à la variable les enregistrements qui remplissent la condition de filtre. Concrètement, le filtre est configuré ou obtenir uniquement les nœuds Contact où le prénom est égal ou plus grand que "M". Pour ce faire, la fonction greater compare chaque item last avec la valeur de constante "M".

La variable a l'entrée compute-when connectée à l'item de racine du composant de source (BranchOffices). Lors du runtime, cela entraîne une réévaluation de la variable à chaque fois qu'un nouvel item est lu depuis la séquence dans le composant de source. Dans ce mappage, néanmoins, la connexion ou la non-connexion de l'item compute-when ne fait aucune différence. La raison est que la variable est connectée à l'item de source contact (indirectement par le filtre), et il calculera autant de fois qu'il y a d'instances de Contact qui se conforment à la condition de filtre.

Les fonctions <u>position</u><sup>(20)</sup> retournent, pour chaque itération de la variable, le nombre de séquences actuelles. Seuls huit contacts remplissent la condition de filtre ; c'est pourquoi, si vous consultez le mappage et regardez la sortie, il est recommandé de noter comment les ID 1 à 8 ont été écrites dans l'élément ID du composant de cible.

La variable a été nécessaire, à cause de l'exigence de numéroter tous les enregistrements. Si nous avions connecté le résultat de filtre directement au composant de cible, il n'y aurait pas moyen de numéroter chaque occurrence de Contact. L'objectif de la variable dans ce mappage est donc de stocker chaque instance de Contact temporairement dans le mappage, elle peut donc être numérotée avant d'être écrite dans la cible.

# 5.3.5 Exemple : Grouper et sous-grouper des enregistrements

Le mappage illustré dans cet exemple est disponible sous **DividePersonsByDepartmentIntoGroups.mfd** dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\**.

Ce mappage traite un fichier XML qui contient des enregistrements d'employés d'une compagnie fictionnelle. L'entreprise a deux bureaux : "Nanonull, Inc." et "Nanonull Partners, Inc". Chaque bureau possède plusieurs départements (par exemple, "IT", "Marketing", etc.), et chaque département compte un ou plusieurs employés. L'objectif du mappage est de créer des groupes d'un maximum de trois personnes de chaque département, quelque soit le bureau. La taille de chaque groupe est de trois par défaut ; néanmoins, cela devrait être simple à changer le cas échéant. Chaque groupe doit être enregistré en tant que fichier XML séparé, et le nom doit présenter le format "<Department Name>\_GroupN" (par exemple, **Marketing\_Group1.xml**, **Marketing\_Group2.xml**, etc.).



DividePersonsByDepartmentIntoGroups.mfd

Comme illustré ci-dessus, afin de pouvoir atteindre l'objectif de mappage, une variable complexe a été ajoutée au mappage, et quelques autres types de composant (surtout des fonctions). La variable présente la même structure qu'un item Department dans le XML de source. Si vous cliquez avec la touche de droite sur la variable pour consulter ses propriétés, vous remarquerez qu'elle utilise le même schéma XML que le composant de source, et contient Department en tant qu'élément de racine. Chose importante, la variable comporte deux items parent-context imbriqués, qui assurent que la variable est calculée tout d'abord dans le contexte de chaque département, et puis dans le contexte de chaque groupe dans le cadre de chaque département (voir aussi <u>Changer le contexte et l'étendue des variables</u>).

Au début, le mappage itère à travers tous les départements pour obtenir le nom de chaque département (cela sera ensuite exigé pour créer le nom de fichier correspondant à chaque groupe). Cela est réalisé en connectant la fonction <u>group-by</u> dans l'item de source Department, et en fournissant le nom du département en tant que clé de regroupement.

Ensuite, dans le contexte de chaque département, un deuxième groupement a lieu : le mappage appelle la fonction <u>group-into-blocks</u> appelle la groupe into-blocks pour créer les groupes des personnes requis. La taille de chaque groupe est fournie par un composant d'entrée simple d'une valeur par défaut de "3". La valeur par défaut est fournie par une constante. Dans cet exemple, afin de changer la taille de chaque groupe, il est facilement possible de modifier la valeurs constante selon vos besoins. Néanmoins, le composant d'entrée "size" peut aussi être modifié de manière à ce que, si le mappage est exécuté par le code généré ou avec MapForce Server, la taille de chaque groupe peut être fournie en tant qu'un paramètre dans le mappage. Pour plus d'informations, voir <u>Fournir des paramètres au mappage</u>.

Enfin, la valeur de la variable est fournie dans le composant XML PersonList cible. Le nom de fichier pour chaque groupe créé a été calculé en concaténant les parties suivantes, par le biais de la fonction <u>concat</u><sup>709</sup> :

- 1. Le nom de chaque département
- 2. Le string "\_Group"
- 3. Le numéro du groupe dans la séquence actuelle (par exemple, "1" s'il s'agit du premier groupe pour ce département)
- 4. Le string ".xml"

Le résultat de cette concaténation est stocké dans l'item Name de la variable, puis est fourni en tant que nom de fichier dynamiques dans le composant cible. Cela entraîne la création d'un nouveau nom de fichier pour chaque valeur reçue. Dans cet exemple, la variable calcule huit groupes au total, donc huit fichiers de sortie sont créés lorsque le mappage est exécuté, le cas échéant. Pour plus d'informations concernant cette technique, voir Traiter plusieurs fichiers d'entrée ou de sortie dynamiquement.
# 5.4 Joindre des données

Parfois, vous souhaitez combiner des données depuis deux structures ou plus sur la base de conditions (par exemple, si le champ A dans la première structure a la même valeur que les champ B dans la deuxième structure). Pour ce type de mappage, vous pouvez utiliser un composant Join.

Un composant Join est un composant MapForce qui permet de joindre deux structures ou plus dans le mappage sur la base de conditions définies par l'utilisateur. Il retourne l'association (ensemble joint) d'items qui satisfont à la condition. Les composants Join sont particulièrement utiles pour combiner des données provenant de deux structures qui partagent un champ commun (comme une identité).

Par exemple, dans le mappage illustré ci-dessous, le composant central est un composant "Join". Dans ce mappage, deux structures XML (une liste de personnes et une liste d'adresses) sont jointes. L'objectif est d'obtenir tous les détails de chaque personne dans un fichier XML cible. Les champs FirstName et LastName agissent en tant que clés de jointure. Si la valeur de FirstName et LastName (sous Person) est la même que celle de FirstName et LastName (sous Address), les détails d'adresse correspondent à une seule et même personne et ils deviennent "joins". Tous les items provenant de la structure jointe peuvent aussi être mappés dans une autre cible (dans ce cas, un fichier XML). La condition Join elle-même est définie dans les propriétés du composant Join, en cliquant sur la touche **Définir condition Join** ( ). Cet exemple s'accompagne d'un échantillon de mappage et est expliqué de manière détaillée dans <u>Exemple : Joindre des structures XML</u>



### JoinPeopleInfo.mfd

Comme illustré ci-dessus, les structures de source et le composant Join sont liées par des connexions "Copier tout", qui réduisent l'encombrement visuel du mappage. En général, ce type de connexion est créé automatiquement par MapForce lorsque le contexte est pertinent (pour plus d'informations, voir <u>Connexions</u> <u>Copier tout</u><sup>(95)</sup>).

Les structures qui doivent être jointes peuvent provenir soit de composants séparés (comme dans le mappage ci-dessus), soit appartenir au même composant. Les structures à joindre peuvent aussi être de genres différents (par exemple, une structure XML et une table de base de données). Pour plus d'informations concernant des jointures liées aux bases de données, voir <u>Joindre les données de base de données</u><sup>333</sup>.

#### Pour ajouter un composant Join :

- 1. Définir le langage de transformation de mappage sur BUILT-IN (pour ce faire, cliquer soit sur la touche de la barre outils with, soit utiliser la commande de menu **Sortie | Moteur d'exécution Built-In**).
- de la barre outils LM, soit utiliser la commande de menu Sortie | Moteur d'execution Built-In).
   Dans le menu Insérer, cliquer sur Join. En alternative, cliquer sur la touche de la barre outils Join (
  - I. Le composant Join apparaît dans le mappage. Par défaut, il accepte des données provenant de deux structures, il a donc deux entrées nodes/rows. Si nécessaire, vous pouvez ajouter de nouvelles entrées dans la jointure en cliquant sur la touche Ajouter entrée ( ), voir Joindre trois structures ou plus<sup>510</sup>.

	🕅 join
	📮 🕪 joined
ľ	nodes/rows
[	nodes/rows
ľ	> will condition

- 3. Connecter les structures qui doivent être jointes dans l'item nodes/rows du composant join.
- 4. Ajouter la condition pour la jointure (ou plusieurs conditions). Pour ce faire, cliquer avec la touche de droite dans le composant Join et choisir **Propriétés**. Les conditions Join peuvent aussi être ajoutées directement depuis le mappage, en connectant le résultat Booléen de certaines fonctions à l'item condition du composant Join. Dans certains cas. lorsque des tables de base de données sont jointes, la condition (ou les conditions) de jointure peut être créée automatiquement par MapForce. Pour plus d'informations, voir <u>Ajouter des conditions Join</u><sup>672</sup>.

Notes :

- Les composants Join sont pris en charge lorsque la langue cible du mappage est configurée sur BUILT-IN. La génération de code dans C#, C++, ou Java n'est pas prise en charge.
- Lorsqu'une structure n'est pas une source d'entrée valide ou prise en charge pour la jointure, MapForce affiche des indices soit immédiatement et directement dans le mappage, soit dans la fenêtre Messages, lorsque vous validez le mappage (voir <u>Valider des mappages</u><sup>103</sup>).
- Les composants Join ne doivent pas être connectés aux paramètres d'entrée ou aux résultats des fonctions définies par l'utilisateur inline. Si de telles connexions existent, des erreurs de validation se produiront au cours de la validation de mappage.
- Lorsque vous vous connectez à des composants de base de données éligibles (comme des tables ou des modes) directement dans un composant Join, une touche mode SQL (<sup>SQL</sup>) apparaît automatiquement en haut à droite du composant Join. Une fois activée, cette touche offre des fonctions SQL spéciales applicables à l'opération de jointure (voir <u>Joins dans le mode SQL</u><sup>GGO</sup>).
- Il n'est pas possible de connecter la sortie de l'item joined à un autre composant Join. Néanmoins, le cas échéant, vous pouvez connecter un résultat partiel à une jointure d'un autre.

## Composants Join comparés à d'autres types de composant

Dans certains cas, des variables complexes ou des filtres peuvent peuvent être utilisés pour obtenir les mêmes résultats (voir <u>Utiliser des variables</u><sup>432</sup> et <u>Filtres et conditions</u><sup>533</sup>, respectivement). Néanmoins, contrairement à d'autres types de composant, les composants Join permettent d'obtenir un mappage plus aisément compréhensible, étant donné que vous pouvez voir en un coup d'œil les données jointes. De plus, si le mode SQL est activé dans le composant Join, la performance de mappage s'améliore considérablement (cela s'applique aux jointures de base de données, voir <u>Joindre les données de base de données</u><sup>533</sup>).

### Ajouter un contexte parent

Dans certains cas particuliers, afin d'obtenir un résultat de mappage spécifique, vous pouvez fournir explicitement un contexte de mappage (soit-disant "contexte parent") pour les données connectées au composant Join. Pour ajouter un contexte parent, cliquer avec la touche de droite sur l'item joined du composant Join, et choisir **Ajouter Contexte Parent** depuis le menu contextuel. Le composant Join change d'apparence pour inclure une entrée parent-context supplémentaire dans laquelle vous pouvez connecter l'item de source requis. Pour plus d'informations, voir <u>Exemple : Changer le contexte Parent</u>.

parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.

## 5.4.1 Ajouter des conditions Join

Un "join" fonctionne en combinant des items de deux ou plusieurs structures conformément à une condition, de manière qu'un join exige toujours au moins une condition. Il existe plusieurs moyens d'ajouter des conditions "join", voir ci-dessous.

**Note :** Lorsque des tables de base de données sont jointes dans le mode SQL, MapForce créera la condition Join (ou les conditions) automatiquement, sur la base des relations de clé étrangère détectée entre les tables. Pour que des conditions Join automatiques se produisent, les tables de base de données doivent se trouver dans une relation enfant-parent dans le composant MapForce (c'est à dire, qu'une table doit être "parent" ou "child" d'un autre dans le composant), voir <u>Exemple: Joindre des tables dans le mode SQL</u><sup>442</sup>.

## Approche 1: Ajouter une condition join depuis les propriétés de composant

- Dans le mappage, veuillez vous assurer qu'au moins deux structures (ou tables de base de données) sont connectées au composant Join. Le composant Join illustré dans cet exemple fait partie du mappage JoinPeopleInfo.mfd disponible dans le dossier
   Cocuments>\Altova\MapForce2023\MapForceExamples\Tutorial\. Ce mappage est discuté en plus grands détails dans Exemple : Joindre des structures XML<sup>611</sup>.
- 2. Dans le composant Join, cliquer sur la touche **Définir condition Join** ( ) (ou cliquer avec la touche de droite sur le composant et sélectionner **Propriétés** depuis le menu contextuel).
- 3. Choisir un item depuis la structure de gauche et un autre depuis la structure de droite (ainsi, lorsque la comparaison de cette paire retourne vrai, les structures de gauche et de droite sont jointes).

	nation connec	tee optionnelle sont vraies :	Ajouter Conditio
Person/FirstName	=	Address/FirstName	X
	1 =		
r définir une condition, sélectionner un item depuis chaque ucture 1 :	structure pour	comparer l'égalité. Structure 2 :	
Person	-	Address	$\sim$ +
type restriction of xs:anyType [1∞] ann.		Address	
FirstName		type restriction of xs:anyType [1	∞]
type xs:string ann.		ann.	
LastName		type xs:string	
type xs:string ann.		ann.	
Email		type xs:string	
type xs:string ann.		ann.	
Phone		type xs:string	
type xs:int ann.		ann.	
		type xs:string	
		ann.	
		type xs:string	
		ann.	
	-		
	•	4	•

- Si vous souhaitez ajouter plusieurs conditions, cliquer sur **Ajouter condition**, puis sélectionner une nouvelle paire d'items. Par exemple, dans l'image ci-dessus, deux conditions join sont définies :
- 1. FirstName dans la Structure 1 doit être égale à FirstName dans la structure 2, et
- 2. LastName dans la structure 1 doit être égale à LastName dans la structure 2.

Pour supprimer une condition join, cliquer sur la touche **Supprimer** X située juste à côté.

Notes :

- Si plusieurs conditions join existent, toutes doivent être satisfaites pour que les deux structures soient jointes. En d'autres termes, plusieurs conditions sont jointes par une opération logique AND. Cela comprend également des conditions optionnelles qui ont été ajoutées depuis le mappage (voir Approche 2 ci-dessous).
- Si plus de deux structures sont connectées au composant Join, ce type de structure supplémentaire apparaît dans la liste déroulante ci-dessous "Structure 2". Lorsque vous sélectionnez une telle

structure supplémentaire depuis la liste déroulante, le panneau de gauche affiche toutes les structures qui se produisent *avant* dans le composant Join. Ainsi, vous pouvez définir des conditions "join" se trouvant entre une des structures multiples. Pour consulter un exemple, voir <u>Exemple : Créer un</u> rapport <u>CSV</u> à partir de plusieurs tables<sup>(351)</sup>.

 Pour consulter le type de données des items dans chaque structure, cocher la case Afficher types. L'option Afficher annotations affiche des informations supplémentaires concernant les items, si de telles informations existent dans le schéma sous-jacent (ou la base de données). Si les deux cases sont cochées, la mise en page change pour accommoder la vue des deux annotations et types, par exemple :

Person/FirstName		= 🗘 Address/FirstName	×
Person/LastName		Address/LastName	x
r definir une condition, selectionner un item depuis ucture 1 :	s chaque structure pou	structure 2 :	
Person	<b>A</b>	Address	~ +
type restriction of xs:anyType [1∞]			
FirstName		type restriction of xs:anvType [	1
type xs:string		ann.	
LastName		FirstName     type vsistring	
type xs:string		ann.	
ann.		LastName	
type xs:string		ann.	
ann.		City	
type xsint		type xs:string	
ann.		<>>Street	
		type xs:string	
		type xs:string	
		ann.	
	-		
	•	4	•

### Approche 2 : Ajouter une condition join depuis le mappage

• Dans le mappage, ajouter des composants qui produisent une valeur booléenne, puis connecter la sortie booléenne à l'entrée de l'item condition. Par exemple, la fonction equal peut comparer une valeur avec des items de mappage, et fournir le résultat booléen en tant qu'entrée dans l'item condition du composant join.



**Note :** Si aucune condition n'est définie depuis les propriétés de composant join (Approche 1), l'item condition du composant join doit être connecté (Approche 2).

## Approche 3: Approche mixte

Dans le même mappage, il est possible de définir des conditions join dans les propriétés de composant (Approche 1) et de les combiner avec celles provenant du mappage (Approche 2). Néanmoins, si vous avez l'intention de joindre des tables de base de données dans le mode SQL, les conditions doivent être définies strictement conformément à l'Approche 1, voir aussi Joins dans le mode SQL.<sup>339</sup>.

## 5.4.2 Joindre trois structures ou plus

Lorsque vous ajoutez un composant Join au mappage en utilisant la commande de menu **Insérer | Join**, il accepte deux structures par défaut (c'est à dire que le composant ne contient que deux entrées nodes/rows).



Si vous nécessitez plus de deux structures, cliquer sur la touche **Ajouter entrée** (•) et créer autant de nodes/rows que nécessaire. Si vous devez supprimer une entrée nodes/rows, cliquer sur la touche **Supprimer entrée** (•). Veuillez noter qu'une jointure nécessite au moins deux structures, la touche est uniquement disponible si plus de deux entrées existent.

ķ
8
8
8

Lorsqu'une jointure a plusieurs entrées, les conditions de jointure doivent donc prendre en compte chacune des entrées que vous souhaitez joindre, voir <u>Ajouter des condition Join</u><sup>607</sup>. Pour un exemple étape par étape de la manière de joindre plusieurs tables de base de données, voir <u>Exemple : Créer un rapport CSV depuis plusieurs tables</u><sup>(51)</sup>.

## 5.4.3 Exemple : Joindre des structures XML

Cet exemple illustre comment combiner des données provenant de deux structures XML conditionnellement, en utilisant un composant Join. L'exemple est accompagné par un échantillon de mappage, disponible sous : 

Cocuments>\Altova\MapForce2023\MapForceExamples\Tutorial\JoinPeopleInfo.mfd.

L'objectif de ce mappage est de collecter des données personnelles (prénom, nom de famille, adresse, e-mail et téléphone) depuis deux fichiers XML de source dans un seul fichier XML cible.

Le premier fichier XML stocke le prénom et le nom de chaque personne, ainsi que leur adresse e-mail et leur numéro de téléphone, comme indiqué dans la liste de code ci-dessous (veuillez noter que la déclaration XML, les espaces de noms et certains enregistrements ont été omis pour des raisons de simplicité) :

```
<People>
<Person>
<FirstName>Marquita</FirstName>
<LastName>Bailey</LastName>
<Email>m.bailey@nanonull.com</Email>
<Phone>555323698</Phone>
</Person>
<Person>
<FirstName>Totie</FirstName>
<LastName>Rea</LastName>
<Email>t.rea@nanonull.com</Email>
<Phone>555598653</Phone>
</Person>
</Person>
</Person>
</Person>
```

People.xml

Le deuxième fichier XML stocke le prénom et le nom de chaque personne, ainsi que leur adresse e-mail :

```
<Addresses>

    <Address>
    <FirstName>Marquita</FirstName>
```

```
<LastName>Bailey</LastName>

<City>Bridgedell</City>

<Street>Olive Street</Street>

<Number>4</Number>

</Address>

<Address>

<FirstName>Totie</FirstName>

<LastName>Rea</LastName>

<City>Roseford</City>

<Street>Evergreen Lane</Street>

<Number>34</Number>

</Address>

</Address>
```

Addresses.xml

L'objectif du mappage est de combiner les informations <Person> depuis le premier fichier avec l'information <Address> provenant du second fichier, à chaque fois que les prénom et nom correspondent. Spécifiquement, pour chaque <Person> dans le premier fichier et pour chaque <Address> dans le deuxième fichier, les FirstName et LastName doivent être comparés. Si les deux valeurs sont les mêmes, les enregistrements <Person> et <Address> correspondants se réfèrent à la même personne et doivent être joints. La structure XML de cible doit ressembler à :

```
<PeopleInfo>
  <Row>
      <FirstName>Marquita</FirstName>
      <LastName>Bailey</LastName>
      <City>Bridgedell</City>
      <Street>Olive Street</Street>
      <Number>4</Number>
      <Email>m.bailey@nanonull.com</Email>
      <Phone>555323698</Phone>
  </Row>
  <Row>
      <FirstName>Totie</FirstName>
      <LastName>Rea</LastName>
      <City>Roseford</City>
      <Street>Evergreen Lane</Street>
      <Number>34</Number>
      <Email>t.rea@nanonull.com</Email>
      <Phone>555598653</Phone>
  </Row>
</PeopleInfo>
```

PeopleInfo.xml

Le but de ce mapping est facilement atteignable en ajoutant un composant Join dans le mappage. Veuillez noter qu'il est également possible d'atteindre le même résultat en utilisant les autres types de composant ; néanmoins, dans les étapes ci-dessous, vous utiliserez un composant Join, qui est le sujet de cet exemple.

Pour créer le mappage requis, suivre les étapes suivantes.

## Étape 1: Ajouter les fichiers XML de source au mappage

- 1. Dans le menu **Insérer**, cliquer sur **Schéma XML/Fichier**, et chercher le fichier de source suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\People.xml.
- 2. Répéter l'étape ci-dessus pour Addresses.xml (le deuxième fichier de source).

## Étape 2: Ajouter le fichier de schéma cible au mappage

Dans le menu Insérer, cliquer sur Schéma XML/Fichier, et chercher
 <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\PeopleInfo.xsd (le fichier de schéma XSD cible). Lorsque vous serez invité à fournir un fichier XML d'échantillon, cliquez sur Sauter. Lorsque vous serez invité à choisir un élément racine, choisissez PeopleInfo en tant qu'élément racine.

## Étape 3: Ajouter le composant Join

1. Dans le menu **Insérer**, cliquer sur **Join**. (ou bien cliquer sur la touche de barre d'outils **Join** №). À ce niveau, le mappage devrait ressembler à l'exemple suivant (vous devrez glisser et redimensionner les composants pour les faire ressembler au graphique ci-dessous) :



Observez la structure du composant Join. Elle contient deux items nodes/rows, qui permettent d'y connecter les deux structures qui doivent être comparées (dans ce cas, les structures Person et Address).

2. Tracer une connexion depuis Person vers le premier item nodes/rows du composant Join. De même, connecter Address au deuxième item nodes/rows.



- 3. Comme mentionné plus tôt, la jointure doit avoir lieu uniquement si les valeurs FirstName et LastName sont égales dans les deux structures. Pour définir cette condition, cliquer sur la touche **Définir Condition Join**
- 4. Choisir la paire des items qui définit la première condition Join (FirstName sous Structure 1, et FirstName sous Structure 2).
- 5. Cliquer Ajouter condition, et répéter l'étape ci-dessus pour LastName.

uctures Join lorsque toutes les conditions suivantes et la cond	ion connectée optionnelle sont vra	ies :	Ajouter Condition
Person/FirstName	= 💙 Address/FirstNa	me	×
r ✔> Person/LastName	= 💙 Address/LastNar	ne	X
ur définir une condition, sélectionner un item depuis chaque st ucture 1 :	ucture pour comparer l'égalité. Structure 2 :		
✓Person	Address		~ +
type restriction of xs:anyType [1∞] ann. type xs:string ann. <b>⇒LastName</b> type xs:string ann. <b>⇒Email</b> type xs:string ann. <b>⇒Phone</b> type xs:string ann.	□       Address         type       restriction         ann.       FirstName         type       xs:st         ann.       LastName         type       xs:st         ann.       City         type       xs:st         ann.       Street         type       xs:st         ann.       Number         type       xs:st         ann.       Street	tion of xs:anyType [1] ne ring ring ring	
	× (		Þ
Afficher annotations	•		

Dans certains mappage, une condition consistant en une comparaison peut suffire à effectuer la jointure. Néanmoins, dans cet exemple, il est important de créer deux comparaisons :

- 1) FirstName dans Structure1 = FirstName dans Structure 2
- 2) LastName dans Structure 1 = LastName dans Structure 2.

Si plusieurs conditions sont définies, *elles doivent toutes être vraies pour que la jointure puisse avoir lieu*. C'est pourquoi, dans cet exemple, une jointure n'aura lieu que si les deux comparaisons sont vraies (ce qui est le comportement souhaité). Si une seule des comparaisons ci-dessus a été définie, une jointure peut se produire pour des personnes ayant le même prénom mais des noms de famille différents.

## Étape 4: Mapper le composant Join vers le schéma de cible

Maintenant que les deux structures sont jointes, vous pouvez définir quels items de la structure jointe doivent être mappés dans la cible. Pour ce faire, créer des connexions depuis les items dans les deux structures jointes vers le composant de cible, comme indiqué ci-dessous. La connexion entre joined et Row a l'objectif suivant : lorsque la condition Join est satisfaite, elle crée un nouvel item Row dans la cible.



Pour consulter la sortie de mappage, cliquer sur l'onglet **Sortie**. Comme prévu, chaque enregistrement de personne (<Row>) comprend maintenant tous les détails d'adresse, joints depuis deux sources différentes.

# 5.5 Trier les données

Afin de trier les données basées sur une clé de tri spécifique, utilisez un composant Sort (Tri). Le composant Tri prend en charge les langues cible suivantes : XSLT2, XQuery, et Built-in. Lorsque le langage de transformation est "Built-in", le composant Tri peut être utilisé pour trier les données de table de la base de données. Néanmoins, il est possible d'obtenir une meilleure performance par le biais d'un composant SQL-WHERE/ORDER. Pour plus de détails, voir <u>Filtrer et trier des données de base de données (SQL</u> <u>WHERE/ORDER</u>).

### Afin d'ajouter un composant de tri au mappage, suivez une des étapes suivantes :

 Cliquez avec la touche de droite sur une connexion existante et sélectionnez Insérer Sort : Nœuds/Lignes depuis le menu contextuel. Cela permet d'insérer le composant Tri et de le connecter automatiquement aux composants de source et de cible. Par exemple, dans le mappage ci-dessous, le composant Tri a été inséré entre une variable et un composant XML. La seule chose qui reste à connecter manuellement est la clé de triage (le champ avec lequel vous souhaitez trier).



• Dans le menu **Insérer**, cliquer sur **Sort** (ou bien cliquer sur la touche de la barre d'outils **Sort**). Le composant Sort est inséré dans sa forme "non-connectée".

	A⊈sort	
կ	nodes/rows	recutt
Ę	key (A=>Z)	result.

Dès qu'une connexion a été établie dans le composant de source, le nom de barre de titre change pour prendre celui de l'item connecté à l'item nodes/rows.

#### Pour définir l'item par lequel vous souhaitez trier :

• Connecter l'item par lequel vous souhaitez trier le paramètre key du composant Tri. Par exemple, dans le mappage ci-dessous, les nœuds/lignes Person sont triées par le champ Last.



#### Pour modifier l'ordre de tri :

Cliquer sur l'icône A⇒Z dans le composant Sort. Elle passe à Z→A pour montrer que l'ordre de tri a été inversé.

#### Pour trier des données d'entrée consistant en des items de type simple :

• Connecter l'item aux deux paramètres nœuds/lignes et clé du composant de tri sort. Dans le mappage ci-dessous, l'élément de type simple first est en cours de tri.



#### Pour trier les strings en utilisant des règles spécifiques au langage :

• Double-cliquer sur l'en-tête du composant Sort pour ouvrir le dialogue Trier les propriétés.

😢 Trier les propri	étés	×								
Sélectionner la collation que vous souhaitez utiliser pour le tri de chaîne.										
Collation de point de code Unicode										
O Collation spécifi	que au langage									
Locale										
Langue :	×									
Pays :	×									
	OK Annuler									

**Collation point de code Unicode :** Cette option (par défaut) compare/trie des strings basés sur des valeurs de point de code. Les valeurs de point de code sont des entiers qui ont été attribués à des caractères abstraits dans le Universal Character Set adopté par l'Unicode Consortium. Cette option permet un tri pour de nombreuses langues et scripts.

**Collation spécifique à la langue** : Cette option vous permet de définir la langue et les variantes régionales spécifiques par lesquelles vous souhaitez trier. Cette option est prise en charge lors de l'utilisation du moteur d'exécution BUILT-IN. Pour XSLT, la prise en charge dépend du moteur spécifique utilisé pour exécuter le code.

## 5.5.1 Trier par clés multiples

Une fois avoir ajouté un composant Sort au mappage, une clé de tri appelée key sera créée par défaut.

AZ	sort	
	des/rows	reeu#D
Ekey	/ (A=>Z	result

Composant Sort par défaut

Si vous souhaitez trier par clés multiples, ajuster le composant Sort comme suit :

- Cliquer sur l'icône **Ajouter clé** ( ) pour ajouter une nouvelle clé (par exemple, key2 dans le mappage ci-dessous).
- Cliquer sur l'icône **Delete clé** ( 🗵 ) pour supprimer une clé.
- Déposer une connexion dans l'icône 
  pour ajouter une clé et pour s'y connecter.

Un mappage qui illustre le tri avec plusieurs clés est disponible dans le chemin suivant : <br/>
<Documents>\Altova\MapForce2023\MapForceExamples\SortByMultipleKeys.mfd.



SortByMultipleKeys.mfd

Dans le mappage ci-dessus, les enregistrements Person sont triés avec trois clés de tri :

- 1. Shares (nombre de parts que détient une personne)
- 2. Last (nom de famille)
- 3. First (prénom)

Veuillez noter que la position de la clé de tri dans le composant Sort détermine sa priorité de tri. Par exemple, dans le mappage ci-dessus, les enregistrements sont tout d'abord triés par le nombre de parts. Il s'agit de la clé de tri avec la priorité la plus élevée. Si le nombre de parts est le même, les gens sont ensuite triés par leur nom de famille. Enfin, si plusieurs personnes ont le même nombre de parts et le même nom de famille, le prénom de la personne sera pris en compte.

L'ordre de tri de chaque clé peut être différent. Dans le mappage ci-dessus, la clé shares a un ordre de tri descendant (Z-A), alors que les deux autres clés ont un ordre de tri ascendant (A-Z).

## 5.5.2 Trier par variables

Dans certains cas, il peut s'avérer nécessaire d'ajouter des variables intermédiaires au mappage pour aboutir au résultat escompté. Cet exemple illustre comment extraire des enregistrements d'un fichier XML et les trier, avec l'aide des variables intermédiaires. L'exemple est accompagné par un échantillon de mappage situé dans

#### le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Altova\_Hierarchical\_Sort.mfd.



Altova\_Hierarchical\_Sort.mfd

Ce mappage lit des données provenant d'un fichier XML de source appelé **Altova\_Hierarchical.xml** et l'écrit dans un fichier XML cible. Comme affiché ci-dessus, le XML source contient l'information à propos d'une entreprise fictive. L'entreprise est divisée en bureaux. Les bureaux sont sous-divisés en départements, et les départements sont répartis en personnes.

Le composant XML cible, PersonList, contient une liste des enregistrements Person. L'item Details est censé stocker des informations concernant le bureau et le département auquel la personne appartient.

L'objectif est d'extraire toutes les personnes depuis le XML de source et de les trier par ordre alphabétique avec leur nom de famille. De plus, le nom du bureau et de département auquel chaque personne appartient doit être écrit dans l'item Details.

Pour aboutir cet objectif, cet exemple utilise les types de composant suivants :

1. La fonction concat. Dans ce mappage, cette fonction retourne un string dans le format Office(Department). Il prend en tant qu'entrée le nom du bureau, le nom du département et deux constantes qui donnent les crochets de début et de fin. Voir aussi Ajouter une fonction au mappage 52

- 2. Une variable intermédiaire. Le rôle de la variable est d'apporter toutes les données concernant une personne dans le même contexte de mappage. La variable a l'effet suivant : le mappage consulte le département et le bureau de chaque personne dans le contexte de chaque personne. Autrement dit, la variable "se souvient" du nom du bureau et du département auquel appartient une personne. Sans la variable, le contexte serait incorrect, et le mappage produirait une sortie non désirée (pour plus d'informations concernant comment un mappage est exécuté, voir <u>Mapper les règles et les stratégies</u>). Veuillez noter que la variable copie la structure du fichier XML cible (elle utilise le même schéma XML). Cela permet de connecter le résultat de tri vers la cible, par le biais d'une connexion Copier tout. Voir aussi <u>Utiliser des variables</u> et <u>Connexions Copier tout</u>.
- 3. Un composant Sort, qui effectue le tri lui-même. Veuillez noter que l'entrée de clé du composant Sort est connecté à l'item Last de la variable, qui trie tous les enregistrements de personnes par leur nom de famille.

# 5.6 Filtres et conditions

Si vous souhaitez filtrer des données ou obtenir une valeur par condition, vous pouvez utiliser un des types de composant suivants :

- Filtre : Nœuds/Lignes ( 축 )
- SQL WHERE/ORDER ( 🖾 )
- Condition If-Else ( ই )

Vous pouvez ajouter ces composants dans le mappage soit depuis le menu **Insérer**, soit depuis la barre d'outils **Insérer composant**. Attention, chaque composant indiqué ci-dessus possède son propre comportement et ses propres exigences. Les différences sont expliquées dans les sections suivantes.

## Filtrer les nœud ou les lignes

Si vous souhaitez filtrer des données, y compris des nœuds XML ou des lignes CSV, utiliser un composant **Filtrer nœuds/lignes**. Celui-ci vous permet d'extraire un sous-ensemble de nœuds depuis un ensemble de données plus important, sur la base d'une condition vrai ou faux. Sa structure dans la zone de mappage le reflète :

😴 filter		
(>node/ro	w	on-true(>
>bool		on-false(>

Dans la structure ci-dessus, la condition connectée à **bool** détermine si le **nœud/ligne** connecté va bien vers la sortie **on-true** ou **on-false**. Concrètement, si la condition est vraie, le **nœud/ligne** sera redirigé vers la sortie **on-true**. Ainsi, si la condition est fausse, le **nœud/ligne** sera redirigé vers la sortie **on-false**.

Si votre mappage nécessite de consommer uniquement des items qui *correspondent* à la condition de filtre, vous pouvez laisser la sortie **on-false** sans connexion. Si vous devez traiter les items qui *ne correspondent pas* à la condition de filtre, connecter la sortie **on-false** vers une cible dans laquelle ces items doivent être redirigés. Si vous souhaitez ajouter une exception lorsque la condition du filtre n'est pas remplie, il est obligatoire de connecter la sortie **on-false** (voir <u>Ajouter des exceptions</u>).

Pour un exemple de mappage étape par étape, voir <u>Exemple : Filtrer des nœuds</u><sup>525</sup>.

## Filtrer les données de base de données

Les composants **Filtrer nœuds/lignes** peuvent filtrer des données depuis toute autre structure de composant prise en charge par MapForce, y compris des bases de données. Néanmoins, si vous souhaitez filtrer des données depuis une base de données, il est recommandé d'utiliser un composant **SQL WHERE/ORDER** à la place. Ce composant **SQL WHERE/ORDER** est optimisé pour travailler avec des bases de données et fournit une meilleure performance qu'un composant **Filtrer nœuds/lignes**.



Pour plus d'informations concernant ces composants, voir Composant SQL WHERE / ORDER 355.

#### Retourner une valeur par condition

Si vous souhaitez obtenir une seule valeur (pas un nœud ou une ligne) de manière conditionnelle, utiliser une **Condition If-Else**. Veuillez noter que celles-ci ne sont pas appropriées pour filtrer les nœuds ou les lignes. Contrairement aux composants **Filtrer nœuds/lignes**, une **Condition If-Else** retourne une valeur de type simple (comme un string ou un entier). C'est pourquoi, les **Conditions If-Else** ne se prêtent que pour des scénarios dans lesquels vous devez traiter une valeur simple par condition. Par exemple, partons du principe que vous avez une liste de températures moyennes par mois, dans le format :

```
<Temperatures>

<data temp="19.2" month="2010-06" />

<data temp="22.3" month="2010-07" />

<data temp="19.5" month="2010-08" />

<data temp="14.2" month="2010-09" />

<data temp="7.8" month="2010-10" />

<data temp="6.9" month="2010-11" />

<data temp="-1.0" month="2010-12" />

</Temperatures>
```

Une **Condition If-Else** vous permettrait de retourner, pour chaque item dans la liste, la valeur "high" si la température dépasse 20 degrés Celsius, et la valeur "low" si la température est inférieure à 5 degrés Celsius.

Dans le mappage, la structure de la Condition If-Else ressemble à l'exemple suivant :



Si la condition connectée à **bool** est vraie, alors la valeur connectée à **value-true** est sorti sous la forme de **result**. Si la condition est fausse, la valeur connectée à **value-false** est sortie sous la forme de **result**. Le type de donnée de **result** n'est pas connu à l'avance ; il dépend du type de données de la valeur connectée à **value-true** ou **value-false**. Attention : ce doit toujours être un type simple (string, entier, etc). La connexion de valeurs d'entrée de type complexe (comme des nœuds ou des lignes) n'est pas prise en charge par la **Condition If-Else**.

Les **Conditions If-Else** sont extensibles. Cela signifie que vous pouvez ajouter plusieurs conditions dans le composant en cliquant sur la touche **Ajouter** ( • ). Pour supprimer une condition préalablement ajoutée, cliquer sur la touche **Supprimer** ( ). Cette fonction vous permet de contrôler plusieurs conditions et de retourner une valeur différente pour chaque condition, si elle est vraie.

	∬ <sub>8 </sub> if-else	
հ	bool1 🛛	
1	value-true1	
հ	bool2 🗵	
1	value-true2	result[:
հ	>bool3 🛛 🛛	
1	value-true3	
1	otherwise	

Les **Conditions If-Else** étendues sont évaluées du haut en bas (la première condition est contrôlée en premier, puis la deuxième, etc.). Si vous souhaitez retourner une valeur si aucune des conditions ne sont vraies, connectez-vous à **otherwise**.

Pour un exemple de mappage étape par étape, voir <u>Exemple: Retourner une valeur par condition <sup>527</sup></u>.

# 5.6.1 Exemple : Filtrer des nœuds

Cet exemple vous montre comment filtrer des nœuds basés sur une condition true/false. Un composant **Filtre : Nœuds/Lignes** (  $\stackrel{\clubsuit}{\Rightarrow}$  ) est utilisé pour atteindre cet objectif. La technique illustrée dans cet exemple fonctionne de la même manière non seulement pour XML, mais aussi pour d'autres types de composant, comme CSV ou texte. Dans le cas des bases de données, bien que vous pouvez utiliser un filtre, il est recommandé d'utiliser à la place un composant SQL WHERE/ORDER, pour une meilleure performance (voir <u>Composant SQL WHERE /</u> <u>ORDER</u> ).

Le mappage décrit dans cet exemple est disponible sous le chemin suivant : <Documents>\Altova\MapForce2023\MapForceExamples\MarketingExpenses.mfd.



Comme indiqué ci-dessus, le mappage lit des données depuis un XML de source qui contient les notes de frais ("ExpReport") et écrit les données dans un XML cible ("MarketingExpenses"). Il existe plusieurs autres composants entre la cible et la source. Le composant le plus pertinent est le filtre **expense-item** ( T), qui représente le sujet de cette rubrique.

Le but du mappage est de filtrer uniquement les postes de dépense qui appartiennent au département Marketing. Pour atteindre cet objectif, un composant de filtre a été ajouté au mappage. (Pour ajouter un filtre, cliquez dans le menu **Insérer**, puis cliquez sur **Filtre : Nœuds/Lignes**.)

Afin d'identifier si chaque poste de dépense doit être attribué à Marketing, ce mappage consulte la valeur de l'attribut "expto" dans la source. Cet attribut a la valeur "Marketing" dès que la dépense est une dépense de marketing. Par exemple, dans la liste de code ci-dessous, le premier et le troisième poste de dépense correspond à Marketing, le deuxième appartient à Development, et le quatrième appartient à Sales :

```
<expense-item type="Meal" expto="Marketing">
     <Date>2003-01-01
     <expense>122.11</expense>
  </expense-item>
  <expense-item type="Lodging" expto="Development">
     <Date>2003-01-02</Date>
     <expense>122.12</expense>
  </expense-item>
  <expense-item type="Lodging" expto="Marketing">
     <Date>2003-01-02</Date>
     <expense>299.45</expense>
  </expense-item>
  <expense-item type="Entertainment" expto="Sales">
     <Date>2003-01-02</Date>
     <expense>13.22</expense>
  </expense-item>
. . .
```

Entrée XML avant exécution du mappage

Dans la zone de mappage, l'entrée **nœud/ligne** du filtre est connectée au nœud **expense-item** dans le composant de source. Cela assure que le composant de filtre obtient la liste des nœuds qu'il doit traiter.

Pour ajouter la condition sur la base de laquelle le filtrage doit se produire, nous avons ajouté la fonction equal depuis la bibliothèque principale de MapForce (pour plus d'informations, voir aussi <u>Ajouter une fonction au mappage</u>). La fonction equal compare la valeur de l'attribut expto à une constante qui a la valeur Marketing. (Pour ajouter une constante, cliquez sur le menu **Insérer**, puis cliquez sur **Constante**.)

Puisque nous souhaitons filtrer uniquement les items qui satisfont à la condition, nous connectons uniquement la sortie **on-true** du filtre sur le composant de cible.

Lorsque vous consultez le résultat de mappage, en cliquant sur l'onglet **Sortie**, MapForce évalue, pour chaque nœud expense-item, la condition connectée à l'entrée **bool** du filtre. Lorsque la condition est vraie, le nœud expense-item est passé sur le cible ; sinon, elle sera ignorée. Par conséquent, seuls les items de dépense correspondant aux critères seront affichés dans la sortie :

Sortie XML après exécution du mappage

## 5.6.2 Exemple: Retourner une valeur par condition

Cet exemple vous montre comment retourner une valeur simple depuis un composant, basée sur une condition vraie/faux. Une **Condition If-Else** ( ) est utilisée pour atteindre l'objectif. Veuillez noter que les **Conditions If-Else** ne doivent pas être confondues avec les composants de filtre. Les **Conditions If-Else** sont uniquement applicables si vous souhaitez traiter des valeurs simples de manière conditionnelle (string, entier, etc.). Si vous souhaitez filtrer des valeurs complexes comme des nœuds, utiliser un filtre à la place (voir <u>Exemple : Filtrer</u> des nœuds).

Le mappage décrit dans cet exemple est disponible sous le chemin suivant : <br/>
<Documents>\Altova\MapForce2023\MapForceExamples\ClassifyTemperatures.mfd.



Ce mappage lit des données depuis un XML de source qui contient des données de température ("Temperatures") et écrit des données vers un XML de cible XML qui se conforme au même schéma. Il existe plusieurs autres composants entre la cible et la source, un d'entre eux est la condition **if-else** (marquée en rouge), qui est aussi le sujet de cette rubrique.

L'objectif de ce mappage est d'ajouter une brève description à chaque enregistrement de température dans la cible. Spécifiquement, si la température se situe au-dessus de 20 degrés Celsius, la description doit être "high". Si la température se situe à 5 degrés Celsius, la description doit être "low". Pour tous les autres cas, aucune description ne doit être écrite.

Pour atteindre cet objectif, un traitement par condition est nécessaire ; c'est pourquoi une Condition If-Else a été ajoutée au mappage. (Pour ajouter une Condition If-Else, cliquer le menu **Insérer**, puis cliquer sur la **Condition If-Else**.) Dans ce mappage, la Condition If-Else a été étendue (avec l'aide de la touche •) pour accepter deux conditions : **bool1** et **bool2**.

Les conditions elles-mêmes sont fournies par les fonctions greater et less, qui ont été ajoutées depuis la bibliothèque principale MapForce (pour plus d'informations, voir aussi <u>Ajouter une fonction au mappage</u><sup>652</sup>). Ces fonctions évaluent les valeurs fournies par deux composants d'entrée, appelés "upper" et "lower". (Pour ajouter un composant d'entrée, cliquer sur le menu **Insérer**, puis sur **Insérer entrée**. Pour plus d'informations à propos des composants d'entrée, voir <u>Fournir des paramètres au mappage</u><sup>477</sup>.

Les fonctions greater et less retournent soit true soit false. Le résultat de fonction détermine ce qui est écrit dans l'instance cible. Concrètement, si la valeur de l'attribut "temp" dans la source est supérieur à 20, la valeur de constante "high" est passée à la condition **if-else**. Si la valeur de l'attribut "temp" dans la source est

inférieure à 5, la valeur de constante "low" est passée dans la condition **if-else**. L'entrée **otherwise** n'est pas connectée. C'est pourquoi, si aucune des conditions ci-dessus n'est remplie, rien ne sera passé dans le connecteur de sortie **result**.

Enfin, le connecteur de sortie **result** fournit cette valeur (une fois pour chaque enregistrement de température) vers l'attribut "desc" dans la cible.

Une fois que vous êtes prêt à consulter le résultat de mappage, cliquer sur l'onglet **Sortie**. Veuillez noter que la sortie XML résultante contient maintenant l'attribut "desc", que la température soit supérieure à 20 ou inférieure à 5.

```
<data temp="-3.6" month="2006-01" desc="low"/>
<data temp="-0.7" month="2006-02" desc="low"/>
<data temp="7.5" month="2006-03"/>
<data temp="12.4" month="2006-04"/>
<data temp="16.2" month="2006-05"/>
<data temp="19" month="2006-06"/>
<data temp="22.7" month="2006-06"/>
<data temp="23.2" month="2006-08" desc="high"/>
</array>
```

Sortie XML après exécution du mappage

# 5.7 Value-Maps

Le composant Value-Map vous permet de transformer une valeur par une autre valeur à l'aide d'une table de consultation prédéfinie. Un tel composant ne traite qu'une seule valeur à la fois ; c'est pourquoi il a une **entrée** et un **résultat** dans le mappage.



Un Value-Map est utile lorsque vous souhaitez mapper des items individuels dans deux ensembles pour pouvoir remplacer des items. Par exemple, vous pourriez mapper les jours de la semaine exprimés en chiffres (1, 2, 3, 4, 5, 6 et 7) pour le nom de chaque jour de la semaine ("Lundi", "Mardi", etc). De même, vous pourriez mapper les noms des mois ("Janvier", "Février", "Mars", etc) dans la représentation numérique de chaque mois (1, 2, 3, etc). Au moment de l'exécution du mappage, les valeurs correspondantes seront remplacées conformément à votre table de consultation personnalisée. Les valeurs dans les deux ensembles peuvent être de types différents, mais chaque ensemble doit stocker des valeurs du même type de données.

Les composants Value-Map conviennent pour des consultations simples, dans lesquelles chaque valeur dans le premier ensemble correspond à une seule valeur dans le second ensemble. Si une valeur ne peut pas être trouvée dans la table de consultation, vous pouvez soit la remplacer avec une valeur personnalisée ou une valeur vide, soit la transmettre telle quelle. Si vous souhaitez consulter ou filtrer des valeurs sur la base de critères plus complexes, utiliser un des <u>composants de filtre</u><sup>620</sup> à la place.

Plus important, lorsque vous générez du code ou que vous compilez un fichier d'exécution MapForce Server à partir du mappage, les données de la table de consultation sont intégrées dans le code ou le fichier généré. Par conséquent, la définition d'une table de consultation directement sur le mappage est un bon choix uniquement si vos données ne changent pas fréquemment et ne sont pas trop volumineuses (moins de quelques centaines d'enregistrements). Si les données changent fréquemment, il peut s'avérer difficile d'entretenir le mappage et le code généré régulièrement, il est plus simple d'entretenir les données de consultation en tant que texte, XML, base de données ou éventuellement en Excel.

Si la table de consultation est très volumineuse, l'exécution de mappage sera ralentie par la table de consultation. Dans ce cas, il est recommandé d'utiliser un composant de base de données avec <u>SQL-</u> <u>Where</u> a la place. Les bases de données SQLite sont donc prédestinées, étant donné leur portabilité. Du côté serveur, vous pouvez améliorer la performance des tables de consultation en exécutant un mappage avec MapForce Server ou MapForce Server Advanced Edition.

## Créer des Value-Maps

Pour ajouter un composant Value-Map dans le mappage, procéder comme suit :

- Cliquer sur la touche de barre d'outils Insérer Value-Map
- Dans le menu Insertion, cliquer sur Value-Map.
- Cliquer avec la touche de droite sur une connexion, et sélectionner Insérer Value-Map depuis le menu contextuel.

Cela ajoute un nouveau composant Value-Map dans le mappage. Vous pouvez maintenant commencer à ajouter des paires d'items dans la table de consultation. Pour ce faire, double-cliquer sur la barre de titre du composant ou cliquer dessus avec la touche de droite et choisir **Propriétés** depuis le menu contextuel.

😢 V	aleur-Propriétés Map	×
Table	Value-Map pour mapper les valeurs spéci	fiques à d'autres valeurs :
	input 🥖	result 🥖
	integer 💌	string 💌
	1	Sunday
	2	Monday
	3	Tuesday
	4	Wednesday
	5	Thursday
	6	Friday
	7	Saturday
*>	(nouvelle entrée)	
	Sinon	Valeur: incorrect date
		O Valeur d'entrée originale
		OK Annuler

Au moment de l'exécution du mappage, MapForce contrôle chaque valeur qui atteint l'**entrée** de Value-Map. S'il existe une valeur correspondante *dans la colonne gauche* de la table de consultation, alors la valeur d'entrée originale sera remplacée avec la valeur *provenant de la colonne de droite*. Sinon, vous pouvez la configurer en option pour retourner un des éléments suivants :

- Une valeur de remplacement. Dans l'exemple ci-dessus, la valeur de remplacement est le texte "incorrect date" (date non correcte). Vous pouvez également définir la valeur de remplacement pour qu'elle soit vide, en ne saisissant aucun texte.
- La valeur d'entrée originale. Cela signifie que, si aucune correspondance n'est trouvée dans la table de consultation, la valeur d'entrée originale sera transmise plus loin dans le mappage, sans être modifiée.

Si vous ne configurez pas une condition "Otherwise", le Value-Map retourne un **nœud vide** à chaque fois qu'une correspondance n'est pas trouvée. Dans ce cas, rien ne sera transmis dans le composant cible et la sortie contiendra des champs manquants. Pour éviter que cela ne se produise, vous devriez soit configurer la condition "Otherwise" soit utiliser la fonction <u>substitute-missing</u><sup>707</sup>.

Il existe une différence entre configurer une valeur de remplacement vide et ne pas spécifier la condition "Otherwise". Dans le premier cas, le champ sera généré dans la sortie, mais il aura une valeur vide. Dans le deuxième cas, le champ (ou l'élément XML) entourant la valeur ne sera pas créé du tout. Pour plus d'informations, voir <u>Exemple : Remplacer les titres d'emploi</u><sup>538</sup>.

## Remplir un Value-Map

Dans une table de consultation, vous pouvez définir autant de paires de valeurs que nécessaire. Vous pouvez soit saisir les valeurs manuellement, soit copier-coller des données tabulaires depuis des fichiers texte, CSV, ou Excel. Le copier-coller de tables depuis une page HTML en utilisant un navigateur commun fonctionnera aussi dans la plupart des cas. Vous pouvez aussi coller des données depuis la grille de la base de données dans le <u>volet de Requête BD</u><sup>(322)</sup>. Si vous copiez des données depuis des fichiers de texte, les champs doivent être séparés par des caractères de tabulateur. De plus, MapForce reconnaîtra du texte séparé par des virgules ou des points-virgules dans la plupart des cas.

Veuillez considérer les éléments suivants lorsque vous créez des tables de consultation :

- 1. Tous les items dans la colonne de gauche doivent être uniques. Sinon, il ne serait pas possible de déterminer pour quel item vous souhaitez effectuer une correspondance.
- Les items qui correspondent à la même colonne doivent être du même type de données. Vous pouvez choisir le type de données depuis la liste déroulante au sommet de chaque colonne dans la table de consultation. Si vous souhaitez convertir des types booléens, saisir le texte "true" ou "false" littéralement. Pour consulter une illustration de ce cas, voir <u>Exemple : Remplacer les jours de la semaine</u><sup>633</sup>.

Si MapForce rencontre des données invalides dans la table de consultation, il affiche un message d'erreur et souligne les lignes invalides en rose, par exemple :

#### Pour importer des données depuis une source externe dans le composant Value-Map :

- 1. Sélectionner les cellules pertinentes dans le programme de source (par exemple, Excel). Cela peut être soit une colonne de données simple ou deux colonnes adjacentes.
- 2. Copier des données dans le presse-papiers en utilisant la commande **Copier** du programme externe.
- 3. Dans le composant Value-Map, cliquer sur la ligne précédent celle dans laquelle vous souhaitez coller les données.
- 4. Cliquer sur la touche **Coller la table depuis le presse-papiers** adans le composant Value-Map. En alternative, appuyer sur **Ctrl+V** ou **Shift+Insert**.
- **Note :** La touche **Coller la table depuis le presse-papiers** est uniquement activée si vous avez tout d'abord copié des données depuis une source (donc, si des données se trouvent dans le presse-papiers).

Lorsque vous données contenues dans le presse-papiers contient plusieurs colonnes, alors seules les données provenant des premières deux colonnes sont insérées dans la table de consultation ; toutes les autres colonnes seront ignorées. Si vous collez des données provenant d'une seule colonne sur des valeurs existantes, un menu contextuel apparaît, vous demandant si les données contenues dans le presse-papiers doivent être insérées dans les nouvelles lignes ou si les lignes existantes doivent être écrasées. Ainsi, si vous souhaitez écraser les valeurs existantes dans la table de consultation (par opposition à l'insertion de nouvelles lignes), veuillez vous assurer que le presse-papiers ne contient qu'une seule colonne.

Pour	insérer	des	lignes	manuelleme	nt avant	une	ligne	existante,	cliquer	tout	d'abord	sur	la ligne	concei	rnée,
puis	cliquer	sur la	a touch	ne Insertion			0						Ū		

Pour déplacer une ligne existante à une autre position, glisser la ligne sur la nouvelle position (vers le haut ou vers le bas) tout en maintenant appuyée la touche de gauche de la souris.

#### Composants de transformation

Pour copier ou couper des lignes pour les coller ensuite sur une autre position, sélectionner tout d'abord la ligne, puis cliquer sur la touche **Copier** (ou sur la touche **Couper**), respectivement). Vous pouvez aussi copier ou couper plusieurs lignes qui ne sont pas forcément consécutives. Pour sélectionner plusieurs lignes, maintenir appuyée la touche **Ctrl** tout en cliquant sur la ligne. Veuillez noter que le texte coupé ou copié

contient toujours des valeurs provenant des deux colonnes ; vous ne pouvez pas couper ou copier des valeurs provenant d'une seule colonne.

Pour supprimer une ligne, cliquer dessus puis cliquer sur la touche Supprimer 🔀.

Pour changer l'ordre de deux colonnes, cliquer sur la touche Échanger

## Renommer des paramètres de Value-Map

Par défaut, le paramètre d'entrée d'un composant Value-Map est appelé "entry" et le paramètre de sortie est appelé "result". Pour rendre le mappage plus clair, vous pouvez renommer en option un de ces paramètres en cliquant sur la touche Édition située à côté du nom respectif. Le graphique suivant est un exemple d'un Value-Map avec des noms de paramètre personnalisés :

value-map P day-as-number day-as-text P

### Préconsulter un Value-Map

Une fois le Value-Map créé, vous pouvez rapidement consulter sa mise en place directement dans le mappage en plaçant la souris au-dessus de la barre de titre du composant :



# 5.7.1 Exemple : Remplacer les jours de la semaine

Cet exemple illustre un Value-Map qui remplace des valeurs entières avec les noms du jour de la semaine (1 = Dimanche, 2 = Lundi, etc.). Cet exemple est accompagné par un mappage qui est disponible dans le chemin suivant : **Compage: Compage: C** 



#### Expense-valmap.mfd

Ce mappage extrait le jour de la semaine depuis l'item **Date** dans le fichier de source, convertit la valeur numérique dans le texte et l'écrit dans l'item **Weekday** du composant de cible. Plus spécifiquement, la chose suivante se produit :

- La fonction weekday extrait le numéro du jour de la semaine depuis l'item **Date** dans le fichier de source. Le résultat de cette fonction sont des entiers allant de 1 à 7.
- Le premier composant Value-Map transforme les entiers dans les jours de la semaine (1 = Dimanche, 2 = Lundi, etc.). Si le composant rencontre un entier invalide en-dehors de la plage 1-7, il retournera le texte "incorrect date".

	input 🥖	result 🥖
	integer 💌	string 💌
	]	Sunday
	2	Monday
	3	Tuesday
	4 5	Thursday
	5	Friday
	7	Saturday
*>	(nouvelle entrée)	Saturday
-	Sinon	Valeur: incorrect date
		O Valeur d'entrée originale
		O Valeur d'entrée originale

• Si le jour de la semaine contient "Tuesday", alors le texte "Prepare Financial Reports" sera écrit dans l'item **Notes** dans le composant de cible. Cela s'effectue avec l'aide de la fonction contains, qui passe une valeur booléenne **true** ou **false** dans un second composant Value-Map. Le second Value-Map présente la configuration suivante :

aleur-Propriétés Map		×
· Value-Map pour mapper les v	valeurs spécifiques à d'autres valeurs :	×
input	🥒 result	1
boolean	✓ string	•
true	Prepare financial reports !	
(nouvelle entrée)		
Sinon	Valeur:	
1	O Valeur d'entrée originale	
	aleur-Propriétés Map Value-Map pour mapper les v Value-Map pour les v Value-Map	aleur-Propriétés Map         Value-Map pour mapper les valeurs spécifiques à d'autres valeurs :         ▶ ● ● ●         input       ✓         boolean       ✓         true       Prepare financial reports !         (nouvelle entrée)       ●         Valeur:       ●         Valeur d'entrée originale

Le Value-Map illustré ci-dessus doit être compris comme suit :

• Lorsqu'une booléenne **true** est rencontrée, la convertir dans le texte "-- Prepare financial reports -- ! ". Pour tous les autres cas, elle retourne le texte "--".

Veuillez noter que le type de données de la première colonne est définie sur "boolean". Cela garantit que la valeur booléenne d'entrée **true** est reconnue comme telle.

## 5.7.2 Exemple : Remplacer des titres de tâche

Cet exemple vous montre comment remplacer des valeurs d'éléments spécifiques dans un fichier XML à l'aide de composants Value-Map (c'est à dire, en utilisant une table de consultation prédéfinie).

Le fichier XML nécessaire pour cet exemple est disponible dans le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\ MFCompany.xml. Il stocke, entre autres données, des informations concernant les employés de l'entreprise et leurs titres de poste, par exemple :

```
<Person>

<Person>
<last>Butler</Last>
<last>Butler</Last>
Title>Software Engineer</Title>
</Person>
<Person>
<Title>Support Engineer</Title>
</Person>
</Person>
<Person>
<Person>
<person>
</Person>
```

Partons du principe que vous souhaitez remplacer certains des titres de poste dans le fichier XML ci-dessus. En particulier, le titre "Software Engineer" doit être remplacé par "Code Magician". De même, le titre "Support Engineer" doit être remplacé par "Support Magician". Les autres titres de poste ne doivent pas être modifiés.

Pour atteindre cet objectif, ajouter le fichier XML à la surface de mappage, en cliquant sur la touche de la barre

d'outils **Insérer Schéma/Fichier XML** ou en exécutant la commande de menu **Insérer | Schéma/Fichier XML**. Ensuite, copier-coller le composant XML dans le mappage et créer les connexions comme indiqué <u>ci-d</u>essous. Veuillez noter que vous devrez éventuellement tout d'abord désactiver l'option dans

la barre d'outils 🖽 **Basculer l'auto-connexion des enfants**, pour pouvoir éviter l'établissement automatique de connexions inutiles.



Le mappage créé jusqu'à présent copie les éléments **Person** dans le fichier cible XML, sans effectuer de modifications dans les éléments **First**, **Last** et **Title**.

Pour remplacer les titres de postes requis, ajoutons un composant Value-Map. Cliquer avec la touche de droite sur la connexion entre les deux éléments **Title**, et choisir **Insérer Value-Map** depuis le menu contextuel. Configurer les propriétés Value-Map comme indiqué ci-dessous :

🖄 V	/aleur-Propriétés Map			×
Table	Value-Map pour mapper les	valeurs spéci	ìques à d'autres valeurs :	
		×		
	input	Ì	result	1
	string	<b>•</b>	string	•
	Software Engineer		Code Magician	
	Support Engineer		Support Magician	
*>	(nouvelle entrée)			
	Sinon		Valeur:	
			O Valeur d'entrée origin	nale

Conformément à la configuration ci-dessus, chaque occurrence de "Software Engineer" sera remplacée par "Code Magician", et chaque occurrence de "Support Engineer" sera remplacée par "Support Magician". Veuillez noter que la condition **Otherwise** n'est pas encore spécifiée. C'est pour cette raison que Value-Map retourne un *nœud vide* à chaque fois que le titre de poste est différent de "Software Engineer" et de "Support Engineer". Par conséquent, si vous cliquez sur l'onglet **Sortie** et que vous consultez le mappage, certains des éléments **Person** auront un titre manquant **Title**, par exemple :

```
<Person>
<First>Vernon</First>
<Last>Callaby</Last>
</Person>
```

<person></person>
<first>Frank</first>
<last>Further</last>
<person></person>
<first>Michelle</first>
<last>Butler</last>
<title>Code Magician</title>

Comme indiqué auparavant, des nœuds vides entraînent des entrées manquantes dans la sortie générée ; c'est pourquoi, dans le fragment XML ci-dessus, seul le titre de Michelle Butler a été remplacé, étant donné que son titre était présent dans la table de consultation. La configuration créée jusqu'à présent ne remplit toujours pas l'exigence originale. La configuration correcte est la suivante :

🙁 v	/aleur-Propriétés Map		×			
Table	Table Value-Map pour mapper les valeurs spécifiques à d'autres valeurs :					
	Х 🖻 🛱 🗱		×			
	input	🥒 r	esult 🥖			
	string	▼ 5 <sup>4</sup>	tring 🗾			
	Software Engineer	C	ode Magician			
	Support Engineer	S	upport Magician			
*>	(nouvelle entrée)					
	Sinon	(	) Valeur:			
		C	Valeur d'entrée originale			

Avec la configuration ci-dessus, la chose suivante se produit au moment de l'exécution :

- Chaque occurrence de "Software Engineer" sera remplacée par "Code Magician"
- Chaque occurrence de "Support Engineer" sera remplacée par "Support Magician"
- Si le titre original ne se trouve pas dans la table de consultation, le Value-Map le retournera sans le modifier.

À seul titre illustratif, nous pouvons aussi changer tous les titres de poste différents de "Software Engineer" et de "Support Engineer" en une valeur personnalisée, par exemple "N/A". Pour ce faire, définir les propriétés Value-Map comme indiqué ci-dessous :

💙 V	'aleur-Propriétés Map			×
Table	Value-Map pour mapper les	valeurs spécif	ìques à d'autres valeurs :	
冒				×
	input		result	
	string	•	string	•
	Software Engineer		Code Magician	
	Support Engineer		Support Magician	
*>	(nouvelle entrée)			
	Sinon		● Valeur:	
			O Valeur d'entrée originale	

Maintenant, lorsque vous consultez le mappage, chaque titre de poste est présent dans la sortie, mais ceux qui n'avaient pas de correspondance auront la valeur "N/A", par exemple :



Cela complète l'exemple Value-Map. En appliquant la logique ci-dessus, vous pouvez obtenir le résultat désiré dans d'autres mappages.

# 5.8 Regrouper des données

Lorsque votre mappage doit regrouper des nœuds ou des lignes, vous pouvez compter sur les fonctions intégrées de MapForce suivantes :

- group-by
- group-adjacent
- group-into-blocks
- group-starting-with
- group-ending-with

Pour utiliser une de ces fonctions sur le mappage, glissez-les depuis la fenêtre de bibliothèque dans la zone de mappage. Voir aussi <u>Ajouter une fonction au mappage</u><sup>552</sup>.

Note: Les fonctions de regroupement sont disponibles dans les langages suivants : XSLT 2.0, XSLT 3.0, C+ +, C#, Java, Built-In.

Les sections suivantes fournissent des exemples typiques d'utilisation pour des fonctions de regroupement. Ces exemples sont accompagnés par le mappage de démo suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd. Veuillez noter que le mappage de donnée contient plusieurs transformations, une pour chaque fonction. Puisque seule une sortie peut être consultée à la fois, n'oubliez pas de cliquer sur la touche Aperçu applicable pour la transformation désirée avant de cliquer sur l'onglet Sortie.

### group-by

La fonction group-by crée des groupes d'enregistrements conformément à certaines clés de regroupement que vous aurez spécifiées.

	f <sub>x</sub> group-by		
Ç	>nodes/rows	groups 🗘	
Ç	>key	key 🗘	

Par exemple, dans la transformation abstraite illustrée ci-dessous, la clé de regroupement est "Department". Étant donné qu'il y a trois départements au total, l'application de la fonction group-by créera trois groupes :
Department	Name
Administration	Vernon Callaby
Marketing	Susi Sanna
Engineering	Michelle Butler
Engineering	Fred Landis
Administration	Frank Further

group-by()
------------

Department	Name
Administration	Vernon Callaby
Administration	Frank Further

Department	Name
Marketing	Susi Sanna

Department	Name
Engineering	Michelle Butler
Engineering	Fred Landis

Pour plus d'informations, voir la référence à la fonction group-by.689.

#### group-adjacent

La fonction group-adjacent exige une clé de regroupement en tant qu'argument, comme pour la fonction group-by. Contrairement à group-by, cette fonction crée un nouveau groupe à chaque fois que la clé suivante est différente. Si deux enregistrements adjacents ont la même clé, ils seront placés dans le même groupe.

Par exemple, dans la transformation abstraite illustrée ci-dessous, la clé de regroupement est "Department". Le côté gauche du diagramme montre les données d'entrée tandis que le côté droit montre les données de sortie après le regroupement. Les événements suivants se produisent lorsque la transformation est exécutée :

- Tout d'abord, la première clé, "Administration", crée un nouveau groupe.
- La clé suivante est différente, donc un deuxième groupe est créé :"Marketing".
- La troisième clé est aussi différente, donc un groupe supplémentaire est créé : "Engineering".
- La quatrième clé est la même que la troisième, c'est pourquoi cet enregistrement est placé dans le groupe déjà existant.
- Enfin, la cinquième clé est différente de la quatrième et cela crée le dernier groupe.

Comme illustré ci-dessous, "Michelle Butler" et "Fred Landis" ont été regroupés car ils partagent la même clé et sont adjacents. Néanmoins, "Vernon Callaby" et "Frank Further" se trouvent dans des groupes séparés étant donné qu'ils ne sont pas adjacents, même s'ils possèdent la même clé.

Administration Frank Further

			Department	Name
			Administration	Vernon Callaby
Department	Name			
Administration	Vernon Callaby		Department	Name
Marketing	Susi Sanna		Marketing	Susi Sanna
Engineering	Michelle Butler	group-adjacent()		
Engineering	Fred Landis			
Administration	Frank Further		Department	Name
			Engineering	Michelle Butler
			Engineering	Fred Landis
			Department	Name

Pour plus d'informations, voir la référence à la fonction group-adjacent 687.

#### group-into-blocks

La fonction group-into-blocks crée des groupes égaux qui contiennent exactement N items, et où N est la valeur que vous fournissez à l'argument block-size. Veuillez noter que le dernier groupe peut contenir N items ou moins, selon le nombre d'items se trouvant dans la source.

f <sub>x</sub> group-int	o-blocks
hodes/rows	
block-size	groups

Dans l'exemple ci-dessous, block-size est 2. Étant donné qu'il existe au total cinq items, chaque groupe contient exactement deux items, sauf pour le dernier.



Pour plus d'informations, voir la référence à la fonction group-into-blocks 683.

#### group-starting-with

La fonction group-starting-with prend une condition booléenne en tant qu'argument. Si celle-ci est vraie, un nouveau groupe est créé, commençant avec l'enregistrement qui satisfait à la condition.

f <sub>8</sub> group-star	ting-with
nodes/rows	
bool	groups

Dans l'exemple ci-dessous, la condition est que "Key" doit être égal à "heading". Cette condition est vraie pour le premier et le quatrième enregistrement, donc deux groupes sont créés :

			Кеу	Value
Кеу	Value		heading	Intro
heading	Intro	× 1	line	A
line	A		line	в
line	в	group-starting-with()	L	
heading	Body			
line	С		Кеу	Value
		1	heading	Body
			line	C

**Note:** Un groupe supplémentaire est créé si des enregistrements existent après le premier qui satisfait à la condition. Par exemple, s'il existait plus d'enregistrements "line" avant le premier enregistrement "heading", ceux-ci seraient tous placés dans un nouveau groupe.

Pour plus d'informations, voir la référence à la fonction group-starting-with<sup>665</sup>.

#### group-ending-with

La fonction group-ending-with prend une condition booléenne en tant qu'argument. Si la condition booléenne est vraie, un nouveau groupe est créé, terminant avec l'enregistrement qui satisfait la condition.

f <sub>8</sub> group-end	ling-with
nodes/rows	
bool	groups

Dans l'exemple ci-dessous, la condition est que "Key" doit être égal à "trailing". Cette condition est vraie pour les troisième et quatrièmes enregistrements, en résultat, deux groupes sont créés :

			Кеу	Value
Кеу	Value		line	A
line	A	×	line	в
line	В		trailing	Total 1
trailing	Total 1	group-ending-with()		
line	C			
trailing	Total 2		Кеу	Value
-				
			line	C

**Note :** Un groupe supplémentaire est créé si des enregistrements existent après le dernier qui satisfait à la condition. Par exemple, si il existait plus d'enregistrements "line" après le dernier enregistrement "trailing", ceux-ci seraient tous placés dans un nouveau groupe.

Pour plus d'informations, voir la référence à la fonction group-ending-with<sup>691</sup>.

## 5.8.1 Exemple : Regrouper les enregistrements par clé

Cet exemple vous montre comment regrouper des enregistrements avec l'aide de la fonction group-by, de même, il illustre comment agréger des données. Cet exemple est accompagné par un mappage de démonstration disponible sous le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\GroupTemperaturesByYear.mfd. Ce mappage lit les données provenant d'un fichier XML qui contient un journal de températures mensuelles, comme illustré dans le code ci-dessous :

```
<Temperatures>
  <data temp="-3.6" month="2006-01" />
  <data temp="-0.7" month="2006-02" />
  <data temp="7.5" month="2006-03" />
  <data temp="12.4" month="2006-04" />
  <data temp="16.2" month="2006-05" />
  <data temp="19" month="2006-06" />
  <data temp="22.7" month="2006-07" />
  <data temp="23.2" month="2006-08" />
  <data temp="18.7" month="2006-09" />
  <data temp="11.2" month="2006-10" />
  <data temp="9.1" month="2006-11" />
  <data temp="0.8" month="2006-12" />
  <data temp="-3.2" month="2007-01" />
  <data temp="-0.3" month="2007-02" />
  <data temp="6.5" month="2007-03" />
  <data temp="10.6" month="2007-04" />
  <data temp="19" month="2007-05" />
  <data temp="20.3" month="2007-06" />
  <data temp="22.3" month="2007-07" />
   <data temp="20.7" month="2007-08" />
   <data temp="19.2" month="2007-09" />
```

```
<data temp="12.9" month="2007-10" />
        <data temp="8.1" month="2007-11" />
        <data temp="1.9" month="2007-12" />
</Temperatures>
```

Les exigences commerciales de ce mappage sont doubles :

- 1. Regrouper les températures de chaque année.
- 2. Trouver le minimum, le maximum, et la température moyenne de chaque année.

Afin d'atteindre le premier objectif, le mappage appelle la fonction <u>group-by</u><sup>663</sup>. Afin d'atteindre le second objectif, il appelle les fonctions d'agrégation <u>min</u><sup>637</sup>, <u>max</u><sup>666</sup>, et <u>avg</u><sup>634</sup>. Toutes ces fonctions sont des fonctions intégrées MapForce, et vous pouvez les ajouter à tout mappage en la glissant depuis la fenêtre Bibliothèques, voir <u>Comment ajouter une fonction dans le mappage</u><sup>552</sup>.



GroupTemperaturesByYear.mfd

Pour exécuter un mappage, MapForce (et il s'agit là de la méthode recommandée pour commencer à lire un mappage) consulter l'item supérieur du composant de cible. Dans cet exemple, un item **YearlyStats** sera créé pour chaque groupe retourné par la fonction group-by. La fonction group-by prend en tant que premier argument tous les items **data** depuis la source et les regroupe selon ce qui est connecté à l'entrée **key**. Puisque l'exigence est de regrouper des températures par année, il faut tout d'abord obtenir l'année. Pour ce faire, la fonction <u>substring-before</u> extrait la partie de l'année depuis l'attribut **month** de chaque élément **data**. Concrètement, il prend en tant qu'argument la valeur de **month** et retourne la partie avant la première apparition de **substr**. Comme illustré ci-dessus, dans cet exemple, **substr** est défini pour le caractère de tiret ; ainsi, pour une valeur "2006-01", la fonction retournera "2006".

Enfin, les valeurs de **MinimumTemp**, **MaximumTemp** et **AverageTemp** sont obtenues en connectant ces items avec les fonctions d'agrégat respectives : min, max et avg. Les trois fonctions prennent en tant qu'entrée la séquence des températures lues depuis le composant de source. Ces fonctions ne nécessitent pas d'argument **parent-context**, parce qu'elles fonctionnent déjà dans le contexte de chaque groupe. En d'autres

mots, il existe une connexion parent, de **data** à **YearlyStats** qui fournit le contexte pour chaque fonction d'agrégation sur laquelle vous souhaitez travailler.

Pour consulter la sortie de mappage, cliquez sur l'onglet **Sortie**. Veuillez noter que le nombre de groupes doit coïncider avec le nombre d'années obtenues en lisant le fichier de source, par exemple :

```
<Temperatures>

<YearlyStats Year="2006">

<MinimumTemp>-3.6</MinimumTemp>

<MaximumTemp>23.2</MaximumTemp>

<AverageTemp>11.375</AverageTemp>

</YearlyStats>

<YearlyStats Year="2007">

<MinimumTemp>-3.2</MinimumTemp>

<AverageTemp>11.5</AverageTemp>

</YearlyStats>

</YearlyStats>
```

**Note :** Pour une plus grande simplicité, les listes de code ci-dessus contiennent moins de données que l'entrée et la sortie véritables utilisés dans le mappage de démonstration.

# 5.9 Exceptions

Une exception est un type de composant spécial qui vous permet d'arrêter le processus de mappage et d'afficher une erreur lorsqu'une condition retournée par un filtre se produit. Vous pouvez ajouter une exception lorsque votre mappage inclut un filtre qui vérifie une condition vrai/faux (voir <u>Filtres et Conditions</u>). Par exemple, vous pouvez vouloir lancer une exception si la valeur de certains item de mappage est supérieure à un seuil personnalisé.

#### Pour ajouter une exception dans le mappage :

- 1. Dans le menu Insérer, cliquer sur Exception.
- 2. Cliquer sur la touche de barre d'outils Insérer Exception ( 🖤 ).
- 3. Connecter l'entrée throw de l'exception soit dans une sortie on-true ou on-false d'un filtre.
- 4. En option, connecter l'entrée **error-text** de l'exception à un autre composant (généralement, une constante) qui fournit le texte de l'erreur lorsque cette exception est lancée.
- Note : Les deux sorties on-true et on-false du filtre doit être connecté. Spécifiquement, une de ces sorties doit être connectée directement à l'exception (sans fonctions ou composants intermédiaires). L'autre sortie doit être connectée au composant de cible, soit directement ou par le biais des composants intermédiaires.

Lorsque le mappage rencontre une exception, vous serez prévenus comme ci-dessous :

• Dans MapForce, la fenêtre Messages affiche une erreur, et le texte d'exception (dans ce cas, "Expense limit exceeded" (La limite de dépense est dépassée)).



Si la langue de mappage est XSLT 2.0 ou XQuery, une erreur "Execution failed" (Échec d'exécution) apparaît dans la fenêtre Messages, et l'onglet XSLT2 ou XQuery respectif est ouvert. La ligne d'erreur est souligné dans la fenêtre Messages.

 Si vous exécutez le mappage avec MapForce Server, l'erreur "Exception was thrown!" (L'exception a été lancée !) est retournée, suivie par le texte d'exception personnalisé que vous avez défini dans MapForce.

c:\codegen\mfx\ExpenseLimit>"C:\Program Files (x86)\Altova\MapForceServer2016\bi n\MapForceServer.exe" run ExpenseLimit.mfx Exception was thrown! Expense limit exceeded!

• Si vous exécutez le mappage depuis le code C#, C++ ou Java généré, l'erreur "USER EXCEPTION" est retournée, suivie par le texte d'exception personnalisé que vous avez défini dans MapForce.

C:\codegen\cs\ExpenseLimit\Mapping\bin\Debug>Mapping.exe Mapping Application USER EXCEPTION: Expense limit exceeded!

# 5.9.1 Exemple : Exception sur condition "Greater Than" (Supérieur à)

Cet exemple illustre un mappage qui lance une exception lorsqu'une condition "Greater Than" (Supérieur à) se produit. Le mappage échantillon qui accompagne cet exemple peut être consulté sous : <br/>
<Documents>\Altova\MapForce2023\MapForceExamples\ExpenseLimit.mfd.



Ce mappage lance une exception à chaque fois que l'item **expense** dans l'instance XML de source a une valeur supérieure à 200. La valeur "200" est fournie par une constante. La fonction **less** est utilisée pour comparer les deux valeurs. Si la valeur de **expense** est inférieure à 200, alors son parent, l'**expense-item**, est transmis sur le filtre, et aucune exception n'est lancée. Sinon, une exception est lancée, avec le texte personnalisé "Expense limit exceed" (La limite de dépense est dépassée).

Comme affiché ci-dessus, l'exception est identifiée par l'icône <sup>(1)</sup> et elle consiste en deux items : **throw** et **error-text**. L'item **throw** doit être connecté dans la sortie **on-false** ou **on-true** d'un filtre. L'**error-text** est connecté à une constante qui fournit le texte personnalisé de l'exception.

Surtout, les deux sorties du filtre sont connectées ; sinon, l'exception ne sera pas lancé. Dans cet exemple particulier, la sortie **on-false** est connectée à l'exception, alors que la sortie **on-true** est connectée au composant cible.

# 5.9.2 Exemple : Exception lorsque le nœud n'existe pas

Cet exemple illustrate comment lancer une exception lorsqu'un nœud dans le schéma XML de source n'existe pas. Pour des raisons de simplicité, cet exemple utilise le même schéma XML aussi bien en tant que composant source et cible.

#### Pour ajouter le schéma de source dans le mappage :

- 1. Dans le menu Insérer, cliquez sur Schéma XML/Fichier, et cherchez <br/>
  Cocuments>\Altova\MapForce2023\MapForceExamples\ BookList.xsd.
- 2. Lorsque vous serez invité à fournir un fichier d'instance, cliquez sur Sauter.
- 3. Lorsque vous serez invité à sélectionner un élément de racine de schéma, sélectionnez **BookList** en tant qu'élément de racine.

Pour ajouter le schéma de cible, suivre les mêmes étapes. Ensuite, en utilisant les commandes correspondantes depuis le menu **Insérer** (ou les touches de barre d'outils correspondantes), ajoutez les éléments suivants :

- Un composant Filtre : Nœuds/Lignes (voir aussi <u>Filtres et conditions</u><sup>523</sup>)
- Une constante avec le texte "No year defined!" (Aucune année définie !)
- Une exception

Enfin, tirer la fonction **exists** depuis la fenêtre **Bibliothèques** dans la zone de mappage, et établir les connexions comme illustré ci-dessous.



Conformément au schéma XML, tous les attributs de l'élément **Book** sont optionnels, sauf le titre du livre. C'est pourquoi, l'attribut "Year" peut ou pet ne pas exister dans une instance XML valide. L'objectif du mappage est de traiter avec succès une instance XML où l'attribut "Year" existe pour chaque livre. Sinon, le mappage doit lancer une exception.

#### Pour tester l'exécution réussie du mappage :

- 1. Double-cliquer sur l'en-tête du composant de source et, à côté de **Fichier XML d'entrée**, chercher le fichier suivant : <Documents>\Altova\MapForce2023\MapForceExamples\ BookList.xml.
- 2. Cliquer sur la touche **Sortie** pour exécuter le mappage.

#### Pour tester l'exception :

- 1. Créer, dans le même répertoire, une copie du fichier **BookList.xml** appelée **BookListInvalid.xml**.
- 2. La modifier de manière à supprimer l'attribut "Year" d'un élément Book.
- 3. Double-cliquer sur l'en-tête du composant de source, et, à côté du **fichier XML d'entrée**, chercher le fichier **BookListInvalid.xml**.
- 4. Cliquer sur la touche Sortie pour exécuter le mappage.

Examinons à présent de plus près le fonctionnement d'un mappage.

La connexion **A** veille à ce qu'un livre soit créé dans l'instance cible pour chaque livre dans l'instance de source. Les connexions **B**, **C**, **D**, **E** garantissent que "Title", "Year", "Price" et "Author" sont copiés depuis la source vers la cible, pour chaque livre.

La connexion **F** déclenche la fonction **exists** pour contrôler l'existence de l'attribut "Year". La connexion **G** fait passer le résultat de fonction (true ou false) dans le filtre. Si le résultat est true, l'attribut "Year" existe, et le livre passe dans le filtre, et ensuite dans la cible par le biais de la connexion **H**.

Veuillez noter que le filtre n'a pas été connecté directement à la sortie **Year** du composant de source. Si nous l'avions connecté de cette manière, le filtre filtrerait **Year** du fait de sa propre existence, ce qui n'est pas significatif, et l'exception ne serait jamais lancée.

La connexion I est là parce que l'exception doit être connectée soit à une sortie **on-false** ou **on-true** d'un filtre, conformément aux règles. Enfin, la connexion **K** fait passer le texte d'erreur personnalisé de la constante au composant d'exception.

# 6 Fonctions

Dans MapForce, vous pouvez utiliser les catégories de fonction suivantes pour transformer des données conformément à vos besoins :

- Fonctions intégrées MapForce ces fonctions sont prédéfinies dans MapForce et vous pouvez les utiliser dans vos mappage pour effectuer une gamme très étendue de tâches de traitement impliquant des strings, des nombres, des dates, et d'autres types de données. Vous pouvez aussi les utiliser pour effectuer des regroupements, des agrégations, des numérotations automatiques ainsi que d'autres tâches. Pour consulter une référence à toutes les fonctions intégrées disponibles, voir Référence des bibliothèques de fonctions<sup>[631]</sup>.
- Fonctions et défauts de nœuds il s'agit de fonctions plus spécialisées qui vous permettent de créer et d'appliquer une logique de traitement personnalisée à un ou plusieurs nœuds descendants dans un composant de mappage. Elles permettent de traiter soit *avant* d'atteindre un nœud d'une structure de mappage, ou *immédiatement après* qu'elle quitte un nœud. Pour plus de détails, voir Fonctions par défaut et de nœud <sup>660</sup>.
- Fonctions définies par l'utilisateur (FDU) il s'agit de fonctions MapForce que vous pouvez créer vous-même, en utilisant en tant que base les types de composants natifs et les fonctions intégrées déjà disponibles dans MapForce, voir <u>Fonctions définies par l'utilisateur</u><sup>575</sup>.
- Fonctions personnalisées il s'agit des fonctions que vous pouvez importer depuis des sources externes comme des bibliothèques XSLT, des modules de bibliothèque XQuery, des fichiers .class Java, des fichiers .NET .dll, et les adapter à MapForce. Veuillez noter, afin de pouvoir les réutiliser dans MapForce, que vos fonctions personnalisées doivent retourner des données de type simple (comme des string ou des entiers) et elles doivent aussi prendre des paramètres de type simple. Pour plus d'informations, voir Importer des fonctions XSLT personnalisées <sup>602</sup>, Importer des fonctions XQuery 1.0 personnalisées <sup>604</sup>.
- Note : Vous pouvez importer des bibliothèques externes de fonctions soit directement (aucune configuration nécessaire) soit en configurant un MFF (MapForce Function File) reconnu par MapForce. Si vous utilisez la deuxième possibilité, vous pouvez aussi importer des bibliothèques C++, en plus des classes Java et des .NET assemblies. Veuillez noter que les bibliothèques importées via des fichiers .mff doivent remplir les conditions préalables mentionnées dans <u>Référencer les bibliothèques Java, C# et C++ manuellement</u><sup>G12</sup>.

# 6.1 Notions fondamentales de base

Les sous-sections ci-dessous donnent un aperçu des actions liées à la fonction de base. Les fonctions que vous voyez dans la fenêtre **Bibliothèques** dépend du langage de transformation que vous avez sélectionné. Pour plus d'information, voir <u>Langages de transformation</u><sup>25</sup>.

## Ajouter une fonction

MapForce inclut un grand nombre de fonctions built-in que vous pouvez ajouter au mappage. Pour plus d'informations concernant toutes les fonctions intégrées disponibles, voir <u>Référence des bibliothèques de fonctions</u><sup>(631)</sup>. Pour ajouter une fonction au mappage, vous pouvez choisir une des méthodes suivantes :

- Appuyez et gardez appuyer la fonction requise dans la fenêtre **Bibliothèques** glissez-la dans la zone de mappage. Pour filtrer les fonctions par leur nom, commencez à saisir le nom de la fonction dans le champ de texte situé en bas de la fenêtre.
- Double-cliquez où vous voulez dans la zone vide du mappage et commencez à saisir le nom de la fonction (*voir la capture d'écran ci-dessous*). Pour voir une info-bulle avec plus de détails concernant une fonction, choisir une fonction parmi la liste. Pour ajouter une fonction à votre mappage, double-cliquez sur la fonction pertinente dans la zone de liste déroulante.

con
core.concat core.contains xpath2.seconds-from-dateTime xpath2.seconds-from-duration xpath2.seconds-from-time

#### <u>Ajouter une UDF</u>

Vous pouvez aussi ajouter des fonctions définies par l'utilisateur (UDF) à votre mappage en utilisant les mêmes approches que décrites ci-dessous si (i) une UDF a déjà été créée dans le même mappage ou (ii) si vous avez importé le mappage qui contient une UDF comme bibliothèque locale ou globale.

### Ajouter une constante

Les constantes vous permettent de fournir du texte personnalisé et les nombres au mappage. Pour ajouter une constante au mappage, vous pouvez choisir une des options suivantes :

- Cliquez avec la touche de droite sur la zone de mappage vide et sélectionnez **Insérer constante** depuis le menu contextuel. Saisissez la valeur et sélectionnez un des types de données : *string, nombre* ou *tous les autres*.
- Sélectionnez la commande de menu **Insérer | Constante**. Saisissez la valeur et sélectionnez un des types de données : *string, nombre* ou *tous les autres*.
- Cliquez sur la commande **Constante** de la barre d'outils. Saisissez la valeur et sélectionnez un des types de données : *string*, *nombre* ou *tous les autres*.
- Double-cliquer où vous voulez dans une zone de mappage vide. Saisissez les doubles guillemets anglais suivis de la valeur constante. Le double guillemet anglais de fermeture est optionnel. Pour ajouter une constante numérique, il suffit de saisir le nombre.

## Chercher une fonction

Pour rechercher une fonction dans la fenêtre **Bibliothèques**, commencez à saisir le nom de la fonction dans le champ de texte en bas de la fenêtre. Par défaut, MapForce effectue la recherche par le biais du nom de fonction et du texte de description. Si vous voulez exclure la description de fonction de la recherche, cliquez sur la flèche déroulante vers le bas et désactivez l'option Recherche dans les descriptions de fonction. Pour annuler la recherche, appuyer sur la touche Esc ou cliquez sur X.

Pour trouver toutes les occurrences d'une fonction dans le mappage actif actuellement, cliquez avec la touche de droite sur le nom de la fonction dans la fenêtre Bibliothèques et choisissez Trouver tous les appels depuis le menu contextuel. Les résultats de recherche sont affichés dans la fenêtre Messages.

### Consulter un type et une description de fonction

Pour consulter le type de données d'un argument d'entrée/de sortie, déplacez votre souris au-dessus de la partie argument d'une fonction (voir la capture d'écran ci-dessous). Veuillez vous assurer que la touche de la barre d'outils 💯

(Afficher astuces est activée.



Pour afficher la description d'une fonction, déplacez le curseur au-dessus de l'en-tête de la fonction (voir la

capture d'écran ci-dessous). Veuillez vous assurer que la touche de la barre d'outils 🕮 (Afficher astuces est activée.

Result is	true if a is equal b, otherwise false.	
💃 equal		
[>a [>b [>b		

### Ajouter/supprimer les arguments de fonction

Pour quelques fonctions intégrées de MapForce, il est possible d'ajouter autant de paramètres dont vous avez besoin pour vos objectifs de mappage. Un de ces exemples est la fonction concat<sup>709</sup>. Pour ajouter ou supprimer les arguments de fonction (pour les fonctions qui les prennent en charge), cliquez sur Ajouter paramètre ( •) ou Supprimer paramètre ( 🗵 ) à côté du paramètre que vous voulez ajouter ou supprimer respectivement (voir ci-dessous). Le fait de déposer une connexion dans le symbole et ajoute le paramètre et le connecte.



# 6.2 Gérer les bibliothèques de fonction

Dans MapForce, vous pouvez importer et utiliser les types de bibliothèques suivants dans un mappage :

- Tout fichier design de mappage (\*.mfd) qui contient les fonctions définies par l'utilisateur (UDF). Cela réfère spécifiquement aux fichiers de mappage qui contient des Fonctions définies par l'utilisateur (UDF) créés avec MapForce, en utilisant les fonctions intégrées MapForce et les composants en tant que blocs de construction. Pour plus d'information, voir <u>Créer des fonctions définies par l'utilisateur</u> <sup>(576)</sup>.
- Les fichiers XSLT personnalisés qui contiennent des fonctions. Cela se réfère à des fonctions XSLT écrites en dehors de MapForce qui se qualifient pour l'importation dans MapForce comme décrit dans <u>Importer des fonctions personnalisées XSLT 1.0 ou 2.0</u><sup>592</sup>.
- Les fichiers XQuery 1.0 personnalisés qui contiennent des fonctions. Cela réfère aux fonctions XQuery écrites en dehors de MapForce qui qualifient pour l'importation dans MapForce comme décrit dans <u>Importer des fonctions personnalisées XQuery 1.0</u><sup>539</sup>.
- Les fichiers Java .class et bibliothèques .NET .dll qui qualifient pour l'importation dans MapForce tel que décrit dans <u>Importer des bibliothèques Java et .NET personnalisées</u><sup>604</sup>.
- Note : Vous pouvez importer des bibliothèques externes de fonctions soit directement (aucune configuration nécessaire) soit en configurant un MFF (MapForce Function File) reconnu par MapForce. Si vous utilisez la deuxième possibilité, vous pouvez aussi importer des bibliothèques C++, en plus des classes Java et des .NET assemblies. Veuillez noter que les bibliothèques importées via des fichiers .mff doivent remplir les conditions préalables mentionnées dans <u>Référencer les bibliothèques Java, C#</u> et C++ manuellement <sup>G12</sup>.

### Gérer la fenêtre Bibliothèques

Vous pouvez consulter et gérer toutes les bibliothèques utilisées par un fichier de mappage provenant de la fenêtre Gérer des bibliothèques. Cela comprend les UDF et des bibliothèques personnalisées.

Par défaut, la fenêtre Gérer des bibliothèques n'est pas visible. Pour l'afficher, choisir une des deux options suivantes :

- Dans le menu View, cliquer sur Gérer Bibliothèques.
- Cliquer Ajouter/Supprimer Bibliothèques en bas de la fenêtre Bibliothèques.

Manage Libraries	x
🔁 🛃 FormatNumber.mfd	*
User-Defined Functions	
Own Library Imports Add	
🛱 😢 Global Library Imports 🛛 Add	
🖓 Format.dll 🔀	
🔁 Functions	
f <sub>x</sub> Format, Version=0.0.0.0, Culture=neutral, PublicKey1	
f <sub>8</sub> Format, Version=0.0.0.0, Culture=neutral, PublicKey1	
Own Library Imports	

Vous pouvez choisir de consulter des UDF et des bibliothèques uniquement pour le document de mappage (fichier .mfd) qui est activé actuellement, ou pour tous les mappages ouverts. Pour consulter des fonctions et des bibliothèques importées pour tous les documents de mappages ouverts actuellement, cliquer avec la

touche de droite dans la fenêtre et sélectionner Afficher les documents ouverts depuis le menu contextuel.

Pour afficher le chemin du document de mappage ouvert au lieu du nom, cliquer avec la touche de droite dans la fenêtre et sélectionner **Afficher les chemins de fichier** depuis le menu contextuel.

Les données affichées dans la fenêtre Gérer Bibliothèques sont organisées en tant que hiérarchie arborescente comme suit :

- Tous les documents de mappage ouverts actuellement sont affichés en tant qu'entrées de niveau supérieur. Chaque entrée a deux branches: Fonctions définies par l'utilisateur et Importations Bibliothèques propres.
  - La branche **Fonctions définies par l'utilisateur** affiche des UDF contenues dans ce document.
  - La branche Importations Bibliothèques propres affiche des bibliothèques importées localement dans le document du mappage actuel. Le terme "bibliothèques" signifie d'autres documents de mappage (fichiers .mfd qui contiennent des fonctions définies par l'utilisateur) ou des bibliothèques externes personnalisées écrites en XSLT 1,0 XSLT 2.0, XQuery 1.0\*, Java\*, C#\*, ou des fichiers .mff mentionnés précédemment. Notez que la structure Importations de bibliothèque propre pourrait se trouver à plusieurs niveaux profonds, puisque tout document de mappage peut importer tout autre document de mappage en tant que bibliothèque.
- La saisie **Global Library Imports** encadre les bibliothèques personnalisées que vous avez importées *globalement* au niveau de l'application. Là aussi, dans le cas de fichiers .mfd, la structure pourrait contenir plusieurs niveaux profonds pour les raisons mentionnées ci-dessus.
- \* Ces langages sont pris en charge uniquement dans les éditions Professional ou Enterprise de MapForce.
- **Note :** Les bibliothèques XSLT, XQuery, C# et Java peuvent avoir leur propres dépendances. Ces dépendances ne sont pas affichées dans la fenêtre Bibliothèques.

### Commandes de menu contextuel

Vous pouvez effectuer rapidement plusieurs opérations par rapport à des objets dans la fenêtre Gérer des bibliothèques en cliquant avec la touche de droite sur un objet et en sélectionnant une des options de menu contextuel suivantes :

Commande	Description	Applicable pour		
Ouvrir	Ouvre le mappage.	Mappages		
Ajouter	Ouvre un dialogue dans lequel vous pouvez chercher une bibliothèque personnalisée des fonctions.			
Localiser une fonction dans la fenêtre Bibliothèques	Change le focus sur la fenêtre Bibliothèques et choisir la fonction.	Fonctions		
Couper, Copier, Supprimer	Ces commandes standard de Windows sont uniquement applicables aux fonctions définies par l'utilisateur de MapForce. Vous ne pouvez pas copier-coller les fonctions de fichiers XSLT externes ou d'autres genres de bibliothèques.	Fonctions définies par l'utilisateur		

Commande	Description	Applicable pour
Coller	Vous permet de coller une fonction définie par l'utilisateur qui a été copiée précédemment dans le presse-papiers à l'intérieur de la bibliothèque actuelle.	Bibliothèques (UDF)
Options	Ouvre une boîte de dialogue où vous pouvez définir ou changer des options pour la bibliothèque actuelle.	Bibliothèques
Afficher tous les Documents ouverts	Lorsque cette option est activée, la fenêtre Gérer les Bibliothèques affichera tous les mappages ouverts actuellement. Cela est généralement utile si vous souhaitez copier- coller des fonctions entre les mappages. Sinon, seul le mappage actuellement en cours est affiché.	Toujours
Afficher les chemins de fichier	Lorsque cette option est activée, les objets contenus dans la fenêtre Gérer les Bibliothèques sont affichés avec leur chemin complet. Sinon, seul le nom de l'objet est affiché.	Toujours

# 6.2.1 Bibliothèques locales et globales

Vous pouvez importer des bibliothèques *localement* ou *globalement*. Les importations globales ont lieu au niveau de l'application. Si une bibliothèque a été importée globalement, vous pouvez utiliser ses fonctions depuis n'importe quel mappage.

Les importations locales ont lieu au niveau du fichier du mappage. Par exemple, supposons que vous êtes en train de travailler sur le mappage **A.mfd**, et que vous décidez d'importer toutes les fonctions définies par l'utilisateur du mappage **B.mfd**. Dans ce cas, le mappage **B.mfd** est considéré être importé en tant que bibliothèque locale dans **A.mfd** et vous pouvez utiliser les fonctions provenant de **B.mfd** dans **A.mfd** également. De même, si vous importez des fonctions depuis un fichier XSLT dans **A.mfd**, il s'agit également d'une importation locale.

Vous pouvez consulter et gérer toutes les importations locales et globales depuis la fenêtre Gérer des bibliothèques. Pour importer des bibliothèques, suivre une des étapes suivantes :

Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u>
 La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques ×		
🗆 🧱 New Design1	*	
Fonctions définies par l'utilisateur		
Importations Bibliothèque propre Ajouter		
Simportations Bibliothèques globales Ajouter		

 Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
 Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.

#### Noms de fonction conflictuels

Vous pouvez être confronté à des situation dans lesquelles le même nom de fonction est défini dans un des niveaux suivants :

- dans le mappage principal
- dans une bibliothèque qui a été importée localement
- dans une bibliothèque qui a été importée globalement

Lorsque ce genre de cas survient, MapForce tentera d'appeler la fonction exactement dans l'ordre ci-dessus, pour éviter toute ambiguïté. C'est à dire, la fonction définie directement dans le mappage prend précédence si le même nom de fonction existe dans une bibliothèque importée localement. De même, la fonction importée localement prend précédence sur la fonction importée globalement (en admettant que les deux fonctions portent le même nom).

S'il existe plusieurs fonctions portant le même nom, seule la fonction "vainqueur" sera appelée, conformément à la règle ci-dessus ; tout autre nom de fonction ambigu sera bloqué. Ces fonctions bloquées apparaissent grisées dans la fenêtre Bibliothèques, et il n'est pas possible de les utiliser dans le mappage.

# 6.2.2 Chemins de bibliothèque relatifs

Vous pouvez définir le chemin de tout fichier de bibliothèque importé pour qu'il soit relatif au fichier de design du mappage (fichier .mfd), si la bibliothèque a été importée localement (pas globalement), tel que décrit dans <u>Bibliothèque Locales et Globales</u><sup>557</sup>.

La configuration d'un chemin de bibliothèque relatif est uniquement applicable pour les bibliothèques qui ont été importées *localement* au niveau du document. Si un mappage a été importé *globalement* au niveau du programme, son chemin est toujours absolu.

#### Pour définir un chemin de bibliothèque comme relatif par rapport au fichier de design du mappage :

- 1. Cliquer sur **Ajouter/Supprimer des bibliothèques** en bas de la fenêtre Bibliothèques. La <u>fenêtre de</u> <u>gestion des bibliothèques</u> <sup>33</sup> s'ouvre.
- 2. Cliquer sur **Options** à côté de la bibliothèque qui vous intéresse. (En alternative, cliquer avec la touche de droite sur la bibliothèque, et sélectionner **Options** depuis le menu contextuel.)

🙁 Library Im	nport Options	×
File:	C:\Libraries\MyCustomLibrary.xslt	
∑ <u>S</u> ave file pa	ath relative to MFD file	
	OK	Cancel

- 3. Cocher la case à cocher Enregistrer chemin de fichier en tant que relatif au fichier MFD.
- **Note :** si la case à cocher est grisée, veuillez vous assurer que la bibliothèque a bien été importée localement et pas globalement.

Si la case à cocher est cochée, MapForce tentera de suivre et de mettre à jour le chemin à tout fichier de bibliothèque référencé lorsque vous enregistrez le mappage fichier dans un nouveau répertoire en utilisant la commande de menu **Enregistrer sous**. De même, si les fichiers de bibliothèque se trouvent dans le même répertoire que le fichier de mappage, le chemin de référence ne sera pas cassé si vous déplacez tout le répertoire vers un autre emplacement sur le disque, voir aussi <u>Utiliser des chemins relatifs sur un</u> composant<sup>®3</sup>.

Veuillez noter que la case à cocher **Enregistrer chemin de fichier comme relatif sous le fichier MFD** spécifie que des chemins sont *relatifs au fichier de mappage*, et qu'elle n'affecte pas les chemins dans le code généré. Pour plus d'informations concernant la manière dont les références de bibliothèque sont gérées dans le code généré, voir <u>Chemins dans des environnement d'exécution divers</u><sup>(65)</sup>.

# 6.3 Valeurs par défaut et fonctions de nœud

Les fonctions par défaut et les fonctions de nœud sont particulièrement utiles lorsque vous souhaitez appliquer la même logique de traitement à plusieurs items descendants dans une structure, par exemple : Normalement, pour ce faire, vous devriez copier-coller la même fonction plusieurs fois sur le mappage. Néanmoins, cela encombrerait le mappage et le rendrait plus compliqué à lire. Les fonctions par défaut et fonctions de nœud peuvent être appliqués à un seul item ou à plusieurs items à la fois. Les défauts remplacent les séquences vides. Si la connexion transporte une valeur, le défaut sera ignoré.

Les défauts et fonctions de nœud sont idéales pour presque la plupart des composants qui ont une arborescence avec des nœuds (par ex., XML, EDI, composants Join, variables).

Les défauts et fonctions de nœud sont compatibles uniquement avec le langage de transformation Built-In. L'exécution de tels mappages provenant du code de programme C#, C++, Java, ou avec les transformations XSLT/XQuery générées n'est pas prise en charge. Du côté du serveur, vous pouvez exécuter ces mappages avec MapForce Server Advanced Edition.

#### Les avantages de fonctions de nœud et les défauts

Créer des fonctions de nœud et des défauts signifient définir une *règle*. Les règles ont les caractéristiques importantes suivantes qui les rendent particulièrement flexibles et faciles à utiliser :

- Héritage. Lorsque vous définissez une règle sur un item qui a des descendants, la règle sera héritée par des descendants par défaut sauf si vous avez choisi de désactiver cette option. Si l'item dans lequel vous définissez la fonction est doté de plusieurs niveaux d'items enfants imbriqués sous lui, vous pouvez choisir d'appliquer la règle uniquement aux items d'enfant directs, ou à tous les items descendants.
- *Filtrer* MapForce applique les règles de manière conditionnelle, basées sur le type de données de chaque item. Cela permet par exemple, d'appliquer certaines valeurs par défaut ou une fonction pour tous les items de type string, et un défaut différent ou une fonction pour tous les items de type décimal Pour les détails, voir le *Scénario 2* dans <u>Scénarios Use-Case</u><sup>666</sup>. Vous pouvez également définir des options de filtrage plus avancées : Par exemple, vous pouvez spécifier un type de données à laquelle votre fonction doit correspondre (ceci pourrait être une catégorie de types de données tels que numérique), puis filtrer les nœuds de ce type de données basé sur le nom de nœud ou le type de nœud (par ex., integer). Pour plus de détails, voir le <u>Scénario 5</u><sup>669</sup>.

Par exemple, vous pouvez demander à MapForce de faire la chose suivante :

- Chaque fois qu'une valeur vide ou nulle survient, la remplacer avec une autre valeur et le faire récursivement pour tous les items descendants.
- Chaque fois qu'une valeur spécifique survient, la remplacer avec une autre valeur (ou avec un string vide), et le faire récursivement pour tous les items descendants.
- Remplacer toutes les valeurs null de la base de données avec des strings vides et du texte personnalisé.
- Ajouter un préfixe ou un suffixe à toutes les valeurs qui sont rédigées dans un fichier cible ou une base de données

#### Côté sortie vs. entrée

Vous pouvez définir des fonctions de nœud et des défauts du côté entrée, du côté sortie, ou des deux côtés d'un composant, dépendant de vos besoins. Dans MapForce, un mappage fonctionne de la manière suivante :

(i) d'abord, il lit les données du composant source (par ex., un fichier XML), (ii) puis, en option, il les traite d'une manière (par ex., en utilisant une fonction), et (iii) finalement, il écrit les données dans un composant cible (par ex., une base de données). En tenant compte de ce principe de base, vous pouvez définir les fonctions de nœud et les défauts à des niveaux différents :

 Immédiatement après la lecture des données depuis la source, mais avant qu'elles ne soient traitées par votre mappage, ce qui signifie que la fonction/le défaut a été défini du côté de la sortie du composant source (voir l'exemple ci-dessous).



 Immédiatement avant l'écriture des données au composant cible (et une fois tous les traitements intermédiaires terminés), ce qui signifie que la . fonction/le défaut a été défini du côté de l'entrée du composant cible (voir l'exemple ci-dessous).



• À un stage intermédiaire du processus de mappage. Par exemple, si le mappage contient une variable intermédiaire de type complexe (par ex., une structure XML), vous pourriez réduire toutes les valeurs avant qu'elles soient fournies à la structure XML, ou juste après qu'elles soient retournées par la structure XML (*voir l'exemple ci-dessous*).



• Du côté sortie d'un composant de source et du côté entrée du composant cible. Dans l'exemple cidessous, un défaut a été défini pour tous les nœuds de type string dans le composant ArticlesOfClothing. Dans le composant ArticlesInfo, nous avons défini une fonction de nœud qui transformera toutes les valeurs de nœud e string en caractères majuscules.



### Dans cette section

Cette section explique comment configurer une règle, décrit les scénarios de vie réelle dans lesquels les défauts et fonctions de nœud peuvent être utiles, et montre comment ajouter des métadonnées de nœud aux fonctions de nœud. La section est organisée en rubriques suivantes :

- <u>Configuration de la règle</u>
   <sup>562</sup>
- Scénarios Use-Case
- Métadonnées de nœud dans les fonctions de nœud<sup>571</sup>

# 6.3.1 Configuration de la règle

Vous pouvez créer des fonctions de nœud et des défauts pour presque chaque item dans le mappage. Créer des fonctions de nœud et des défauts signifient définir une *règle*. Afin de créer une règle, sélectionnez un item (nœud ou champ) dans lequel vous souhaitez définir une règle. Il peut s'agir d'un seul nœud ou d'un nœud avec des nœuds enfant. Si vous choisissez de créer une règle pour un nœud avec enfants, la règle s'appliquera à tous les enfants sauf si vous désactivez explicitement cette option.

### Aspects importants de défauts et fonctions de nœud

Les défauts et fonctions de nœud ont les aspects importants suivants :

- Vous pouvez créer des défauts ou des fonctions de nœud soit du côté d'entrée d'un composant cible, soit du côté sortie d'un composant source. Afin d'établir quel côté est celui qui convient à vos besoins, voir <u>Choisir le côté d'entrée ou de sortie</u><sup>660</sup>.
- Lorsque plusieurs règles existent pour un seul et même item, MapForce appliquera une règle à un item qui est plus proche de cet item. Pour savoir comment écraser les règles, voir le *Scénario 4* dans <u>Scénarios Use-Case</u><sup>569</sup>.
- Les défauts et les fonctions de nœud peuvent être créés si le type de connexion entre la source et la cible est Orienté vers la source ou Orienté vers la cible. Toutefois, les connexions copier-tout ne sont pas prises en charge. En particulier, les fonctions de nœud et les défauts ne sont pas appliquées aux descendants des connexions copier-tout. Le nœud parent avec une connexion copier-tout peut avoir des fonctions de nœud et des défauts, mais uniquement s'il a une valeur simple, par exemple, un

élément XML avec un contenu de type simple et des attributs. Pour plus d'informations sur les types de connexion, voir <u>Types de connexion</u><sup>(9)</sup>.

- Créer des fonctions par défaut et fonctions de nœud n'est pas prise en charge pour le nœud File.
- À l'intérieur d'une fonction de nœud, seuls certains composants importants dans ce contexte sont pris en charge (par ex., des fonctions built-in, des variables, des conditions if-else). Les structures complexes telles que XML, JSON, EDI ou des bases de données ne sont pas prises en charge. Ajouter des fonctions définies par l'utilisateur inline et des composants join à une fonction de nœud n'est pas pris en charge.
- Une fonction de nœud peut avoir un paramètre d'entrée maximum, ou aucun paramètre. Le paramètre d'entrée est toujours appelé raw\_value. Ne jamais supprimer le paramètre d'entrée, même si vous n'avez pas besoin d'entrée pour votre fonction ; sinon, des erreurs de validation apparaîtront lorsque vous exécutez le mappage. La même chose s'applique à la sortie de la fonction. Si vous devez rétablir un composant d'entrée supprimé par accident, exécutez la commande de menu Fonction | Insérer entrée.
- MapForce vous permet de fournir des métadonnées de nœud (par ex., un nom de nœud, une annotation) à une fonction de nœud. Pour les détails, voir <u>Métadonnées de nœud dans les fonctions de</u> <u>nœud</u><sup>[57]</sup>.
- La boîte de dialogue **Filtrer les fonctions de nœud et Défauts** vous permet de choisir d'une large gamme de types de données, dont quelques unes représentent des catégories de types. Ceci signifie qu'ils correspondront à une plus grande sélection de types. Par exemple, le type string correspond à d'autres types de données variés depuis le string, tel que normalizedString, jeton, NCName, NMTOKEN, IDREF, ENTITY et d'autres. De même, le type decimal correspondra aux types dérivés integer, long, short et autres. La hiérarchie des types est spécifiée dans la <u>Recommandation XML</u> <u>Schema W3C</u>.

#### Créer une règle

Pour créer une règle, suivez les étapes suivantes :

 Cliquez avec la touche de droite sur le nœud d'intérêt et sélectionnez Fonctions de nœud et défauts | Fonctions de nœud de sortie et défauts depuis le menu contextuel. Qu'il s'agisse de fonctions de nœud et défauts Entrée ou Sortie dépend du côté du composant sur lequel vous devez créer une fonction de nœud ou un défaut. En alternative, cliquez avec la touche de droite sur un connecteur et sélectionnez une commande de fonction de nœud pour ce côté. Le volet Mappage affichera la fenêtre de fonctions de nœud, dans laquelle vous pouvez définir des défauts et fonctions de nœud (rectangle rouge dans la capture d'écran ci-dessous).

Ð	Fonctions de nœud et défauts définis du côté entrée de l'item (> <u>Email</u> dans le composant 🔮 <u>ExpReport-Target</u> :
⊠н	ériter les règles des ancêtres
* f* ×	Appliquer à 1 Valeur par défaut / Description de la fonction

Si un nœud parent a des règles définies et que vous voulez transférer ces règles du parent à un enfant, sélectionnez la case à cocher *Hériter les règles des ancêtres (voir la capture d'écran ci-dessus)*. Pour plus d'informations sur l'héritage, voir <u>Scénarios Use-Case</u><sup>665</sup>.

2. La prochaine étape est de définir si vous voulez ajouter un défaut ( ) ou une fonction ( ). Dès que vous cliquez sur l'option pertinente, une nouvelle règle sera créée (une ligne dans la grille de la fenêtre de fonction de nœud). Pour des informations sur la configuration des règles, voir *Configuration de la règle* ci-dessous. Si vous définissez une fonction, la zone de mappage changera pour afficher l'entrée de la fonction et les paramètres de sortie.

#### Éditer/supprimer les règles

Pour voir, modifier ou supprimer une règle, cliquez sur l'icône  $f_{\mathbf{x}}$  (noire ou rouge) à côté d'un nœud d'intérêt. La fenêtre des fonctions de nœud affichera toutes les règles définies pour ce nœud. Si vous avez ajouté une fonction de nœud, vous verrez le bouton  $\boxed{Edit}$  dans la grille, qui vous permet de modifier la mise en œuvre de la fonction. Si la touche  $\boxed{Edit}$  n'est pas présente, alors la fonction est probablement définie sur un ancêtre. Dans ce cas, cliquez sur l'icône  $f_{\mathbf{x}}$  à côté de l'item pour lequel la règle a été définie.

Pour supprimer une règle, sélectionnez-la depuis la grille dans la fenêtre de fonction de nœud, puis cliquez sur le bouton ×.

### Configurer une règle

Dès que vous sélectionnez Fonctions de nœud et Défauts | Entrée/Sortie Fonctions de nœud et Défauts depuis le menu contextuel (*voir les instructions ci-dessus*), vous verrez la fenêtre des fonctions de nœud dans laquelle vous pouvez configurer des fonctions de nœud et des défauts. Les paramètres disponibles sont décrits ci-dessous.

- Appliquer à : Une règle peut être appliquée à l'item actuel, ses items enfant directs ou à tous les items descendants. Si l'item que vous avez sélectionné n'a pas de descendant, seule l'option *Item actuel* sera disponible.
- *Type de données :* Cliquez sur la touche et choisissez un type de données depuis le dialogue. La règle ne s'appliquera qu'aux items qui ont ce type de données (ou un type de données dérivé). Si l'item que vous avez sélectionné n'a pas de descendants, alors le type de données de l'item est le seul choix possible.
- Valeur par défaut/Description de la fonction : Si vous définissez un défaut, saisissez ici la valeur par défaut que vous souhaitez définir pour l'item sélectionné (et tous les descendants, le cas échéant).
   Pour définir un string vide comme défaut, laissez ce champ vide. Si vous définissez une fonction, ce champ est réservé uniquement à des informations : Il affiche un résumé de la fonction.

Pour quitter la fenêtre de fonction de nœud, cliquez sur le bouton <sup>4</sup> dans le coin supérieur gauche de la fenêtre et appuyez sur **Escape**.

#### Indice visuel

MapForce affiche différents indices visuels pour aider à comprendre quelles règles sont définies et si elles s'appliquent à un nœud spécifique (*voir la table ci-dessous*).

Icône	Description
fя	Cet icône indique qu'une règle est définie pour son item et peut affecter tous ses descendants. Cliquez sur l'icône pour modifier ou supprimer la règle.
f <sub>8</sub>	Cet icône indique que l'item hérite la règle définie au niveau d'ancêtre.
fя	Cet icône indique que la règle est définie pour et s'applique à cet élément. Cette icône apparaît généralement lorsqu'une fonction ou un défaut est défini pour un seul nœud.
×.	Cette icône indique que même si une règle s'applique à cet item, elle est délibérément bloquée. Cette icône est affichée uniquement si l'héritage est bloqué et aucune autre règle n'est définie pour ce nœud. Si une règle provenant d'un ancêtre s'applique, l'icône $\frac{f_{s}}{2}$ a une priorité.
fx	Cette icône indique qu'une règle définie pour cet item est inactive. Par exemple, cette icône peut apparaître pour des items enfants qui ne sont pas encore connectés.
<u>í</u>	Cette icône indique que la règle est actuellement inactive, car aucune correspondance et nœuds connectés n'ont été identifiés. Le terme <i>matching</i> signifie la chose suivante : Avant d'appliquer une fonction ou un défaut, MapForce recherche le type de données que vous avez défini ; si vous avez également spécifié un filtre qui doit correspondre à un nom de nœud spécifique, par exemple, MapForce recherchera également le nom de nœud.

# 6.3.2 Scénarios Use-Case

Cette rubrique décrit les différents scénarios dans lesquels les défauts et fonctions de nœud peuvent être utiles. Pour tester ces scénarios, vous aurez besoin du mappage d'échantillon suivant : comments>\Altova\MapForce2023\MapForceExamples\Tutorial\MissingFields.mfd. Notre point de départ est le mappage illustré ci-dessous.



Certains des nœuds dans le fichier source sont vides. À cette étape, le fichier de sortie a l'aspect suivant :

T-Shirt	25.3	20	Disponible en noir, bleu et rouge
Chemise	70.3	60	
Pantalons			Stock limité
Veste	57.5	40	

Les scénarios décrits ci-dessous vous montreront comment définir un défaut pour tous les nœuds de type string, définir les défauts pour les différents types de données, bloquer les règles pour des nœuds spécifiques, écraser les règles héritées, et définir une fonction de nœud, et appliquer cette fonction de manière conditionnelle aux nœuds pertinents.

#### Scénario 1 : Définir un défaut pour tous les nœuds de string

Dans notre mappage, quelques uns des nœuds du type string sont vides. Notre objectif est de définir une valeur par défaut n/a à la place de strings vides. Suivez les étapes suivantes :

- 1. Cliquez avec la touche de droite sur l'élément Article et sélectionnez Fonctions de nœud et défauts | Fonctions de nœud de sortie et défauts depuis le menu contextuel.
- 2. Cliquez sur l'option fai dans la fenêtre des Fonctions de nœuds et saisissez n/a dans le champ Valeur par défaut (voir la capture d'écran ci-dessous).



Dans le composant ArticlesOfClothing, les éléments Nom et Description sont de type string, ce qui rend la valeur par défaut appropriée pour les deux. Pour cette raison, l'icône  $\int_{x}$  apparaît à côté de ces nœuds. Puisque les valeurs par défaut s'appliquent uniquement aux valeurs vides, seuls deux nœuds Field4 ont reçu les valeurs n/a dans la sortie (*en surbrillance jaune ci-dessous*).

T-Shirt	25.3	20	Disponible en noir, bleu et rouge
Chemise	70.3	60	n/a
Pantalons			Stock limité
Veste	57.5	40	n/a

#### Scénario 2 : Définir les défauts pour différents types de données

Outre la définition d'un défaut pour des nœuds d'un type spécifique, vous pouvez en plus définir un autre défaut pour un autre type de données. Par exemple, un des articles a des valeurs vides dans Prix et ItemsInStock, dont les deux sont de type numérique. Maintenant, notre objectif est de remplacer ces valeurs vides avec 0. Suivez les étapes ci-dessous :

- 1. Répétez les étapes depuis le *Scénario 1*. Ceci ajoutera une valeur par défaut pour tous les nœuds de type string qui ont des valeurs vides.
- 2. Ajoutez un autre défaut et sélectionnez numérique comme type de données.
- 3. Tapez 0 dans le champ Valeur par défaut (voir la capture d'écran ci-dessous).



La sortie a maintenant l'air de ceci (notez les parties en surbrillance) :

T-Shirt	25.3	20	Disponible en noir, bleu et rouge
Chemise	70.3	60	n/a
Pantalons	0	0	Stock limité
Veste	57.5	40	n/a

**Note :** Conformément à la <u>Spécification de schéma XML</u>, integer est dérivé de décimal, et les deux appartiennent à un type de données numérique. Dans notre exemple, la règle s'appliquera aux deux éléments ItemsInStock et Prix si vous sélectionnez numeric comme type de données dans la fenêtre de fonctions de nœud. Si vous sélectionnez decimal comme type de données, la règle continuera à s'appliquer aux deux éléments. Toutefois, si vous sélectionnez integer comme type de données, la règle s'appliquera uniquement à l'élément ItemsInStock.

### Scénario 3 : Bloquer la règle pour des nœuds spécifiques

Dans ce scénario, nous voulons appliquer les défauts pour tous les nœuds string et numeric, mais nous ne voulons pas définir un défaut dans l'élément Prix. Pour atteindre cet objectif, reproduisez les étapes du *Scénario 2*, cliquez sur l'élément Prix et désactivez la case à cocher *Hériter des règles des ancêtres*. Dans le mappage ci-dessus, l'item Prix n'hérite plus des règles depuis son parent, Article. Pour cette raison, l'icône **X** apparaît à côté de l'élément Prix (*voir la capture d'écran ci-dessous*).

Fonctions de nœud et défauts définis du côté sortie de l'item	> <u>SinglePrice</u> dans le composant 섙 <u>Articles</u> :
Hériter les règles des ancêtres	
Appliquer à Valeur par défaut / Description de la fonction	1
$f_{R}^{+}$	
$\times$	
Si Antintes	Taut file
Fichier: MissingFields.xml Fichier/Ct	E f Fichier: (défaut) Fichier/Chaîne
Articles restriction of xs:anyType [0.	→ · · · · · · · · · · · · · · · · · · ·
Article ArticleType $[1]$	Field1 string [0.1]
Number xstnteger [01]	Field3 string [01]
SinglePrice xs:decimal [01]	Field4 string [01]
Description xs:string [01]	

La sortie ci-dessous montre que la valeur vide de l'élément Prix, pour laquelle la règle a été bloquée, a été mappée vers le Champ2 (*notez les parties en surbrillance*).

T-Shirt Chemise	25.3 70.3	20 60	Disponible en noir, bleu et rouge <mark>n/a</mark>
Pantalons		0	Stock limité
Veste	57.5	40	n/a

## Scénario 4 : Écraser les règles héritées

Dans ce scénario, nous voulons définir les défauts pour tous les nœuds string et numériques ; toutefois, pour l'élément Prix exclusivement, nous voulons définir 100 comme valeur par défaut. Pour atteindre cet objectif, répétez les étapes décrites dans le *Scénario 2*, cliquez sur Prix dans le composant ArticlesOfClothing et ajoutez un nouveau défaut, tel qu'affiché dans la capture d'écran ci-dessous. Les règles héritées ont un fond jaune.

Û	Node functions and defaults defined on output side of item 🗅 <u>Price</u> in component 🔮 <u>ArticlesOfClothing</u> :						
	Inherit rules from ancestors						
तन	Apply to	Data typ		Default value / Function description	*		
f.t	Current item 💌	decimal	del Default value:	100			
~		string	del Default value:	n/a			
~		numeric	Default value:	0			
					Ψ.		

La règle définie directement dans un nœud a la priorité sur les règles créées au niveau parent. Pour cette raison, la sortie a maintenant l'aspect suivant (*notez les parties surlignées*) :

T-Shirt	25.3	20	Disponible en noir, bleu et rouge
Chemise	70.3	60	n/a
Pantalons	<mark>100</mark>	0	Stock limité
Veste	57.5	40	n/a

Si vous avez défini plus d'une règle pour le même nœud, la règle au niveau supérieur de la grille s'applique au nœud actuel. Vous pouvez permuter l'ordre de ces règles en glissant manuellement les règles dans la grille.

#### Scénario 5 : Créer une fonction de nœud et l'appliquer aux conditions

Ce scénario expliquera comment créer une fonction de nœud et l'appliquer aux conditions. Nous utiliserons la fonction <u>concat</u> pour ajouter du texte aux nœuds du type string. Suivez les instructions ci-dessous :

- 1. Cliquez avec la touche de droite sur l'élément Article et sélectionnez Fonctions de nœud et défauts | Fonctions de nœud de sortie et défauts depuis le menu contextuel.
- 2. Ajoutez une fonction de nœud en cliquant sur l'icône 👫 et créez la fonction affichée ci-dessous.



À cette étape, nous n'avons pas filtré les nœuds et la sortie a l'aspect suivant (*notez les parties en surbrillance*) :

T-Shirt <mark>XXL)</mark>	(XXS, L-XXL)	25.3	20	Disponible en noir, bleu et rouge <mark>(XXS, L-</mark>
Chemise	(XXS, L-XXL)	70.3	60	
Pantalons	(XXS, L-XXL)			Stock limité <mark>(XXS, L-XXL)</mark>
Veste	(XXS, L-XXL)	57.5	40	

La sortie affiche que notre règle a été appliquée aux deux éléments de type string : Nom et Description. Maintenant, nous voulons que la règle s'applique uniquement à l'élément Nom. À cette fin, allez à la fenêtre des fonctions de nœud et cliquez sur la touche dans la colonne *Type et filtre de données*. Ceci ouvre la boîte de dialogue **Filtrer Fonctions de nœud et Défauts**, dans laquelle vous pouvez changer les types de données et spécifier les options de filtrage. Définissez le filtre tel qu'affiché dans la capture d'écran ci-dessous.

😢 Filtrer fonction:	s et défauts de nœud	$\times$
Ce dialogue vous per défaut de manière co l'item.	rmet d'appliquer les fonctions ou les onditionnelle, selon le type de donnée	es de
Appliquer la fonction	ou le défaut aux items de ce type :	
string numeric decimal		
integer date		
dateTime		
gDay		
gMonthDay		
gYear gYearMonth		
boolean		
QName		
Filtrer par :	Nom du nœud	$\sim$
Correspond à :		
	Respecter la casse	
	Expression régulière	
	OK Appuls	ar.
		-

#### La sortie a désormais l'aspect suivant :

T-Shirt Chemise	(XXS, L-XXL) (XXS, L-XXL)	25.3 70.3	20 60	Disponible en noir, bleu et rouge
Pantalons	(XXS, L-XXL)			Stock limité
Veste	(XXS, L-XXL)	57.5	40	

#### Filtrer par type de nœud

Cette option vous permet de réduire le type de données que vous avez sélectionné dans le dialogue **Filtrer Fonctions de nœud et Défauts**. Par exemple, vous avez choisi d'appliquer votre fonction de nœud aux nœuds numériques. Ensuite, vous pouvez spécifier un sous-type (par ex., *Filtrer par/nom de nœud :* xs:decimal).

#### Filtrer avec des expressions régulières

Pour plus d'options de filtrage avancées, vous pouvez utiliser des expressions régulières (par ex., pour faire correspondre des noms de nœud multiples). Pour plus d'informations, voir <u>Expressions régulières</u>. Veuillez prendre note des points suivants :

- Les ancres A et s sont implicites et ne doivent pas être saisies dans la case Correspond à.
- Le respect de la casse peut être spécifié dans la case à cocher *Correspond à*. Pour cette raison, l'indicateur i n'est pas pris en charge.
- La correspondance sur plusieurs lignes n'est pas pertinente pour le filtrage de nœud. Pour cette raison, l'indicateur m n'est pas pris en charge.

### Scénario 6 : Créer des défauts pour des nœuds non connectés

MapForce vous permet d'appliquer les défauts aux nœuds non connectés, mais uniquement si le nœud parent et/ou un frère sont connectés. Considérez l'exemple suivant :



Dans le mappage ci-dessus, nous avons défini une valeur par défaut pour le nœud Champ4 dans le composant ArticlesInfo. À cette étape, Champ4 n'a pas de connexion d'entrée. Les nœuds parent et frère sont connectés. Si le Champ4 reste non connecté, la valeur par défaut écrasera toutes les valeurs de Description dans la sortie (*voir ci-dessous*). Si vous connectez Description et Champ4, la valeur par défaut sera appliquée aux séquences vides.

T-Shirt	25.5	20	n/a
Chemise	70.3	60	n/a
Pantalons			n/a
Veste	57.5	40	n/a

Note : les défauts ne sont pas appliqués aux nœuds non connectés dans des composants JSON.

## 6.3.3 Métadonnées de nœud dans des fonctions de nœud

Dans certains cas, il se peut que vous souhaitiez que la fonction de nœud fasse quelque chose sur la base d'informations concernant le nœud actuel, à savoir les *métadonnées de nœud*. Le terme « métadonnées de nœud » signifie des informations diverses concernant le nœud auquel la fonction s'applique (par ex., le nom de nœud, la longueur de valeur et la précision), qui est stocké dans le schéma. La table ci-dessous recense tous les paramètres de métadonnées y pouvez utiliser dans une fonction de nœud. Notez que certains paramètres de métadonnées s'appliquent uniquement à des types de données spécifiques. MapForce affichera un avertissement lorsque vous tentez d'utiliser des métadonnées qui sont incompatibles avec le nœud actuel.

Paramètres de métadonnées	Description
node_name	Fournit le nom du nœud actuel tel qu'affiché dans le composant. Ces métadonnées sont prise en charge pour tous les nœuds. Dans les fichiers XML, node_name fait référence au nom de l'élément ou à l'attribut actuel. Dans les composants CSV, node_name fait référence au nom d'un champ CSV. Dans les base de données, node_name fait référence au nom d'une colonne de table.
node_annotation	Affiche un texte d'annotation à côté de la valeur du nœud dans la sortie. Ces métadonnées sont prises en charge pour tous les nœuds.
node_minLength	Fournit la valeur de la facette minLength du type de données du nœud. Applicable aux nœuds XML et de texte avec des types appropriés. Pour plus d'information sur les facettes de contrainte Length, voir la <u>Recommandation de schéma XML</u> .
node_maxLength	Fournit la valeur de la facette maxLength du type de données du nœud. Applicable aux nœuds XML et de texte avec des types appropriés. Pour plus d'information sur les facettes de contrainte Length, voir la <u>Recommandation de schéma XML</u> .
node_totalDigits	Fournit la valeur de la facette totalDigits du type de données du nœud. Applicable aux nœuds XML avec des types appropriés. Pour les détails sur la facette totalDigits, voir la <u>Recommandation de schéma XML</u> .
node_fractionDigits	Fournit la valeur de la facette fractionDigits du type de données du nœud. Applicable aux nœuds XML avec des types appropriés. Pour les détails sur la facette fractionDigits, voir la <u>Recommandation de schéma</u> XML.
node_length	Fournit la length (en octets) d'un nombre. Applicable aux champs de base de données avec des types appropriés. Pour en savoir plus sur length, voir la <u>documentation Microsoft</u> .
node_precision	Fournit le nombre de tous les chiffres d'un nombre. Applicable aux champs de base de données avec des types appropriés. Pour plus d'informations, voir la <u>documentation Microsoft</u> .
node_scale	Fournit les chiffres du numéro à droite du point décimal d'un nombre. Applicable aux champs de base de données avec des types appropriés. Pour plus d'informations, voir la <u>documentation Microsoft</u> .

**Note :** le type de données d'un paramètre de métadonnées doit coïncider avec le type de données du nœud pertinent. Autrement, l'exécution de la fonction échouera.

### Ajouter les métadonnées à fonction de nœud

Pour ajouter un paramètre de métadonnées à une fonction de nœud, vous devez <u>créer une fonction de</u> <u>nœud</u> an<u>eud</u>, puis cliquer sur l'option **Ajouter des spécificités du nœud**, qui est située sous la fenêtre des fonctions de nœud. Ceci ouvrira le dialogue **Insérer entrée avec spécificité de nœud**, dans laquelle vous pouvez sélectionner un paramètre de métadonnées pertinent. À la place de l'option **Ajouter les spécificités du nœud**, vous pouvez aussi choisir une des options suivantes :

- Cliquez sur la touche de droite dans une zone vide dans le mappage et sélectionnez **Insérer entrée** depuis le menu contextuel.
- Cliquez sur la commande **H** de la barre d'outils.
- Cliquez Insérer entrée dans le menu Fonction.

#### Lorsque les métadonnées ne sont pas prises en charge par un nœud

Lorsqu'un paramètre de métadonnées n'est pas pris en charge par un nœud, vous pouvez donner l'instruction à MapForce de renvoyer une séquence vide ou de ne pas appliquer la fonction de nœud (les deux options sont disponibles dans le dialogue **Insérer entrée avec la spécificité de nœud**). La séquence vide doit être gérée ; autrement, la fonction nœud peut ne retourner aucune valeur. Généralement, vous devez utiliser les fonctions de séquence (comme <u>substitute-missing</u><sup>(00)</sup> et <u>exists</u><sup>(303)</sup>), ou d'autres types de composant pour le traiter ultérieurement.

## Exemple

Cet exemple ci-dessous explique comment fournir le texte d'annotation à une fonction de nœud. Pour cet exemple, vous allez avoir besoin du mappage suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\MissingFields.mfd. Dans MissingFields.xsd (Sur lequel MissingFields.xml est basé), nous avons ajouté l'annotation suivante pour l'élément Name : Nom de l'article du catalogue Printemps 2022. Notre objectif est de fournir cette annotation à une fonction de nœud et de voir l'annotation dans la sortie. Suivez les instructions ci-dessous :

- 1. <u>Créer une fonction de nœud</u><sup>663</sup> pour l'élément Name dans le composant ArticlesOfClothing.
- 2. Répliquer la fonction affichée ci-dessous. Pour ajouter un paramètre node\_annotation, cliquez sur Ajouter des spécifiques de nœud (*voir le rectangle rouge ci-dessous*) et sélectionnez ce paramètre depuis la liste. Pour savoir comment ajouter et manipuler des fonctions, voir <u>Notions fondamentales de</u> <u>bases</u> <sup>652</sup>.



**Note :** par défaut, des structures imbriquées profondément ne sont pas entièrement scannées, afin de pouvoir préserver la mémoire et d'améliorer l'expérience de l'utilisateur. Si le composant dans lequel vous appliquez la fonction de nœud possède de telles structures imbriquées, vous pouvez agrandir les nœuds pertinents sur le mappage afin de rendre MapForce sensible aux nœuds. Dans ce cas,

MapForce prendra en compte les nœuds agrandis lorsque vous ajoutez un nouveau paramètre de métadonnées, et l'avertissement peut disparaître. La fonction de nœud doit être connectée pour être appliquée ; élargir des éléments non connectés n'est pas pertinent.

#### <u>Sortie</u>

La liste ci-dessous montre que l'annotation a été ajoutée aux valeurs des éléments Name.

```
T-Shirt (nom de l'article du catalogue Printemps 2022) 25.5 20 Dispo en noir, bleu
et rouge Chemise (nom de l'article du catalogue Printemps 2022) 70.3 60
Pantalons (nom de l'article du catalogue Printemps 2022)
Stock limité
Veste (nom de l'article du catalogue Printemps 2022) 57.5 40
```

# 6.4 Fonctions définies par l'utilisateur

Les Fonctions définies par l'utilisateur (UDF en abrégé) sont des fonctions personnalisées définies une fois et réutilisables plusieurs fois dans le cadre du même mappage ou sur de plusieurs mappages. Les FDU sont de mini-mappages eux-mêmes : elles consistent généralement en un ou plusieurs paramètres, certains composants intermédiaires pour traiter les données et une sortie pour retourner les données à l'appelant. L'appelant est le mappage principal ou une autre UDF

### Avantages des FDU

Les FDU ont les avantages suivants :

- Les FDU sont réutilisables à l'intérieur d'un mappage ou dans de multiples mappages.
- Les FDU peuvent faciliter la lecture de votre mappage : Par exemple, vous pouvez mettre en packages des parties du mappage en composants plus petits et abstraire des détails de mise en œuvre. Le diagramme ci-dessous illustre ce principe.



- Les UDF sont des fonctions flexibles qui vous permettent de procéder des strings, nombres, dates et d'autres données de manière personnalisée qui agrandit les fonctions built-in de MapForce. Par exemple, vous voulez concaténer ou séparer du texte, réaliser des calculs en avance, manipuler des dates et heures, etc.
- Une autre utilisation commune des FDU est de rechercher un champ dans un fichier XML, une base de données ou tout autre format de données pris en charge par votre édition de MapForce et présenter ces données de manière utile. Pour les détails, voir <u>Look-up Implementation</u><sup>589</sup>.
- Les FDU peuvent être appelées récursivement (par ex., la FDU s'appelle elle-même). Ceci requiert que la FDU soit définie comme <u>fonction régulière (pas inline)</u> <sup>570</sup>. Les <u>FDU récursives</u> <sup>580</sup> peuvent remplir plusieurs exigences de mappage avancées, telles l'iération sur des structures de données ayant une profondeur d'enfants *N*, où *N* n'est pas connu à l'avance.

### Exemple

L'exemple suivant montre une FDU simple qui partage un string en deux strings séparés. Cette FDU fait partie d'un plus grand mappage appelé MapForceExamples\ContactsFromPO.mfd. Il prend un nom en tant que paramètre (par exemple, Helen Smith), applique les fonctions intégrées substring-before et substringafter, puis retourne deux valeurs : Helen et Smith.



#### Dans cette section

Cette section vous explique comment travailler avec des FDU et est organisé comme suit :

- Notions fondamentales des FDU<sup>576</sup>
- Paramètres FDU<sup>581</sup>
- FDU récursives 586
- Implémentation de la consultation 588

## 6.4.1 Notions de bases des FDU

Cette rubrique explique comment créer, importer, éditer, copier-coller et supprimer les fonctions définies par l'utilisateur (abrégé UDF).

### Créer des UDF

Cette sous-section explique comment créer une FDU de zéro et de composants déjà existants. L'exigence minimum est un compostant de sortie auquel quelques données sont connectées. Pour les paramètres d'entrée, une fonction peut avoir zéro, une ou plusieurs entrées. Les paramètres d'entrée et de sortie peuvent être de type simple (comme un string) ou de type complexe (une structure). Pour plus d'information concernant les paramètres simples et complexes, voir <u>Paramètres UDF</u><sup>631</sup>.

#### UDF à partir de zéro

Pour créer une FDU à partir de zéro, suivez les instructions ci-dessous :

- 1. Sélectionnez Fonction | Créer des fonctions définies par l'utilisateur. En alternative, cliquez sur le bouton de la barre d'outils  $\vec{\Box}$ .
- 2. Saisissez l'information pertinente dans le dialogue **Créer une fonction définie par l'utilisateur** (*voir la capture d'écran ci-dessous*).
| Créer une fonction définie par l'utilisateur                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | < |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| Paramètres<br>Nom de la LookupPerson<br>fonction :<br>Nom de user<br>bibliothèque :                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |   |
| Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |   |
| Syntaxe : LookupPerson( OfficeName, First, Last )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |   |
| Détail : Recherche des informations détaillées sur la personne<br>dans Altova_Hierarchical.xml                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |   |
| Implémentation<br>Utilisation inline<br>L'« Utilisation inlined » conseille à MapForce d'extraire les contenus de<br>cette fonction dans tous les emplacements où vous l'utiliserez. Cela<br>rendra le code généré un peu plus long mais il sera généralement un<br>peu plus rapide et permettra de définir des Sorties multiples en une<br>fonction.<br>Décocher « Utilisation inline » si vous souhaitez appeler cette fonction<br>récursivement. Si vous devez retourner des valeurs multiples, vous<br>pouvez toujours utiliser, par exemple, une structure XML contenant<br>des éléments multiples. |   |
| OK Annuler                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |   |

Les options disponibles sont recensées ci-dessous :

- Nom de la fonction : Champ obligatoire. Pour le nom de votre UDF, vous pouvez utiliser le caractère suivant : caractères alphanumériques (a-z, A-Z, 09), un trait de soulignement (\_), un trait d'union/tiret (-), et un deux-points (:).
- Nom de la bibliothèque : Champ obligatoire. Il s'agit du nom de la bibliothèque de fonction (dans la fenêtre Bibliothèques<sup>(31)</sup>) dans laquelle votre fonction sera enregistrée. Si vous ne spécifiez pas le nom de bibliothèque, la fonction sera placée dans une bibliothèque par défaut appelée user.
- Syntaxe : Champ optionnel. Saisir un texte décrivant brièvement la syntaxe de la fonction (par ex., paramètres attendus). Ce texte sera affiché à côté de la fonction dans la fenêtre **Bibliothèques** et n'influera pas sur la mise en place de la fonction.
- *Détail :* Champ optionnel. Cette description sera affichée lorsque vous déplacez le curseur sur la fonction dans la fenêtre **Bibliothèques** ou dans d'autres contextes.
- Utilisation Inline : Sélectionnez cette case à cocher si la fonction doit être créée comme inline. Pour plus d'informations, voir UDF régulières vs. Inline ci-dessous.
- 3. Cliquez sur **OK**. La fonction devient immédiatement visible dans la fenêtre **Bibliothèques** sous le nom de bibliothèque spécifié ci-dessus. De même, la fenêtre de mappage est maintenant redessinée afin de

#### 578 Fonctions

vous permettre de créer une nouvelle fonction (il s'agit d'un mappage autonome dont la référence est le *mappage de la fonction*). Le mappage de la fonction inclut un composant de sortie par défaut.

4. Ajoutez tous les composants requis du mappage de la fonction. Vous pouvez faire ceci comme pour un mappage standard.

Pour utiliser l'UDF dans un mappage, glissez-la depuis la fenêtre **Bibliothèques** vers la zone principale de mappage. Voir aussi *Appeler et importer les UDF* ci-dessous.

#### UDF de composants existants

Pour créer une UDF depuis des composantes existantes, suivez les étapes suivantes :

- 1. Choisir les composants pertinents sur le mappage en effectuant une sélection rectangulaire avec la souris. Vous pouvez aussi choisir plusieurs composants en cliquant sur chacun d'entre eux tout en maintenant la touche **Ctrl** appuyée.
- Sélectionnez la commande de menu Fonction | Créer une fonction définie par l'utilisateur depuis la sélection. En alternative, cliquez sur le bouton de la barre d'outils <sup>100</sup>/<sub>2</sub>.
- 3. Suivez les étapes 2 à 4 depuis l'UDF à partir de zéro.

#### UDF régulières vs. Inline

Il existe deux types de FDU : *regular* et *inline*. Vous pouvez spécifier le type de votre FDU quand vous le créez. Les fonctions inline et regular se comportent différemment en termes de génération de code, de récursivité et la capacité d'avoir plusieurs paramètres de sortie. La table ci-dessous résume les différences principales entre les FDU regular et inline.

Fonctions Inline (bordure en pointillés)	Fonctions régulières (bordure solide)
Avec les fonctions inline, le code FDU est inséré dans tous les emplacements là où la fonction est appelée. Si la FDU est appelée plusieurs fois, le code de programme généré serait plus long de manière significative.	Le code de la FDU est généré une fois et les entrées sont passées comme valeurs de paramètre. Si la FDU est appelée plusieurs fois, la FDU est évaluée chaque fois avec les valeurs de paramètre correspondantes.
Les fonctions inline peuvent avoir plusieurs sorties et donc retourner plusieurs valeurs.	Les fonctions régulières ne peuvent avoir qu'une sortie. Pour retourner de multiples valeurs, vous pouvez déclarer la sortie comme étant de type complexe (par exemple, structure XML), qui vous permettrait de passer de multiples valeurs à l'appelant.
Les fonctions inline ne peuvent pas être appelées récursivement.	Les fonctions régulières peuvent être appelées récursivement.
Les fonctions inline ne prennent pas en charge la configuration d'un <u>contexte de priorité</u> <sup>884</sup> dans un paramètre.	Les fonctions régulières prennent en charge la configuration un contexte de priorité dans un paramètre.

**Note :** Basculer un formulaire FDU de inline à régulier, et vice versa, peut affecter le <u>contexte de mappage</u><sup>674</sup>, et ceci pourrait amener le mappage à produire un résultat différent.

## Appeler et importer les UDF

Après avoir créé une FDU, vous pouvez l'appeler du même mappage dans lequel vous l'avez créé ou depuis tout autre mappage.

### Appeler la FDU du même mappage

Pour appeler une FDU depuis le même mappage, suivez les étapes suivantes ci-dessous :

- Trouver la fonction pertinente dans la fenêtre Bibliothèques sous la bibliothèque que vous avez spécifiée quand vous avez créé la fonction. Pour ce faire, commencez à taper le nom dans la fenêtre Bibliothèques.
- 2. Glissez la fonction depuis la fenêtre **Bibliothèques** dans le mappage. Vous pouvez désormais connecter tous les paramètres requis à la fonction.

#### Importer la FDU d'un mappage différent

Pour importer une FDU à partir d'un autre mappage, suivez les instructions ci-dessous :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u> <sup>(31)</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques	×
🖻 🛃 New Design1	*
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
😢 Importations Bibliothèques globales Ajouter	

- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- Cherchez le fichier .mfd qui contient les fonctions, et cliquez sur Ouvrir. Un message apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée et pouvant être accédée dans la fenêtre Bibliothèques.

Vous pouvez maintenant utiliser une des fonctions importées dans le mappage actuel en les glissant depuis la fenêtre **Libraries** dans le mappage. Pour plus informations concernant la consultation et l'organisation des bibliothèques de fonction, voir <u>Gérer des bibliothèques de fonction</u><sup>555</sup>.

#### Mappage avec identifiants (Enterprise Edition)

Si le fichier .mfd importé contient des identifiants, ceux-ci sont affichés en tant qu'importés (avec un arrièreplan jaune) dans le Gestionnaire d'identifiants. Par défaut, les identifiants importés ne sont pas enregistrés avec le mappage principal, mais vous pouvez créer en option une copie locale et les enregistrer sous le mappage principal.

# Éditer les UDF

Pour éditer une FDU, suivez les étapes ci-dessous :

- 1. Ouvrir le mappage qui contient une FDU.
- 2. Double-cliquez sur la barre de titre de la FDU dans le mappage pour voir les contenus de la fonction là où vous pouvez ajouter, éditer ou supprimer les composants, tel que requis.
- Pour changer les propriétés de la fonction (par ex., le nom de la description), cliquez avec la touche de droite sur une zone vide dans le mappage et sélectionnez Paramètres de fonction depuis le menu contextuel. En alternative, cliquez sur la touche de la barre d'outils <sup>1</sup>/<sub>4</sub>.

Vous pouvez aussi éditer une fonction en double-cliquant sur son nom dans la fenêtre **Bibliothèques**. Néanmoins, seules des fonctions se trouvant dans le document actif actuellement peuvent être ouvertes de cette manière. Double-cliquer sur une fonction définie par l'utilisateur qui a été créée dans un autre mappage ouvre ce mappage dans une autre fenêtre. Si vous éditez ou supprimez une fonction définie par l'utilisateur qui a été importée dans plusieurs mappages, tous ces mappages seront touchés par la modification.

Pour retourner au mappage principal, cliquer sur 🗮 la touche dans le coin supérieur gauche de la fenêtre de mappage.

MapForce vous permet de naviguer à travers différents mappages et FDU en utilisant les touches de la barre d'outils et Les raccourcis de clavier correspondants pour ces touches sont **Alt+Left** et **Alt+Right**, respectivement.

# Copier-coller les UDF

Pour copier une FDU à le collez dans un d'un autre mappage, suivez les étapes ci-dessous :

- 1. Ouvre la <u>fenêtre de gestion des bibliothèques</u><sup>555</sup>.
- 2. Cliquez avec la touche de droite dans une zone vide de la fenêtre **Bibliothèques** et choisissez l'option **Afficher tous les documents ouverts**.
- Ouvrir les mappages de source et destination. Assurez-vous que les deux mappages sont enregistrés sur le disque. Ceci assure une résolution correcte des chemins. Voir aussi <u>Copier-coller et les</u> <u>chemins relatifs</u><sup>(83)</sup>.
- 4. Cliquez avec la touche de droite sur la FDU pertinente depuis le mappage de source dans la fenêtre **Gérer les bibliohèques** et sélectionnez **Copier** depuis le menu contextuel (*voir la capture d'écran cidessous*) ou appuyez sur **Ctrl+C**. Laissez la fenêtre **Gérer les bibliothèques** ouverte.

Gérer les bibliothèques			
🖽 🖶 BuildHierarchyRecursive.mfd			
🕀 Fonctions défir	nies p	ar l'utilisateur	
😹 user.Bu		Ouncrie	
Importations I		Ouvin	
📮 🌄 New.mfd		Localiser la fonction dans la fenêtre Biblic	othèques
······Fonctions défu		P <u>r</u> opriétés	
Importations I	u	_	
🖃 😋 Importation	ቆ	<u>C</u> ouper	Strg+X
🔄 🖶 BooksTol	Ð	C <u>o</u> pier	Strg+C
······Fonctions de	X	<u>S</u> upprimer	Entf
Importation			
	~	Montrer tous les documents ouverts	
		Mo <u>n</u> trer les chemins de fichier	

5. Basculer à la fenêtre de de mappage de destination (et la fenêtre **Gérer les bibliothèques** sera modifiée conformément), cliquez avec la touche de droite sur *Fonctions définies par l'utilisateur*, et sélectionnez **Coller** depuis le menu contextuel.

## Supprimer les UDF

Pour supprimer une FDU, suivez les étapes ci-dessous :

- 1. Double-cliquer sur la barre de titre de la FDU dans le mappage.
- 2. Cliquer sur la touche 🚵 dans le coin supérieur droit de la fenêtre Mappage.
- 3. Si la fonction est utilisée dans le mappage ouvert actuel, MapForce demandera si vous voulez remplacer toutes les instances avec des composants internes. Cliquer sur **Oui** si vous souhaitez supprimer la fonction et remplacer toutes les instances dans lesquelles elle est appelée avec les composants de la fonction. Cela vous permet de garder valide le mappage principal même si la fonction est supprimée. Néanmoins, si la fonction supprimée est utilisée dans d'autres mappages externes, ceux-ci ne seront pas valides. Cliquez sur **Non** si vous voulez supprimer la fonction est utilisée ne sera pas valide.

# 6.4.2 Paramètres UDF

Lorsque vous créez une FDU, vous devez spécifier quels paramètres d'entrée celle-ci doit prendre (le cas échéant) et quelle sortie elle doit retourner. Alors que des paramètres d'entrée ne sont pas toujours nécessaires, un paramètre de sortie est obligatoire dans tous les cas, c'est-à-dire qu'une fonction doit toujours retourner quelque chose. Les paramètres de fonction peuvent être de type simple (par ex., string ou integer) ou posséder une <u>structure complexe</u><sup>682</sup>. Par exemple, la fonction définie par l'utilisateur FindArticle illustrée ci-dessous a deux paramètres d'entrée et un de sortie :

- POArtNr est un paramètre d'entrée de type simple string;
- Amount est un paramètre d'entrée de type integer.
- CompletePO est un paramètre de sortie de structure complexe XML.



#### Ordre du paramètre

Lorsqu'une FDU a plusieurs paramètres d'entrée ou de sortie, vous pouvez modifier l'ordre dans lequel les paramètres doivent apparaître aux appelants de cette fonction. L'ordre des paramètres dans le mappage de la fonction (en commençant par le haut) dicte l'ordre dans lequel il apparaît aux appelants de la fonction.

#### Important

- Les paramètres d'entrée et de sortie sont triés par leur position du haut au bas. C'est pourquoi, si vous déplacez le paramètre input3 en haut du mappage de la fonction, il deviendra le premier paramètre de cette fonction.
- Si deux paramètres ont la même position verticale, le paramètre le plus à gauche prend la précédence.
- Dans le cas inhabituel que deux paramètres ont exactement la même position, l'ID de composant interne est utilisé automatiquement.

## Structures de type complexe

Les structures sur lesquelles un paramètre dans les FDU peut être basé sont résumées dans la liste cidessous.

#### MapForce Basic Edition

- Structure de schéma XML
- MapForce Professional Edition
  - Structure de schéma XML
  - Structure de base de données

#### MapForce Enterprise Edition

- Structure de schéma XML
- Structure de base de données
- Structure EDI
- Structure FlexText
- Structure de schéma JSON

#### UDF basées sur les structures de base de données (éditions Professional et Enterprise)

MapForce vous permet de créer des paramètres UDF basés sur BD avec une arborescence de tables associées. L'arborescence de tables associées représente une structure in-memory qui n'a pas de connexion

vers la base de données lors de l'exécution. Ceci signifie également qu'il n'y a pas de gestion automatique de la clé étrangère et pas d'actions de table dans les paramètres ou variables.

## Ajouter des paramètres

Pour ajouter un paramètre d'entrée ou de sortie, suivez les étapes suivantes :

- 1. <u>Créer une FDU<sup>576</sup> ou ouvrir une tâche existante<sup>579</sup>.</u>
- Exécuter la commande de menu Fonction | Insérer entrée, ou Fonction | Insérer sortie, respectivement (*voir la capture d'écran ci-dessous*). De manière alternative, cliquez sur *i* (Insérer entrée) ou *i* (Insérer sortie) dans la barre d'outils.

🙁 Create Input			×
Name: ShortPC	)		
Туре			
○ <u>S</u> imple type (in	teger, string, etc.)		
Datatype:	string $\vee$		
• <u>C</u> omplex type	(tree structure)		
Str <u>u</u> cture:	e2019\MapForceExamples\ShortPO.xsd	C <u>h</u> oose	Edit
<u>R</u> oot:	ShortPO/LineItems/LineItem	Ch <u>o</u> ose	
	$\square$ Save structure file path relative to MFD	file	
✓ Input is required			
Input is a Sequer	nce		
		OK	Cancel

3. Choisissez si les paramètres d'entrée ou de sortie doivent être de type simple ou complexe (voir la boite de dialogue ci-dessus). Voir la liste des structures complexes disponibles ci-dessous. Par exemple, pour créer un paramètre qui est de type XML complexe, cliquez sur Choisir à côté de Structure et cherchez le schéma XML qui décrit la structure requise.

Si le mappage de la fonction contient déjà des schémas XML, ils sont disponibles pour la sélection en tant que structures. Sinon, vous pouvez sélectionner un nouveau schéma qui fournira la structure du paramètre. La même chose vaut pour les bases de données ou autres structures complexes si elles sont prises en charge par votre édition de MapForce. Avec des structures XML, il est possible de sélectionner un élément root pour votre structure, si le schéma XML le permet. Afin de spécifier un élément root, cliquez sur **Choisir** à côté de *Root* et sélectionner l'élément root à partir duquel le dialogue qui s'ouvre.

Une fois cochée, la case *Enregistrer le chemin de fichier de la structure relatif au fichier MFD* modifiera le chemin absolu du fichier de structure en un chemin relatif au mappage actuel, lorsque vous enregistrez le mappage. Pour plus d'informations, voir <u>Chemins relatifs et absolus</u><sup>(33)</sup>. Les cases à cocher *Entrée requise* et *Entrée est une séquence* sont expliquées dans les sous-sections suivantes.

#### L'entrée est requise

Pour rendre un paramètre obligatoire dans une FDU, sélectionnez la case *Entrée est requise* (voir la boîte de dialogue ci-dessus). Si vous effacez la case à cocher *Entrée requise*, le paramètre deviendra optionnel et aura une bordure en pointillés dans le mappage.

Vous pouvez aussi spécifier une valeur de paramètre par défaut en la connectant à l'entrée par default d'un paramètre (*voir l'exemple ci-dessous*). La valeur par défaut sera uniquement appliquée s'il n'y a pas d'autre valeur. Si le paramètre optionnel reçoit une valeur lorsque la fonction est appelée, alors cette valeur prend le dessus sur le défaut.



## L'entrée est une séquence

En option, vous pouvez définir si un paramètre de fonction doit être traité en tant que valeur simple (option par défaut) ou en tant que séquence. Une séquence est une plage de zéro ou de plusieurs valeurs. Une séquence peut être utile quand votre fonction définie par l'utilisateur s'attend à des données d'entrée en tant que séquence afin de calculer les valeurs dans cette séquence, par exemple, en appelant les fonctions telles que avg, min, max. Pour traiter l'entrée du paramètre en tant que séquence, sélectionnez la case à cocher *Entrée est une séquence*. Veuillez noter que cette boîte à cocher est activée uniquement si la FDU est regular <sup>GTB</sup>.

L'usage d'une séquence est illustrée dans le mappage suivant : MapForceExamples\InputIsSequence.mfd. Dans l'extrait de ce mappage (*voir la capture d'écran ci-dessous*), le filtre des données est connecté à la FDU appelée Calculer. La sortie du filtre est une séquence d'items. C'est la raison pour laquelle le paramètre d'entrée de la fonction est défini en tant que séquence.



Vous trouverez ci-dessous l'illustration de la fonction Calculer qui agrège toutes les valeurs de séquence : Elle exécute toutes les fonctions avg, min, et max dans la séquence d'entrée. Pour voir la structure interne de la fonction Calculer, double-cliquez sur l'en-tête du composant Calculer dans le mappage ci-dessus.



En règle générale, les données d'entrée, soit séquence, (séquence ou non séquence) détermine combien de fois la fonction est appelée :

- Lorsque les données d'entrée sont connectées à un paramètre séquence, la fonction définie par l'utilisateur est appelée uniquement une seule fois et la séquence complète est passée dans la fonction définie par l'utilisateur.
- Lorsque les données d'entrée sont connectées à un paramètre *non-sequence*, la fonction définie par l'utilisateur est appelée *une seule fois dans chaque item unique dans la séquence*.
- Si vous connectez une séquence vide dans un paramètre de non-séquence la fonction ne sera pas appelée du tout. Cela peut se produire si la structure de source dispose d'items optionnels ou si une condition de filtre ne retourne pas d'items correspondants. Pour éviter ceci, utilisez la fonction

<u>substitute-missing</u><sup>(00)</sup> avant l'entrée de fonction pour assurer que la séquence n'est jamais vide. En alternative, spécifiez le paramètre comme a séquence et ajoutez la gestion pour la séquence vide à l'intérieur de la fonction.

La case à cocher Sortie est une séquence peut aussi être nécessaire pour des paramètres de sortie. Lorsqu'une fonction passe une séquence de plusieurs valeurs dans son composant de sortie, et que le composant de sortie n'est pas défini sur séquence, la fonction ne retournera que le premier item dans la séquence.

# 6.4.3 FDU récursives

Cette rubrique explique comment rechercher des données dans le fichier XML source avec l'aide d'une FDU récursive. Pour tester la FDU récursive, vous aurez besoin du mappage suivant :

MapForceExamples\RecursiveDirectoryFilter.mfd. Dans le mappage ci-dessous, la FDU FilterDirectory reçoit des données depuis le fichier source Directory.xml et le composant d'entrée simple SearchFor qui fournit l'extension .xml. Une fois que les données ont été traitées par la FDU, elles sont mappées vers le fichier cible.

Le mappage principal (*voir la capture d'écran ci-dessous*) décrit la mise en page du mappage général. La manière dont la FDU traite les données est définie séparément dans le mappage de fonction (*voir l'implémentation FDU ci-dessous*).



## **Objectifs**

Notre objectif est de recenser des fichiers avec une extension .xml dans la sortie tout en préservant toute la structure du répertoire. Les sous-sections ci-dessous expliquent le processus de mappage en détail.

## **Fichier source**

Ci-dessous, vous trouverez un extrait du fichier XML source (Directory.xml) qui contient des informations de fichiers et répertoires. Notez que les fichiers dans la liste ont différentes extensions (par ex., .xml, .dtd, .sps). Conformément au schéma (Directory.xsd), l'élément directory peut avoir des enfants fichier et des enfants répertoire. Tous les éléments directory sont récursifs.

```
<directory name="ExampleSite">
      <file name="blocks.sps" size="7473"/>
      <file name="blocks.xml" size="670"/>
      <file name="block_file.xml" size="992"/>
      <file name="block_schema.xml" size="1170"/>
      <file name="contact.xml" size="453"/>
      <file name="dictionaries.xml" size="206"/>
      <file name="examplesite.dtd" size="230"/>
      <file name="examplesite.spp" size="1270"/>
      <file name="examplesite.sps" size="20968"/>
       . . .
       <directory name="output">
              <file name="examplesite1.css" size="3174"/>
              <directory name="images">
                    <file name="blank.gif" size="88"/>
                    <file name="block_file.gif" size="13179"/>
                    <file name="block_schema.gif" size="9211"/>
                    <file name="nav_file.gif" size="60868"/>
                    <file name="nav_schema.gif" size="6002"/>
              </directory>
      </directory>
</directory>
```

## Implémentation FDU

Pour voir l'implémentation interne de la FDU, double-cliquez sur son en-tête dans le mappage principal. La FDU est récursive, c'est-à-dire qu'elle contient un appel vers elle-même. Étant donné qu'elle est connectée à l'élément récursif directory, cette fonction sera appelée autant de fois qu'il existe des éléments directory imbriqués dans l'instance XML de source. Pour prendre en charge des appels récursifs, la fonction doit être regular <sup>678</sup>.



L'implémentation de la FDU est constituée de deux parties : (i) définir les fichiers et (ii) définir le répertoire à rechercher.

# 588 Fonctions

### Définir des fichiers

La FDU traite les fichiers comme suit : La fonction contains vérifie si le premier string (le nom de fichier) contient le sous-string .xml (fourni par le composant d'entrée simple SearchFor). Si la fonction retourne true, le nom de fichier avec une extension .xml est écrite dans la sortie.

### Traiter des répertoires enfant

Les répertoires enfant du répertoire actuel sont envoyés comme entrée dans la FDU actuelle. La FDU itère donc à travers tous les éléments du directory et vérifie si les fichiers avec l'extension .xml existent.

## Sortie

Lorsque vous cliquez sur le volet **Sortie**, MapForce affichera uniquement des fichiers avec l'extension .xml (*voir l'extrait ci-dessous*).

# 6.4.4 Implémentation de la consultation

Cette rubrique explique comment rechercher des données sur les employés et présenter cette information de manière utile. Pour tester l'implémentation de la consultation, vous aurez besoin du mappage suivant : MapForceExamples\PersonListByBranchOffice.mfd.



## <u>Objectifs</u>

Nos objectifs sont les suivants :

- Pour rechercher des données sur chaque employé (leur poste, adresse e-mail et titre) dans un fichier XML.
- Pour présenter ces données en tant que liste séparée par une virgule et mapper cette liste dans l'élément Détails de l'XML cible.
- Pour extraire l'information sur les employés uniquement d'un branch office dénommé Nanonull, Inc.

Pour atteindre ces objectifs, nous avons conçu notre mappage de la manière suivante :

- Pour filtrer uniquement des employés de Nanonull, Inc., le mappage utilise le filtre Office.
- Pour consulter des informations sur les employés dans un autre fichier XML, le mappage appelle la FDU LookupPerson. L'implémentation de cette UDF est décrite dans la sous-section ci-dessous.
- Pour traiter des données d'employés, la fonction LookupPerson appelle en interne d'autres fonctions qui extraient et concatène l'information sur chaque employé. Toutes ces opérations se trouvent dans le mappage de la fonction et ne sont pas visibles dans le mappage principal. La fonction LookupPerson mappe ensuite les données de l'employé vers l'élément Détails dans PersonList.

## Implémentation de consultation

La fonction de consultation est fournie par la fonction LookupPerson, dont la définition est illustrée ci-dessous. Pour voir l'implémentation interne de la FDU, double-cliquez sur son en-tête dans le mappage principal.



La FDU est définie comme suit :

- Les données sont extraites depuis le fichier XML Altova\_Hierarchical.xml: (i) le nom de l'office et les prénom et nom de famille des employés, qui sont utilisés pour sélectionner les employés de Nanonull, Inc., et (ii) l'e-email, le titre et l'extension téléphonique qui sont concaténés en un string. Les définitions des fonctions EqualAnd et Person2Detail sont décrites ci-dessous.
- L'UDF a trois paramètres d'entrée qui fournissent les valeurs look-up Office\_Name, First\_Name et Last\_Name. La valeur du paramètre Office\_Name est extraite depuis l'entrée OfficeName du mappage principal, et les valeurs de First\_Name et Last\_Name sont fournies par le composant BranchOffices du mappage principal.
- La valeur de la fonction EqaulAnd (true ou false) est passée au filtre Details à chaque fois que de nouveaux détails liés aux employés (titre, e-mail, téléphone) sont traités. Lorsque le filtre Détails obtient la valeur true, l'opération look-up a du succès et les détails de l'employé sont extraits et retournés dans le mappage principal. Sinon, l'item suivant est examiné dans le contexte, et cette procédure continue jusqu'à la fin de la boucle.

#### Implémentation EqualAnd

La fonction EqualAnd est une FDU séparée (*voir ci-dessous*) définie à l'intérieur de la FDU LookupPerson. Pour voir la structure interne de l'UDF EqualAnd, double-cliquez sur l'en-tête de la fonction.



#### Fonctions

La FDU EqualAnd vérifie d'abord si a est égale à b; si le résultat est true, il est passé comme premier paramètre de la fonction logical-and. Si les deux valeurs dans la fonction logical-and sont true, le résultat est également true et passé à la prochaine fonction EqualAnd. Le résultat de la troisième fonction EqualAnd (voir LookupPerson UDF ci-dessus) est passé au filtre Details.

### Implémentation Person2Detail

La FDU Person2Details est une autre fonction à l'intérieur de la FDU LookupPerson. La FDU Person2Details (*voir ci-dessous*) concatène trois valeurs (extraites de Altova\_Hierarchical.xml) et deux constantes de texte.



## Sortie

Lorsque vous cliquez sur le volet **Sortie**, MapForce afficher les prénom et nom, et les détails des employés uniquement pour Nanonull, Inc (*voir l'extrait ci-dessous*).

```
<PersonList>

<Person>

<p
```

# 6.5 Custom Functions

Cette section explique comment importer les bibliothèques <u>XSLT</u><sup>592</sup>, <u>XQuery</u><sup>599</sup>, <u>Java et .NET</u><sup>604</sup>. Cette section montre également comment <u>référencer des bibliothèquesC#, C++ et Java manuellement</u><sup>612</sup>.

# 6.5.1 Importer des fonctions XSLT personnalisées

Vous pouvez étendre les bibliothèques de fonction XSLT 1.0 et 2.0 disponibles dans MapForce avec vos propres fonctions personnalisées, si vos fonctions personnalisées retournent des types simples.

Seules des fonctions personnalisées qui retournent des types de données simples (par exemple, strings) sont prises en charge.

#### Pour importer des fonctions depuis un fichier XSLT :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u><sup>31</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques	×
🗆 🧱 New Design1	*
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
😢 Importations Bibliothèques globales Ajouter	

 Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
 Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.

Chercher le fichier .xsl qui contient les fonctions, et cliquer sur **Ouvrir**. Un message apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée.

Les fichiers XSLT importés apparaissent sous forme de bibliothèques dans la fenêtre Bibliothèques, et affichent tous les modèles nommés en tant que fonctions sous le nom de bibliothèque. Si vous ne voyez pas la bibliothèque importée, veuillez vous assurer que vous avez bien sélectionné XSLT en tant que <u>langage de</u> transformation<sup>23</sup>. Voir aussi <u>Gérer les Bibliothèques de fonction</u><sup>555</sup>.

Veuillez noter que :

- Pour pouvoir importer dans MapForce, les fonctions doivent être déclarées en tant que modèles nommés conformément aux spécifications XSLT dans le fichier XSLT. Vous pouvez aussi importer des fonctions qui se produisent dans un document XSLT 2.0 sous la forme de <xsl:function name="MyFunction">. Si le fichier XSLT importé ou comprend d'autres fichiers XSLT, alors ces fichiers et fonctions XSLT seront également importés.
- Les connecteurs d'entrée mappables des fonctions personnalisées importées dépendent du nombre de paramètres utilisés dans l'appel du modèle ; des paramètres optionnels sont aussi pris en charge.
- Les espaces de noms sont pris en charge.
- Si vous effectuez des mises à jour dans des fichiers XSLT que vous avez déjà importé dans MapForce, des modifications seront détectées automatiquement et MapForce vous invite à recharger les fichiers.
- Lorsque vous écrivez des modèles nommés, assurez-vous que les instructions XPath utilisés dans le modèle sont liés dans les espaces de noms corrects. Pour voir les liaisons d'espace de noms du mappage, <u>consulter le code XSLT généré</u><sup>105</sup>.

## Types de données dans XPath 2.0

Si votre document XML référence un schéma XML et est valide, vous devez construire explicitement ou lancer des types de données qui ne sont pas implicitement convertis dans le type de données requis par une opération.

Dans le XPath 2.0 Data Model utilisé par l'Altova XSLT 2.0 Engine, toutes les valeurs de nœud **atomisées** depuis le document XML sont attribuées au type de données xs:untypedAtomic. Le type xs:untypedAtomic fonctionne bien avec des conversions de type implicite.

Par exemple,

- l'expression xs:untypedAtomic("1") + 1 résulte dans une valeur de 2 parce que la valeur xdt:untypedAtomic est implicitement promotée dans xs:double par l'opérateur d'addition.
- Les opérateurs arithmétiques promeuvent implicitement des opérandes dans xs:double.
- Les opérateurs de comparaison de valeur promeuvent des opérandes dans xs:string avant la comparaison.

## Voir aussi :

Exemple : Ajouter des fonctions XSLT 1.0 personnalisées <sup>633</sup> Exemple : Totaliser les valeurs de nœud <sup>636</sup> Implémentation de moteur XSLT 1.0 <sup>1479</sup> Implémentation de moteur XSLT 2.0 <sup>1480</sup>

# 6.5.1.1 Exemple : Ajouter des fonctions XSLT personnalisées

Cet exemple illustre comment importer des fonctions XSLT 1.0 personnalisés dans MapForce. Les fichiers nécessaires pour cet exemple sont disponibles dans le dossier suivant : c: \Users\cusername>\Documents\Altova\MapForce2023\MapForceExamples.

- Name-splitter.xslt. Ce fichier XSLT défini un modèle nommé appelé "tokenize" avec un seul paramètre "string". Le modèle fonctionne par le biais d'un string d'entrée et sépare les caractères en majuscule avec un espace pour chaque occurrence.
- Name-splitter.xml (le fichier d'instance XML de source à traiter)
- Customers.xsd (le schéma XML de source)
- CompletePO.xsd (le schéma XML de cible)

#### Pour ajouter une fonction XSLT personnalisée :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u> <sup>(31)</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques	×
🗆 🧱 New Design1	-
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
Importations Bibliothèques globales Ajouter	

- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- Parcourez le fichier .xsl ou .xslt qui contient le modèle nommé que vous voulez utiliser pour qu'il agisse comme fonction, dans ce cas Name-splitter.xslt, ouis cliquez sur Ouvrir. Une boîte de message apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée, et le nom du fichier XSLT apparaît dans la fenêtre Bibliothèques, avec les fonctions définies comme modèles nommés (dans cet exemple, Name-splitter avec la fonction tokenize).

Bibliothèques	×
🔁 core	^
E Summing-nodes	
Total	result = Total( string )
± xsit	~
Chercher les fonctions	•
Ajou	ter/Supprimer les bibliothèques
🔚 Bibliothèques 🕞 🛅 Proje	t

#### Pour utiliser la fonction XSLT dans votre mappage :

1. Glisser la fonction tokenize dans la fenêtre de Mappage et mapper les items comme indiqué cidessous.



2. Cliquer sur l'onglet **XSLT** pour voir le code XSLT généré.



- Note : Dès qu'un modèle nommé est utilisé dans un mappage, le fichier XSLT contenant le modèle nommé est included dans le code XSLT généré (xsl:include href...), et est appelé en utilisant la commande xsl:call-template.
  - 3. Cliquer sur l'onglet Sortie pour voir le résultat du mappage.

#### Pour supprimer des bibliothèques XSLT personnalisées depuis MapForce :

- 1. Cliquer sur la touche **Ajouter/Supprimer Bibliothèques**, dans la zone inférieure de la fenêtre Bibliothèques. La fenêtre Gérer les Bibliothèques s'ouvre.
- 2. Cliquer sur **Supprimer Bibliothèque** 🔀 à côté de la bibliothèque à supprimer.

# 6.5.1.2 Exemple : Totaliser les valeurs de nœud

Cet exemple montre comment traiter plusieurs nœud d'un document XML et faire mapper le résultat comme une valeur unique dans un document XML de cible. Plus spécifiquement, l'objectif du mappage est de calculer le prix de tous les produits dans un fichier XML de source et l'écrire en tant que valeur unique dans un fichier XML de sortie. Les fichiers utilisés dans cet exemple sont disponibles dans le dossier

- Summing-nodes.mfd le fichier de mappage
- input.xml le fichier XML de source
- input.xsd le schéma XML de source
- output.xsd le schéma XML de cible
- Summing-nodes.xslt Une feuille de style XSLT personnalisée contenant un modèle nommé pour additionner les nœuds individuels.

Il existe deux moyens différents d'atteindre l'objectif du mappage :

- En utilisant la fonction <u>sum</u><sup>640</sup>. Cette fonction intégrée MapForce est disponible dans la fenêtre Bibliothèques.
- En important une feuille de style XSLT personnalisée dans MapForce.

## Solution 1: Utiliser la fonction d'agrégation "sum"

Pour utiliser la fonction sum dans le mappage, la glisser depuis la fenêtre **Bibliothèques** dans le mappage. Veuillez noter que les fonctions disponibles dans la fenêtre **Bibliothèques** dépendent de la version de langage XSLT que vous avez sélectionné (XSLT 1 ou XSLT 2). Ensuite, créer les connexions de mappage comme indiqué ci-dessous.



Pour plus d'informations concernant les fonctions d'agrégation de la bibliothèque core, voir aussi core la aggregate functions<sup>633</sup>.

## Solution 2: Utiliser une feuille de style XSLT personnalisée

Comme mentionné ci-dessus, l'objectif de l'exemple est d'additionner les champs Price des produits dans le fichier XML de source, dans ce cas, les produits A et B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Input xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

L'extrait de code ci-dessous montre une feuille de style XSLT personnalisée qui utilise le modèle nommé "Total" et un seul paramètre string. Le modèle fonctionne par le biais du fichier d'entrée XML et additionne toutes les valeurs obtenues par l'expression XPath /Product/Price.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
<xsl:template match="*">
<xsl:template match="*">
<xsl:for-each select=".">
<xsl:for-each select=".">
<xsl:call-template name="Total">
</xsl:call-template name="Total">
</xsl:call-template>
</xsl:for-each>
</xsl:for-each>
</xsl:template>
<xsl:template name="Total">
<xsl:template name="Total">
<xsl:template>
</sl:template>
</sl:template name="string"/>
</sl:template>
```

**Note :** Pour additionner les nœuds dans XSLT 2.0, modifier la déclaration de feuille de style en version="2.0".

Avant d'importer la feuille de style XSLT dans MapForce, choisir XSLT 1.0 en tant que <u>langage de</u> <u>transformation</u><sup>25</sup>. Vous êtes maintenant prêt à importer la fonction personnalisée, comme suit :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u><sup>31</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).



- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- Chercher <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Summingnodes.xslt, et cliquer sur Ouvrir. Un message apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée et la nouvelle bibliothèque apparaît dans la fenêtre Bibliothèques.

Bibliothèques		×
E core		^
E Summing-nodes		
Total	result = Total( string )	
E xsit		~
Chercher les fonction	S	•
	Ajouter/Supprimer les bibliothèques	
🔚 Bibliothèques 🥤	Projet	

4. Glisser la fonction **Total** depuis les Bibliothèques dans le mappage, et créer les connexions de mappage comme indiqué ci-dessous.



Pour consulter le résultat de mappage, cliquer sur l'onglet **Sortie**. La somme de deux champs Price est maintenant affiché dans le champ Total.

# 6.5.2 Importer une fonction XQuery 1.0 personnalisée

Lorsque XQuery est sélectionné en tant que langage de transformation de mappage, MapForce affiche les bibliothèques de fonctions intégrées disponibles pour XQuery dans la fenêtre de Bibliothèque. Si nécessaire, vous pouvez élargir la liste avec des fonctions Xquery personnalisées, en important des modules de bibliothèque 1.0 XQuery personnalisées dans MapForce.

Pour être éligible à l'importation dans MapForce, un fichier XQuery doit satisfaire aux exigences suivantes :

- Cela doit être un module de bibliothèque valide conformément à la spécification XQuery. En d'autres termes, il doit commencer avec une déclaration de module comme module namespace
   <prefix>="<namespace name"</p>
- Toutes les fonctions déclarées dans le module de bibliothèque importé doivent retourner des types de données atomiques (par exemple, xs:string, xs:boolean, xs:integer, etc). Les paramètres de fonction doivent aussi avoir des types atomiques.

#### Pour importer un module de bibliothèque XQuery :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u><sup>31</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques	×
🕞 🧱 New Design1	*
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
😢 Importations Bibliothèques globales Ajouter	

- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- 3. Recherchez le fichier de bibliothèque .xq ou .xquery et cliquez sur Ouvrir.

Les modules de bibliothèques importées apparaissent dans la fenêtre Bibliothèques, et vous pourrez glisser des fonctions spécifiques dans la zone de mappage et les utiliser comme tout autre composant de fonction MapForce, voir aussi <u>Ajouter une fonction au mappage</u><sup>652</sup>.

Si vous ne voyez pas le module de bibliothèque XQuery importé, veuillez vous assurer que XQuery est sélectionné en tant que <u>langage de transformation</u><sup>25</sup>.

Voir aussi :

Mise en place du moteur XQuery<sup>1482</sup>

# 6.5.2.1 Example: Import Custom XQuery Function

Cet exemple vous montre comment importer un module de bibliothèque XQuery de démonstration dans MapForce et appeler ses fonctions depuis un mappage. Le module de démonstration contenu dans cet exemple consiste en une seule fonction qui calcule des taxes sur les montants décimaux à 20% du montant. Dans un scénario de production, un module XQuery peut contenir plusieurs fonctions.

Toutes les fonctions déclarées dans le module XQuery doivent retourner des types atomiques et leurs paramètres doivent aussi être de types de données atomiques. Sinon, le module n'est pas éligible pour une importation dans MapForce.

Vous trouverez le fichier de module XQuery de démonstration dans le chemin suivant relatif à votre dossier personnel "Documents" sur l'ordinateur dans lequel MapForce est installé :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\module.xq.

```
xquery version "1.0";
module namespace demo="http://www.altova.com/mapforce/demo";
declare function demo:calculatetax($val as xs:decimal) as xs:decimal {
    $val*0.2
};
```

module.xq

Une fois avoir importé le fichier de module XQuery dans MapForce, vous pourrez appeler la fonction <u>demo:calculatetax</u> depuis un mappage. Veuillez noter que le calcul du montant des impôts à l'aide d'une fonction XQuery n'est destinée qu'à des fins de démonstration, vous pouvez obtenir le même résultat en utilisant les fonctions built-in de MapForce. Un mappage de démonstration qui appelle la fonction <u>demo:calculatetax</u> ci-dessus est disponible sous le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\CalculateTax\_XQuery.mfd. Lorsque vous ouvrez tout d'abord ce mappage, MapForce affiche un avertissement qu'il contient un ou plusieurs composants qui ne sont pas disponibles dans XQuery. Cet avertissement est normal et se produit parce que le mappage référence une fonction provenant d'un module de bibliothèque XQuery personnalisé qui n'a pas encore été importé. Afin de supprimer l'avertissement et exécuter le mappage, nous allons importer le module XQuery manquant dans MapForce comme indiqué ci-dessous.

## Pour importer le module XQuery dans MapForce :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u><sup>31</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques	×
🗆 🧱 New Design1	*
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
😢 Importations Bibliothèques globales Ajouter	

- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- 3. Rechercher <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\module.xq et cliquer sur Ouvrir. Un message apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée.

MapFor	ce X
	Une nouvelle bibliothèque a été ajoutée. Voir la fenêtre de Messages pour plus de détails.
	Si vous ne voyez pas les fonctions récemment ajoutées dans la fenêtre des Bibliothèques, veuillez vous assurer que le langage correct a été sélectionné dans le menu Sortie.
	Ne plus afficher ce message.
	ОК

La bibliothèque importée et la fonction demo:calculatetax sont maintenant visibles dans la fenêtre Bibliothèques.

Bibliothèques	×
🔁 core	~
🔁 module	
demo:calculatetax result = demo:calculatetax(val)	_
距 xpath2	~
Chercher les fonctions	] -
Ajouter/Supprimer les bibliothèques	
🗮 Bibliothèques	

De plus, le mappage peut maintenant être validé et exécuté sans avertissements. La fonction demo:calculatetax illustrée dans l'image ci-dessous provient du module XQuery importé et elle peut être ajoutée au mappage comme toute autre fonction intégrée, voir <u>Ajouter une fonction dans le mappage</u><sup>552</sup>.



CalculateTax\_XQuery.mfd

#### Explication du mappage

Le mappage **CalculateTax\_XQuery.mfd** illustré ci-dessus prend en tant qu'entrée un fichier XML qui stocke des articles. Chaque article possède un seul prix, exprimé dans une valeur décimale, par exemple :

```
<Articles>
<Article>
<Article>
<Number>1</Number>
<Name>T-Shirt</Name>
<SinglePrice>25</SinglePrice>
</Article>
<Article>
<Number>2</Number>
<SinglePrice>2.30</SinglePrice>
</Article>
</Article>
```

Fonctions

```
<Article>
    <Number>3</Number>
        <Number>3</Number>
        <Name>Pants</Name>
        <SinglePrice>34</SinglePrice>
        </Article>
        <Number>4</Number>
        <Number>4</Number>
        <SinglePrice>57.50</SinglePrice>
        </Article>
</Article>
</Article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article><//article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article></article>
```

### Articles.xml

Le mappage produit un fichier XML qui suit les mêmes règles que le fichier XML source. Ainsi, les composants de source et de cible ont tous les deux la même structure dans le mappage. Comme le suggère les connexions de mappage, presque tous les éléments sont mappés de manière claire de la cible à la source ; par exemple, pour chaque **Article** dans la source, il y aura un **Article** dans la cible. Les valeurs de tous les items sont copiés verbatim depuis le XML de source, sauf en ce qui concerne **SinglePrice**. La valeur de **SinglePrice** est calculée à l'aide des deux fonctions :

- La fonction XQuery demo: calculatetax calcule le montant de l'impôt en prenant le **SinglePrice** original en tant qu'entrée.
- La fonction intégrée MapForce add ajoute le montant de l'impôt au montant **SinglePrice** original et retourne le montant final.

Chose importante, le type de données de l'item **SinglePrice** est xs:decimal; ce qui correspond au type de paramètre d'entrée et au type de retour de la fonction XQuery.

La sortie produite par le mappage lorsque vous cliquez sur l'onglet **Sortie** est illustrée ci-dessous. Veuillez noter l'augmentation de 20% appliquée à chaque prix comparé au XML source.



# 6.5.3 Importer des bibliothèques Java et .NET personnalisées

Cette section explique comment importer les fichiers de classe Java compilés et les assemblies .NET DLL (y compris les assemblies .NET 4.0) dans MapForce. Si les bibliothèques importées contiennent des fonctions qui utilisent des types de données de base en tant que paramètres et retournent des types simples, ces fonctions apparaissent dans la fenêtre **Bibliothèques**, et peuvent être utilisées dans des mappages comme dans tout autre fonction disponible dans MapForce. La sortie de mappage des fonctions importées Java et .NET peuvent être consultées dans le volet **Sortie** et les fonctions sont disponibles dans le code généré. Pour en savoir plus sur comment importer les bibliothèques personnalisées, voir les exemples fournis dans Importer <u>Classe Java personnalisée</u> et Importer un .NET DLL Assembly personnalisé

## Important :

- Afin d'importer des fonctions Java ou .NET, vous avez besoin de classes Java compilées (.class) ou des fichiers .NET.dll assembly. L'importation de fichiers .jar files ou .dll qui ne sont pas un .NET assembly n'est pas pris en charge.
- Les fichiers .NET assembly sont pris en charge lorsque le langage de mappage est défini sur C#. Les .NET assemblies peuvent être écris dans des langages .NET différents de C# (par exemple, C++.NET ou VB.NET), s'ils utilisent uniquement des types de données de base depuis le System Assembly en tant que paramètres et types de retour. Pour les détails, voir <u>Prise en charge de la fonction .NET</u><sup>608</sup>.
- Si vous voulez utiliser des fonctions .NET personnalisées dans l'aperçu de sortie built-in (dans le volet **Sortie**), ces fonctions doivent être compilées pour .NET Framework 4.x ou .NET Standard 2.0.

- Des fichiers de classes Java compilées .class) sont prises en charge lorsque le langage de mappage est défini sur Java. Java Runtime Environment 7 ou plus récent doit être installé sur votre ordinateur. Seuls des types et membres spécifiques sont pris en charge (voir <u>Prise en charge de la fonction</u> <u>Java</u><sup>(005)</sup>).
- Vous ne pouvez pas définir le langage de mappage sur C++ si le mappage utilise .class ou des assemblies .NET DLL importés.
- Vous ne pouvez pas définir le langage de mappage sur XSLT si le mappage utilise des .class Java importées ou des assemblies .NET DLL importés (une fonction XSLT personnalisée qui agit en tant qu'un adaptateur devra être rédigée).
- L'importation de fonctions provenant de DLL C++ natifs est limitée et nécessite une approche particulière. Pour plus d'informations, voir <u>Référencer les bibliothèques Java, C# et C++</u> <u>manuellement</u><sup>612</sup>.
- Toutes les fonctions appelées depuis un mappage MapForce doivent retourner la même valeur à chaque fois que la fonction est appelée avec les même paramètres d'entrée. L'ordre exact et le nombre de fois qu'une fonction est appelée par MapForce ne sont pas définis.
- Dans le cas de Java, les fichiers de classe importés et leurs packages n'ont pas besoin d'être ajoutés à la variable CLASSPATH puisque le moteur d'exécution Built-In et le code Java généré ajouteront automatiquement des packages importés respectivement au classpath du moteur Java ou à Ant. Néanmoins, toute dépendance des fichiers de classe importés et des packages ne seront pas gérés automatiquement. C'est pourquoi, si des fichiers de classe Java importés ou des packages dépendent d'autres fichiers de classe, assurez-vous d'ajouter les répertoires de parent de tous les packages dépendants de la variable d'environnement CLASSPATH.

## Prise en charge de la fonction Java

Les classes de niveau supérieur, les classes membre statiques et les classes de membre non-statiques sont prises en charge :

- new <classname>(<arg1>, <arg2>, ...)
- <object>.new <member-class>(<arg1>, <arg2>, ...)

Les fonctions membre et les fonctions statiques sont prises en charge :

- <function>(<arg1>, <arg2>, ...)
- <object>.<method>(<arg1>, ...)

Les connexions prises en charge entre Schéma XML et les types Java :

Type Schéma	Type Java
xs:string	String
xs:byte	byte
xs:short	short
xs:int	int

Type Schéma	Type Java
xs:long	long
xs:boolean	booléenne
xs:float	float
xs:double	double
xs:decimal	java.math.BigDecimal
xs:integer	java.math.BigInteger

Des connexions dans les deux sens sont possibles. D'autres types Java (y compris les types array) ne sont pas pris en charge. Les méthodes pour utiliser de tels paramètres ou des valeurs de retour seront ignorées. Les types d'objet sont pris en charge en appelant leur constructeur, ou en tant que valeur de retour d'une méthode. Ils peuvent être mappés à d'autres méthodes Java. Il n'est pas possible de manipuler l'objet en utilisant MapForce.

## Prise en charge de la fonction .NET

Les classes de niveau supérieur et les classes membre sont prises en charge :

• new <classname>(<arg1>, <arg2>, ...)

Les fonctions membre et les fonctions statiques sont prises en charge :

- <function>(<argl>, <arg2>, ...)
- <object>.<method>(<arg1>, ...)

Les connexions prises en charge entre Schéma XML et les types .NET/C# :

Type Schéma	Type .NET	Туре С#
xs:string	System.String	string
xs:byte	System.SByte	sbyte
xs:short	System.Int16	short
xs:int	System.Int32	int
xs:long	System.Int64	long
xs:unsignedByte	System.Byte	byte
xs:unsignedShort	System.UInt16	ushort
xs:unsignedInt	System.UInt32	uint
xs:unsignedLong	System.UInt64	ulong
xs:boolean	System.Boolean	bool

Type Schéma	Type .NET	Type C#
xs:float	System.Single	float
xs:double	System.Double	double
xs:decimal	System.Decimal	decimal

Des connexions dans les deux sens sont possibles. Les autres types .NET/C# (y compris les types array) ne sont pas pris en charge. Les méthodes pour utiliser de tels paramètres ou des valeurs de retour seront ignorées. Les types d'objet sont pris en charge en appelant leur constructeur, ou en tant que valeur de retour d'une méthode. Ils peuvent être mappés à d'autres méthodes .NET. Il n'est pas possible de manipuler l'objet en utilisant MapForce.

## Problèmes et contournements de types de données

Lorsqu'une fonction dans votre bibliothèque personnalisée attend des types d'entier, connectant des constantes de type Number aux arguments de la fonction, ceci peut causer une erreur d'incompatibilité semblable à celle-ci : No match for MyCustomClassLibrary, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null.MyCustomClassLibrary.Converter.AddValues(MyCustomClassLibrary.Converter, xs:decimal, xs:decimal). Vérifier types d'argument. Ce problème est spécifique aux constantes de type Number uniquement. Un exemple de mappage qui pourrait générer cette erreur est affiché ci-dessous. Dans ce mappage, deux constantes de type Number sont connectées aux arguments de type de la fonction Integer.



Les contournements possibles sont décrits ci-dessous :

- 1. Changer le type de constante de **Number** à **All other**. Pour ce faire, double-cliquez sur la barre de titre du composant de la constante.
- 2. Au lieu d'une constante, utilisez un composant de source (par exemple, un fichier XML) qui fournit des valeurs du type de données attendues par la fonction.



3. Dans votre code externe, créez une fonction wrapper qui accepte une valeur décimale et retourne une valeur d'entier. La solution wrapper peut être importée en tant que bibliothèque séparée. C'est pourquoi vous ne devez pas changer le code source d'origine de la fonction cible pour utiliser cette approche.

# 6.5.3.1 Exemple : Importer une classe Java personnalisée

Cet exemple illustre comment importer un fichier Java .class personnalisé dans MapForce.

**Note :** Java SE 8 Runtime Environment ou une version plus élevée est nécessaire pour compléter cet exemple.

#### Importation de Java .class

Pour ajouter un fichier Java .class en tant que bibliothèque MapForce, suivez les étapes suivantes :

Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u>
 La fenêtre Gérer les Bibliothèques s'ouvre (voir la capture d'écran ci-dessous).

Gérer les bibliothèques	×
🕞 🚋 New Design1	*
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
😢 Importations Bibliothèques globales Ajouter	

- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- 3. Recherchez le fichier suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Java\Format.Format.class. Un message
apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée. La bibliothèque importée est
maintenant visible dans la fenêtre Bibliothèques (voir la capture d'écran ci-dessous).

Libraries		×	
E core		$\sim$	
🛃 db			
🖅 edifact			
E Format.F	ormat		
Forma	t.Format		
Forma	t.Format.FormatNumber		
記 記 記 記 記 文 文 文 文			
		$\sim$	
Search for fu	nction	•	
	Add/Remove Libraries		
🗮 Libraries	🛅 Project		

Si vous ne voyez pas la nouvelle bibliothèque importée dans la fenêtre **Bibliothèques**, veuillez vous assurer que le <u>langage de transformation</u><sup>(26)</sup> est configuré sur Java. Pour ajouter la fonction au-dessus du mappage, glissez-la depuis la fenêtre **Bibliothèques** vers la zone de mappage. Pour plus d'informations, voir <u>Ajouter une fonction au mappage</u><sup>(52)</sup>.

### <u>Sortie de mappage</u>

Pour réaliser cet sortie de mappage dans MapForce, suivons les étapes suivantes :

1. Ouvrir le mappage suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Java\FormatNumber.mfd. || s'agit d'un mappage complet qui importe d'ores et déjà la bibliothèque .class Java mentionnée plus haut.

2. Cliquez sur la touche Sortie pour voir le résultat du mappage (voir la capture d'écran ci-dessous).

1	Start date,End date,Region,Amount	
2	2008-01-01,2008-01-31,CA,"110.400,00"	
3	2008-01-01,2008-01-31,MA,"75.300,00"	
4	2008-02-01,2008-02-29,CA,"114.300,00"	
5	2008-02-01,2008-02-29,MA,"65.200,00"	
6	2008-03-01,2008-03-31,CA,"134.200,00"	
7	2008-03-01,2008-03-31,MA,"86.100,00"	
8	2008-04-01,2008-04-30,CA,"107.300,00"	
9	2008-04-01,2008-04-30,MA,"112.100,00"	
10	2008-05-01,2008-05-31,CA,"114.400,00"	
11	2008-05-01,2008-05-31,MA,"93.800,00"	

#### Mappage dans Java

Pour exécuter le mappage dans Java, suivez les instructions ci-dessous :

- 1. Cliquez sur Générer code dans | Java dans le menu fichier.
- 2. Choisissez un répertoire cible dans lequel le code doit être généré et cliquez sur OK.
- 3. Importer les bibliothèques générées dans votre projet Java et construisez l'application Java. Pour plus d'informations, voir <u>Exemple : Générer et exécuter du Code Java</u><sup>1057</sup>.

# 6.5.3.2 Exemple : Importer un .NET DLL Assembly personnalisé

Cet exemple illustre comme importer dans MapForce un DLL .NET personnalisé créé dans C#. Le code source de cet échantillon est disponible dans le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\C#\Format. Le fichier assembly .dll qui sera importé dans MapForce se trouve dans le répertoire ..\bin\Debug. Vous pouvez aussi ouvrir le fichier de solution .sln dans Visual Studio et compiler un nouveau fichier .dll.

**Note :** si vous voulez utiliser des fonctions .NET personnalisées dans l'aperçu de sortie built-in (dans le volet **Sortie**), ces fonctions doivent être compilées pour .NET Framework 4.x ou .NET Standard 2.0.

#### importation de .NET assembly

Pour importer un fichier .NET assembly, suivez les étapes suivantes :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u><sup>(31)</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).

Gérer les bibliothèques	×
🗆 🛃 New Design1	*
Fonctions définies par l'utilisateur	
Importations Bibliothèque propre Ajouter	
😢 Importations Bibliothèques globales Ajouter	

- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- 3. Recherchez Format.dll dans le répertoire suivant : ...\MapForceExamples\C#\Format\bin\Debug\. Un message apparaît vous informant qu'une nouvelle bibliothèque a été ajoutée. La bibliothèque importée est maintenant visible dans la fenêtre **Bibliothèques**.

Libraries	×
n core	$\sim$
and the second s	
and the second s	
E Format, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null	
Format.FormatNumber	_
new Format	
至 lang 至 xbrl 至 xlsx	
Search for function	•
Add/Remove Libraries	
🗮 Libraries    Project	

Si vous ne voyez pas la nouvelle bibliothèque importée dans la fenêtre **Bibliothèques**, veuillez vous assurer que le <u>langage de transformation</u><sup>25</sup> est configuré sur C#. Pour ajouter la fonction au-dessus du mappage, glissez-la depuis la fenêtre **Bibliothèques** vers la zone de mappage. Pour plus d'information, voir <u>Ajouter une fonction au mappage</u><sup>552</sup>.

### Sortie de mappage

Pour réaliser cet sortie de mappage, suivons les étapes suivantes :

- 1. Ouvrir le fichier FormatNumber.mfd disponible dans le dossier suivant : ...\MapForceExamples\C#. Il s'agit d'un mappage échantillon qui a une bibliothèque .dll importée mentionnée ci-dessus.
- 2. Cliquez sur la touche Sortie pour voir le résultat du mappage (voir la capture d'écran ci-dessous).

1	Start date,End date,Region,Amount	
2	2008-01-01,2008-01-31,CA,"110.400,00"	
3	2008-01-01,2008-01-31,MA,"75.300,00"	
4	2008-02-01,2008-02-29,CA,"114.300,00"	
5	2008-02-01,2008-02-29,MA,"65.200,00"	
6	2008-03-01,2008-03-31,CA,"134.200,00"	
7	2008-03-01,2008-03-31,MA,"86.100,00"	
8	2008-04-01,2008-04-30,CA,"107.300,00"	
9	2008-04-01,2008-04-30,MA,"112.100,00"	
10	2008-05-01,2008-05-31,CA,"114.400,00"	
11	2008-05-01,2008-05-31,MA,"93.800,00"	

#### Mappage dans C#

Pour exécuter le mappage depuis une application C# personnalisée, suivez les instructions ci-dessous :

- 1. Cliquez sur Générer code dans | C# dans le menu fichier.
- 2. Choisissez un répertoire cible dans lequel le code doit être généré et cliquez sur OK.
- Créez l'application avec Visual Studio, et exécutez l'application de console générée. Pour plus de détails, voir <u>Générer le code C#</u><sup>1074</sup>.

# 6.5.4 Référencer les bibliothèques Java, C# et C++ manuellement

Cette section explique comment référencer des bibliothèques personnalisées dans un fichier .mff (fichier de fonction MapForce). Le fichier .mff contenant la référence peut ensuite être importé en tant que bibliothèque MapForce. Un fichier .mff est un fichier XML dans lequel vous définissez manuellement le lien entre les définitions de classe dans votre code personnalisé et MapForce. Une fois que vous avez créé le fichier .mff personnalisé, vous pouvez l'importer dans MapForce, de la même manière que vous importez un fichier .NET DLL ou un fichier de classe Java.

### Important :

- Si vous voulez utiliser des fonctions .NET personnalisées dans l'aperçu de sortie built-in (dans le volet **Sortie**), ces fonctions doivent être compilées pour .NET Framework 4.x ou .NET Standard 2.0.
- Vous pouvez importer une fonction dans MapForce uniquement si votre type de retour et les paramètres sont de type simple. Pour en savoir plus sur la liste de types de données disponible dans chaque langage, voir <u>Mappage de type de données</u><sup>618</sup>.
- Lorsque vous importez des librairies de fonction depuis les fichiers .mff personnalisés, la consultation préalable du mappage directement dans MapForce (en cliquant sur la touche Sortie) est limitée. Pour des bibliothèques écrites dans C++, la consultation du mappage dans MapForce n'est pas prise en charge. Pour Java et C#, la consultation est disponible quand votre bibliothèque utilise des types de langage natif, mais n'est pas disponible si votre bibliothèque importe les classes générées d'Altova. Toutefois, vous pouvez générer du code dans le langage spécifique ciblé par votre bibliothèque. Les fonctions personnalisées seront disponibles dans le code généré, vous permettant d'exécuter le mappage depuis le code généré.
- L'ordre exact dans lequel des fonctions sont appelées par le code de mappage généré n'est pas défini. MapForce peut nécessiter de mettre en cache des résultats calculés à réutiliser ou évaluer des expressions dans n'importe quel ordre. Nous vous recommandons donc fortement de n'utiliser que les fonctions personnalisées qui ne présentent pas d'effets secondaires.
- Il est important de distinguer entre des fonctions définies par l'utilisateur et des bibliothèques de fonction personnalisée. Les fonctions définies par l'utilisateur sont créées graphiquement dans un mappage, et elles ne peuvent pas et de doivent pas être enregistrées dans un fichier .mff, parce qu'elles sont enregistrées ensemble avec le fichier de mappage .mfd dans lequel elles ont été créées. Pour plus d'informations, voir <u>appeler et importer des FDU</u><sup>576</sup>.
- Si vous êtes en train de mettre à jour depuis une version antérieure à 2010, vous devrez mettre à jour les types de données utilisées dans vos fonctions personnalisées. Pour plus de détails, voir <u>Mappage</u> <u>de type de données</u><sup>613</sup>.

Pour plus d'instructions concernant la création et la configuration d'un fichier personnalisé .mff, voir <u>Configurer</u> <u>le fichier .mff</u><sup>613</sup>. Les exemples sont fournis dans les rubriques suivantes :

- <u>Référencer C# dans .mff</u><sup>621</sup>
- <u>Référencer C++ dans .mff</u><sup>622</sup>
- Référencer Java dans .mff<sup>625</sup>
# 6.5.4.1 Configurer le fichier .mff

Cette rubrique fournit des instructions comment configurer un fichier MapForce Function File (.mff). Un fichier .mff est un fichier de configuration de format XML qui permet d'importer des fonctions depuis les bibliothèques Java, C# ou C++ personnalisées dans MapForce pour qu'elles apparaissent dans la fenêtre **Bibliothèques**. Un fichier .mff est un intermédiaire entre vos bibliothèques personnalisées et MapForce. Le fichier .mff doit être configuré pour spécifier i) les interfaces pour les fonctions personnalisées et ii) pour trouver la mise en œuvre dans le code généré.

### Important :

- Les fichiers \*.mff doivent être valides par rapport au schéma suivant : C:\Program Files\MapForceLibraries\mff.xsd. Le schéma mff.xsd définit la configuration de bibliothèque personnalisée et est pour utilisation interne uniquement. Altova GmbH se réserve le droit de modifier ce format de fichier lors des nouvelles publications.
- Il est possible de ne définir qu'une seule classe C#, C++, ou Java par fichier .mff.

### Échantillon .mff pour C#

L'extrait de code suivant illustre un fichier d'échantillon .mff pour C++ :

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="cpp">
         <setting name="namespace" value="mylib"/>
         <setting name="class" value="Greetings"/>
         <setting name="path" value="C:\Libraries\cpp"/>
         <setting name="include" value="Greetings.h"/>
         <setting name="source" value="Greetings.cpp"/>
      </implementation>
   </implementations>
   <group name="greetings">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="cpp">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
```

</group> </mapping>

#### Bibliothèque personnalisée importée

L'image ci-dessous montre à quoi pourrait ressembler un fichier .mff personnalisé après l'importation dans MapForce. Veuillez noter que la bibliothèque personnalisée mylib apparaît en tant qu'entrée de bibliothèque (triée par ordre alphabétique), contenant la fonction string sayhello.

Libraries			×
🔁 edifact			^
🔁 lang			
🔁 mylib			
sayhe	llo	result = sayhello(ismorning)	
Irdx 편			~
Search for fu	inction		•
	A	dd/Remove Libraries	
🗮 Libraries	Pro	oject	

## Étapes de configuration

Pour configurer le fichier .mff, suivez les instructions ci-dessous.

#### Étape 1. Configurer le nom de bibliothèque

Vous trouverez le nom de bibliothèque du fichier .mff file (*voir ci-dessous*). Par convention, les noms de bibliothèque sont écrits en minuscule dans MapForce ; néanmoins, vous pouvez aussi utiliser des majuscules.

```
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="mff.xsd">
```

Dans l'échantillon ci-dessus, l'entrée qui apparaîtra dans la fenêtre Bibliothèques sera appelée mylib.

#### Étape 2. Configurer les mises en œuvre du langage

L'élément <implementations> est un élément obligatoire qui spécifie quels langages votre bibliothèque doit prendre en charge et qui doit être ajoutée en tant qu'enfant de <mapping>, (voir l'exemple ci-dessous)

```
</implementation>
</implementations>
```

Les paramètres dans le cadre de chaque élément <implementation> permettent au code généré d'appeler les fonctions spécifiques définies dans Java, C++ ou C#. Un fichier .mff peut être rédigé de manière à ce qu'il cible plus d'un langage de programmation. Dans ce cas, chaque langage supplémentaire doit contenir un élément <implementation> supplémentaire. Les paramètres spécifiques pour chaque langage de programmation sont discutés ci-dessous.

Référence de bibliothèque Java

Il est important que le code généré puisse trouver votre fichier **Greetings.class**. C'est pourquoi vous devez vous assurer que votre classe a été ajoutée au chemin de classe Java. <u>Référence de bibliothèque C#</u>

Pour C#, il est important que l'espace de noms dans le code corresponde à l'espace de noms défini dans le fichier .mff (dans le listing du code ci-dessus, l'espace de noms est MyLibrary). La même chose vaut pour le nom de classe (dans l'extrait de code ci-dessus, le nom de classe est Greetings). Le troisième paramètre, reference, fournit le chemin du dll qui doit être lié au code généré. Référence de bibliothèque C++

```
<!-- ... -->
    <implementation language="cpp">
        <setting name="namespace" value="MyLibrary"/>
            <setting name="class" value="Greetings"/>
            <setting name="path" value="C:\Libraries\cpp"/>
            <setting name="include" value="Greetings.h"/>
            <setting name="source" value="Greetings.cpp"/>
            </implementation>
<!-- ... -->
```

En ce qui concerne C++, veuillez noter :

- namespace est l'espace de noms dans lequel votre classe Greetings sera défini. Il doit être égal à l'attribut library dans l'élément mapping.
- path est le chemin dans lequel se trouvent l'include et les fichiers de source.

 Lors de la génération de code pour un mappage, l'include et les fichiers de source seront copiés dans le répertoire targetdir/libraryname, qui est défini lors de la sélection de la commande Fichier | Générer code dans | C++, et inclus dans le fichier de projet.

Tous les fichiers include que vous fournissez seront inclus dans l'algorithme généré.

#### Étape 3. Ajouter un composant

Dans la fenêtre **Bibliothèques**, chaque fonction apparaît imbriquée sous un groupe de fonctions, par exemple des "string functions". Dans le fichier .mff, une fonction correspond à un élément <component>. En revanche chaque <component> doit être imbriqué sous un élément <group>, par exemple :

```
<!-- ... -->
<group name="string functions">
        <component name="sayhello">
        <!-- ... -->
        </component>
</group>
<!-- ... -->
```

Le code affiché ci-dessous définit une fonction d'échantillon (composant) appelée sayhello.

```
<!-- ... -->
<component name="sayhello">
   <sources>
      <datapoint name="ismorning" type="xs:boolean"/>
   </sources>
   <targets>
      <datapoint name="result" type="xs:string"/>
   </targets>
   <implementations>
   <!-- ... -->
   </implementations>
   <description>
     <short>result = sayhello(ismorning)</short>
     <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
   </description>
</component>
<!-- ... -->
```

Voici à quoi ressemblerait le composant ci-dessus dans MapForce :



Dans le listing de code ci-dessus, l'élément <datapoint> peut être librement défini en tant que paramètre d'entrée ou de sortie d'une fonction (aussi nommé connecteur d'entrée ou de sortie). L'argument type de l'élément <datapoint> spécifie le type de données du paramètre ou le type de données de la valeur de retour. Un seul point de cible est autorisé pour chaque fonction. Le nombre de points de données source que vous pouvez définir n'est pas limité.

#### Fonctions

Le type de données de chaque point de données doit être un des types de Schéma XML (par exemple, xs:string, xs:integer, etc.). Ces types de données doivent correspondre aux types de données des paramètres de fonction que vous avez défini dans votre bibliothèque Java, C++ ou C#. Pour en savoir plus sur le mappage des types de données de schéma XML dans les types de langage, voir <u>Mappage de type de données</u>

Les fonctions sont accompagnées par des descriptions brèves et longues dans la fenêtre **Bibliothèques**. La description brève est toujours affichée à droite du nom de fonction, alors que la description longue est affichée en tant qu'infobulle lorsque vous placez le curseur de la souris sur la description brève (*voir la capture d'écran ci-dessous*).

Libraries	×
🔁 core	~
置 db	result = sayhello(ismorning)
🛃 edifact	
🖅 lang	Returns "Good morning" or "Good day", depending on the input parameter.
🔁 mylib	
sayhello result = sayhello(isr	morning)
I	×
Search for function	▼
Add/Remove Libraries	
崖 Libraries	

#### Étape 4. Définir les mises en œuvre de langage

Nous pouvons désormais connecter la fonction dans la fenêtre **Bibliothèques** avec la fonction dans les classes personnalisées Java, C# ou C++. Cela s'effectue par le biais de l'élément <implementation>. Une fonction peut avoir plusieurs éléments <implementation> - un pour chaque langage de programmation pris en charge. Une fonction peut être appelée Hello dans Java ou SayHello dans C++. C'est pourquoi vous devez spécifier un nom de fonction séparé pour chaque langage de programmation. Une fonction pour chacun des langages de programmation pourrait ressembler à l'exemple suivant :

```
<!-- ... -->
<component name="sayhello">
<!-- ... -->
   <implementations>
      <implementation language="cs">
         <function name="HelloFunction"/>
      </implementation>
      <implementation language="java">
         <function name="Hello"/>
      </implementation>
      <implementation language="cpp">
         <function name="SayHello"/>
      </implementation>
   </implementations>
<!-- ... -->
</component>
<!-- ... -->
```

La valeur que vous fournissez en tant que nom de fonction doit correspondre à la méthode dans la classe Java, C# ou C++.

# 6.5.4.2 Importer le fichier .mff dans MapForce

Une fois que vous avez <u>créé<sup>613</sup></u> un fichier .mff personnalisé, vous pouvez l'importer dans MapForce comme suit :

 Cliquez sur la touche Ajouter/Supprimer des bibliothèques en bas de la <u>fenêtre Bibliothèques</u> <sup>(31)</sup>. La fenêtre Gérer les Bibliothèques s'ouvre (*voir la capture d'écran ci-dessous*).



- Pour importer des fonctions en tant que bibliothèque *locale* (uniquement dans le cadre du document actuel), cliquez sur Ajouter sous le nom actuel du mappage. Pour importer les fonctions en tant que bibliothèque *globale* (au niveau du programme), cliquez sur Ajouter à côté des Importations
   Bibliothèques globales. Lorsque vous importez une bibliothèque *localement*, vous pouvez définir le chemin du fichier de bibliothèque pour qu'il soit relatif au fichier de mappage. Avec des bibliothèques importées globalement, le chemin de la bibliothèque importée est toujours absolu.
- 3. Cherchez le fichier .mff personnalisé et cliquez sur Ouvrir.

La bibliothèque importée devient visible dans la fenêtre **Bibliothèques** après avoir défini le langage de mappage sur un langage ciblé par la bibliothèque personnalisée.

Si vous enregistré le fichier \*.mff dans ...\Altova\MapForce2023\MapForceLibraries, qui est relatif aux fichiers de programme (ou au dossier fichiers de programme (x86)), la bibliothèque est automatiquement chargée dans la fenêtre **Bibliothèques** lorsque vous démarrez MapForce. Les bibliothèques et leurs fonctions peuvent être activées ou désactivées, en supprimant ou en ajoutant le fichier de bibliothèque respectif (\*.mff).

# 6.5.4.3 Mappage de type de données

La table ci-dessous recense les types de données pris en charge en tant que types de retour de fonction et types de paramètres lorsque vous créez des fichiers .mff personnalisées qui référence des biblitohèques Java, C# et C++. La table contient les types de données natifs et non-natifs. Si vous nécessitez une prise en charge pour des types de données non-natifs comme les types de date, heure et durée Altova, vos bibliothèques Java et C# personnalisées doivent inclure une référence aux bibliothèques Altova. Dans le cas de C++, les bibliothèques Altova doivent toujours être importées. Pour plus d'informations concernant la génération des bibliothèques Altova, voir Générateur de code

Type de Schéma XML	Type Java	Туре С#	Туре С++
anyAtomicType	String	string	string_type
anySimpleType	String	string	string_type
anyURI	String	string	string_type
base64Binary	byte[]	byte[]	altova::mapforce::blo b
booléen	booléenne	bool	bool
byte	int	int	int
date	com.altova.types.Date Time	Altova.Types.DateTime	altova::DateTime
dateTime	com.altova.types.Date	Altova.Types.DateTime	altova::DateTime <sup>(153)</sup>
dayTimeDuration	com.altova.types.Dura tion <sup>(189)</sup>	Altova.Types.Duration	altova::Duration <sup>1156</sup>
decimal	java.math.BigDecimal	decimal	double
double	double	double	double
duration	com.altova.types.Dura tion	Altova.Types.Duration	altova::Duration <sup>1156</sup>
ENTITIES	String	string	string_type
ENTITY	String	string	string_type
float	double	double	double
gDay	com.altova.types.Date Time	Altova.Types.DateTime	altova::DateTime <sup>1153</sup>
gMonth	com.altova.types.Date	Altova.Types.DateTime	altova::DateTime
gMonthDay	com.altova.types.Date Time <sup>[1184]</sup>	Altova.Types.DateTime	altova::DateTime <sup>(1153)</sup>
gYear	com.altova.types.Date Time <sup>[184]</sup>	Altova.Types.DateTime	altova::DateTime <sup>1153</sup>
gYearMonth	com.altova.types.Date	Altova.Types.DateTime	altova::DateTime <sup>(153)</sup>
hexBinary	xBinary byte[]		altova::mapforce::blo b
ID String		string	string_type

Type de Schéma XML	Type Java	Туре С#	Туре С++
IDRef	String	string	string_type
IDREFS	String	string	string_type
int	int	int	int
integer	java.math.BigInteger	décimal	int64
language	String	string	string_type
long	long	long	int64
Nom	String	string	string_type
NCName	String	string	string_type
negativeInteger	java.math.BigInteger	decimal	int64
NMTOKEN	String	string	string_type
NMTOKENS	String	string	string_type
nonNegativeInteger	java.math.BigInteger	decimal	unsignedint64
nonPositiveInteger	java.math.BigInteger	decimal	int64
normalizedString	String	string	string_type
NOTATION	String	string	string_type
positiveInteger	java.math.BigInteger	decimal	unsignedint64
QName	javax.xml.namespace.Q Name	Altova.Types.QName	altova::QName
short	int	int	int
string	String	string	string_type
time	com.altova.types.Date	Altova.Types.DateTime	altova::DateTime <sup>(153)</sup>
token	String	string	string_type
unsignedByte	long	ulong	unsignedint64
unsignedInt	long	ulong	unsignedint64
unsignedLong	java.math.BigInteger	ulong	unsignedint64
unsignedShort	long	ulong	unsignedint64
untypedAtomic	String	string	string_type

Type de Schéma XML	Type Java	Туре С#	Туре С++
yearMonthDuration	com.altova.types.Dura tion <sup>1189</sup>	Altova.Types.Duration	altova::Duration <sup>1156</sup>

# 6.5.4.4 Référencer la bibliothèque C# dans .mff

Cet exemple vous montre comment créer une bibliothèque C# d'échantillon et la référencer à un fichier MapForce Function File (.mff). Le fichier .mff peut ensuite être importé en tant que bibliothèque MapForce. Référencer une bibliothèque C# au format .mff ne constitue qu'un des moyens pour importer des bibliothèques C# dans MapForce. Une alternative plus simple est d'importer des .NET assemblies directement. Pour plus d'informations, voir <u>Exemple : Importer un .NET DLL Assembly personnalisé</u><sup>610</sup>.

# Étapes de configuration

Pour référencer une bibliothèque C# dans un fichier .mff, suivez les instructions ci-dessous.

**Note :** si vous voulez utiliser des fonctions .NET personnalisées dans l'aperçu de sortie built-in (dans le volet **Sortie**), ces fonctions doivent être compilées pour .NET Framework 4.x ou .NET Standard 2.0.

#### Étape 1 : Créer une nouvelle bibliothèque de classe dans VS

Créer un nouveau projet de bibliothèque de classe dans Visual Studio. Veuillez noter que la fonction a été définie en tant que public static.

### Étape 2 : Ajouter une référence à Altova.dll

Si vous avez besoin de types de Schéma XML spéciaux (tels date et durée), ajoutez une référence depuis votre projet Visual Studio dans la bibliothèque Altova.dll. Pour obtenir cette bibliothèque, générez du code C# provenant d'un mappage sans fonctions personnalisées. Le fichier Altova.dll se trouvera dans le répertoire ... **Altova.bin/debug** relatif au répertoire dans lequel le code a été généré. Pour ajouter la référence à Altova.dll dans Visual Studio, cliquez sur **Ajouter référence** dans le menu du **Projet** et recherchez Altova.dll. Puis, ajoutez la ligne **utilisant Altova.Types**; à votre code. Pour toute information sur le mappage des types de données de schéma XML dans les types C#, voir <u>Mappage de type de données</u>

#### Étape 3 : Crééz votre projet VS

Créer votre projet Visual Studio. Le fichier MyLibrary.dll est généré dans votre répertoire de sortie de projet.

#### Étape 4. Créer un .mff et référencez votre bibliothèque C#

En utilisant un éditeur XML, créer un nouveau fichier .mff et valider le par rapport au schéma suivant : **c:** \Program Files\MapForceLibraries\mff.xsd. Assurez-vous que toutes les références sous implementation language="cs" pointent vers les membres C# et les chemins corrects créés précédemment. La ligne function name="SayHello" doit référer au nom de la fonction exactement tel qu'il a été défini dans C#. Pour les détails, voir <u>Configurer le fichier .mff</u><sup>GIS</sup>.

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="cs">
         <setting name="namespace" value="MyLibrary" />
         <setting name="class" value="Greetings" />
         <setting name="reference" value="C:</pre>
\Libraries\cs\MyLibrary\bin\debug\MyLibrary.dll" />
      </implementation>
   </implementations>
   <group name="string functions">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="cs">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
   </group>
</mapping>
```

#### Étape 5 : Importer le fichier .mff en tant que bibliothèque

Maintenant que notre bibliothèque personnalisée est référencée dans le fichier .mff, vous pouvez importer le fichier .mff dans MapForce en tant que bibliothèque. Pour plus d'informations, voir <u>Importer les bibliothèques</u> .mff<sup>618</sup>.

# 6.5.4.5 Référencer C++ dans .mff

Cet exemple vous montre comment créer une bibliothèque C# d'échantillon et la référencer à un fichier MapForce Function File (.mff). Le fichier .mff peut ensuite être importé en tant que bibliothèque MapForce.

## Étapes de configuration

Pour référencer une bibliothèque C++ dans un fichier .mff, suivez les instructions ci-dessous.

#### Étape 1 : Créer un fichier d'en-tête

Créer un fichier d'en-tête (.h) pour votre bibliothèque de classe. La liste de code suivante illustre un fichier d'échantillon d'en-tête appelé Greetings.h.

```
#ifndef MYLIBRARY_GREETINGS_H_INCLUDED
#define MYLIBRARY_GREETINGS_H_INCLUDED
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
using namespace altova;
namespace mylib {
   class ALTOVA_DECLSPECIFIER Greetings
   {
     public:
        static string_type SayHello(bool isMorning);
   };
   } // namespace mylib
#endif // MYLIBRARY_GREETINGS H_INCLUDED
```

Veuillez noter que la fonction a été déclarée comme étant statique, et que l'espace de noms altova est importé. Ne pas oublier d'écrire ALTOVA\_DECLSPECIFIER devant le nom de classe : cela garantit que vos classes compileront correctement, que vous utilisiez des liens dynamiques ou statiques dans le code généré.

#### Étape 2 : Créer un fichier .cpp

Créer un fichier .cpp portant le même nom que le fichier d'en-tête. Le fichier .cpp doit se trouver dans le même répertoire que le fichier .h. La liste de code suivante illustre un fichier d'échantillon .cpp appelé **Greetings.cpp** qui contient le fichier **Greetings.h** créé précédemment :

```
#include "StdAfx.h"
#include "../Altova/Altova.h"
#include "../Altova/AltovaException.h"
#include "Greetings.h"
#include "Greetings.h"
namespace mylib {
    string_type Greetings::SayHello(bool isMorning)
    {
        if( isMorning )
            return _T("Good morning!");
        return _T("Good day!");
    }
}
```

Veuillez noter les lignes qui importent le **stdAfx.h** et plusieurs bibliothèques Altova. Ces lignes ne doivent pas être modifiées. Si les chemins menant aux bibliothèques Altova sont correctes dans le code généré, ces chemins pointeront vers les fichiers respectifs. Contrairement à Java ou C#, vous ne devez pas compiler vos fichiers source C++. Ils seront copiés dans le code généré et seront compilés avec le reste du code de mappage généré.

#### Étape 3 : Créer un .mff et référencer votre bibliothèque C++

En utilisant un éditeur XML, créer un nouveau fichier .mff et valider le par rapport au schéma suivant : C: \Program Files\MapForceLibraries\mff.xsd. Veuillez vous assurer que les espaces de noms, noms de fonction ainsi que les types de données définis ici correspondent à ceux contenus dans le code C++, comme décrit dans <u>Configurer le fichier .mff</u><sup>613</sup>. Pour information sur la prise en charge des types de données, voir <u>Mappage de type de données</u><sup>613</sup>.

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="cpp">
         <setting name="namespace" value="mylib"/>
         <setting name="class" value="Greetings"/>
         <setting name="path" value="C:\Libraries\cpp"/>
         <setting name="include" value="Greetings.h"/>
         <setting name="source" value="Greetings.cpp"/>
      </implementation>
   </implementations>
   <group name="greetings">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="cpp">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
   </group>
</mapping>
```

#### Étape 4 : Importer le fichier .mff en tant que bibliothèque

Maintenant que notre bibliothèque personnalisée est référencée dans le fichier .mff, vous pouvez importer le fichier .mff dans MapForce en tant que bibliothèque. Pour plus d'information, voir <u>Importer le fichier .mff<sup>618</sup></u>.

#### Erreurs de compilation C++

Afin de pouvoir exécuter des mappages qui utilisent des bibliothèques C++ natives, vous devrez générer du code C++ et exécuter le mappage depuis votre code C++ ou votre application, comme décrit dans <u>Générer</u> <u>code C++</u><sup>(103)</sup>. S vous recevez une erreur de compliation dans <u>#import "msado15.dll" rename("EOF",</u> "EndofFile"), modifiez les propriétés de projet pour inclure une référence à <u>msado15.dll</u> dans <u>C:\Program</u> Files\Common Files\System\ADO.

# 6.5.4.6 Référencer Java dans .mff

Cet exemple vous montre comment créer une bibliothèque Java library d'échantillon et la référencer à un fichier MapForce Function (.mff). Le fichier .mff peut ensuite être importé en tant que bibliothèque MapForce. Référencer une bibliothèque Java dans un fichier .mff ne constitue qu'un des moyens pour importer des bibliothèques Java dans MapForce. Une alternative plus simple est d'importer des fichiers de .class Java directement. Pour plus d'informations, voir <u>Exemple : Importer une classe Java personnalisée</u>

## Étapes de configuration

Pour référencer une bibliothèque C# dans un fichier .mff, suivez les instructions ci-dessous.

#### Étape 1. Créer un nouveau projet

Créer un nouveau projet Java dans votre environnement de développement de prédilection (par exemple, Eclipse).

#### Étape 2. Ajouter le package com.mylib

Ajouter au projet un nouveau package appelé mylib qui consiste en une classe appelée Greetings. Dans l'extrait de code ci-dessous, veuillez noter que la fonction SayHello a été définie en tant que public static.

```
package com.mylib;
public class Greetings {
    public static String SayHello ( boolean isMorning ) {
        if( isMorning )
            return "Good Morning!";
        return "Good Day!";
    }
}
```

#### Étape 3. Importer les com.altova.types

En option, si votre projet nécessite une prise en charge pour des types de schéma spéciaux comme, par exemple, la date, l'heure et la durée, importer le package com.altova.types. Pour obtenir ce package, générer du code Java depuis un mappage sans fonctions personnalisées : importer com.altova.types.\*;

#### Étape 4. Compilez votre bibliothèque personnalisée

Compilez votre bibliothèque personnalisée dans un fichier de classe et l'ajouter au classpath Java.

#### Étape 5. Créez un .mff et référencez votre bibliothèque Java library

En utilisant un éditeur XML, créer un nouveau fichier .mff et valider le par rapport au schéma suivant : C: \Program Files\MapForceLibraries\mff.xsd. Veuillez vous assurer que toutes les références sous implementation language="java" pointent vers les membres Java corrects créés précédemment. Aussi, la ligne function name="sayHello" doit référer au nom de la fonction exactement tel qu'il a été défini dans Java. Pour les plus détails, voir <u>Configurer le fichier .mff</u><sup>613</sup>.

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="custom" xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="java">
         <setting name="package" value="com.mylib"/>
         <setting name="class" value="Greetings"/>
      </implementation>
   </implementations>
   <group name="greetings">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="java">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
   </group>
</mapping>
```

#### Étape 6. Importer le fichier .mff en tant que bibliothèque

Maintenant que notre bibliothèque personnalisée est référencée dans le fichier .mff, vous pouvez importer le fichier .mff dans MapForce en tant que bibliothèque. Pour plus d'information, voir <u>Importer les Bibliothèques</u> .mff<sup>618</sup>.

# 6.6 Expressions régulières

Lors de la conception d'un mappage MapForce, vous pouvez utiliser des expressions régulières ("regex") dans les contextes suivants :

- Dans les paramètres **pattern** des fonctions match-pattern<sup>769</sup> et tokenize-regexp<sup>720</sup>
- Pour filtrer les nœuds dans lesquels une fonction de nœud doit s'appliquer. Pour plus d'informations, voir <u>Appliquer des fonctions de nœud et les défauts par condition</u><sup>565</sup>.

La syntaxe et la sémantique d'expression régulière pour XSLT et XQuery sont tels que définis dans <u>Annexes F</u> <u>de "XML Schema Part 2: Datatypes Second Edition"</u>.

**Note :** Lors de la génération de code C++, C# ou Java, les fonctions avancées de la syntaxe d'expression régulière peuvent différer légèrement. Voir la documentation regex de chaque langage pour plus d'informations.

## Terminologie

Examinons la terminologie d'expression régulière basique en analysant la fonction tokenize-regexp en tant qu'un exemple. Cette fonction partage du texte dans une séquence de strings, avec l'aide des expressions régulières. Pour ce faire, la fonction prend les paramètres d'entrée suivants :

input	Le string d'entrée à traiter par la fonction. L'expression régulière fonctionnera sur ce string.
pattern	Le pattern d'expression régulière à appliquer.
flags	Paramètre optionnel qui définit des options supplémentaires (flags) qui déterminent comment l'expression régulière est interprétée, voir "Flags" ci-dessous.

Dans le mappage ci-dessous, le string d'entrée est "Altova MapForce". Le paramètre **pattern** est un caractère d'espace et aucun flag d'expression régulière n'est utilisé.



En conséquence, le texte est partagé à chaque fois que le caractère d'espace se produit, le résultat de mappage est donc le suivant :

<items></items>	
<item>Altova</item>	
<item>MapForce</item>	

Veuillez noter que la fonction tokenize-regexp exclut les caractères correspondants du résultat. Autrement dit, le caractère d'espace dans cet exemple est omis de la sortie.

L'exemple ci-dessus est très basique et le même résultat peut être obtenu sans expression régulière, avec la fonction <u>tokenize</u><sup>715</sup>. Dans un scénario plus pratique, le paramètre **pattern** contiendrait une expression régulière plus complexe. L'expression régulière peut consister en :

- Littéraux
- Classes de caractère
- Gammes de caractère
- Classe niées
- Caractères Meta
- Quantificateurs

### Littéraux

Utiliser des littéraux pour faire correspondre les caractères exactement tels qu'ils sont écrits (littéralement). Par exemple, si le string d'entrée est abracadabra, et **pattern** est le littéral br, le résultat sera :

```
<items>
<item>a</item>
<item>acada</item>
<item>a</item>
</items>
```

L'explication est que le littéral **br** avait deux correspondances dans le string d'entrée **abracadabra**. Une fois avoir retiré les caractères correspondants du résultat, la séquence de trois strings illustrée ci-dessus est produite.

## Classes de caractère

Si vous contenez un ensemble de caractères dans des crochets ([ et ]), cela crée une classe de caractère. Un seul (et uniquement un seul) des caractères contenu dans la classe de caractère a une correspondance, par exemple :

- Le pattern [aeiou] correspond à toute voyelle de casse minuscule.
- Le pattern [mj]ust correspond à "must" et "just".
- **Note :** Le pattern est sensible à la casse, donc un "a" de casse minuscule ne correspond pas à la casse majuscule "A". Pour rendre la correspondance insensible à la casse, utiliser le flag **i**, voir ci-dessous.

#### Gammes de caractère

Utiliser [a-z] pour créer une gamme entre les deux caractères. Seul un des caractères trouvera une correspondance un à la fois. Par exemple, le pattern [a-z] correspond à tout caractère de casse minuscule entre "a" et "z".

#### Classes niées

L'utilisation du caret ( ^ ) en tant que le premier caractère après le crochet ouvert nie la classe de caractère. Par exemple, le pattern [^a\_z] correspond à tout caractère ne se trouvant pas dans la classe de caractère, y compris les caractères de nouvelle ligne.

## Correspondant à un caractère

Utiliser le méta-caractère du point (.) pour faire correspondre chaque caractère unique, sauf pour le caractère de nouvelle ligne. Par exemple, le pattern . correspond à tout caractère unique.

## Quantificateurs

Dans le cadre d'une expression régulière, des quantificateurs définissent combien de fois le caractère ou sousexpression précédente est autorisée à se produire pour que la correspondance puisse avoir lieu.

?	Correspond à zéro ou une occurrence de l'item qui précède. Par exemple, le pattern mo? fera se correspondre "m" et "mo".
+	Correspond à une ou plusieurs occurrences de l'item qui précède. Par exemple, le pattern mo+ correspondra à "mo", "moo", "mooo", etc.
*	Correspond à zéro or plusieurs occurrences de l'item qui précède.
{min,max}	Correspond à n'importe quel nombre d'occurrences entre <i>min</i> et <i>max</i> . Par exemple, le pattern mo{1,3} fait se correspondre "mo", "moo" et "mooo".

## Parenthèses

Les parenthèses ( et ) sont utilisées pour regrouper des parties d'un regex. Elles peuvent être utilisées pour appliquer des quantificateurs dans une sous-expression (contrairement à uniquement un seul caractère), ou avec une alternance (voir ci-dessous).

## Alternance

La barre verticale | signifie "ou". Elle peut être utilisée pour faire correspondre n'importe laquelle des différentes sous-expressions séparées par |. Par exemple, le pattern (horse |make) sense fera se correspondre les deux "horse sense" et "make sense".

## Flags

Il s'agit de paramètres optionnels qui définissent comme l'expression régulière doit être interprétée. Chaque flag correspond à une lettre. Les lettres peuvent se trouver dans n'importe quel ordre et peuvent être répétées.

s	Si ce flag est présent, le processus de correspondance fonctionne dans le mode "dot-all".
	Si le string d'entrée contient "hello" et "world" dans deux lignes <i>différentes</i> , l'expression régulière hello*world correspondra uniquement si le flag s est défini.
m	Si ce flag est présent, le processus de correspondance fonctionne dans le mode multi-ligne.
	Dans le mode multi-ligne, le symbole caret <pre>ocrespond au début d'une ligne, c'est à dire le début du string entier et le premier caractères après un caractère de nouvelle ligne.</pre>
	Le caractère dollar s correspond à la fin d'une ligne, c'est à dire la fin du string entier et le caractère juste avant un caractère de nouvelle ligne.

	Nouvelle ligne est le caractère <b>#x0A</b> .
i	Si ce flag est présent, le processus de correspondance fonctionne dans le mode insensible à la casse. Par exemple, l'expression régulière [a-z] plus le flag i font se correspondre toutes les lettres a-z et A-Z.
	C= "[a-z]" files C= "[a-z]" files C= "[a-z]" files C= "[a-z]" files C= "i" files
x	Si ce flag est présent, les caractères d'espace blanc sont supprimés de l'expression régulière avant le processus de mise en correspondance. Les caractères d'espace blanc sont <b>#x09</b> , <b>#x0A</b> , <b>#x0D</b> et <b>#x20</b> .
	Note: Les caractères d'espace blanc se trouvant dans une classe de caractère ne sont pas supprimés, par exemple, [#x20].

#### Fonctions

# 6.7 Référence des bibliothèques de fonctions

Cette section de référence décrit les fonctions built-in de MapForce disponibles dans la <u>fenêtre</u> <u>Bibliothèques</u><sup>31</sup>. Les fonctions sont organisées par bibliothèque. La disponibilité des bibliothèques de fonction dans la fenêtre **Bibliothèques** dépend du langage de transformation du mappage que vous choisissez pour votre mappage. Pour en savoir plus sur la liste des langages de transformation disponibles, voir <u>cette</u> <u>rubrique</u><sup>25</sup>.

L'information sur la compatibilité des fonctions et les langages de transformation est fournie dans les soussections ci-dessous.

#### fonctions core

La liste ci-dessous résume la comptabilité des fonctions core avec les langages de transformation. *fonctions core | aggregate* 

- avg, max, max-string, min, min-string: XSLT 2.0, XSLT 3.0, XQuery 1.0, C#, C++, Java, Built-In;
- count, sum: tous les langages de transformation.

#### fonctions core | conversion

- **boolean**, string, **number**: tous les langages de transformation ;
- format-date, format-dateTime, format-time: XSLT 2.0, XSLT 3.0, C#, C++, Java, Built-In;
- format-number : XSLT 1.0, XSLT 2.0, XSLT 3.0, C#, C++, Java, Built-In;
- parse-date, parse-dateTime, parse-number, parse-time: C#, C++, Java, Built-In.

#### core | file path functions

Toutes les fonctions de chemin d'accès au fichier sont compatibles avec les langages de transformation.

#### <u>core | generator functions</u>

La fonction **auto-number** est disponible pour tous les langages de transformation.

#### core | logical functions

les fonctions de chemin d'accès au fichier sont compatibles avec les langages de transformation.

#### core | math functions

- add, ceiling, divide, floor, modulus, multiply, round, subtract: tous les langages de transformation;
- round-precision : C#, C++, Java, Built-In.

#### core | node functions

- is-xsi-nil, local-name, static-node-annotation, static-node-name: tous les langages de transformation;
- node-name, set-xsi-nil, substitute-missing-with-xsi-nil: XSLT 2.0, XSLT 3.0, XQuery 1.0, C#, C++, Java, Built-In.

#### core | QName functions

Les fonctions QName sont compatibles avec les langages de transformation à l'exception de XSLT1.0.

#### 632 Fonctions

#### fonctions core | sequence

- exists, not-exists, position, substitute-missing: tous les langages de transformation ;
- distinct-values, first-items, generate-sequence, item-at, items-from-till, last-items, replicate-item, replicate-sequence, set-empty, skip-first-items: XSLT 2.0, XSLT 3.0, XQuery 1.0, C#, C++, Java, Built-In;
- group-adjacent, group-by, group-ending-with, group-into-blocks, group-starting-with: XSLT 2.0, XSLT 3.0, C#, C++, Java, Built-In.

#### fonctions core | string

- concat, contains, normalize-space, starts-with, string-length, substring, substring-after, substring-before, translate: tous les langages de transformation;
- char-from-code, code-from-char, tokenize, tokenize-by-length, tokenize-regexp: XSLT 2.0, XSLT 3.0, XQuery 1.0, C#, C++, Java, Built-In.

### fonctions bson (uniquement MapForce Enterprise Edition)

Toutes les fonctions BSON sont compatibles uniquement avec Built-In.

### fonctions db (MapForce Professional et Enterprise Edition)

Les fonctions db sont compatibles avec C#, C++, Java, Built-In.

## fonctions edifact (uniquement MapForce Enterprise Edition)

Les fonctions edifact sont compatibles avec C#, C++, Java, Built-In.

## fonctions lang (MapForce Professional et Enterprise Edition)

La liste ci-dessous résume la comptabilité des fonctions lang avec les langages de transformation. <u>lang | datetime functions</u> Les fonctions lang | datetime sont compatibles avec C#, C++, Java, Built-In.

#### fonctions lang | file

Les fonctions read-binary-file et write-binary-file sont uniquement compatibles avec Built-In.

#### <u>fonctions lang | generator</u> La fonction <u>create-guid</u> est disponible pour le C#, C++, Java, Built-In.

#### <u>fonctions lang | logical</u> Les fonctions lang | Logique sont disponibles pour C#, C++, Java, Built-In.

#### <u>fonctions lang | math</u> Les fonctions lang | math sont disponibles pour C#, C++, Java, Built-In.

#### fonctions lang | QName

Les fonctions lang | QName sont compatibles avec C#, C++, Java, Built-In.

#### fonctions lang | string

- charset-decode, charset-encode: Built-In;
- match-pattern : C#, Java, Built-In.
- capitalize, count-substring, empty, find-substring, format-guid-string, left, left-trim, lowercase, pad-string-left, pad-string-right, repeat-string, replace, reversefindsubstring, right, right-trim, string-compare, string-compare-ignore-case, uppercase: C#, C++, Java, Built-In.

### fonctions mime (uniquement MapForce Enterprise Edition)

Les fonctions mime sont uniquement disponibles pour Built-In.

#### fonctions xbrl (uniquement MapForce Enterprise Edition)

Les fonctions xbrl sont compatibles avec C#, C++, Java, Built-In.

### fonctions xlsx (uniquement MapForce Enterprise Edition)

Les fonctions xlsx sont compatibles avec XSLT 2.0, XSLT 3.0, C#, Java, et Built-In.

### fonctions xpath2

Toutes les fonctions xpath2 sont compatibles avec XSLT 2.0, XSLT 3.0 et XQuery 1.0.

### fonctions xpath3

Toutes les fonctions xpath3 sont compatibles uniquement avec XSLT 3.0.

## fonctions xslt10

La liste ci-dessous résume la comptabilité des fonctions xslt10 avec les langages de transformation. *fonctions xslt10 | xpath* 

- local-name, name, namespace-uri: XSLT 1.0, XSLT 2.0, et XSLT 3.0.
- lang, last, position: XSLT 1.0.

#### fonctions xslt10 | xslt

- generate-id, system-property: XSLT 1.0, XSLT 2.0, et XSLT 3.0.
- current, document, element-available, function-available, unparsed-entity-uri: XSLT 1.0.

# 6.7.1 core | aggregate functions

"Aggregate" signifie traiter plusieurs valeurs du même type de manière à obtenir un seul résultat, comme une somme, un décompte ou une moyenne. Vous pouvez effectuer des regroupements (aggregation) de données dans MapForce avec l'aide des fonctions de rassemblement, comme avg, count, max, etc.

Les deux arguments suivants sont communs à toutes les fonctions de rassemblement:

- 1. **parent-context**. Cet argument est optionnel ; il vous permet de contourner le contexte de mappage par défaut (et donc de changer l'étendue de la fonction, ou les valeurs que la fonction doit itérer). Pour voir un exemple, consulter <u>Exemple : Changer le contexte de Parent</u><sup>(80)</sup>.
- 2. values. Cet argument doit être connecté à un item de source qui fournit les valeurs à traiter. Par exemple, dans le mappage illustré ci-dessous, la fonction sum prend en entrée une séquence de valeurs numériques qui provient d'un fichier XML de source. Pour chaque item dans le fichier XML de source, la fonction multiply reçoit le prix de l'item multiplié par la quantité et transmet le résultat à la fonction sum. La fonction sum rassemblera toutes les valeurs d'entrée et produira un résultat total qui est également la sortie du mappage. Vous trouverez ce mappage dans le dossier MapForceExamples.



SimpleTotal.mfd

Certaines fonctions aggregate, comme min, max, sum et avg, fonctionnement exclusivement avec des valeurs numériques. Les données d'entrée de ces fonctions sont converties dans le type de données **decimal** pour traitement.

# 6.7.1.1 avg

Retourne la somme moyenne de toutes les valeurs dans la séquence d'entrée. La moyenne d'un set vide est un set vide.

	$f_{\rm x}$ avg	
ļ	parent-context	recult b
0	>values	result

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description	
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>880</sup> .	
valeurs	Cet argument doit être connecté à un item de source qui fournit les données actuelles. Veuillez noter que la valeur d'argument fournie doit être numérique.	

## Exemple

Voir Exemple : Regrouper les enregistrements par clé<sup>544</sup>.

# 6.7.1.2 count

Retourne le nombre des items individuels constituant la séquence d'entrée. Le décompte d'un set vide est zéro.

f <sub>8</sub> count	
parent-context	
hodes/rows	result

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Veuillez noter que cette fonction a une fonctionnalité limitée dans XSLT 1.0.

## Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>880</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
nodes/rows	Cet argument doit être connecté à l'item de source à compter.

#### Exemple

Voir <u>Exemple : Modifier le contexte Parental</u><sup>(880</sup>, <u>Exemple : Compter les lignes de table de base de</u> <u>données</u><sup>(500)</sup>.

# 6.7.1.3 max

Retourne la valeur maximum de toutes les valeurs numériques dans la séquence d'entrée. Le maximum d'un ensemble vide est un ensemble vide.

f <sub>x</sub> max	
parent-context	
values result i	

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>600</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
valeurs	Cet argument doit être connecté à un item de source qui fournit les données actuelles. Veuillez noter que la valeur d'argument fournie doit être numérique. Pour obtenir le maximum d'une séquence de strings, utiliser la fonction <u>max-string</u> <sup>639</sup> .

#### Exemple

Voir Exemple : Regrouper les enregistrements par clé<sup>644</sup>.

# 6.7.1.4 max-string

Retourne la valeur maximum de toutes les valeurs string dans la séquence d'entrée. Par exemple, maxstring("a", "b", "c") retourne "c". La fonction retourne un ensemble vide si l'argument **strings** est un ensemble vide.

	$\mathbf{f}_{\mathbf{x}}$	max-string	
1	<sup>&gt;</sup> pa	rent-context	rocult [
Ę	> st	rings	result L

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>(R0)</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
strings	Cet argument doit être connecté à un item de source qui fournit les données actuelles. La valeur d'argument fourni doit être une séquence (zéro ou plusieurs) de xs:string.

# 6.7.1.5 min

Retourne la valeur minimum de toutes les valeurs numériques dans la séquence d'entrée. Le minimum d'un ensemble vide est un ensemble vide.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

# Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>680</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core

Argument	Description
	MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
valeurs Cet argument doit être connecté à un item de source qui fournit les données actur Veuillez noter que la valeur d'argument fournie doit être numérique. Pour obtenir le minimum d'une séquence de strings, utiliser la fonction <u>min-string</u> <sup>660</sup> .	

# Exemple

Voir Exemple : Regrouper les enregistrements par clé<sup>544</sup>.

# 6.7.1.6 min-string

Retourne la valeur minimum de toutes les valeurs de string dans la séquence d'entrée. Par exemple, <u>min-string("a", "b", "c")</u> retourne <u>"a"</u>. La fonction retourne un ensemble vide si l'argument **strings** est un ensemble vide.

$f_{\rm R}$ min-string	
parent-context	
strings	result

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

# Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>880</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
strings	Cet argument doit être connecté à un item de source qui fournit les données actuelles. La valeur d'argument fourni doit être une séquence (zéro ou plusieurs) de xs:string.

# 6.7.1.7 string-join

Concatène toutes les valeurs de la séquence d'entrée dans un string délimité par le string que vous avez choisi d'utiliser en tant que le délimiteur. La fonction retourne un string vide si l'argument **strings** est un ensemble vide.

	$f_{\rm x}$ string-join	
ţ	parent-context	
Ç	> strings	result 🛱
ţ	>delimiter	

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>880</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
strings	Cet argument doit être connecté à un item de source qui fournit les données actuelles. La valeur d'argument fourni doit être une séquence (zéro ou plusieurs) de xs:string.
delimiter	Argument optionnel. Spécifie le délimiteur à insérer entre deux strings consécutifs.

## Exemple

Dans l'exemple ci-dessous, le fichier XML de source contient quatre items **Article**, avec les nombres suivants : 1, 2, 3 et 4.



La constante fournit le caractère "#" en tant que délimiteur. Le résultat de mappage est donc 1#2#3#4. Si vous ne fournissez pas de délimiteur, le résultat devient 1234.

# 6.7.1.8 sum

Retourne la somme arithmétique de toutes les valeurs dans la séquence d'entrée. La somme d'un set vide est zéro.

f <sub>8</sub> sum	
parent-context	
▷values	result

## Langues

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
parent-context	Argument optionnel. Fournit le contexte parent. Voir aussi <u>Exemple : Changer le</u> <u>contexte de Parent</u> <sup>680</sup> .
	parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.
valeurs	Cet argument doit être connecté à un item de source qui fournit les données actuelles. Veuillez noter que la valeur d'argument fournie doit être numérique.

Exemple

Voir Exemple : Additionner les valeurs de nœud<sup>596</sup>.

# 6.7.2 core | conversion functions

Pour prendre en charge explicitement la conversion de type de données, plusieurs fonction de conversion de type sont disponibles dans la bibliothèque **conversion**. Veuillez noter que les fonctions de conversion ne sont pas toujours nécessaires parce que, dans la plupart des cas, MapForce crée les conversion nécessaires automatiquement. Les fonctions de conversion sont généralement utiles pour formater des valeurs de date et d'heures, ou pour comparer des valeurs. Par exemple, si certains items de mappage sont de types différents (comme des entiers et des strings), vous pouvez utiliser la fonction de conversion <u>number</u> pour forcer une comparaison numérique.

# 6.7.2.1 boolean

Convertit la valeur de **arg** dans une valeur Booléenne Cela peut être utile pour travailler avec des fonctions logiques (comme equal, greater, etc.), ainsi qu'avec des <u>filtres et des conditions if-else</u><sup>523</sup>. Pour obtenir une Booléenne **false**, fournir un string vide ou un numérique 0 en tant qu'argument. Pour obtenir une Booléenne **true**, fournir un string non-vide ou un numérique 1 en tant qu'argument.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètre

Argument	Description
arg	Argument obligatoire. Fournit la valeur à convertir.

# 6.7.2.2 format-date

Convertit une valeur d'entrée xs:date dans un string et le formate conformément à des options spécifiées.

∫ <sub>8</sub> format-date		
<b>⊳</b> value		
format	result 🕻	
language		

## Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
value	La valeur xs:date à formater.
format	Un string de format identifiant la manière avec laquelle la date doit être formatée. Cet argument est utilisé de la même manière que l'argument <b>format</b> dans la fonction <u>format-dateTime</u> <sup>642</sup> .

#### 642 Fonctions

Argument	Description			
language	Argument optionnel. Lorsqu'il est fourni, le nom du mois est le jour de la semaine sont retournés dans un langage spécifique. Valeurs valides :			
	de	de Allemand		
	en (défaut)	Anglais		
	es	Espagnol		
	fr	Français		
	ja	Japonais		

## Exemple

Le mappage suivant sort la date actuel dans un format comme : "25 March 2020, Wednesday". Pour traduire cette valeur en espagnol, définir la valeur de l'argument de **language** sur **es**.



Veuillez noter que le mappage ci-dessus est conçu pour les langages de transformation Built-in, C++, C# ou Java. Dans XSLT 2.0, le même résultat peut être obtenu par le mappage suivant :



# 6.7.2.3 format-dateTime

Convertit une valeur de type xs:dateTime en un string. La représentation string de date et time est formatté conformément à la valeur de l'argument **format**.

f <sub>x</sub> format-dateTime		
<b>⊳</b> value		
⊳format result		
language		

## Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description		
value	La valeur xs:date	La valeur xs:dateTime à formater.	
format	Un string de format identifiant la manière avec laquelle <b>value</b> doit être formatée. Voir "Remarques" ci-dessous.		
language	Argument optionnel. Lorsqu'il est fourni, le nom du mois est le jour de la semaine sont retournés dans un langage spécifique. Valeurs valides :		
	de	Allemand	
	en (défaut)	Anglais	
	es	Espagnol	
	fr	Français	
	ja	Japonais	

**Note :** Si la sortie de la fonction (résultat) est connectée à un item de type différent d'un string, le formatage peut être perdu lorsque la valeur est amenée vers le type de cible. Pour désactiver ce cast automatique, supprimer les cases à cocher **Valeurs cibles cast vers les types de cible** dans les <u>Paramètres de composant</u><sup>(80)</sup> du composant cible.

## Remarques

L'argument **format** consiste en un string contenant de soit-disant marqueurs de variable contenue dans des crochets, par exemple <code>[Y]/[M]/[D]</code>. Les caractères se trouvant en dehors des crochets sont des caractères littéraux. Si les crochets sont nécessaires en tant que caractères littéraux dans le résultat, alors ils devraient être doublés.

Chaque marqueur de variable consiste en un spécificateur de composant identifiant lequel des composants de date ou time doit être affiché, un modificateur de formatage optionnel, un autre modificateur de présentation optionnel et un modificateur de largeur optionnel, précédé par une virgule si elle est présente.

```
format := (literal | argument)*
argument := [component(format)?(presentation)?(width)?]
width := , min-width ("-" max-width)?
```

Les composants sont les suivants :

Spécificateu r	echo Hello, World!	Présentation par défaut
Y	année (valeur absolue)	quatre chiffres (2010)
Μ	mois de l'année	1-12
J	jour du mois	1-31
d	jour de l'année	1-366
F	jour de la semaine	nom du jour (dépendant du langage)
W	semaine de l'année	1-53
w	semaine du mois	1-5
н	heure (24 heures)	0-23
h	heure (12 heures)	1-12
Р	A.M. ou P.M.	alphabétique (selon le langage)
m	minutes dans l'heure	00-59
S	secondes dans la minute	00-59
f	secondes fractionnelles	numérique, une place décimale
Z	fuseau horaire en tant qu'offset d'heure depuis UTC	+08:00
Z	fuseau horaire en tant qu'offset d'heure en utilisant GMT	GMT+n

Le modificateur de format peut être un des suivants :

Caractère	echo Hello, World!	Exemple
1	Format numérique décimal sans zéros au début	1, 2, 3
01	Format numérique, deux chiffres	01, 02, 03
Ν	Nom du composant , casse majuscule <sup>1</sup>	MONDAY, TUESDAY
n	Nom du composant, casse minuscule <sup>1</sup>	monday, tuesday
Nn	Nom du composant, casse de titre <sup>1</sup>	Monday, Tuesday

Notes de bas de page :

1. Les modificateurs N, n et Nn sont pris en charge par les composants suivants uniquement : M, d, D.

Le modificateur de largeur, si nécessaire, est introduit par une virgule, suivi par un chiffre qui exprime la largeur minimum. En option, vous pouvez ajouter une barre oblique suivie par un autre chiffre qui exprime la largeur maximum. Par exemple :

- [D,2] est le jour du mois, avec des zéros en début de ligne (deux chiffres).
- [MNn, 3-3] est le nom du mois, écrit avec trois lettres, par ex. Jan, Feb, Mar, etc.

### Exemples

La table ci-dessous illustre quelques exemples de valeurs de formatage xs:dateTime avec l'aide de la fonction format-dateTime. La colonne "Value" spécifie la valeur fournie à l'argument **value**. La colonne "Format" spécifie la valeur de l'argument **format**. La colonne "Result" illustre ce qui est retourné par la fonction.

Valeur	Format	Résultat
2003-11- 03T00:00:00	[D]/[M]/[Y]	11/03/2003
2003-11- 03T00:00:00	[Y]-[M,2]-[D,2]	03/11/2003
2003-11- 03T00:00:00	[Y]-[M,2]-[D,2] [H,2]:[m]:[s]	2003-11-03 00:00:00
2010-06-02T08:02	[Y] [MNn] [D01] [F,3-3] [d] [H]:[m]:[s].[f]	2010 June 02 Wed 153 8:02:12.054
2010-06-02T08:02	[Y] [MNn] [D01] [F,3-3] [d] [H]:[m]:[s].[f] [z]	2010 June 02 Wed 153 8:02:12.054 GMT+02:00
2010-06-02T08:02	[Y] [MNn] [D1] [F] [H]:[m]:[s].[f] [Z]	2010 June 2 Wednesday 8:02:12.054 +02:00
2010-06-02T08:02	[Y] [MNn] [D] [F,3-3] [H01]:[m]:[s]	2010 June 2 Wed 08:02:12

# 6.7.2.4 format-number

Convertit un nombre en un string et le formate conformément aux options spécifiées.

${ m f}_{ m s}$ format-number	
<b>⊳</b> value	
⊳format	
decimal-point-character	result
grouping-separator	

### Langages

Built-in, C++, C#, Java, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
value	Argument obligatoire. Fournit le nombre à formater.
format	Argument obligatoire. Fournit un string de format qui identifie la manière avec laquelle le nombre doit être formaté. Voir "Remarques" ci-dessous.
decimal-point-format	Argument optionnel. Fournit le caractère à utiliser en tant que le caractère de point décimal. La valeur par défaut est le caractère de point final ( . ).
grouping-separator	Argument optionnel. Fournit le caractère utilisé pour séparer des groupes de nombres. La valeur par défaut est le caractère de virgule (, ).

Note : Si la sortie de la fonction (résultat) est connectée à un item de type différent d'un string, le formatage peut être perdu lorsque la valeur est amenée vers le type de cible. Pour désactiver ce cast automatique, supprimer les cases à cocher Valeurs cibles cast vers les types de cible dans les <u>Paramètres de composant</u><sup>(80)</sup> du composant cible.

## Remarques

L'argument format prend la forme suivante :

```
format := subformat (;subformat)?
subformat := (prefix)? integer (.fraction)? (suffix)?
prefix := any characters except special characters
suffix := any characters except special characters
integer := (#)* (0)* ( allowing ',' to appear)
fraction := (0)* (#)* (allowing ',' to appear)
```

Le premier *subformat* est utilisé pour formater des nombres positifs, et le second subformat pour des nombres négatifs. Si un seul *subformat* est spécifié, alors le même sous-format sera utilisé pour des nombres négatifs, mais avec un signe négatif rajouté avant le *prefix*.

Caractère spécial	Défaut	echo Hello, World!
zero-digit	0	Un nombre apparaîtra toujours à cet endroit du résultat
digit	#	Un nombre apparaîtra à cet endroit du string de résultat à moins qu'il s'agisse d'un zéro de début ou de fin redondant
decimal-point		Sépare l'entier et la partie fractionnelle du nombre
grouping-separator	,	Sépare des groupes de chiffres.

Caractère spécial	Défaut	echo Hello, World!
percent-sign	%	Multiplie le nombre par 100 et le montre en tant que pourcentage.
per-mille	‰	Multiplie le nombre par 1000 et le montre en tant que pourmille.

La table ci-dessous illustrate des exemples des strings de format et leur résultat.

**Note:** La méthode d'arrondissement utilisée par la fonction <u>format-number</u> est "half up", ce qui signifie que la valeur est arrondie si la fraction est supérieure ou égale à 0.5 La valeur est arrondie vers le bas si la fraction est inférieure à 0.5. Cette méthode d'arrondissement s'applique uniquement au code de programme généré et le moteur d'exécution Built-In. Dans XSLT 1.0, le mode d'arrondissement est non-défini. Dans XSLT 2.0, le mode d'arrondissement est "round-half-to-even".

Numéro	String de Format	Résultat
1234,5	#,##0.00	1 234,50
123,456	#,##0.00	123,46
100000	#,##0.00	1 000 000,00
-59	#,##0.00	-59,00
1234	###0.0###	1234,0
1234,5	###0.0###	1234,5
.00025	###0.0###	0,0003
.00035	###0.0###	0,0004
0,25	#00%	25%
0,736	#00%	74%
1	#00%	100%
-42	#00%	-4200%
-3,12	#.00;(#.00)	(3.12)
-3,12	#.00;#.00CR	3.12CR

## Exemple

Le mappage illustré ci-dessous lit des données provenant d'un XML de source et l'écrit dans un XML cible. Il existe plusieurs éléments **SinglePrice** dans la source qui contient les valeurs décimales suivantes : **25**, **2.30**, **34**, **57.50**. Le mappage a deux objectifs :

1. Remplir toutes les valeurs avec des zéros à gauche de manière à ce que la partie importante prenne exactement 5 chiffres

2. Remplir toutes les valeurs avec des zéros à droite de manière à ce que la partie importante prenne exactement 2 chiffres

Pour ce faire, le string de format 00000.00 a été fourni en tant qu'argument à la fonction format-number.



#### PreserveFormatting.mfd

Par conséquent, les valeurs dans la cible sont devenues :

00025,00 00002,30 00034,00 00057,50

Vous pouvez trouver le fichier de design de mappage sous le chemin suivant: <br/>
<Documents>\Altova\MapForce2023\MapForceExamples\PreserveFormatting.mfd.

# 6.7.2.5 format-time

Convertit la valeur d'entrée xs:time dans un string.

$f_8$ format-time	
⊳value	rocult C
⊳format	result

## Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
value	Argument obligatoire. Fournit la valeur $xs:time$ à formater.
Argument	Description
----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
format	Argument obligatoire. Fournit un string de format. Cet argument est utilisé de la même manière que l'argument <b>format</b> dans la fonction <u>format-</u> <u>dateTime</u> <sup>642</sup> .

## Exemple

Le mappage suivant produit l'heure actuelle dans un format comme 2:15 p.m. . Pour y parvenir, il utilise le string de format [h]:[m] [P], où :

- [h] est l'heure actuelle dans un format 12 heures
- [m] est la minute actuelle
- [P] est la partie "a.m." ou "p.m."



Veuillez noter que le mappage ci-dessus est conçu pour les langages de transformation Built-in, C++, C# ou Java. Dans XSLT 2.0, le même résultat peut être obtenu par le mappage suivant :



# 6.7.2.6 number

Convertit la valeur de **arg** en un nombre, où **arg** est un string ou une valeur Booléenne. Si **arg** est un string, MapForce tentera de le parser en tant que nombre. Par exemple, un string comme "12.56 » est converti dans la valeur décimale 12.56. Si **arg** est booléenne **true**, est converti dans le numérique 1. Si **arg** est booléenne **false**, est converti dans le numérique 0.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
arg	Argument obligatoire. Fournit la valeur à convertir.

### Exemple

Dans l'exemple ci-dessous, la première constante est de type string et elle contient le string "4". La seconde constante contient la constante numérique 12. Pour que les deux valeurs puissent être comparées en tant que nombres, les types doivent s'accorder.



En ajoutant une fonction number à la première constante convertir le string "4" dans la valeur numérique de 4. Le résultat de la comparaison est alors "true". Si la fonction number n'a pas été utilisée (c'est à dire, si "4" a été connecté directement à **a**), une comparaison de string se produirait, le résultat étant "false".

## 6.7.2.7 parse-date

Convertit un string en une date. Cette fonction utilise la fonction <u>parse-dateTime</u><sup>651</sup> en tant que base tout en ignorant le composant de temps. Le résultat est de type xs:date.



## Langages

Built-in, C++, C#, Java.

Argument	Description
value	Argument obligatoire. Fournit la valeur string à convertir.
format	Argument obligatoire. Fournit un string de format. Cet argument est utilisé de la même manière que l'argument <b>format</b> dans la fonction <u>parse-</u> dateTime <sup>651</sup> .

### Exemple

Le mappage ci-dessous parse le string "01 Apr 2015", le convertit en une date et écrit le résultat en un item de cible (**pubdate**) de type xs:date. Cela a été obtenu en utilisant le format [D01] [MNn, 3-3] [Y], où :

- [D01] est la date du mois, exprimé en deux chiffres
- [MNn, 3-3] est le nom du mois, avec une largeur minimum et maximum de 3 caractères
- [Y] est l'année



Le résultat est le suivant (excluant les déclaration XML et d'espace de noms) :



## 6.7.2.8 parse-dateTime

Convertit une valeur date/heure exprimée en tant que string dans une valeur de type xs:dateTime.



Langages Built-in, C++, C#, Java.

Argument	Description
value	La valeur string à convertir.
format	Spécifie le masque de format à appliquer sur la value.

#### Remarques

Un masque de format peut consister dans les composants suivants :

Composant	echo Hello, World!	Présentation par défaut
Y	année (valeur absolue)	quatre chiffres (2010)
М	mois de l'année	1-12
D	jour du mois	1-31
d	jour de l'année	1-366
н	heure (24 heures)	0-23
h	heure (12 heures)	1-12
Р	A.M. ou P.M.	alphabétique (selon le langage)
m	minutes dans l'heure	00-59
S	secondes dans la minute	00-59
f	secondes fractionnelles	numérique, une place décimale
Z	fuseau horaire en tant qu'offset d'heure depuis UTC	+08:00
Z	fuseau horaire en tant qu'offset d'heure en utilisant GMT	GMT+n

Certains des composants ci-dessus prennent des modificateurs (par exemple, ils peuvent être utilisés pour interpréter une date soit en tant que chiffre unique, soit en tant que deux chiffres) :

Modificateur	echo Hello, World!	Exemple
1	format numérique décimal sans zéros au début : 1, 2, 3,	1, 2, 3
01	format numérique, deux chiffres : 01, 02, 03,	01, 02, 03
N	nom du composant, casse majuscule	FEBRUARY, MARCH
n	nom du composant, casse minuscule	february, march
Nn	nom du composant, casse de titre	February, March

Note : Les modificateurs

*N*, *n*, et*Nn* ne prennent en charge que le composant *M* (mois).

Le modificateur de largeur, si nécessaire, est introduit par une virgule, suivi par un chiffre qui exprime la largeur minimum. En option, vous pouvez ajouter une barre oblique suivie par un autre chiffre qui exprime la largeur maximum. Par exemple :

- [D,2] est le jour du mois, avec des zéros en début de ligne (deux chiffres).
- [MNn, 3-3] est le nom du mois, écrit avec trois lettres, par ex. Jan, Feb, Mar, etc.

La table ci-dessous recense quelques exemples de format :

Valeur	Format	Résultat
21-03-2002 16:21:12.492 GMT+02:00	[D]-[M]-[Y] [H]:[m]:[s].[f] [z]	2002-03- 21T16:21:12.492+02:00
315 2004 +01:00	[d] [Y] [Z]	2004-11-10T00:00:00+01:00
1.December.10 03:2:39 p.m. +01:00	[D].[MNn].[Y,2-2] [h]:[m]:[s] [P] [Z]	2010-12-01T15:02:39+01:00
20110620	[Y,4-4][M,2-2][D,2-2]	2011-06-20T00:00:00

## Exemple

Dans le mappage ci-dessous, la valeur string 2019-12-24 19:43:04 +02:00 est convertie dans son équivalent dateTime en appliquant le masque de format [Y]-[M]-[D] [H]:[m]:[s] [Z].



Le résultat est le suivant (excluant les déclaration XML et d'espace de noms) :



## 6.7.2.9 parse-number

Convertit un string en un nombre décimal, conformément au format spécifié.

f <sub>8</sub> parse-number	
🗟 value	
format	
decimal-point-character	result
separator	

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Argument	Description
value	Le string à convertir en un nombre.
format	Argument optionnel. Un string format qui identifie la manière dont le nombre est formaté actuellement Le string format est le même que celui utilisé dans <u>format-number</u> <sup>645</sup> . La valeur par défaut est "#,##0.#"
decimal-point-character	Argument optionnel. Spécifie le caractère à utiliser en tant que le caractère de point décimal. La valeur par défaut est le caractère '.'
grouping-separator	Argument optionnel. Le séparateur/délimiteur est utilisé pour séparer les groupes de nombres. La valeur par défaut est le caractère "," (optionnel).

## Exemple

Le mappage suivant parse la valeur de string "1,234.50" vers un équivalent décimal, en utilisant le masque de format #,##0.00. Dans ce mappage, il n'y a pas besoin de connecter les arguments **decimal-point-character** et **grouping-separator**, puisque leur valeur par défaut correspond au format du string d'entrée.



Le résultat de mappage est le suivant (en excluant les déclaration XML et d'espace de noms) :



## 6.7.2.10 parse-time

Convertit un string en une valeur xs:time. Cette fonction utilise la fonction <u>parse-dateTime</u><sup>(61)</sup> en tant que base tout en ignorant le composant de date.

$f_{\rm x}$ parse-time	
<b>⊳value</b>	
<b>⊳</b> format	result L

## Langages

Built-in, C++, C#, Java.

Argument	Description
value	Argument obligatoire. Fournit la valeur string à convertir.
format	Argument obligatoire. Fournit un string format. Cet argument est utilisé de la même manière que l'argument <b>format</b> dans la fonction <u>parse-</u> <u>dateTime</u> <sup>651</sup> .

## 6.7.2.11 string

Convertit la valeur d'entrée dans un string. La fonction peut aussi être utilisée pour extraire le contenu de texte d'un nœud. Si le nœud d'entrée est un types complexe XML, alors tous les descendants sont aussi sortis en tant que string unique.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
arg	Argument obligatoire. Fournit la valeur à convertir.

# 6.7.3 core | file path functions

Les fonctions **file path** vous permettent d'accéder directement et de manipuler des données de chemin de fichier, comme des dossiers, des noms de fichier et des extensions pour un traitement ultérieur dans vos mappages. Elles peuvent être utilisées dans tous les langages pris en charge par MapForce.

## 6.7.3.1 get-fileext

Retourne l'extension du chemin de fichier contenant le caractère ".".



Langages Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Argument	Description
filepath	Argument obligatoire Fournit le chemin de fichier à traiter.

#### Exemple

Si vous fournissez "c:\data\Sample.mfd" en tant qu'argument, le résultat est .mfd.

## 6.7.3.2 get-folder

Retourne le nom du dossier du chemin de fichier y compris la barre oblique de fin, ou la barre oblique inversée.

∫<sub>∗</sub> get-folder ⊳filepath folder ⊳

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
filepath	Argument obligatoire Fournit le chemin de fichier à traiter.

#### Exemple

Si vous fournissez "c:\data\Sample.mfd" en tant qu'argument, le résultat est c:\data\.

## 6.7.3.3 main-mfd-filepath

Retourne le chemin complet du fichier de design de mappage (.mfd) contenant le mappage principal. Un string vide est retourné si le .mfd n'est pas enregistré actuellement.



Langages Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## 6.7.3.4 mfd-filepath

Si la fonction est appelée dans le mappage principal, elle retourne la même chose que la fonction <u>main-mfd-filepath</u>, c.à.d. le chemin complet du fichier .mfd contenant le mappage principal. Un string vide est retourné si le fichier .mfd n'est actuellement pas enregistré. Si appelé dans le cadre d'une fonction définie par

l'utilisateur qui est *importée* par un mfdfile, il retourne le chemin complet du fichier mfd *importé* qui contient la définition de la fonction définie par l'utilisateur.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## 6.7.3.5 remove-fileext

Supprime l'extension du chemin de fichier contenant le caractère dot.

f <sub>8</sub> remove-fileext		re-fileext	
Ç	>fil	epath	result-filepath 🕻

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
filepath	Argument obligatoire Fournit le chemin de fichier à traiter.

#### Exemple

Si vous fournissez "c:\data\Sample.mfd" en tant qu'argument, le résultat est c:\data\sample.

## 6.7.3.6 remove-folder

Supprime le répertoire du chemin de fichier contenant le caractère de la barre oblique de fin, ou la barre oblique inversée.

#### ∫<sub>∗</sub> remove-folder ⊳filepath filename ⊳

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description	
filepath	Argument obligatoire Fournit le chemin de fichier à traiter.	

#### Exemple

Si vous fournissez "c:\data\Sample.mfd" en tant qu'argument, le résultat est sample.mfd.

# 6.7.3.7 replace-fileext

Remplace l'extension du chemin de fichier fourni par le paramètre **filepath** celui fourni par la connexion du paramètre d**extension**.

	f <sub>x</sub> replace-fileext			
0	>filepath		recult filemath [	ί
0	> extensio	n	result-mepath L	ſ

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
filepath	Argument obligatoire Fournit le chemin de fichier à traiter.
extension	Argument obligatoire Fournit la nouvelle extension à utiliser.

## Exemple

Si vous fournissez "c:\data\Sample.log" en tant que **filepath**,et ".txt" en tant qu'**extension**, le résultat est c: \data\Sample.txt.

## 6.7.3.8 resolve-filepath

Résoud un chemin de fichier relatif contre un dossier de base. La fonction prend en charge '.' (répertoire actuel) et '..' (répertoire parent).

$f_8$ resolve-filepath		
0	>basefolder	secult files ath
C	>filepath	result-mepath

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description	
basefolder	Argument obligatoire Fournit le répertoire de base relatif au chemin qui doit être résolu. Il peut s'agir d'un chemin absolu ou relatif.	
filepath	Argument obligatoire Fournit le fichier relatif à résoudre.	

#### Exemples

Dans le mappage ci-dessous, le chemin de fichier relatif **..\route.gpx** est résolu par rapport au répertoire **C: \data**.



Le résultat de mappage est C:\route.gpx.

# 6.7.4 core | generator functions

La bibliothèque de fonctions core / generator contient des fonctions qui génèrent des valeurs.

## 6.7.4.1 auto-number

Génère des nombres entiers dans une séquence (par exemple, 1,2,3,4, …). Il est possible de définir l'entier de démarrage, la valeur d'incrémentation et d'autres options par le biais des paramètres.

f <sub>x</sub> auto-number	
global-id	
start-with	
increment	result
restart-on-change	

L'ordre exact dans lequel des fonctions sont appelées par le code de mappage généré n'est pas défini. MapForce peut nécessiter de dissimuler des résultats calculés à réutiliser ou évaluer des expressions dans n'importe quel ordre. De même, contrairement à d'autres fonctions, la fonction <u>auto-number</u> retourne un résultat différent lorsqu'ils sont appelés plusieurs fois avec les même paramètres d'entrée. C'est pourquoi il est fortement recommandé d'utiliser la fonction <u>auto-number</u> avec précaution. Dans certains cas, il est possible d'obtenir le même résultat en utilisant la fonction <u>position</u> à la place.

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description	
global-id	Paramètre optionnel. Si un design de mappage contient plusieurs fonctions auto-number, elles généreront des séquences avec des nombres dupliqués (se chevauchant). Pour rendre toutes les fonctions auto-number conscientes l'une de l'autre, et donc générer des séquences qui ne se chevauchent pas, connecter un string commun (par exemple une constante) dans l'entrée <b>global-id</b> de chaque fonction auto-number.	
start-with	Paramètre optionnel. Spécifie l'entier avec lequel la séquence générée commence. La valeur par défaut est de <b>1</b> .	
increment	Paramètre optionnel. Spécifie la valeur d'incrément. La valeur par défaut est de 1.	
restart-on-change	Paramètre optionnel. Réinitialise le décompte sur <b>start-with</b> , lorsque le contenu de l'item connecté change.	

## Exemple

Le mappage suivant est une variation du mappage **ParentContext.mfd** discuté dans l'<u>Exemple : Changer le</u> <u>contexte de Parent</u><sup>(30)</sup>.

L'objectif du mappage illustré ci-dessous est de générer plusieurs fichiers XML, un pour chaque département dans le fichier XML de source. Il existe des département portant le même nom (parce qu'ils appartiennent à des bureaux de parents différents). Pour cette raison, chaque nom de fichier généré doit commencer avec un nombre séquentiel, par exemple **1-Administration.xml**, **2-Marketing.xml**, etc.



Pour atteindre l'objectif de mappage, la fonction auto-number a été utilisée. Le résultat de cette fonction est concaténé avec un caractère de tiret, suivi par le nom du département, suivi par le string ".xml" afin de créer le nom unique du fichier généré. Chose importante, le troisième paramètre de la fonction concat (le nom de département) a un priority context a qui s'applique. Cela a pour conséquence que la fonction auto-number est appelée dans le contexte de chaque département et produit les valeurs séquentielles requises. Si le contexte de priorité n'a pas été utilisé, la fonction auto-number garderait le nombre 1 (dans l'absence de tout contexte), et des noms de fichier doubles seraient générées en conséquence.

# 6.7.5 core | logical functions

Les fonctions logiques sont (généralement) utilisées pour comparer des données d'entrée avec le résultat étant une valeur booléenne true ou false. Elles sont généralement utilisées pour tester des données avant d'être transmises à un sous-ensemble vers le composant de cible à l'aide d'un <u>filtre</u>. Presque toutes les fonctions logiques ont la structure suivante :

- paramètres d'entrée : a | b OU value1 | value2
- paramètres de sortie : résultat

Le résultat d'évaluation dépend des valeurs d'entrée et des types de données utilisés pour la comparaison. Par exemple, la comparaison 'inférieur à' des valeurs d'entier 4 et 12 donne la valeur booléenne true, étant donné que 4 est inférieur à 12. Si les deux paramètres d'entrée contiennent les valeurs de string 4 et 12, l'analyse lexicale résulte dans la valeur de sortie false, puisque 4 est alphabétiquement supérieur au premier caractère 1 du second opérande (12).

Si toutes les valeurs d'entrée sont de même type de données, alors la comparaison est effectuée pour le type commun. Si des valeurs d'entrée sont de types différents (par exemple, integer et string ou string et date), alors le type de données utilisé pour la comparaison est le plus général (le moins restrictif) des deux.

Avant de comparer deux valeurs de types différents, toutes les valeurs d'entrée sont converties en un type de données commun. En reprenant l'exemple précédent ; le type de données string est moins restrictif que integer. Comparer la valeur d'entier 4 avec le string 12 convertit la valeur d'entier 4 vers le string 4, qui est ensuite comparé avec le string 12.

**Note:** Les fonctions logiques ne peuvent pas être utilisées pour tester l'existence de valeurs nulles. Si vous fournissez une valeur nulle en tant qu'argument pour une fonction logique, elle retourne une valeur nulle. Pour plus d'informations concernant les valeurs nulles, voir <u>Nil Values / Nillable</u><sup>128</sup>.

# 6.7.5.1 equal

La fonction equal (voir la capture d'écran ci-dessous) retourne Booléenne true si a est la même que b ; autrement false. La comparaison est sensible à la casse.



#### Exemple :

a = hi b = hi

Dans cet exemple, les deux valeurs sont les mêmes. Pour cette raison, le résultat est true. Si, par exemple, b égalisait Hi, la fonction retournerait false.

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Argument	Description	
a	Paramètre obligatoire. Fournit la première valeur à comparer.	
b	Paramètre obligatoire. Fournit la seconde valeur à comparer.	

## 6.7.5.2 equal-or-greater

Retourne Booléenne true si a est égal ou supérieur à b; false sinon.

	$\mathbf{f}_{\mathbf{z}}$	equa	l-or-greater
Ç	>a		
ç	> b		result L

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
а	Paramètre obligatoire. Fournit la première valeur à comparer.
b	Paramètre obligatoire. Fournit la seconde valeur à comparer.

## 6.7.5.3 equal-or-less

Retourne Booléenne true si a est égal ou inférieur à b; false sinon.

	$f_{x}$	eq	ual-or-less
0	>a		and the P
C	>b		result

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Argument	Description
а	Paramètre obligatoire. Fournit la première valeur à comparer.
b	Paramètre obligatoire. Fournit la seconde valeur à comparer.

## 6.7.5.4 greater

Retourne Booléenne true si a est supérieur à b; false sinon.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Argument	Description
а	Paramètre obligatoire. Fournit la première valeur à comparer.
b	Paramètre obligatoire. Fournit la seconde valeur à comparer.

## 6.7.5.5 less

Retourne Booléenne true si a est inférieur à b; false sinon.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Argument	Description
а	Paramètre obligatoire. Fournit la première valeur à comparer.
b	Paramètre obligatoire. Fournit la seconde valeur à comparer.

# 6.7.5.6 logical-and

Retourne Booléenne **true** uniquement si chaque valeur d'entrée est true ; **false** sinon. Vous pouvez connecter le résultat à une autre fonction **logical-and** et rejoindre un nombre arbitraire de conditions avec la logique AND, afin de tester qu'elles retournent toutes **true**. De même, cette fonction peut être élargie pour prendre des arguments supplémentaires, voir <u>Ajouter ou supprimer des arguments de fonction</u><sup>653</sup>.



### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
value1	Paramètre obligatoire. Fournit la première valeur à comparer.
value2	Paramètre obligatoire. Fournit la seconde valeur à comparer.

## Exemple

Le mappage illustré ci-dessous retourne **true** parce que toutes les valeurs d'entrée dans la fonction logicaland sont **true** également. Si une de ces valeurs d'entrée était **false**, le résultat de mappage serait **false** également.



Voir aussi Exemple : Consultation et Concaténation<sup>688</sup>.

## 6.7.5.7 logical-not

Invertit ou renverse le résultat logique de la valeur d'entrée. Par exemple, si value est true, le résultat de la fonction est faux. Si value est false, alors le résultat est true

∫<sub>8</sub> logical-not ≥value result ⊃

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
value	Paramètre obligatoire. Fournit la valeur d'entrée.

## 6.7.5.8 logical-or

Cette fonction exige que les deux valeurs d'entrée soient booléennes. Si au moins une des valeurs d'entrée est **true**, alors le résultat est **true**. Sinon le résultat est **false**.

Cette fonction peut être étendue pour prendre des arguments supplémentaires, voir <u>Ajouter ou supprimer des</u> arguments de fonctions<sup>553</sup>.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
value1	Paramètre obligatoire. Fournit la première valeur à comparer.
value2	Paramètre obligatoire. Fournit la seconde valeur à comparer.

### Exemple

Le résultat du mappage ci-dessous est true, parcequ'au moins un des arguments de la fonction est true.



## 6.7.5.9 not-equal

Retourne booléenne true si a n'est pas égal à b; false sinon.

	$\mathbf{f}_{\mathbf{x}}$	not-equal		
Ç	>a			ļ
C	>b		result L	ĺ

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
а	Paramètre obligatoire. Fournit la première valeur à comparer.
b	Paramètre obligatoire. Fournit la seconde valeur à comparer.

## 6.7.6 core | math functions

Les fonctions math sont utilisées pour effectuer des opérations mathématiques de base sur des données. Veuillez noter qu'elles ne peuvent pas être utilisées pour effectuer des calculs sur les durées ou des valeurs datetime.

La plupart des fonctions math prennent deux paramètres d'entrée (**value1**, **value2**) qui sont des opérandes de l'opération mathématique. Les valeurs d'entrée sont converties automatiquement au type decimal pour un traitement ultérieur. Le résultat des fonctions math est aussi de type decimal.



L'exemple indiqué ci-dessus ajoute une taxe de vente de 20% pour chacun des articles mappés dans le composant de cible.

## 6.7.6.1 add

Ajoute **value1** à **value2** et retourne le résultat en tant que valeur décimale. Cette fonction peut être étendue pour prendre des arguments supplémentaires, voir <u>Ajouter ou supprimer des arguments de fonctions</u>



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
value1	Paramètre obligatoire. Fournit le premier opérande.
value2	Paramètre obligatoire. Fournit la seconde opérande.

## 6.7.6.2 ceiling

Retourne l'entier le plus petit qui est supérieur à ou égal à value.



Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Argument	Description
value	Paramètre obligatoire. Fournit la valeur d'entrée de la fonction.

## Exemple

Si la valeur d'entrée est **11.2**, y appliquer la fonction **ceiling** pour faire le résultat **12**, c.à.d. l'entier le plus petit qui est supérieur à **11.2**.

## 6.7.6.3 divide

Divise **value1** à **value2** et retourne le résultat en tant que valeur décimale. La précision du résultat dépend du langage cible. Utiliser la fonction <u>round-precision</u> <sup>673</sup> pour définir la précision du résultat.

$f_{\!x}$ divide	
¢value1	
¢value2	result

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
value1	Paramètre obligatoire. Fournit le premier opérande.
value2	Paramètre obligatoire. Fournit la seconde opérande.

## 6.7.6.4 floor

Retourne l'entier le plus grand qui est inférieur à ou égal à value.



Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Argument	Description
value	Paramètre obligatoire. Fournit la valeur d'entrée de la fonction.

## Exemple

Si la valeur d'entrée est **11.7**, y appliquer la fonction **floor** pour faire le résultat **11**, c.à.d. l'entier le plus grand qui est inférieur à **11.7**.

## 6.7.6.5 modulus

Retourne le reste de la division value1 par value2.

$f_{\rm s}$	f <sub>x</sub> modulus		
₽va	lue1	rocult [	ļ
₽va	lue2	result L	ſ

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
value1	Paramètre obligatoire. Fournit le premier opérande.
value2	Paramètre obligatoire. Fournit la seconde opérande.

## Exemple

Si les valeurs d'entrée sont **1.5** et **1**, alors le résultat de la fonction modulus est **0.5**. L'explication est que **1.5** / **1** laisse un restant de **0.5**.

Si les valeurs d'entrée sont 9 et 3, alors le résultat est 0, puisque 9 / 3 ne laisse aucun reste.

## 6.7.6.6 multiply

Multiplie value1 par value2 et retourne le résultat en tant que valeur décimale.



### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Argument	Description
value1	Paramètre obligatoire. Fournit le premier opérande.
value2	Paramètre obligatoire. Fournit la seconde opérande.

## 6.7.6.7 round

Retourne la valeur arrondie à l'entier le plus proche. Lorsque la valeur se trouve exactement entre deux entiers, l'algorithme "Round Half Towards Positive Infinity" est utilisé. Par exemple, la valeur "10.5" est arrondie à "11", et la valeur "-10.5" est arrondie à "-10".



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Argument	Description
value	Paramètre obligatoire. Fournit la valeur d'entrée de la fonction.

# 6.7.6.8 round-precision

Arrondit la valeur d'entrée à N N décimales, où N est l'argument decimals.

f <sub>8</sub> round-p	recision
₽ value	
decimals	result

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Argument	Description
value	Paramètre obligatoire. Fournit la valeur d'entrée de la fonction.
decimals	Paramètre obligatoire. Spécifie le nombre des décimales pour arrondir.

### Exemple

Arrondir la valeur 2.777777 à deux décimales 2.78. Arrondir la valeur 0.1234 à 3 décimales 0.123.

## 6.7.6.9 subtract

Soustrait value2 à value1 et retourne le résultat en tant que valeur décimale.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Argument	Description
value1	Paramètre obligatoire. Fournit le premier opérande.
value2	Paramètre obligatoire. Fournit la seconde opérande.

# 6.7.7 core | node functions

Les fonctions provenant de la bibliothèque **core | node functions** vous permettent d'accéder à des informations d'accès à propos des nœuds sur un composant de mappage (comme le nom de nœud ou l'annotation), ou pour traiter des éléments nillables, voir aussi <u>Nil Values / Nillable</u><sup>128</sup>.

Veuillez considérer qu'il existe un moyen alternatif d'accéder aux noms de nœud, qui ne nécessite pas du tout de fonctions de nœud, voir <u>Mapper des noms de nœud</u><sup>(851)</sup>.

Le mappage illustré ci-dessous montre quelques fonctions de nœud qui obtiennent l'information depuis le nœud **msg:InterchangeHeader** du fichier XML de source. Plus spécifiquement, les informations suivantes sont extraites :

- 1. La fonction node-name retourne le nom qualifié du nœud, qui inclut le préfixe du nœud.
- 2. La fonction local-name retourne uniquement la partie locale.
- 3. La fonction **static-node-name** est semblable à la fonction **node-name**, mais elle est disponible également dans XSLT 1.0.
- 4. La fonction **static-node-annotation** obtient l'annotation de l'élément tel qu'il a été défini dans le schéma he XML.



La sortie du mappage est le suivant (excluant les déclarations XML et d'espace de noms) :



# 6.7.7.1 is-xsi-nil

Retourne true si le nœud element a l'attribut xsi:nil défini sur true.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Argument	Description
element	Paramètre obligatoire. Doit être connecté au nœud de source qui doit être vérifié.

## Exemple

Le design de mappage illustré ci-dessous copie conditionnellement des données depuis une source vers un fichier XML de cible, et illustre également l'utilisation de plusieurs fonctions, y compris <u>is-xsi-nil</u>. Ce mappage est appelé **HandlingXsiNil.mfd** et peut être trouvé dans le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\**.



Comme illustré ci-dessus, la fonction **is-xsi-nil** vérifie si l'attribut xsi:nil est "true" pour l'item **state** dans le fichier de source. Si cet attribut est "false", le filtre copiera l'élément parent **Address** dans la cible. Le fichier XML de source ressemble à l'exemple suivant (sauf les déclarations XML et d'espace de noms) :

```
<BranchOffices>
   <Name>Nanonull</Name>
  <Office>
      <Name>Nanonull Research Outpost</Name>
      <EMail>sp@nanonull.com</EMail>
      <Fax xsi:nil="true"/>
      <Phone>+8817 3141 5926</Phone>
      <Address>
         <city>South Pole</city>
         <state xsi:nil="true"/>
         <street xsi:nil="true"/>
         <zip xsi:nil="true"/>
      </Address>
      <Contact>
         <first>Scott</first>
         <last>Amundsen</last>
      </Contact>
  </Office>
</BranchOffices>
```

Le résultat du mappage est qu'aucun **Address** n'est copié dans la cible, parce qu'il n'y a qu'un seul **Address** dans la source, et l'attribut xsi:nil est défini sur "true" pour l'élément **state**. Par conséquent, la sortie de mappage est le suivant :

```
<BranchOffices>

<Name>Nanonull</Name>

<Office>

<Name>Nanonull Research Outpost</Name>

<EMail xsi:nil="true"/>

<Fax>n/a</Fax>

<Phone>+8817 3141 5926</Phone>

<Contact>

<first>Scott</first>

<last>Amundsen</last>

</Contact>

</Office>

</BranchOffices>
```

## 6.7.7.2 node-name

Retourne le nom qualifié (QName) du nœud connecté. Si le nœud est un nœud XML **text()** un QName vide sera retourné. Cette fonction ne marche que sur les nœuds qui ont un nom. Si XSLT est le langue cible (qui appelle fn:node-name), la fonction retourne une séquence vide pour les nœuds qui n'ont pas de nom

**Note :** Obtenir le nom de nœud n'est pas pris en charge pour les nœuds "File input", les tables de base de données ou les champs, XBRL, Excel, JSON, ou Protocol Buffers.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
node	Paramètre obligatoire. Connecter cette entrée dans le nœud dont vous souhaitez obtenir le nom.

## 6.7.7.3 set-xsi-nil

Défini le nœud de cible sur xsi:nil.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.7.4 static-node-annotation

Retourne le string avec l'annotation du nœud connecté. L'entrée doit être : (i) un composant de source, ou (ii) une fonction définie par l'utilisateur de type "<u>inline</u><sup>570</sup>" qui est directement connectée à un <u>paramètre</u><sup>581</sup> qui est à son tour directement connecté à un nœud dans le mappage d'appel.

La connexion doit être directe. Elle ne peut pas passer à travers un filtre ou une fonction définie par l'utilisateur régulière (pas "inline"). Il s'agit d'une pseudo-fonction, qui est remplacée au moment de la génération par le texte obtenu depuis le nœud connecté, et est donc disponible pour tous les langages

$f_8$ static-nod	e-annotation
<mark>⊳node</mark>	name 🗘

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
node	Paramètre obligatoire. Connecter cette entrée vers le nœud dont vous souhaitez obtenir l'annotation.

## 6.7.7.5 static-node-name

Retourne le string avec le nom du nœud connecté. L'entrée doit être : (i) un composant de source, ou (ii) une fonction définie par l'utilisateur de type "inline<sup>576</sup>" qui est directement connectée à un paramètre<sup>531</sup> qui est, à son tour, directement connecté à un nœud dans le mappage d'appel.

La connexion doit être directe. Elle ne peut pas passer par un filtre ou une fonction non-inlined définie par l'utilisateur. Il s'agit d'une pseudo-fonction, qui est remplacée au moment de la génération par le texte obtenu depuis le nœud connecté, et est donc disponible pour tous les langages.

	$f_{\rm s}$ static-node-name			
ç	>nc	ode	name 🗘	>

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Argument	Description
node	Paramètre obligatoire. Connecter cette entrée dans le nœud dont vous souhaitez obtenir le nom.

## 6.7.7.6 substitute-missing-with-xsi-nil

Pour des nœuds avec un contenu simple, cette fonction remplace toute valeur manquante (ou nulle) du composant de source, avec l'attribut xsi:nil dans le nœud cible.

f <sub>x</sub> substitute-missing-with-xsi-nil		ssing-with-xsi-nil	
ç	>in	put	result 🗅

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Argument	Description
input	Paramètre obligatoire. Connecter cette entrée dans le nœud dont vous souhaitez obtenir le nom.

# 6.7.8 core | QName functions

Les fonctions QName permettent de manipuler les Qualified Names (QName) dans les documents XML.

## 6.7.8.1 QName

Construit un QName depuis un URI d'espace de noms et une partie locale. Utiliser cette fonction pour créer un QName dans un composant de cible. Les paramètres **uri** et **node-name** peuvent être fournis par une fonction constante.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description
uri	Obligatoire. Fournit l'URI.
node-name	Obligatoire. Fournit le nom du nœud.

# 6.7.8.2 local-name-from-QName

Extrait le nom local d'une valeur de type xs:QName. Veuillez noter que, contrairement à la fonction local-name qui retourne le nom local du nœud de *nœud*, cette fonction traite le *contenu* de l'item connecté à l'entrée **qname**.

f <sub>8</sub> local-name	from-QName
🗟 qname	result 🗘

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description
qname	Obligatoire. Fournit la valeur d'entrée de la fonction, de type xs:QName.

## 6.7.8.3 namespace-uri-from-QName

Retourne la partie d'espace de noms d'URI de la valeur QName fournie en tant qu'argument.

f <sub>8</sub> namespace-u	ri-from-QName
qname	result 🖓

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
qname	Obligatoire. Fournit la valeur d'entrée de la fonction.

#### Exemple

Le fichier XML suivant contient une valeur QName, **o:name**. Veuillez noter que le préfixe "o" est mappé à l'espace de noms <a href="http://NamespaceTest.com/Order">http://NamespaceTest.com/Order</a>.

Un mappage qui traite la valeur QName et obtient l'URI d'espace de noms est illustrée ci-dessous :



La sortie de ce mappage est http://NamespaceTest.com/Order.

# 6.7.9 core | sequence functions

Les fonctions de séquence permettent le traitement des <u>sequences</u><sup>673</sup> d'entrée et du regroupement de leur contenu.

## 6.7.9.1 distinct-values

Traite la séquence des valeurs connectées aux entrées **values** et retourne uniquement les valeurs distinctes, en tant que séquence. Cela est utile lorsque vous devez supprimer des valeurs doubles d'une séquence et ne copier que les items uniques dans le composant cible.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description
valeurs	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> <sup>1673</sup> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source , un champ CSV, un enregistrement de base de données, etc.

#### Exemple

Le fichier XML suivant contient des informations concernant les employés d'une entreprise fictive. Certains employés ont le même rôle ; c'est pourquoi le rôle d'attribut "role" contient des valeurs doubles. Par exemple, "Loby Matise" et "Susi Sanna" ont toutes deux le rôle "Support".

```
<?xml version="1.0" encoding="UTF-8"?>
<KeyValueList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="KeyValueList.xsd">
   <Ttem>
      <Property Key="role">Manager</Property>
      <Property Key="First">Vernon</Property>
      <Property Key="Last">Callaby</Property>
   </Item>
   <Ttem>
      <Property Key="role">Programmer</Property>
      <Property Key="First">Frank</Property>
      <Property Key="Last">Further</Property>
   </Ttem>
   <Item>
      <Property Key="role">Support</Property>
      <Property Key="First">Loby</Property></property>
      <Property Key="Last">Matise</Property>
   </Item>
   <Item>
      <Property Key="role">Support</Property>
      <Property Key="First">Susi</Property>
      <Property Key="Last">Sanna</Property>
   </Item>
</KeyValueList>
```

Supposons que vous souhaitez extraire une liste de tous les noms de rôle *unique* qui se produisent dans ce fichier XML. Cela peut être obtenu à l'aide d'un mappage comme celui ci-dessous :



Dans le mappage ci-dessus, les choses suivantes se produisent :

- Chaque élément Property provenant du fichier XML de source est traité par un filtre.
- La connexion à l'entrée bool du filtre assure que seuls des éléments Property où l'attribut Key est égale à "role" sont fournis dans le composant de la cible. Le string "role" est fourni par une constante. Veuillez noter que la sortie du filtre produit encore des doubles à ce niveau (puisqu'il y a deux propriétés "Support" qui remplissent la condition du filtre).
- La séquence produite par le filtre est traitée par la fonction **distinct-values** qui exclut toute valeur double.

Par conséquent, la sortie de mappage est la suivante (à l'exception des déclarations XML et de schéma) :

<items></items>
<item>Manager</item>
<pre><item>Programmer</item></pre>
<item>Support</item>

# 6.7.9.2 exists

Retourne **true** si le nœud connecté existe; **false** sinon. Puisqu'elle retourne une valeur booléenne, cette fonction est généralement utilisée avec <u>filters</u> pour filtrer uniquement les enregistrements qui ont (ou peutêtre qui n'ont pas) un élément ou un attribut enfant.



### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description
node	L'existence du nœud doit être testée.

## Exemples

Le mappage suivant illustre comment filtrer des données avec l'aide de la fonction **exists**. Ce mappage est appelé **PersonListsForAllBranchOffices.mfd** et peut être trouvé dans le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\**.



PersonListsForAllBranchOffices.mfd

Dans le fichier source **BranchOffices.xml**, vous trouverez trois éléments **Office**. Notablement, un des trois bureaux n'a pas d'éléments enfant **Contact**. L'objectif de ce mappage est multiple :

- a) pour chaque bureau, extraire une liste des contacts qui existent dans ce bureau
- b) pour chaque bureau, créer un fichier XML séparer avec le même nom dans le bureau
- c) ne pas générer le fichier XML si le bureau n'a pas de contacts.

Pour atteindre ces objectifs, un filtre a été ajouté au mappage. Le filtre fait passer vers la cible uniquement les items **Office** où au moins un item **Contact** existe. Cette condition booléenne est fournie par la fonction **exists**. Si le résultat de la fonction est true, le nom du bureau est concaténé avec le string **.xml** afin de produire le nom de fichier de cible. Pour plus d'informations concernant la génération des noms de fichier depuis le mappage, voir <u>Traiter plusieurs fichiers d'entrée ou de sortie dynamiquement</u>

Un autre exemple est le mappage suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\HasMarketingExpenses.mfd. lci, si un expense-item existe dans le XML source, alors l'attribut hasExpenses est défini sur true dans le fichier XML de cible.


HasMarketingExpenses.mfd

Voir aussi Exemple : Exception lorsque le nœud n'existe pas<sup>648</sup>.

## 6.7.9.3 first-items

Retourne les premiers items N de la séquence d'entrée, où N est fourni par le paramètre count.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
count	Paramètre optionnel. Spécifie combien d'items doivent être extraits depuis la séquence d'entrée. La valeur par défaut est de <b>1</b> .

Le mappage fictif suivant génère une séquence de 10 valeurs. La séquence est traitée par la fonction **first**items et le résultat est écrit dans un fichier XML de cible.



Étant donné que l'argument **count** n'a pas de valeur, la valeur par défaut de **1** s'applique. Par conséquence, seule la première valeur provenant de la séquence est générée dans la sortie du mappage :



Pour un exemple plus réaliste, voir le mappage **FindHighestTemperatures.mfd** discuté dans <u>Fournir les</u> paramètres au mappage<sup>477</sup>.

## 6.7.9.4 generate-sequence

Crée une séquence des entiers en utilisant les paramètres "from" et "to" en tant que les limites.

	f <sub>8</sub> generat	te-sequence
Ì	>from	and the b
[	>to	results

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

Nom	Description
from	Paramètre optionnel. Spécifie l'entier avec lequel la séquence doit démarrer (bord inférieur). La valeur par défaut est de <b>1</b> .
to	Paramètre obligatoire. Spécifie l'entier avec lequel la séquence doit se terminer (bord supérieur).

## 6.7.9.5 group-adjacent

La fonction group-adjacent regroupe les items connectés aux entrées **nodes/rows** par la clé connectée à l'entrée **key**. Veuillez noter que cette fonction place les items qui partagent la même clé dans des groupes séparés s'ils ne sont pas adjacents. Si plusieurs items consécutifs (adjacents) partagent la même clé, ils sont placés dans le même groupe.

f <sub>8</sub> group-adj	jacent
nodes/rows	groups Þ
key	key Þ

Par exemple, dans la transformation abstraite illustrée ci-dessous, la clé de regroupement est "Department". Le côté gauche du diagramme montre les données d'entrée tandis que le côté droit montre les données de sortie après le regroupement. Les événements suivants se produisent lorsque la transformation est exécutée :

- Tout d'abord, la première clé, "Administration", crée un nouveau groupe.
- La clé suivante est différente, donc un deuxième groupe est créé :"Marketing".
- La troisième clé est aussi différente, donc un groupe supplémentaire est créé : "Engineering".
- La quatrième clé est la même que la troisième, c'est pourquoi cet enregistrement est placé dans le groupe déjà existant.
- Enfin, la cinquième clé est différente de la quatrième et cela crée le dernier groupe.

Comme illustré ci-dessous, "Michelle Butler" et "Fred Landis" ont été regroupés car ils partagent la même clé et sont adjacents. Néanmoins, "Vernon Callaby" et "Frank Further" se trouvent dans des groupes séparés étant donné qu'ils ne sont pas adjacents, même s'ils possèdent la même clé.

			Department	Name
			Administration	Vernon Call
epartment	Name			
dministration	Vernon Callaby		Department	Name
Marketing	Susi Sanna		Marketing	Susi Sanna
Engineering	Michelle Butler	group-adjacent()		
Engineering	Fred Landis			
Administration	Frank Further		Department	Name
			Engineering	Michelle Bu
			Engineering	Fred Landis

Department	Name	
Administration	Frank	Further

#### Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> <sup>173</sup> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
key	La clé avec laquelle regrouper des items.

### Exemple

Partons du principe que vos données de source existent sous la forme d'un fichier XML contenant les éléments suivants (veuillez noter que dans l'extrait de code suivant, l'espace de noms et des déclarations XML ont été supprimés pour plus de simplicité).

<company></company>	
<person< th=""><th><pre>department="Administration" name="Vernon Callaby"/&gt;</pre></th></person<>	<pre>department="Administration" name="Vernon Callaby"/&gt;</pre>
<person< th=""><th>department="Marketing" name="Susi Sanna"/&gt;</th></person<>	department="Marketing" name="Susi Sanna"/>
<person< th=""><th><pre>department="Engineering" name="Michelle Butler"/&gt;</pre></th></person<>	<pre>department="Engineering" name="Michelle Butler"/&gt;</pre>
<person< th=""><th><pre>department="Engineering" name="Fred Landis"/&gt;</pre></th></person<>	<pre>department="Engineering" name="Fred Landis"/&gt;</pre>
<person< th=""><th><pre>department="Administration" name="Frank Further"/&gt;</pre></th></person<>	<pre>department="Administration" name="Frank Further"/&gt;</pre>

L'exigence commerciale est de grouper des enregistrements de personnes par département, s'ils sont adjacents. Pour ce faire, le mappage suivant invoque la fonction group-adjacent et fournit **department** en tant que **key**.



Le résultat de mappage est le suivant :

```
<groups>
<group>
<record key="Administration" value="Vernon Callaby"/>
</group>
<group>
<record key="Marketing" value="Susi Sanna"/>
</group>
<group>
```

Cet exemple, avec d'autres exemples de regroupement, fait partie du fichier de mappage suivant : **Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd**. Ne pas oublier de cliquer sur la touche **Aperçu** applicable à la fonction que vous souhaitez consulter préalablement, avant de cliquer sur l'onglet **Sortie**.

# 6.7.9.6 group-by

La fonction group-by crée des groupes d'enregistrements conformément à certaines clés de regroupement que vous aurez spécifiées.

	f <sub>x</sub> group-by	
Ç	>nodes/rows	groups 🗘
⊳ key		key 🖓

Par exemple, dans la transformation abstraite illustrée ci-dessous, la clé de regroupement est "Department". Étant donné qu'il y a trois départements au total, l'application de la fonction group-by créera trois groupes :

Department	Name
Administration	Vernon Callaby
Marketing	Susi Sanna
Engineering	Michelle Butler
Engineering	Fred Landis
Administration	Frank Further

group-by()	D
	7

¥.

Name
Vernon Callaby
Frank Further

Department	Name
Marketing	Susi Sanna

Department	Name
Engineering	Michelle Butler
Engineering	Fred Landis

## Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> <sup>1373</sup> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
key	La clé avec laquelle regrouper des items.

### Exemple

Partons du principe que vos données de source existent sous la forme d'un fichier XML contenant les éléments suivants (veuillez noter que dans l'extrait de code suivant, l'espace de noms et des déclarations XML ont été supprimés pour plus de simplicité).



L'exigence commerciale est de regrouper des enregistrements de personnes par département. Pour ce faire, le mappage suivant invoque la fonction group-by et fournit **department** en tant que clé.



Le résultat de mappage est le suivant :

```
<groups>

<group>

<record key="Administration" value="Vernon Callaby"/>

<record key="Administration" value="Frank Further"/>

</group>

<group>

<record key="Marketing" value="Susi Sanna"/>

</group>

<group>

<group>
```

Cet exemple, avec d'autres exemples de regroupement, fait partie du fichier de mappage suivant : **Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd**. Ne pas oublier de cliquer sur la touche **Aperçu** applicable à la fonction que vous souhaitez consulter préalablement, avant de cliquer sur l'onglet **Sortie**.

# 6.7.9.7 group-ending-with

La fonction group-ending-with prend une condition booléenne en tant qu'argument. Si la condition booléenne est vraie, un nouveau groupe est créé, terminant avec l'enregistrement qui satisfait la condition.

$f_{x}$ group-ending-with		
nodes/rows		
⊳ <mark>bool</mark>	groups	

Dans l'exemple ci-dessous, la condition est que "Key" doit être égal à "trailing". Cette condition est vraie pour les troisième et quatrièmes enregistrements, en résultat, deux groupes sont créés :

		1	Кеу	Value
Кеу	Value		line	A
line	A		line	в
line	В	1	trailing	Total
trailing	Total 1	group-ending-with()		
line	C	<b>/</b>		
trailing	Total 2		Кеу	Value
		1	line	С
			trailing	Total

**Note :** Un groupe supplémentaire est créé si des enregistrements existent après le dernier qui satisfait à la condition. Par exemple, si il existait plus d'enregistrements "line" après le dernier enregistrement "trailing", ceux-ci seraient tous placés dans un nouveau groupe.

## Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> <sup>673</sup> de zéros ou plus de valeurs. Par exemple, la

Nom	Description	
	connexion peut provenir d'un item XML de source , un champ CSV, un enregistrement de base de données, etc.	
bool	Fournit la condition Booléenne qui lance un nouveau groupe si true.	

Partons du principe que vos données de source existent sous la forme d'un fichier XML contenant les éléments suivants (veuillez noter que dans l'extrait de code suivant, l'espace de noms et des déclarations XML ont été supprimés pour plus de simplicité).

```
<records>

<record key="line" value="A"/>

<record key="line" value="B"/>

<record key="trailing" value="Total 1"/>

<record key="line" value="C"/>

<record key="trailing" value="Total 2"/>

</records>
```

L'exigence commerciale est de créer des groupes pour chaque enregistrement "trailing" Chaque groupe doit aussi inclure des enregistrements "line" qui précède l'enregistrement "trailing". Pour ce faire, le mappage suivant invoque la fonction group-ending-with. Dans le mappage ci-dessous, à chaque fois que le nom **key** est égal à "trailing", l'argument fournit en **bool** devient **true**, et un nouveau groupe est créé.



Le résultat de mappage est le suivant :

```
<groups>
<groups
<group>
<record key="line" value="A"/>
<record key="line" value="B"/>
<record key="trailing" value="Total 1"/>
</group>
<group>
<record key="line" value="C"/>
<record key="line" value="C"/>
<record key="trailing" value="Total 2"/>
```

Cet exemple, avec d'autres exemples de regroupement, fait partie du fichier de mappage suivant : **Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd**. Ne pas oublier de cliquer sur la touche **Aperçu** applicable à la fonction que vous souhaitez consulter préalablement, avant de cliquer sur l'onglet **Sortie**.

# 6.7.9.8 group-into-blocks

La fonction group-into-blocks crée des groupes égaux qui contiennent exactement N items, et où N est la valeur que vous fournissez à l'argument block-size. Veuillez noter que le dernier groupe peut contenir N items ou moins, selon le nombre d'items se trouvant dans la source.

f <sub>8</sub> group-into-blocks		
nodes/rows		
block-size	groups	

Dans l'exemple ci-dessous, block-size est 2. Étant donné qu'il existe au total cinq items, chaque groupe contient exactement deux items, sauf pour le dernier.



## Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source , un champ CSV, un enregistrement de base de données, etc.

Nom	Description
block-size	Spécifie la taille de chaque groupe

Partons du principe que vos données de source existent sous la forme d'un fichier XML contenant les éléments suivants (veuillez noter que dans l'extrait de code suivant, l'espace de noms et des déclarations XML ont été supprimés pour plus de simplicité).

```
<company>
  <person department="Administration" name="Vernon Callaby"/>
  <person department="Marketing" name="Susi Sanna"/>
  <person department="Engineering" name="Michelle Butler"/>
  <person department="Engineering" name="Fred Landis"/>
  <person department="Administration" name="Fred Landis"/>
  <person department="Administration" name="Fred Landis"/>
  </company>
```

L'exigence commerciale est de grouper des enregistrements de personnes en deux items chacun. Pour ce faire, le mappage suivant invoque la fonction group-into-blocks et fournit la valeur d'entier "2" en tant que **block-size**.



Le résultat de mappage est le suivant :

```
<groups>
<groups
<groups
<record key="Administration" value="Vernon Callaby"/>
<record key="Marketing" value="Susi Sanna"/>
</groups
<groups
<groups
<record key="Engineering" value="Michelle Butler"/>
<record key="Engineering" value="Fred Landis"/>
</groups
<groups
</groups
```

Veuillez noter que le dernier groupe ne contient qu'un seul item, puisque le nombre total d'items (5) ne peut pas être divisé par 2.

Cet exemple, avec d'autres exemples de regroupement, fait partie du fichier de mappage suivant : **Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd**. Ne pas oublier de cliquer sur la touche **Aperçu** applicable à la fonction que vous souhaitez consulter préalablement, avant de cliquer sur l'onglet **Sortie**.

## 6.7.9.9 group-starting-with

La fonction group-starting-with prend une condition booléenne en tant qu'argument. Si celle-ci est vraie, un nouveau groupe est créé, commençant avec l'enregistrement qui satisfait à la condition.

$\int_{R}$ group-starting-with		
nodes/rows		
bool	groups	

Dans l'exemple ci-dessous, la condition est que "Key" doit être égal à "heading". Cette condition est vraie pour le premier et le quatrième enregistrement, donc deux groupes sont créés :



**Note:** Un groupe supplémentaire est créé si des enregistrements existent après le premier qui satisfait à la condition. Par exemple, s'il existait plus d'enregistrements "line" avant le premier enregistrement "heading", ceux-ci seraient tous placés dans un nouveau groupe.

### Langages

Built-in, C++, C#, Java, XSLT 2.0, XSLT 3.0.

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source , un champ CSV, un enregistrement de base de données, etc.
bool	Fournit la condition Booléenne qui lance un nouveau groupe si true.

Partons du principe que vos données de source existent sous la forme d'un fichier XML contenant les éléments suivants (veuillez noter que dans l'extrait de code suivant, l'espace de noms et des déclarations XML ont été supprimés pour plus de simplicité).

```
<records>
<record key="heading" value="Intro"/>
<record key="line" value="A"/>
<record key="line" value="B"/>
<record key="heading" value="Body"/>
<record key="line" value="C"/>
</records>
```

L'exigence commerciale est de créer des groupes pour chaque enregistrement "heading". Chaque groupe doit aussi inclure tout enregistrement "line" qui suit l'enregistrement "heading". Pour ce faire, le mappage suivant invoque la fonction group-starting-with. Dans le mappage ci-dessous, dès que le nom **key** est égal à "heading", l'argument fourni en **bool** devient **true** et un nouveau groupe est créé.



Le résultat de mappage est le suivant :

```
<groups>

<groups>

<record key="heading" value="Intro"/>

<record key="line" value="A"/>

<record key="line" value="B"/>

</group>

<group>

<record key="heading" value="Body"/>

<record key="line" value="C"/>

</group>

</group>
```

Cet exemple, avec d'autres exemples de regroupement, fait partie du fichier de mappage suivant : **Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd**. Ne pas oublier de cliquer sur la touche **Aperçu** applicable à la fonction que vous souhaitez consulter préalablement, avant de cliquer sur l'onglet **Sortie**.

## 6.7.9.10 item-at

Retourne un item depuis la séquence de **nodes/rows** fournie en tant qu'argument, à la position fournie par l'argument **position**. Le premier item se trouve à la position **1**.



## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
position	Cet entier spécifie quel item provenant de la séquence des items doit être retournée.

## Exemple

Le mappage fictif suivant génère une séquence de 10 valeurs. La séquence est traitée par la fonction <u>item-at</u> et le résultat est écrit dans un fichier XML de cible.



Étant donné que l'argument **position** est défini sur **3**, seule la troisième valeur provenant de la séquence est transmise vers la cible. Par conséquent, la sortie de mappage est la suivante (à l'exception des déclarations XML et de schéma) :



## 6.7.9.11 items-from-till

Retourne une séquence de **nodes/rows** en utilisant les paramètres "from" et "till" en tant que limites. Le premier item se trouve à la position **1**.

f <sub>8</sub> items-from-till	
hodes/rows	
₽ from	results 🖡
¢till	

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> <sup>B73</sup> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
from	Cet entier spécifie la position de démarrage à partir de laquelle les items doivent être extraits.
till	Cet entier spécifie la position jusqu'à laquelle les items doivent être extraits.

## Exemple

Le mappage fictif suivant génère une séquence de 10 valeurs. La séquence est traitée par la fonction *item-from-till* et le résultat est écrit dans un fichier XML de cible.



Étant donné que les arguments **from** et **till** sont définis sur **3** et **5**, respectivement, seul le sous-ensemble de valeurs provenant de **3** à **5** est transmis vers la cible. Par conséquent, la sortie de mappage est la suivante (à l'exception des déclarations XML et de schéma) :

```
<items>
<item>3</item>
<item>4</item>
<item>5</item>
</items>
```

## 6.7.9.12 last-items

Retourne les premiers items *N* de la séquence d'entrée, où *N* est fourni par le paramètre **count**. Le premier item se trouve à la position "1".

f <sub>8</sub> last-items	
nodes/rows	
count	

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## Paramètres

Nom	Description
nodes/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
count	Paramètre optionnel. Spécifie combien d'items doivent être extraits depuis la séquence d'entrée. La valeur par défaut est de <b>1</b> .

## Exemple

Le mappage fictif suivant génère une séquence de 10 valeurs. La séquence est traitée par la fonction lastitems et le résultat est écrit dans un fichier XML de cible.



Étant donné que l'argument count est défini sur **3**, seule les trois dernières valeurs provenant de la séquence sont transmises vers la cible. Par conséquent, la sortie de mappage est la suivante (à l'exception des déclarations XML et de schéma) :

```
<items>
<item>8</item>
<item>9</item>
<item>10</item>
</items>
```

## 6.7.9.13 not-exists

Retourne **false** si le nœud connecté existe; **true** sinon. Cette fonction est l'inverse de la fonction <u>exists</u><sup>683</sup> mais à part cela elle a la même utilisation.



### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
node	Le nœud à tester pour non-existence.

## 6.7.9.14 position

Retourne la position d'un item dans le cadre de la séquence des items actuellement en cours de traitement. Cela peut être utilisé, par exemple pour numéroter automatiquement des items séquentiellement.



### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Nom	Description
node	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> <sup>873</sup> de zéros ou plus de valeurs. Par exemple, la connexion peut

Nom	Description	
	provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.	

Le mappage suivant illustre l'utilisation de la fonction **position** pour générer des valeurs d'identification uniques dans des données générées par le mappage. Ce mappage est accompagné par un fichier de design de mappage qui est disponible sous le chemin suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\ContactsFromBranchOffices.mfd.



#### ContactsFromBranchOffices.mfd

Dans le mappage ci-dessus, le fichier XML de source contient trois filiales. Une filiale peut contenir un nombre arbitraire d'items enfant **Contact**. Les objectifs du mappage sont les suivants :

- Extraire tous les items **Contact** du fichier XML de source et les écrire dans le fichier XML de cible.
- Chaque contact doit être attribué à un numéro d'identification unique (l'item ID dans le XML de cible).
- L'ID de chaque contact doit prendre la forme de cxx-yyyyy, où X identifie le numéro de bureau et Y identifie le numéro de contact. Si le numéro du bureau a moins de deux caractères, il faut rajouter des zéros à gauche. De même, si le numéro de contact a moins de cinq caractères, il faut rajouter des zéros à gauche. Par conséquent, un numéro d'identification valide du premier contact du premier bureau devrait ressembler à l'exemple suivant : c01-00001.

Pour obtenir les objectifs de mappage, plusieurs fonctions MapForce ont été utilisées, y compris la fonction **position**. La fonction supérieure **position** obtient la position de chaque bureau. La fonction inférieure obtient la position de chaque contact, dans le contexte de chaque bureau.

Lorsque vous utilisez la fonction **position**, il est important de considérer le <u>contexte de mappage</u> actuel. Plus spécifiquement, lorsque le mappage est exécuté, le contexte de mappage initial est établi depuis l'item root du composant cible vers l'item de source qui y est connecté (même indirectement via des fonctions). Dans cet exemple, la fonction supérieure **position** traite *la séquence de tous les bureaux* et elle génère initialement la valeur valeur 1, correspondant au premier bureau dans la séquence. La fonction inférieure **position** génère des nombres séquentiels correspondant à la position du contact *dans le contexte de ce bureau* (1, 2, 3, etc.). Veuillez noter que cette séquence "interne" sera réinitialisée (et donc recommencera avec le numéro 1) lorsque le bureau suivant est traité. Les deux fonctions **pad-string-left** appliquent le remplissage aux numéros générés, conformément aux exigences déclarées précédemment. La fonction

#### 702 Fonctions

concat fonctionne *dans le contexte de chaque contact* (en raison de la connexion parent provenant de la source vers la cible **Contact**). Elle rejoint toutes les valeurs calculées et retourne le numéro d'identification unique de chaque contact.

La sortie générée depuis le mappage ci-dessus est affiché ci-dessous (veuillez noter que certains des enregistrements ont été supprimés pour la lecture) :

```
<Contacts>
  <Contact>
      <ID>C01-00001</ID>
      <First>Vernon</First>
      <Last>Callaby</Last>
  </Contact>
  <Contact>
     <ID>C01-00002</ID>
      <First>Frank</First>
      <Last>Further</Last>
  </Contact>
  <!--->
  <Contact>
      <ID>C02-00001</ID>
      <First>Steve</First>
      <Last>Meier</Last>
  </Contact>
  <Contact>
      <ID>C02-00002</ID>
      <First>Theo</First>
      <Last>Bone</Last>
  </Contact>
  <!-- ... -->
</Contacts>
```

Il peut aussi exister des cas où vous souhaitez obtenir la position d'items résultant après l'application d'un <u>filtre</u><sup>623</sup>. Veuillez noter que le composant de filtre n'est pas une fonction de séquence et qu'il ne peut pas être utilisé *directement* en conjonction avec la fonction de <u>position</u> pour trouver la position des items filtrés. Indirectement, cela est possible en ajoutant un composant de <u>variable</u><sup>622</sup> dans le mappage. Par exemple, le mappage ci-dessous est une version simplifiée de la version précédente. Son fichier de design de mappage est disponible sous le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\PositionInFilteredSequence.mfd.



Le résultat des variables dans MapForce est toujours des séquences. C'est pourquoi, dans le mappage cidessus, la fonction **position** itère à travers la séquence créée par la variable et retourne la position de chaque item dans cette séquence. Ce mapping est discuté plus en détail dans <u>Exemple : Filtrer et Numéroter des</u> <u>nœuds</u>.

## 6.7.9.15 replicate-item

Répète chaque item dans la séquence d'entrée le nombre de fois spécifié dans l'argument **count**. Si vous connectez un seul item dans l'entrée **node/row**, la fonction retournera *N* items, où *N* est la valeur de l'argument **count**. Si vous connectez une séquence d'item dans l'entrée **node/row**, la fonction répètera chaque item individuel dans la séquence **count** fois, en traitant un item à la fois. Par exemple, si count est **2**, la séquence **1**,**2**,**3** produira **1**,**1**,**2**,**2**,**3**,**3**. Il est aussi possible de fournir une valeur **count** différente pour chaque item dans la séquence d'entrée, comme illustré dans l'exemple ci-dessous.

f <sub>8</sub> replicate-item	
node/row	
Count	results L

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

Nom	Description
node/row	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
count	Spécifie le nombre de fois nécessaire pour répliquer chaque item ou séquence connectée à <b>node/row</b> .

Imaginons que vous avez un fichier XML source comportant la structure suivante :

```
<SourceList>
<person>
<name>Michelle</name>
<count>2</count>
</person>
<name>Ted</name>
<count>4</count>
</person>
<person>
<person>
<name>Ann</name>
<count>3</count>
</person>
<sourceList>
```

Avec l'aide de la fonction **replicate-item**, vous pouvez répéter chaque nom de personne un nombre différent de fois dans un composant de cible. Pour ce faire, connecter le nœud **count** de chaque personne à l'entrée **count** de la fonction **replicate-item** :



La sortie est la suivante :

```
<TargetLists>
<TargetList>
<TargetList>
<TargetString>Michelle</TargetString>
<TargetString>Michelle</TargetString>
</TargetList>
<TargetList>
<TargetString>Ted</TargetString>
<TargetString>Ted</TargetString>
<TargetString>Ted</TargetString>
<TargetString>Ted</TargetString>
<TargetList>
<TargetList</TargetList>
<TargetList</td>
<TargetList</td>
<TargetList</td>
<TargetList</td>
<TargetList</td>
<TargetList</td>
```

```
<TargetString>Ann</TargetString>
</TargetList>
</TargetLists>
```

## 6.7.9.16 replicate-sequence

Répéter tous les items dans la séquence d'entrée le nombre de fois spécifié dans l'argument **count**. Par exemple, si le décompte est **2**, la séquence **1,2,3** produit **1,2,3,1,2,3**.

<b>f</b> <sub>8</sub> replicate-sequence	
nodes/rows	
¢ count	

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## Paramètres

Nom	Description
node/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
count	Spécifie le nombre de fois nécessaire pour répliquer la séquence connectée.

## Exemple

Le mappage fictif suivant génère la séquence **1,2,3**. La séquence est traitée par la fonction **replicatesequence** et le résultat est écrit dans un fichier XML de cible.



Puisque l'argument **count** est défini sur **2**, la séquence est répliquée deux fois puis est transmise sur la cible. Par conséquent, la sortie de mappage est la suivante (à l'exception des déclarations XML et de schéma) :



```
<item>2</item>
<item>3</item>
<item>3</item>
<item>1</item>
<item>2</item>
<item>3</item>
</item>3</item>
```

## 6.7.9.17 set-empty

Retourne une séquence vide.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.9.18 skip-first-items

Saute les premiers items/nœuds "N" de la séquence d'entrée, où N est le nombre fourni par le paramètre **count** et retourne le reste de la séquence.

f <sub>8</sub> skip-first-items	
hodes/rows	
count	results L

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

Nom	Description
node/rows	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
count	Argument optionnel. Spécifie le nombre d'items à sauter. La valeur par défaut est de 1.

Le mappage fictif suivant génère la séquence **1,2,3**. La séquence est traitée par la fonction **skip-first-items** et le résultat est écrit dans un fichier XML de cible.



Puisque l'argument **count** est défini sur **2**, les premiers deux items sont sautés et les items restants sont transférés à la cible. Par conséquent, la sortie de mappage est la suivante (à l'exception des déclarations XML et de schéma) :

items>	
<item>3</item>	
/items>	

## 6.7.9.19 substitute-missing

Cette fonction est une combinaison pratique de la fonction de <u>exists</u><sup>683</sup> et <u>if-else condition</u><sup>627</sup>. Si l'item connecté à l'entrée de **node** existe, son contenu sera copié dans la cible. Sinon, le contenu de l'item connecté à l'entrée de **replace-with** sera copié dans la cible.

	$f_{\rm R}$ substitute-missing		
Ç	node		
Ç	replace-with		result

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Nom	Description
node	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit une <u>séquence</u> de zéros ou plus de valeurs. Par exemple, la connexion peut provenir d'un item XML de source, un champ CSV, un enregistrement de base de données, etc.
replace-with	Cette entrée doit recevoir une connexion depuis un item de mappage qui fournit la valeur de remplacement

## 6.7.10 core | string functions

Les fonctions string vous permettent de manipuler des données de string pour extraire des portions de string, tester des sous-string, extraire l'information concernant des strings, partager des strings, etc.

## 6.7.10.1 char-from-code

Retourne la représentation de caractère de la valeur décimale Unicode (code) fournie en tant qu'argument. **Astuce :** Pour trouver le code décimal Unicode d'un caractère, vous pouvez utiliser la fonction <u>code-from-</u> <u>char</u><sup>(10)</sup>.

	$f_{\rm R}$ char-from-code	
Ç	>code	char 🖓

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description
code	La valeur Unicode, en tant que nombre décimal.

### Exemple

Conformément aux graphiques disponibles sur le site web Unicode (<u>https://www.unicode.org/charts/</u>), le point d'exclamation a la valeur hexadécimale de 0021. La valeur correspondante en format décimal est **33**. C'est pourquoi, si vous fournissez **33** en tant qu'argument à la fonction <u>char-from-code</u>, le caractère **1** Sera retourné. Pour un exemple, voir <u>Remplacer des caractères spéciaux</u><sup>278</sup>.

## 6.7.10.2 code-from-char

Retourne la valeur décimale Unicode (code) du caractère fourni en tant qu'argument. Si le string fourni en tant qu'argument a plusieurs caractères, le code du premier caractère est retourné.



#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
char	La valeur de string d'entrée.

### Exemple

Si l'entréechar est le caractère **\$** (signe de dollar), la fonction retourne **36** (qui est la valeur Unicode décimale pour ce caractère).

## 6.7.10.3 concat

Concatène (ajoute) deux ou plusieurs valeurs dans un seul string de résultat. Toutes les valeurs d'entrée sont converties automatiquement dans le type "string". Par défaut, cette fonction a uniquement deux paramètres, mais vous pouvez en ajouter plus. Cliquer sur **Ajouter paramètres** ( • ) ou **Supprimer paramètre** ( • ) pour ajouter ou supprimer des paramètres.



**Note:** Toutes les entrées de la fonction <u>concat</u> doivent avoir une valeur. Si une des entrées n'a pas de valeur, la fonction ne sera pas appelée et une erreur se produira. Veuillez noter qu'un string vide est une valeur d'entrée valide ; néanmoins, une séquence vide (comme le résultat de la fonction <u>set-empty</u>) n'est pas une valeur valide et en conséquence la fonction échouera. Pour éviter cette situation, vous pouvez d'abord traiter les valeurs avec la fonction <u>substitute-missing</u><sup>707</sup> puis fournir le résultat en tant qu'entrée dans la fonction <u>concat</u>.

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
value1	La première valeur d'entrée.
value2	La deuxième valeur d'entrée.
valueN	La N valeur d'entrée.

### Exemple

Dans le mappage illustré ci-dessous, la fonction **concat** réunit le prénom, la constante " " et le nom de famille. La valeur de retour est ensuite écrite dans l'item de cible **FullName**. Le mappage de cette fonction est disponible dans le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\HasMarketingExpenses.mfd.



HasMark etingExpenses.mfd

## 6.7.10.4 contains

Retourne Booléenne true si la valeur de string fournie en tant qu'argument contient le sous-string fourni en tant qu'argument.

f <sub>8</sub> contains	
value	
substring	result

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
value	La valeur d'entrée (c'est à dire la "pile de foin").
substring	Le sous-string à chercher (c'est à dire l'"aiguille").

### Exemple

Si la value d'entrée est "category" et substr est "cat", la fonction retourne true.

## 6.7.10.5 normalize-space

La fonction normalize-space (voir la capture d'écran ci-dessous) supprime des espaces à gauche et à droite d'un string et remplace les espaces blancs internes avec un espace blanc unique. Les espaces blancs incluent un espace (U+0020), un onglet (U+0009), le retour à la ligne (U+000D) et la saut de ligne (U+000A). Pour des détails sur les espaces blancs, voir la <u>Recommandation XML</u>.



#### À propos des espaces insécables

Les fonctions left-trim, right-trim et normalize-space ne suppriment pas les espaces insécables. L'une des solutions possibles pourrait être de remplacer l'espace insécable, dont la représentation est 160, avec l'espace, dont la représentation décimale est 32. Le mappage ci-dessous montre qu'une fois que l'espace insécable a été remplacé, la valeur découpée SomeValue sera mappée vers la cible.



Si votre composant source est un fichier Excel, vous pouvez supprimer les espaces supplémentaires dans Excel utilisant une combinaison de fonctions TRIM, CLEAN et SUBSTITUTE. Pour les détails, voir <u>Supprimer</u> les espaces et caractères non imprimables du texte.

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
string	Le string d'entrée à normaliser.

## Exemple

Si le string d'entrée est The quick brown fox, la fonction retourne The quick brown fox.

## 6.7.10.6 starts-with

Retourne Booléenne **true** si le string fourni en tant qu'argument commence avec le sous-string fourni en tant qu'argument; **false** sinon.



### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description	
string	Туре	
substr	Le sous-string à vérifier.	

### Exemple

Si le string d'entrée est category et substr est cat, la fonction retourne true.

## 6.7.10.7 string-length

Retourne le nombre de caractères dans le string fourni en tant qu'argument.

∫<sub>x</sub> string-length ▷ string result ▷

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Nom	Description
string	Туре

Si le string d'entrée est car, la fonction retourne 3. Si le string d'entrée est string vide, la fonction retourne 0.

## 6.7.10.8 substring

Retourne la portion du string spécifié par les paramètres start et length.

f <sub>8</sub> substring		
string		
⊳ start	result 🕻	
length		

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
string	Туре
start	Spécifie la position de démarrage (index) à partir de laquelle le sous- string.devrait être extrait. Le premier index est <b>1</b> .
length	Optionnel Spécifie le nombre de caractères à extraire. Si le paramètre <b>length</b> n'est pas spécifié, le résultat est un fragment commençant par <b>start</b> jusqu'à la fin du string.

## Exemple

Si le string d'entrée est MapForce, le commencement est 1 et la longueur est 3, la fonction retourne Map. Si le string d'entrée est MapForce, le commencement est 4 et la longueur n'est pas fournie, la fonction retourne Force.

## 6.7.10.9 substring-after

Retourne la portion du string qui se produit après la première occurrence de **substr**. Si **substr** ne se produit pas dans **string**, la fonction retourne un string vide.

	$f_{\rm R}$ substring-after		
Ç	> string		
Ç	> substr	result	

#### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description	
string	Le string d'entrée	
substr	Le sous-string. Tout caractère après la première occurrence de <b>substr</b> est le résultat de la fonction.	

## Exemple

Si le string d'entrée est MapForce et substr est Map, la fonction retourne Force. Si le string d'entrée est 2020/01/04 et substr est /, la fonction retourne 01/04.

# 6.7.10.10 substring-before

Retourne la portion du string qui se produit avant la première occurrence de **substr**. Si **substr** ne se produit pas dans **string**, la fonction retourne un string vide.

$f_{\rm R}$ substring-before		
string		
⊳ substr	result	

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Description	
string	Туре	
substr	Le sous-string. Tout caractère avant la première occurrence de <b>substr</b> est le résultat de la fonction.	

## Exemple

Si le string d'entrée est **MapForce** et **substr** est **Force**, la fonction retourne **Map**. Si le string d'entrée est 2020/01/04 et **substr** est /, la fonction retourne 2020.

## 6.7.10.11 tokenize

Partage le string d'entrée en une séquence de strings en utilisant le délimiteur fourni en tant qu'argument.

	f <sub>8</sub> tokenize		
0	⊳input		
0	delimiter		result

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
input	Туре
delimiter	Le délimiteur à utiliser.

## Exemple

Si le string d'entrée est A,B,C et le délimiteur est , alors la fonction retourne une séquence de trois strings : A, B et C.



Dans le mappage fictif illustré ci-dessus, le résultat de la fonction est une séquence de strings. Conformément au mappage général <u>rules</u><sup>673</sup>, pour chaque item dans la séquence de source, un nouvel **item** est créé dans le composant de cible. Par conséquent, la sortie de mappage ressemble à :



Pour un exemple plus complexe, voir le mappage **tokenizeString1.mfd** disponible dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\**.



#### tokenizeString1.mfd

Un fragment du fichier XML de source est affiché ci-dessous. L'élément **Tool** a deux attributs : **Name** et **Code**. Les données d'élément **Tool** consistent en un texte délimité par virgule.



Ce que fait le mappage :

- La fonction tokenize reçoit des données provenant de l'item source Tool et utilise le délimiteur virgule
   pour partager les données en blocs séparés. Le premier bloc est "XML editor", le deuxième est "XSLT editor", etc.
- Pour chaque bloc résultant de la fonction tokenize, une nouvelle ligne est générée dans la cible. Cela se produit grâce à la connexion entre le résultat de la fonction et l'item **Rows** dans le composant de cible.
- Le résultat de la fonction tokenize est aussi mappé dans la fonction left-trim, qui supprime l'espace de début de chaque bloc.
- Le résultat de la fonction left-trim (chaque bloc) est écrite dans l'item Feature du composant cible.
- Le fichier de sortie du composant cible a été défini en tant que fichier CSV (AltovaToolFeatures.csv) et comporte un délimiteur de champ sous la forme d'un point-virgule (double cliquer le composant pour voir les paramètres.

Le résultat du mappage est que, pour chaque bloc créé par la fonction tokenize, une nouvelle ligne est créée dans le fichier CSV cible. Un fragment de la sortie de mappage ressemble à :

Tool;Feature XMLSpy;XML editor XMLSpy;XSLT editor XMLSpy;XSLT editor XMLSpy;XSLT debugger XMLSpy;XQuery editor XMLSpy;XQuery debugger XMLSpy;XML Schema / DTD editor XMLSpy;WSDL editor XMLSpy;WSDL editor XMLSpy;SOAP debugger MapForce;Data integration MapForce;CAtabase mapping MapForce;text conversion MapForce;EDI translator MapForce;EXCEl mapping

## 6.7.10.12 tokenize-by-length

Partage le string d'entrée dans une séquence de strings. La taille de chaque string résultant est déterminé par le paramètre **length**.

	f <sub>x</sub> tokenize-by-length		
l>input		put	
ę	length		result

## Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
input	Туре
length	Détermine la longueur de chaque string dans la séquence générée des strings.

## Exemple

Si le string d'entrée est ABCDEF et que la longueur est 2, alors la fonction retourne une séquence de trois strings : AB, CD, et EF.





Dans le mappage fictif illustré ci-dessus, le résultat de la fonction est une séquence de strings. Conformément au mappage général <u>rules</u><sup>(373)</sup>, pour chaque item dans la séquence de source, un nouvel **item** est créé dans le composant de cible. Par conséquent, la sortie de mappage ressemble à :



Pour un exemple plus complexe, voir le mappage tokenizeString2.mfd disponible dans le dossier <Documents>\Altova\MapForce2023\MapForceExamples\.



#### tokenizeString2.mfd

Le fichier de source XML est affiché ci-dessous et est le même que celui utilisé dans l'exemple précédent. L'élément **MissionKit** a deux attributs : **Edition** et **ToolCodes**, mais pas de contenu d'élément MissionKit. Veuillez noter que le contenu XML non pertinent pour cet exemple a été supprimé.



```
<Tool Name="StyleVision" Code="SV"><!--..-></Tool>
<Tool Name="UModel" Code="UM"><!--..-></Tool>
<Tool Name="DatabaseSpy" Code="DS"><!--..-></Tool>
<Tool Name="DiffDog" Code="DD"><!--..-></Tool>
<Tool Name="SchemaAgent" Code="SA"><!--..-></Tool>
<Tool Name="SchemaAgent" Code="SW"><!--..-></Tool>
<Tool Name="SemanticWorks" Code="SW"><!--..-></Tool>
<Tool Name="SemanticWorks" Code="SW"><!--..-></Tool>
<Tool Name="Authentic" Code="AU"><!--..-></Tool>
<Tool Name="Authentic" Code="AU"><!--..-></Tool>
<MissionKit Edition="Enterprise Software Architects" ToolCodes="XSMFSVUMDSDDSASW"/>
<MissionKit Edition="Professional Software Architects" ToolCodes="XSMFSVUMDS"/>
<MissionKit Edition="Enterprise XML Developers" ToolCodes="XSMFSVUDSASW"/>
<MissionKit Edition="Professional XML Developers" ToolCodes="XSMFSVUDSASW"/>
</AltovaTools>
```

L'objectif du mappage est de générer une liste montrant quels outils Altova font partie des éditions MissionKit respectives

Comment fonctionne le mappage :

- Le composant d'entrée **SelectMissionKit** agit comme un paramètre dans le mappage, il reçoit sa valeur par défaut depuis une constante, dans ce cas "Enterprise XML Developers".
- La fonction equal compare l'édition fournie en tant que paramètre avec l'item Edition depuis le fichier XML de source et transmet le résultat vers le paramètre bool du filtre ToolCodes.
- L'entrée **node/row** du filtre **ToolCodes** est fourni par l'item **ToolCodes** du fichier de source. La valeur pour l'édition "Enterprise XML Developers" est : **XSMFSVDDSASW**.
- La valeur **XSMFSVDDSASW** est transmise dans le paramètre **on-true**, puis vers le paramètre **input** de la fonction tokenize-by-length.
- La fonctiontokenize-by-length partage la valeur **XSMFSVDDSASW** en plusieurs blocs de deux caractères chacun. Le paramètre **length** est **2** ; c'est pourquoi 6 blocs sont créés.
- Chaque bloc est comparé à la valeur Code de 2 caractères depuis le fichier de source (dont il y a 9 items au total). Le résultat de la comparaison (true/false) est transmis dans le paramètre bool du filtre. Veuillez noter que *tous* les blocs produits par la fonction tokenize-by-length sont transmis dans le paramètre node/row du filtre.
- Les fonctions exists vérifient maintenant les nœuds existants/non-existants transmis vers le paramètre on-true du composant de filtre. Les nœuds existants sont ceux dans lesquels *il y a une correspondance* entre le bloc ToolCodes et la valeur de Code. Des nœuds non-existants se trouvent là où il n'y avait pas de bloc ToolCodes pour correspondre à une valeur Code.
- Chaque résultat bool de la fonction exists est transmis dans le composant if-else qui génèer un "Y" dans la cible si le nœud existe, ou un "N" si le nœud n'existe pas.

Résultat du mappage :

```
Tool;MissionKit for Enterprise XML Developers
XMLSpy;Y
MapForce;Y
StyleVision;Y
UModel;N
DatabaseSpy;N
DiffDog;Y
SchemaAgent;Y
```

SemanticWorks;Y Authentic;N

## 6.7.10.13 tokenize-regexp

Partage le string d'entrée dans une séquence de strings. Tout sous-string qui correspond au **pattern** d'expression régulier fourni en tant qu'argument définit le séparateur. Les strings correspondants (séparateur) ne sont pas inclus dans le résultat retourné par la fonction.

**Note :** Lors de la génération de code C++, C# ou Java, les fonctions avancées de la syntaxe d'expression régulière peuvent différer légèrement. Voir la documentation regex de chaque langage pour plus d'informations.

$f_{\rm x}$ tokenize-regexp		
l>input		
pattern 🖓	result 🖗	
flags		

### Langages

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Description
input	Туре
pattern	Apporte un pattern d'expression régulier. Tout sous-string qui correspond au pattern sera traité en tant que délimiteur. Pour plus d'informations, voir <u>Expressions régulières</u> <sup>627</sup> .
flags	Paramètre optionnel. Fournit l'expression régulière <u>flags</u> <sup>623</sup> à utiliser. Par exemple, le flag "i" instruit le processus de mappage de fonctionner dans le mode insensible à la casse.

## Exemple

L'objectif du mappage illustré ci-dessous est de partager le string **a**, **b c**, **d** dans une séquence des strings, où chaque caractère alphabétique est un item dans la séquence. Tout espace blanc ou virgule redondant doit être supprimé.


Pour obtenir cet objectif, le pattern d'expression régulière [,]+ a été fourni en tant que paramètre dans la fonction tokenize-regexp. Ce pattern a la signification suivante :

- Il correspond à un des caractères dans la classe de caractère [,]. Ainsi, un partage se produira à chaque fois qu'une virgule ou qu'une espace est rencontrée dans le string d'entrée.
- Le quantificateur + spécifie qu'une ou plusieurs occurrences de la classe du caractère précédent doivent trouver correspondance. Sans ce quantificateur, chaque occurrence d'espace ou de virgule créera un item séparé dans la séquence résultante de strings, ce qui n'est pas le résultats intenté.

Le sortie de mappage est la suivante :

```
<items>
<item>a</item>
<item>b</item>
<item>c</item>
<item>d</item>
</items>
```

## 6.7.10.14 translate

Effectue un remplacement caractère par caractère. Il regarde dans la **value** pour les caractères contenus dans **string1**, et remplace chaque caractère avec celui dans la même position dans le **string2**. Lorsqu'il n'y a pas de caractères correspondants dans **string2**, le caractère est supprimé.



Langages Built-in, C++, C#, Java, XQuery, XSLT 1.0, XSLT 2.0, XSLT 3.0.

Nom	Description	
value	Туре	

Nom	Description	
string1	Fournit une liste des caractères de recherche. La position de chaque caractère se trouvant dans le string est importante.	
string2	Fournit une liste de caractères de remplacement. La position de chaque caractère de remplacement doit correspondre à celui dans <b>string1</b> .	

### Exemple

Supposons que vous souhaitez convertir de string [12,3] en (12.3). Concrètement, les crochets doivent être remplacés par des parenthèses, et toute virgule doit être remplacée par le caractère de point. Pour ce faire vous pouvez appeler la fonction translate comme suit :



Dans le mappage ci-dessus, la première constante fournit le string d'ouverture à traiter. La seconde et la troisième constante fournit une liste des caractères en tant que **string1** et **string2**, respectivement.

string1	[,]
string2	(.)

Veuillez noter que les deux **string1** et **string2** ont le même nombre de caractères. Pour chaque caractère dans **string1**, le caractère équivalent à la même position depuis **string2** sera utilisé en tant que remplacement. Par conséquent, le remplacement suivant aura lieu :

- Chaque [ sera remplacé par un (
- Chaque, sera remplacé par un.
- Chaque ] sera remplacé par un ]

Le sortie de mappage est la suivante :

#### (12.3)

Cette fonction peut aussi être utilisée pour retirer certains caractères sélectivement depuis un string. Pour ce faire, définir le paramètre **string1** sur les caractères que vous souhaitez supprimer, et **string2** sur un string vide. Par exemple, le mappage ci-dessous supprime tous les chiffres depuis le string **38ab8a7a65xkh3**.



Le sortie de mappage est la suivante :

abaaxkh

# 6.7.11 db

La bibliothèque **db** contient des fonctions qui vous permettent de définir les résultats de mappage lorsque vous rencontrez des champs null dans des bases de données.

# 6.7.11.1 is-not-null

Retourne false si le champ est null, sinon, retourne true.

∫<sub>8</sub> is-not-null ⊳field result ⊳

Langages Built-in, C++, C#, Java.

### Paramètres

Nom	Description
field	Le champ de base de données.

## 6.7.11.2 is-null

Retourne true si le champ est null, sinon, retourne false.



Built-in, C++, C#, Java.

### Paramètres

Nom	Description	
field	Le champ de base de données.	

# 6.7.11.3 set-null

Définit un champ de base de données sur null. Cette fonction écrasera aussi une valeur par défaut avec null. Si elle est connectée à autre chose, c'est à dire, n'est pas un champ de base de données, elle se comportera comme une séquence vide. Veuillez noter les points suivants :

- Une connexion set-null à une fonction différente résultera généralement dans le fait que l'autre fonction ne sera pas être appelée du tout. Une connexion set-null à une fonction de séquence, par ex. count, appellera la fonction avec une séquence vide.
- Une connexion **set-null** à des filtres et des conditions if-else fonctionnent comme prévu, les champs sont définis sur null. Pour les filtres cela signifie l'entrée "node/row".
- L'utilisation set-null en tant qu'une entrée pour un élément simpleType ne créera pas cet élément dans le composant cible

$\mathbf{f}_{\mathbf{x}}$	set-null	
	result 🕻	2

Langages Built-in, C++, C#, Java.

# 6.7.11.4 substitute-null

Retourne le champ lui-même s'il n'est pas null ; sinon, replace-with est retourné.



Langages Built-in, C++, C#, Java.

### Paramètres

Nom	Description	
field	Le champ de base de données.	
replace-with	La valeur de remplacement.	

### Exemple

Le mappage ci-dessous montre un exemples de la fonction **substitute-null** en cours d'utilisation. Ce mappage est appelé **DB-ApplicationList.mfd** et est disponible dans le dossier **<Documents>\Altova\MapForce2023\MapForceExamples\**.



Le mappage lit des données depuis une base de données Access qui contient un table "Applications".

	AppID •	AppName •	Description •	Category •	URL •
1	1	Altova MapForce	The premier data mapping tool.	IDE	www.altova.com/xmlspy
2	2	Notepad	[NULL]	[NULL]	[NULL]

La première fonction vérifie si le champ **Category** est null dans la table "Applications". Puisque ce champ est null pour l'application Notepad, la valeur de substitution "Misc" est mappée dans l'item **Category** du fichier de texte de cible.

La seconde fonction vérifie si le champ **Description** est null. À nouveau, ce champ est null pour l'application Notepad, la valeur de substitution "No description" est donc mappée dans l'item **Description** du fichier de cible.

# 6.7.12 lang | datetime functions

Les fonctions de date et heure depuis la bibliothèque **lang** peuvent être utilisées pour manipuler des dates, heures et durées. Contrairement aux fonctions date et heure depuis la bibliothèque **core**, ces fonctions sont disponibles uniquement lorsque vous sélectionnez les langages Built-in, Java, C#, ou C++.

# 6.7.12.1 age

Retourne le nombre des années pleines qui se sont écoulées entre la date de naissance fournie en tant que l'argument et maintenant.

	$f_{\rm x}$ age	
Ç	>birthdate	
ļ	now	result

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
birthdate	xs:date	Obligatoire. Fournit la fate de naissance en tant que valeur xs:date.
now	xs:date	Paramètre optionnel. Le défaut est la date de système actuel. Si une valeur est mappée à l'argument now, la fonction retourne la différence entre la date de naissance et maintenant, en années pleines.

### 6.7.12.2 convert-to-utc

Convertit la valeur d'heure fournie en tant qu'argument au UTC (Coordinated Universal Time). La fonction prend en compte le composant de fuseau horaire (par exemple, "+5:00").



Langages Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
time	xs:dateTime	Fournit la valeur xs:dateTime à convertir.

### Exemple

Si la valeur d'entrée est 2001-12-17T09:30:02+05:00, le résultat de la fonction est 2001-12-17T04:30:02.

Si la valeur d'entrée est 2001-12-17T09:30:02z, le résultat de la fonction est 2001-12-17T09:30:02. Dans ce cas, aucune conversion ne s'est produite, parce que le "Z" en fin de ligne a déjà défini cette heure pour être l'heure "Zero" (ou "Zulu"), qui est le même que l'UTC.

## 6.7.12.3 date-from-datetime

Retourne la partie *date* provenant de la valeur xs:dateTime fournie en tant qu'rgument. La partie *time* est définie sur zéro. Le fuseau horaire n'est pas modifié.

∫<sub>8</sub> date-from-datetime ▷ datetime date ▷

#### Langages Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur xs:dateTime à traiter.

### Exemple

Si la valeur d'entrée est 2001-12-17T09:30:02+05:00, le résultat de la fonction est 2001-12-17+05:00.

# 6.7.12.4 datetime-add

Retourne une valeur xs:dateTime obtenue en ajoutant une durée (le second argument) dans un datetime (le premier argument).

$f_{\rm R}$ datetime-add		
datetime	rocult C	
duration	result	

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!	
datetime	xs:dateTime	Fournit la valeur xs:dateTime à utiliser en tant qu'entrée.	
duration	xs:duration	Fournit la valeur xs:duration.	
		<ul> <li>Un exemple de durée est P1Y2M3DT04H05M59s, où :</li> <li>"P" est de désignateur de la période et est obligatoire ;</li> <li>Le reste des caractères dénote, dans cet ordre : 1 an, 2 mois, 3 jours, T (Désignateur de temps), 04 heures, 05 minutes, 59 secondes.</li> </ul>	
		Si le caractère "moins" apparaît avant le désignateur "P", cela indique une durée négative, par exemple : -p1D.	

### Exemple

Partons du principe que la valeur d'entrée **datetime** est 2001-12-17T09:30:02+05:00. Si la **durée** est P10D (10 jours), le résultat de la fonction est 2001-12-27T09:30:02+05:00.

Pour obtenir la date d'hier, connecter la fonction **now** à l'entrée **datetime**. Dans le mappage ci-dessous, la période **-PID** signifie "moins 1 jour", pour que le mappage retourne la date de la veille.



# 6.7.12.5 datetime-diff

Retourne la durée obtenue en soustrayant **datetime2** (second argument) depuis **datetime1** (premier argument). Le résultat peut être mappé dans un type de données string ou durée.

	$f_{\! 8}$ datetime-diff		
Ç	>datetime1		
ç	>datetime2	result	

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!	
datetime1	xs:dateTime	Fournit la première valeur xs:dateTime.	
datetime2	xs:duration	Fournit la seconde valeur xs:dateTime.	

### Exemple

Dans le mappage illustré ci-dessous, la fonction datetime-diff soustrait le datetime de départ de vol 2001-12-17T09:30:02+05:00 depuis le datetime d'arrivée 2001-12-17T19:30:02+05:00. Veuillez noter que le datetime d'arrivée est la valeur supérieure, il est donc connecté à la première entrée de la fonction.



La sortie de mappage est la différence entre les deux (une période de 10 heures) :

PT10H

# 6.7.12.6 datetime-from-date-and-time

Retourne une valeur xs:dateTime contruite depuis une valeur xs:date (premier argument) et une valeur xs:time (second argument). Le résultat peut être mappé dans un string ou un type de données xs:dateTime.

	$f_{\rm x}$ datetime-from-date-and-time			
ç	datevalue			
ł	timevalue		datetimet	

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!	
datevalue	xs:date	Fournit une valeur de type xs:date	
timevalue	xs:time	Fournit une valeur de type xs:time	

#### Exemple

Dans le premier argument est 2012-06-29 et le second argument est 11:59:55, la fonction retourne 2012-06-29T11:59:55.

# 6.7.12.7 datetime-from-parts

Retourne une valeur de type xs:dateTime construite depuis n'importe quelle combinaison des parties suivantes en tant qu'arguments : annéer, mois, jour, heure, minute, seconde, milliseconde et fuseau horaire. Cette fonction normalise automatiquement les paramètres fournis. Par exemple, le 32 janvier sera changé automatiquement en 1er février.

	f <sub>x</sub> datetime-from-parts		
Ę	>year		
Ę	>month		
Ę	> day		
ł	hour	datatima	
ł	> minute	uatetime	
ł	second		
ł	>millisecond		
ł	timezone		

Langages Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!	
year	xs:int	Fournit l'année.	
month	xs:int	Fournit le mois.	
day	xs:int	Fournit le jour du mois.	
hour	xs:int	Optionnel Fournit l'heure.	
minute	xs:int	Optionnel Fournit la minute.	
second	xs:int	Optionnel Fournit la seconde.	
millisecond	xs:decimal	Optionnel Fournit la milliseconde.	
timezone	xs:int	Optionnel Fournit le fuseau horaire, en minutes. Cette valeur peut être négative.	

### Paramètres

### Exemple

Le mappage suivant construit une valeur xs:dateTime provenant de parties qui sont fournies par des constantes.



La sortie de mappage est 2020-04-17T08:58:54.333-01:00.

# 6.7.12.8 day-from-datetime

Retourne le jour en tant que valeur d'entier, depuis la valeur xs:dateTime fournie en tant qu'argument .

	J <sub>x</sub> day-from-datetime		
Ç	datetime		day 🗘

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!	
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.	

### Exemple

Si datetime est 2001-12-17T10:30:03+01:00, la fonction retourne 17.

# 6.7.12.9 day-from-duration

Retourne le jour en tant que valeur d'entier, depuis la valeur xs:duration fournie en tant qu'argument .

∫<sub>x</sub> day-from-duration ▷ duration day ▷

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

### Exemple

Si duration est P1Y2M3DT10H30M, la fonction day-from-duration retourne 3.

# 6.7.12.10 duration-add

Retourne la durée obtenue en ajoutant deux durées.

	$f_{\rm s}$ duration	n-add
0	>duration1	
0	>duration2	result

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!	
duration1	xs:duration	Fournit la première valeur d'entrée de type xs:duration.	
duration2	xs:duration	Fournit la deuxième valeur d'entrée de type xs:duration.	

### Exemple

Si la première durée est **POYOM3DTO3HOM** (3 jours et 3 heures) et la seconde durée est **POYOM3DTO1HOM** (3 jours et 1 heure), la fonction retourne **P6DT4H** (6 jours et 4 heures).

# 6.7.12.11 duration-from-parts

Retourne une valeur de type xs:duration calculée en combinant les parties suivantes fournies en tant qu'arguments : année, mois, heure, minute, seconde, milliseconde, négative.

Un exemple de durée est p1Y2M3DT04H05M59s, où :

- "P" est de désignateur de la période et est obligatoire ;
- Le reste des caractères dénote, dans cet ordre : 1 an, 2 mois, 3 jours, T (Désignateur de temps), 04 heures, 05 minutes, 59 secondes.

Si le caractère "moins" apparaît avant le désignateur "P", cela indique une durée négative, par exemple : -PID.



Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!	
year	xs:int	Fournit l'année.	
month	xs:int	Fournit le mois.	
day	xs:int	Fournit le jour du mois.	
hour	xs:int	Optionnel Fournit l'heure.	
minute	xs:int	Optionnel Fournit la minute.	
second	xs:int	Optionnel Fournit la seconde.	
millisecond	xs:decimal	Optionnel Fournit la milliseconde.	
negative	xs:boolean	Optionnel Doit être true pour une durée négative ; false sinon.	

## Exemple

Le mappage suivant génère une durée négative de 1 an, 4 mois, 17 jours, 8 heures, 58 minutes, et 54.333 secondes.



La sortie de mappage est -P1Y4M17DT8H58M54.333s.

# 6.7.12.12 duration-subtract

Retourne la valeur xs:duration obtenue en soustrayant duration2 depuis duration1.

Un exemple de durée est p1Y2M3DT04H05M59s, où :

- "P" est de désignateur de la période et est obligatoire ;
- Le reste des caractères dénote, dans cet ordre : 1 an, 2 mois, 3 jours, T (Désignateur de temps), 04 heures, 05 minutes, 59 secondes.

Si le caractère "moins" apparaît avant le désignateur "P", cela indique une durée négative, par exemple : -PID.

	$f_{\rm x}$ duration-	subtract	
Ç	>duration1		
Ç	>duration2	result	

### Langages

Built-in, C++, C#, Java.

### Exemple

Si duration1 est POYOMODTO5H07M (5 heures et 7 minutes) et duration2 est PT1H (1 heure), la fonction retourne PT4H7M (4 heures et 7 minutes).

## 6.7.12.13 hour-from-datetime

Retourne l'heure, en tant qu'une valeur d'entier depuis la valeur xs:dateTime fournie en tant qu'argument.

	$f_8$	hour-from	-datetime
datetime		atetime	hour 🖓

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

#### Exemple

Si datetime est 2001-12-17T09:30:02+05:00, la fonction retourne 9.

## 6.7.12.14 hour-from-duration

Retourne l'heure, en tant qu'une valeur d'entier depuis la valeur xs:duration fournie en tant qu'argument.

J <sub>8</sub> hour-fro	om-duration
duration	hour

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

### Exemple

Si duration est POYOMODTO5H07M, la fonction retourne 5.

# 6.7.12.15 leapyear

Retourne Boolean **true** si l'année de la valeur xs:dateTime fournie en tant qu'argument est une année bissextile; **false** sinon.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si **datetime** est 2020-04-17T09:30:02+02:00, la fonction retourne **true**, puisque l'année 2020 est une année bissextile.

# 6.7.12.16 millisecond-from-datetime

Retourne les millisecondes, en tant qu'une valeur xs:decimal, depuis la valeur xs:dateTime fournie en tant qu'argument.

$f_8$ millisecond	millisecond-from-datetime	
datetime	millisecond Þ	

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si datetime est 2001-12-17T09:30:02.544+05:00, la fonction retourne 544.

## 6.7.12.17 millisecond-from-duration

Retourne les millisecondes, en tant qu'une valeur xs:decimal depuis la valeur xs:duration fournie en tant qu'argument.

 f<sub>x</sub> millisecond-from-duration

 P duration

 millisecond P

#### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

### Exemple

Si duration est P0Y0M0DT05H07M02.227s, la fonction retourne 227.

# 6.7.12.18 minute-from-datetime

Retourne les minutes, en tant qu'une valeur d'entier depuis la valeur xs:dateTime fournie en tant qu'argument.

∫ <sub>8</sub> mi	nute-fro	m-datetime
datet	ime	minute Þ

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si datetime est 2001-12-17T09:30:02.544+05:00, la fonction retourne 30.

# 6.7.12.19 minute-from-duration

Retourne les minutes, en tant qu'une valeur d'entier depuis la valeur xs:duration fournie en tant qu'argument.

	$f_{\rm x}$ minute-from-duration		
ę	duration		minute 🗘

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

### Exemple

Si duration est POYOMODTO5H07M02.227s, la fonction retourne 7.

# 6.7.12.20 month-from-datetime

Retourne le mois, en tant qu'une valeur d'entier depuis la valeur xs:dateTime fournie en tant qu'argument.

	$\mathbf{f}_{\mathbf{x}}$	month-from-datetime		
datetime		atetime	month 🕻	>

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si datetime est 2001-12-17T09:30:02.544+05:00, la fonction retourne 12.

## 6.7.12.21 month-from-duration

Retourne le mois, en tant qu'une valeur d'entier depuis la valeur xs:duration fournie en tant qu'argument.

∫ <sub>8</sub> month-fro	month-from-duration	
duration	month 🗘	

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

#### Exemple

Si duration est p0y04m0DT05H07m02.227s, la fonction retourne 4.

### 6.7.12.22 now

Retourne la date et l'heure actuelles (y compris le fuseau horaire), en tant que valeur xs:dateTime.



### Langages

Built-in, C++, C#, Java.

### Exemple

Le mappage suivant sort la date et l'heure actuelles. La sortie change à chaque fois que le mappage est exécuté.



Une sortie d'exemple serait 2020-04-17T11:42:34.684+02:00.

Pour voir un exemple de la manière d'extraire la date d'hier, voir la fonction <u>core | lang | datetime-add</u><sup>[27]</sup>.

## 6.7.12.23 remove-timezone

Supprime le composant de fuseau horaire depuis le paramètre d'entrée time (of type xs:dateTime).

	f <sub>8</sub> remove-timezone	
Ç	>time	time 🗘

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
time	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si time est 2001-12-17T09:30:02+05:00, la fonction retourne 2001-12-17T09 30:02.

# 6.7.12.24 second-from-datetime

Retourne les secondes, en tant qu'une valeur d'entier depuis la valeur xs:dateTime fournie en tant qu'argument.

∫<sup>®</sup> second-from-datetime ▷ datetime second ▷

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si datetime est 2001-12-17T09:30:02.544+05:00, la fonction retourne 2.

## 6.7.12.25 second-from-duration

Retourne les secondes, en tant qu'une valeur d'entier depuis la valeur xs:duration fournie en tant qu'argument.

	$\mathbf{f}_{\mathbf{x}}$	second-from-duration	
duration		uration	second 🖡

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

#### Exemple

Si duration est p0y04M0DT05H07M02.227s, la fonction retourne 2.

# 6.7.12.26 time-from-datetime

Retourne le composant time, en tant qu'une valeur xs:time, depuis la valeur xs:dateTime fournie en tant qu'argument.

	$f_{\!8}$ time-from-datetime			
ç	>da	atetime	time 🕻	>

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

#### Exemple

Si datetime est 2001-12-17T09:30:02+05:00, la fonction retourne 09:30:02+05:00.

# 6.7.12.27 timezone

Retourne l'offset de fuseau horaire, en minutes, depuis la valeur xs:dateTime fournie en tant qu'argument. Retourne 0 pour UTC.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si datetime est 2001-12-17T09:30:02.544+05:00, la fonction retourne 300.

# 6.7.12.28 weekday

Retourne le jour de la semaine depuis la valeur xs:dateTime fournie en tant qu'argument. La fonction retournera la valeur 1 pour lundi, la valeur 2 pour mardi, etc.

∫<sub>x</sub> weekday ⊳datetime weekday⊳

### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

## 6.7.12.29 weeknumber

Retourne le numéro de la semaine dans l'année à partir de la valeur xs:dateTime fournie en tant qu'argument. La fonction retournera la valeur 1 pour la première semaine de l'année, la valeur 2 pour la deuxième semaine, etc.



#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

## 6.7.12.30 year-from-datetime

Retourne l'année, en tant qu'une valeur d'entier depuis la valeur xs:dateTime fournie en tant qu'argument.

f <sub>x</sub> year-from-datetime		-datetime
datetime		year 🗘

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
datetime	xs:dateTime	Fournit la valeur d'entrée de type xs:dateTime.

### Exemple

Si datetime est 2001-12-17T09:30:02.544+05:00, la fonction retourne 2001.

# 6.7.12.31 year-from-duration

Retourne l'année, en tant qu'une valeur d'entier depuis la valeur xs:duration fournie en tant qu'argument.

$f_{\rm s}$ year-from-duration		duration	
Ç	> dı	uration	year 🗘

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
duration	xs:duration	Fournit la valeur d'entrée de type xs:duration.

### Exemple

Si duration est p01Y04M0DT05H07M02.227s, la fonction retourne 1.

# 6.7.13 lang | file functions

Utiliser les fonctions **file** depuis la bibliothèque **lang** pour lire du contenu binaire depuis des fichiers, ou écrire du contenu binaire dans des fichiers.

# 6.7.13.1 read-binary-file

Retourne le contenu du fichier spécifié en tant qu'un BLOB (binary large object) de type xs:base64Binary. Veuillez noter que bien que le type de données est appelé "base64Binary", la représentation interne est juste un BLOB. Ce n'est que lorsque vous mappez le résultat de la fonction dans un nœud XML de type xs:base64Binary qu'il sera encodé en base64. Vous pourriez aussi mapper le résultat de la fonction dans xs:hexBinary, dans un blob de base de données ou dans un champ binaire dans une structure Protocol Buffers.

	$f_{\rm R}$ read-binary-file		
Ç	>filepath	contant [	
ţ	>must-exist	content	

Pour lire un fichier binaire dans un mappage, fournir son chemin en tant qu'entrée dans l'argument **filepath**. Si le **filepath** est relatif, MapForce cherchera le fichier dans le même répertoire que le mappage. L'argument **must-exist** est optionnel ; si le fichier ne peut pas être ouvert et que ce paramètre est **vrai**, le mappage lance une erreur. Si le fichier ne peut pas être ouvert et que ce paramètre est **faux**, un binaire vide est retourné.

Built-in.

#### Paramètres

Nom	Туре	echo Hello, World!
filepath	xs:string	Le chemin de fichier.
must-exist	xs:boolean	Paramètre optionnel. Si le fichier ne peut pas être ouvert et que ce paramètre est <b>true</b> , le mappage lance une erreur. Si le fichier ne peut pas être ouvert et que ce paramètre est <b>false</b> , un binaire vide est retourne. La valeur par défaut est <b>true</b> .

### Exemple

Voir Exemple: Lire des fichiers binaires 473.

# 6.7.13.2 write-binary-file

Cette fonction écrit du contenu binaire dans le chemin de fichier spécifié et retourne le chemin du fichier écrit. Si un fichier binaire est la seule sortie désirée, connecter le résultat de la fonction à un composant <u>sortie</u> <u>simple</u><sup>469</sup>. Étant donné que cette fonction écrit un fichier à chaque fois qu'une sortie est utilisée dans le mappage, il est recommandé de connecter le résultat de la fonction directement dans un composant cible, sans utiliser un autre traitement entre temps.

$f_{\rm s}$ write-binary-file		
⇒ filepath	filemeth N	
content 🗘	Thepath	

Pour écrire des fichiers binaires, fournir leurs chemins en tant qu'entrée dans l'argument **filepath**. Si **filepath** est relatif, MapForce générera le fichier dans le même répertoire que le mappage. L'argument **content** doit être connecté au contenu binaire (par exemple, un champ BLOB dans une base de données).

Lorsque vous consultez le mappage dans MapForce, la fonction génère des fichiers temporaires par défaut, au lieu d'écrire des fichiers directement sur le disque. Pour enregistrer les fichiers temporaires sur le disque, cliquer tout d'abord sur l'onglet **Sortie**, puis cliquer sur la touche de la barre d'outils **Enregistrer sortie générée u** ou **Enregistrer toutes les sorties générées**, selon les cas.

Pour configurer MapForce pour écrire une sortie directement dans des fichiers finaux au lieu des fichiers

temporaires, sélectionner la commande de menu **Outils | Options**, cliquer sur **Général**, puis sélectionner l'option **Écrire directement dans les fichiers de sortie finaux**. Veuillez noter que cette option écrase tous les fichiers existants avec le même nom.

La fonction retourne le nom du fichier final (pas temporaire), même si le fichier final n'est pas encore enregistré sur le disque (c'est le cas lorsque vous consultez le mappage et que l'option Écrire directement dans les fichiers de sortie finaux est désactivée).

Veuillez noter qu'il n'est pas possible pour un mappage de relire son propre fichier de sortie.

### Langages

Built-in.

### Paramètres

Nom	Туре	echo Hello, World!
filepath	xs:string	Le chemin de fichier d'entrée.
contenu	xs:base64Binary	Le contenu binaire de type xs:base64Binary.

### Exemple

Voir Exemple: Écrire des fichiers binaires 474.

# 6.7.14 lang | generator functions

Les fonctions generator provenant de la bibliothèque **lang** sont des fonctions qui génèrent des valeurs (actuellement, **create-guid** est la seule fonction de ce type).

# 6.7.14.1 create-guid

Crée un identificateur global unique (GUID), en tant que string encodé hex. Cette fonction peut être utilisée pour générer des valeurs uniques, directement depuis le mappage, pour des champs de base de données ou d'autres types de composant. Voir aussi la fonction <u>format-guid-string</u><sup>766</sup>.



Langages Built-in, C++, C#, Java.

# 6.7.15 lang | logical functions

Les fonctions logiques provenant de la bibliothèque **lang** contiennent des fonctions qui évaluent des types de valeur divers en utilisant une logique booléenne.

## 6.7.15.1 logical-xor

Retourne true si value1 est différent de value2; false sinon.

	f <sub>8</sub> logical-xor			
⊳value1		lue1	rocult [	ļ
ę	≻va	lue2	result L	

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
value1	xs:boolean	La première valeur d'entrée.
value2	xs:boolean	La deuxième valeur d'entrée.

# 6.7.15.2 negative

Retourne true si la valeur d'entrée est négative (moins de zéro) ; false sinon.



### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
value	xs:decimal	La valeur d'entrée.

# 6.7.15.3 numeric

mff-lib-lang-read-binary-file.txtRetourne true si la valeur d'entrée est un nombre ou un string qui peut être parsdé en tant que nombre ; false sinon.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:decimal	La valeur d'entrée.

### Exemple

Si la valeur d'entrée est le string "4.33", la fonction retourne true. Si la valeur d'entrée est le string "4.33 uso", la fonction retourne false.

## 6.7.15.4 positive

Retourne true si la valeur d'entrée est positive (égale ou supérieure à zéro); false sinon.



### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
value	xs:decimal	La valeur d'entrée.

# 6.7.16 lang | math functions

Les fonctions math provenant de la bibliothèque **lang** peuvent être utilisée pour effectuer diverses opérations mathématiques dans le mappage.

### 6.7.16.1 abs

Retourne la valeur absolue de la valeur numérique fournie en tant qu'argument. Si l'argument n'est pas négatif, l'argument est retourné. Si l'argument est négatif, la négation de l'argument est retournée.



#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
value	xs:decimal	La valeur d'entrée.

## 6.7.16.2 acos

Retourne l'arc cosinus de value, dans la plage de -pi/2 à pi/2.



### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

## 6.7.16.3 asin

Retourne l'arc sinus de value, dans la plage de -pi/2 à pi/2.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.4 atan

Retourne l'arc tangente de value, dans la plage de -pi/2 à pi/2.

$f_8$ ata	n
⊳value	result 🕻

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

## 6.7.16.5 cos

Retourne le cosinus trigonométrique de l'angle donné par value. L'unité de la valeur est radian.



Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.6 degrees

Convertit un angle mesuré en radian à un angle équivalent approximatif mesuré en degrés.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.7 divide-integer

Retourne le résultat entier de la division de value1 par value2.

$f_{\rm R}$ divide-integer	
⊳value1	
¢value2	result

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value1	xs:decimal	La première valeur d'entrée.
value2	xs:decimal	La deuxième valeur d'entrée.

### Exemple

Si la première valeur est 15 et que la seconde valeur est 2, la fonction retourne 7.

## 6.7.16.8 exp

Retourne le nombre d'Euler e élévé à la puissance de value.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.9 log

Retourne le logarithme naturel (base e) de value.



Langages Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.10 log10

Retourne le logarithme décimal (base 10) de value.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.11 max

Retourne la valeur numérique de la plus grande valeur fournie en tant qu'argument. Par défaut, cette fonction a uniquement deux paramètres, mais vous pouvez en ajouter plus. Cliquer sur **Ajouter paramètres** ( ) ou **Supprimer paramètre** ( ) pour ajouter ou supprimer des paramètres, voir aussi <u>Ajouter ou supprimer des arguments de fonction</u>.



### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
value1	xs:decimal	La première valeur d'entrée.

Nom	Туре	echo Hello, World!
value2	xs:decimal	La deuxième valeur d'entrée.
valueN	xs:decimal	La nème valeur d'entrée.

# 6.7.16.12 min

Retourne la valeur numérique de la plus petite valeur fournie en tant qu'argument. Par défaut, cette fonction a uniquement deux paramètres, mais vous pouvez en ajouter plus. Cliquer sur **Ajouter paramètres** ( ) ou **Supprimer paramètre** ( ) pour ajouter ou supprimer des paramètres, voir aussi <u>Ajouter ou supprimer des arguments de fonction</u>.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value1	xs:decimal	La première valeur d'entrée.
value2	xs:decimal	La deuxième valeur d'entrée.
valueN	xs:decimal	La nème valeur d'entrée.

# 6.7.16.13 pi

Retourne la valeur de la constante mathématique pi.



Langages Built-in, C++, C#, Java.

### 6.7.16.14 pow

Retourne la valeur de **a** élevé à la puissance de **b**.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
а	xs:double	Fournit valeur <b>a</b> (la base).
b	xs:double	Fournit valeur <b>b</b> (la puissance).

# 6.7.16.15 radians

Convertit un angle mesuré en degrés à un angle équivalent approximatif mesuré en radians.

$f_{\rm s}$ rad	$f_{\rm x}$ radians	
l¢ value	result 🗘	

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

## 6.7.16.16 random

Retourne une valeur avec un sign positif, supérieur ou égal à 0.0 et inférieur à 1.0. Les valeurs retournées sont choisies pseudo-aléatoirement avec une distribution uniforme (environ) de cette plage.
∫<sub>8</sub> random result ₽

# Langages

Built-in, C++, C#, Java.

## 6.7.16.17 sin

Retourne le sinus de value.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

# 6.7.16.18 sqrt

Retourne la racine carrée positive arrondie correctement de value.



### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

## 6.7.16.19 tan

Retourne la tangente de value.



### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
value	xs:double	La valeur d'entrée.

### 6.7.16.20 unary-minus

Retourne la négation de la valeur d'entrée signée.

∫<mark>x unary-minus</mark> ⊳value result ⊳

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
value	xs:decimal	La valeur d'entrée.

### Exemple

Si la valeur d'entrée est 3, la fonction retourne -3. Si la valeur d'entrée est -3, la fonction retourne 3.

# 6.7.17 lang | QName functions

Les fonctions QName de la bibliothèque **lang** convertissent les valeurs Qualified Name (QName) en strings, et vice-versa. Contrairement aux fonctions provenant de la bibliothèque **core**, ces fonctions sont disponibles uniquement dans les langages Built-in, Java, C# ou C++.

# 6.7.17.1 QName-as-string

Retourne la représentation de string de la valeur Qname fournie en tant qu'argument.

f <sub>8</sub> QName-as-string	
QName	result Þ

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Description
QName	La valeur d'entrée de xs:QName.

# 6.7.17.2 string-as-QName

Reconvertit la représentation de string d'un QName en un QName.

f <sub>8</sub> string-as-QName	
string result	

### Langages

Built-in, C++, C#, Java.

Nom	Description
Paramètres	La valeur de string d'entrée.

# 6.7.18 lang | string functions

Les fonctions string provenant de la bibliothèque **lang** vous permettent de traiter des strings (par exemple, trim, pad, replace, de convertir des strings en casse majuscule ou minuscule, etc.).

## 6.7.18.1 capitalize

Retourne le string d'entrée value, où la première lettre de chaque mot est une majuscule.



### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
value	xs:string	La valeur d'entrée.

### Exemple

Si la valeur d'entrée est the quick brown fox, la fonction retourne The Quick Brown Fox.

# 6.7.18.2 charset-decode

La fonction charset-decode prend en tant qu'entrée des données binaires encodées en tant que texte Base64. Elle décode des données conformément à l'ensemble de caractères spécifié (par exemple, "utf-8") et retourne la valeur de string résultante. Si vous souhaitez encoder des données binaires en tant que texte Base64, utilsier la fonction charset-encode<sup>762</sup>.

$f_{\rm s}$ charset-decode	
binary	
encoding	string C
error-abort	

### Langages

Built-in.

Nom	Туре	echo Hello, World!
binary-data	xs:base64Binary	Les données binaires en tant que texte Base64.

Nom	Туре	echo Hello, World!
encoding	xs:string	L'ensemble de caractère utilisé pour l'encodage (par exemple, "utf- 8").
error-abort	xs:boolean	<ul> <li>Argument optionnel qui spécifie comment le traitement doit continuer en cas d'erreurs. Valeurs valides :</li> <li>true - Terminer le traitement avec une exception sur le caractère invalide</li> <li>false - Continuer le traitement et remplacer des caractères valides avec le caractère de remplacement .</li> </ul>
		La valeur par défaut est true.

### Exemple

Supposons que vous souhaitez décoder des données binaires provenant du fichier XML source suivant. Veuillez noter que l'élément de **message** contient des données binaires encodées en tant que texte Base64.

```
<?xml version="1.0" encoding="UTF-8"?>
<message xsi:noNamespaceSchemaLocation="message.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">TG9yZW0gaXBzdW0=</message>
```

Le type de données de l'élément message est xs:base64Binary, comme illustré par le schéma :



Un mappage qui décode le message ci-dessus comme suit :



Le mappage dans cet exemple produit le texte "Lorem ipsum".

Un mappage peut aussi traiter du texte ou des fichiers XML encodés en tant que données Base64, avec l'aide d'un composant de sérialisation MapForce. Par exemple, le mappage illustré ci-dessous a un paramètre **input** qui nécessite des données de texte Base64. En partant du principe que les données Base64 ont été créées depuis un fichier XML comme indiqué dans l'exemple <u>charset-encode</u><sup>(762)</sup>, vous pouvez recréer le fichier XML original comme indiqué dans le mappage ci-dessous :



Dans ce mappage, l'argument **error-abort** obtient une valeur **false**, qui a été produite avec l'aide de la fonction intégrée **boolean**. Cela garantit que le traitement se poursuivra même si des caractères invalides surviennent. Le résultat de string de la fonction est ensuite transmis à un composant de parsage XML qui le convertit en un fichier XML. Veuillez noter que, pour que le parsage XML soit possible, vous devez avoir le fichier de schéma XSD. Pour plus d'informations, voir <u>Parser et sérialiser des strings</u><sup>®4</sup>.

# 6.7.18.3 charset-encode

La fonction charset-encode prend en tant que string d'entrée des données et l'encode en tant que texte Base64. Les données sont encodées dans l'ensemble de caractère spécifique (par exemple, "utf-8") et retournées en tant que type xs:base64Binary. Si vous souhaitez décoder des données binaires encodées précédemment en tant que texte Base64, utiliser la fonction <u>charset-decode</u><sup>760</sup>.

$f_{\!\!8}$ charset-encode	
⊳ string	
encoding	binary C
substitute	

### Langages

Built-in.

Nom	Туре	echo Hello, World!
string-data	xs:string	Les données de string à encoder.

Nom	Туре	echo Hello, World!
encoding	xs:string	L'ensemble de caractère utilisé pour l'encodage (par exemple, "utf- 8").
substitute	xs:string	Argument optionnel qui spécifie un caractère de remplacement en cas d'émergence de caractères invalides. Cet argument est applicable si vous utilisez un encodage non-Unicode. Pour des encodages Unicode, le caractère de remplacement est .

### Exemple

Imaginons que vous souhaitez encoder le texte "Lorem ipsum" en tant que données Base64, en utilisant le set de caractère UTF-8 et que vous souhaitez l'écrire dans un fichier XML cible. Celui-ci a un élément **message** de xs:base64Binary comme illustré par le schéma :

Un mappage qui effectue l'encodage Base64 ressemble à l'exemple suivant :



Ce mappage produit un résultat XML comme celui indiqué ci-dessous (les références de schéma et la déclaration XML ont été sautées) :

```
<message>TG9yZW0gaXBzdW0=</message>
```

Vous pouvez aussi encoder du texte ou des fichiers XML en tant que Base64, avec l'aide d'un composant de sérialisation MapForce. Par exemple, le mappage illustré ci-dessous sérialise un fichier XML de source dans un string. Le string résultant est ensuite fourni en tant qu'argument dans la fonction charset-encode. Enfin, le résultat de fonction est retourné en tant que sortie de mappage avec l'aide d'un composant de sortie simple, voir <u>Retourner les valeurs String depuis un mappage</u><sup>488</sup>. Pour plus d'informations concernant la sérialisation, voir <u>Parser et sérialiser des strings</u><sup>604</sup>.



# 6.7.18.4 count-substring

Retourne une valeur d'entier exprimant le nombre de fois que substr se produit dans string.

	$f_8$ count-	substring
Ç	> string	
Ç	>substr	result

### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре
substr	xs:string	Le sous-string à tester.

### Exemple

Le mappage suivant retourne **2**. Il s'agit du nombre de fois que la barre verticale se produit dans le cadre du string d'entrée **id name email**.



# 6.7.18.5 empty

Retourne true si la valeur de string d'entrée est vide ; false sinon.



### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:string	La valeur d'entrée.

# 6.7.18.6 find-substring

Retourne la position de la première occurrence de **substr** dans **string**. Par défaut, la fonction lance la recherche depuis le premier caractère, qui a la position (index) 1, mais vous pouvez spécifier en option un index de démarrage spécifique. Si **substr** ne peut pas être trouvé, la fonction retourne **0**.

$f_{s}$ find-substring	
string	
⊳ substr	result 🕻
startindex	

### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре
substr	xs:string	Le sous-string à chercher.
startindex	xs:int	Optionnel Spécifie la position de début (index) de la recherche. Si ce paramètre n'est pas spécifié, la recherche commence à la position 1.

### Exemple

Le mappage suivant sort **3**, qui est la position de la première occurrence du caractère vertical dans le string d'entrée **id** | name | email.



Si vous spécifiez **4** en tant qu'index de démarrage, la fonction commencera la recherche à partir du quatrième caractère. Par conséquent, le mappage ci-dessous produira **8**, qui est la première occurrence du caractère vertical après la recherche à partir du quatrième caractère.



# 6.7.18.7 format-guid-string

Retourne un string d'identifiant unique global (GUID) correctement formaté, typique pour l'utilisation dans des champs de base de données. Voir aussi la fonction <u>create-guid</u><sup>747</sup>.

```
∫<sub>∗</sub> format-guid-string
▷ unformatted_guid |formatted_guid ▷
```

### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
unformatted_guid	xs:string	Le string d'entrée encodé en HEX à formater.

# 6.7.18.8 left

Retourne un string contenant les premiers caractères number du string d'entrée.

	$\mathbf{f}_{\mathbf{z}}$	left		
ç	> sti	ring	racult C	
Ç	>nu	umber	result	

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре
number	xs:int	Spécifie combien de caractères doivent être retournés, en commençant avec le début du string.

### Exemple

Si le string d'entrée est This is a sentence et que le nombre est 4, la fonction retourne This.

# 6.7.18.9 left-trim

La fonction left-trim (voir la capture d'écran ci-dessous) supprime l'espace blanc à gauche d'un string. Les espaces blancs incluent un espace (U+0020), un onglet (U+0009), le retour à la ligne (U+000D) et la saut de ligne (U+000A). Pour des détails sur les espaces blancs, voir la <u>Recommandation XML</u>.



#### À propos des espaces insécables

Les fonctions left-trim, right-trim et normalize-space ne suppriment pas les espaces insécables. L'une des solutions possibles pourrait être de remplacer l'espace insécable, dont la représentation est 160, avec l'espace, dont la représentation décimale est 32. Le mappage ci-dessous montre qu'une fois que l'espace insécable a été remplacé, la valeur découpée SomeValue sera mappée vers la cible.



Si votre composant source est un fichier Excel, vous pouvez supprimer les espaces supplémentaires dans Excel utilisant une combinaison de fonctions TRIM, CLEAN et SUBSTITUTE. Pour les détails, voir <u>Supprimer</u> les espaces et caractères non imprimables du texte.

#### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	Description
string	xs:string	Le string d'entrée.

## 6.7.18.10 lowercase

Convertit le **string** d'entrée en casse minuscule. Pour des caractère Unicode, les caractères en casse minuscule correspondants (définis par le groupe Unicode) sont utilisés.

f<sub>x</sub> lowercase String result P

Langages Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре

# 6.7.18.11 match-pattern

Retourne booléenne **true** si le string d'entrée correspond à l'expression régulière définie par **pattern** ; **false** sinon. Voir aussi <u>Expressions régulières</u><sup>627</sup>.

**Note :** Lors de la génération de code C++, C# ou Java, les fonctions avancées de la syntaxe d'expression régulière peuvent différer légèrement. Voir la documentation regex de chaque langage pour plus d'informations.

	$f_{s}$ match-pattern	
string		
ç	>pattern	result

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре
pattern	xs:string	L'expression régulière à faire correspondre.

### Exemple

Le mappage suivant valide plusieurs titres de personnes. En particulier, le mappage produira **true** pour tous les titres suivants : Mr, Mrs, Mx, Ms, Miss.



SI le string d'entrée est différent de tous les titres listés ci-dessus, le mappage produira false.

# 6.7.18.12 pad-string-left

Retourne le string qui est complété à sa gauche par un seul caractère spécifique, jusqu'à une longueur requise. La longueur de string désirée et le caractère de remplissage sont fournis en tant qu'arguments.

	f <sub>8</sub> pad-string-left	
Ç	> string	
Ç	>desired-length	result 🖡
Ç	>padding-char	

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Spécifie le string d'entrée.
desired-length	xs:int	Définit la longueur désirée du string après le remplissage.
padding-char	xs:string	Définit le caractère à utiliser en tant que caractère de remplissage.

# 6.7.18.13 pad-string-right

Retourne le string qui est complété à sa droite par un seul caractère spécifique, jusqu'à une longueur requise. La longueur de string désirée et le caractère de remplissage sont fournis en tant qu'arguments.

	f <sub>x</sub> pad-string-right		
0	> string		
C	desired-length	result 🕻	>
0	>padding-char		

### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Spécifie le string d'entrée.
desired-length	xs:int	Définit la longueur désirée du string après le remplissage.

Nom	Туре	echo Hello, World!
padding-char	xs:string	Définit le caractère à utiliser en tant que caractère de remplissage.

# 6.7.18.14 repeat-string

Répète le string fourni en tant qu'argument *n* fois. L'argumet **count** spécifie le nombre de fois qu'il faut répéter le string.

$f_{s}$ repeat-string		
string		
count	result	

### Langages

Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре
count	xs:int	Le nombre de fois qu'il faut répéter le string.

# 6.7.18.15 replace

Le résultat est un nouveau string où chaque instance de **oldstring**, dans le string d'entrée **value**, est remplacé par **newstring**.

$f_{\!\scriptscriptstyle 8}$ replace	
⊳value	
oldstring	result 🕻
newstring	

Langages Built-in, C++, C#, Java.

### Paramètres

Nom	Туре	echo Hello, World!
value	xs:string	La valeur d'entrée.
oldstring	xs:string	L'ancien string à remplacer.
newstring	xs:string	Le nouveau string pour utiliser en tant que remplacement.

#### Exemple

Voir <u>Remplacer les caractères spéciaux</u><sup>278</sup>.

# 6.7.18.16 reversefind-substring

Retourne la position de la dernière occurrence de **substr** dans **string**. Par défaut, la fonction lance la recherche depuis le premier caractère, dont la position est (index) 1, et termine la recherche au niveau du dernier caractère, mais vous pouvez spécifier en option un index de fin. Si **substr** ne peut pas être trouvé, la fonction retourne **0**.

$f_8$ reversefind-substring	
string	
substr	result 🗘
endindex	

### Langages

Built-in, C++, C#, Java.

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре
substr	xs:string	Le sous-string à chercher.
endindex	xs:int	Optionnel Spécifie la position de fin (index) de la recherche. Si ce paramètre n'est pas spécifié, la recherche se termine après le dernier caractère dans <b>string</b> .

### Exemple

Le mappage suivant sort **8**, qui est la position de la dernière occurrence du caractère vertical dans le string d'entrée **id** | name | email.



Si vous spécifiez **4** en tant qu'index de fin, la fonction recherchera jusqu'au quatrième caractère. Par conséquent, le mappage ci-dessous produira **3**.



# 6.7.18.17 right

Retourne un string contenant le dernier caractÈre **number** du string d'entrée.



Langages Built-in, C++, C#, Java.

### Exemple

Si le string d'entrée est The brown red fox et le nombre est 3, la fonction retourne fox.

# 6.7.18.18 right-trim

La fonction **right-trim** (*voir la capture d'écran ci-dessous*) supprime l'espace blanc à droite d'un string. Les espaces blancs incluent un espace (U+0020), un onglet (U+0009), le retour à la ligne (U+000D) et la saut de ligne (U+000A). Pour des détails sur les espaces blancs, voir la <u>Recommandation XML</u>.



#### À propos des espaces insécables

Les fonctions left-trim, right-trim et normalize-space ne suppriment pas les espaces insécables. L'une des solutions possibles pourrait être de remplacer l'espace insécable, dont la représentation est 160, avec l'espace, dont la représentation décimale est 32. Le mappage ci-dessous montre qu'une fois que l'espace insécable a été remplacé, la valeur découpée SomeValue sera mappée vers la cible.



Si votre composant source est un fichier Excel, vous pouvez supprimer les espaces supplémentaires dans Excel utilisant une combinaison de fonctions TRIM, CLEAN et SUBSTITUTE. Pour les détails, voir <u>Supprimer</u> les espaces et caractères non imprimables du texte.

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	Description
string	xs:string	Le string d'entrée.

## 6.7.18.19 string-compare

La fonction string-compare (voir la capture d'écran ci-dessous) retourne le résultat d'un caractère en comparant les caractères de deux strings d'entrée : string1 et string2. La comparaison est basée sur des codes ASCII. Les deux string1 et string2 sont de type xs:string. Le fonction est sensible à la casse. Si les strings sont égaux, le résultat est 0. Si string1 est inférieur à string2, le résultat sera -1. Si string1 est supérieur à string2, le résultat sera 1.

$f_{\rm x}$ string-compare	
⊳string1	rocult N
⊳string2	result

#### Fonctions

#### Exemple :

string1 : hi string2 : Hit

La fonction string-compare compare les strings caractère par caractère. La comparaison se termine une fois que la fonction a détecté que le premier caractère de string1 et le premier caractère de string2 sont différents. Le résultat est basé sur la comparaison du premier caractère de chaque string. Puisque h est représenté comme numéro de code ASCII supérieur (104 selon le système décimal) à H (72 dans le système décimal), string1 est supérieur au string2, et le résultat de la comparaison de string est 1. Si le premier caractère du string1 et le premier caractère du string2 étaient les mêmes, la fonction poursuivrait l'analyse du second caractère et ainsi de suite.

Pour une comparaison de string simple avec un résultat booléen, voir <u>core | logical functions |</u> equal<sup>663</sup>.

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	Description
stringl	xs:string	Le premier string d'entrée.
string2	xs:string	Le second string d'entrée.

# 6.7.18.20 string-compare-ignore-case

La fonction string-compare-ignore-case (voir la capture d'écran ci-dessous) retourne le résultat d'un caractère en comparant les caractères de deux strings d'entrée : string1 et string2. Les deux string1 et string2 sont de type xs:string. La focntion ignore la casse. La comparaison est basée sur des codes ASCII. Si les strings sont égaux, le résultat est 0. Si string1 est inférieur à string2, le résultat sera -1. Si string1 est supérieur à string2, le résultat sera 1.

f <sub>x</sub> string-compare-ignore-case	
⊳string1	
⊳string2	result

#### Exemple :

string1 : hi string2 : Hit

La fonction string-compare-ignore-case compare les strings caractère par caractère. Même si h est représenté en tant que numéro de code ASCII supérieur à H, ces deux caractères sont traités en égal dans cette fonction. Le deuxième caractère dans les deux strings est le même. Toutefois, le string2 a un troisième caractère, alors que le string1 n'en a pas. Le troisième caractère dans le string1 a une valeur vide. La valeur

#### 776 Fonctions

t dans le string2 est supérieure à la valeur vide dans le string1. C'est la raison pour laquelle le string1 est inférieur au string2, et le résultat est égale à -1.

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	Description
stringl	xs:string	Le premier string d'entrée.
string2	xs:string	Le second string d'entrée.

# 6.7.18.21 uppercase

Convertit le **string** d'entrée en majuscule. En ce qui concerne les caractères Unicode, les caractères de majuscule correspondants sont utilisés (tels que définis par le groupe Unicode).

	$f_{\rm s}$ upp	ercase	
ç	> string	result 🕻	þ

#### Langages

Built-in, C++, C#, Java.

#### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Туре

# 6.7.19 xpath2 | accessors

Les fonctions provenant de la sous-bibliothèque **xpath2** | **accessors** extraient des informations concernant des nœuds ou des items XML. Ces fonctions sont disponibles lorsque les langages XSLT2 ou XQuery sont sélectionnés.

## 6.7.19.1 base-uri

La fonction **base-uri** prend un nœud en tant qu'entrée et retourne l'URI de la ressource XML contenant le nœud. La sortie est de type xs:string.

∫<sub>x</sub> node-name ⊳node result ⊳

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
node	mf:node	Le nœud d'entrée.

## 6.7.19.2 node-name

La fonction node-name prend un nœud en tant que son argument d'entrée et retourne son QName. Lorsque le QName est représenté en tant que string, il prend la forme de prefix:localname si le nœud prend un préfixe, ou localname si le nœud n'a pas de préfixe. Pour obtenir l'URI d'espace de noms d'un nœud, utiliser la fonction namespace-uri-from-QName

f <sub>8</sub> nod	e-name
<mark>⊳nod</mark> e	result 🖓

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
node	mf:node	Le nœud d'entrée.

# 6.7.19.3 string

La fonction **string** fonctionne comme le constructeur xs:string : elle convertit son argument en xs:string.

Lorsque l'argument d'entrée est une valeur d'un type atomique (par exemple xs:decimal), cette valeur atomique est convertie en une valeur de type xs:string. Si l'argument d'entrée est un nœud, la valeur string du nœud est extraite. (La valeur de string d'un nœud est une concaténation des valeurs des nœuds descendants du nœud.)



#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
item	mf:item	La valeur d'entrée.

# 6.7.20 xpath2 | anyURI functions

La sous-librairie **xpath2** | **anyURI** contient la fonction **resolve-uri**. Cette fonction est disponible lorsque les langages XSLT2 ou XQuery sont sélectionnés.

## 6.7.20.1 resolve-uri

La fonction **resolve-uri** prend un URI en tant que son premier argument et le résout par rapport à l'URI dans le second argument. Le résultat est de type de données xs:string. La mise en place de la fonction traite les deux entrées en tant que strings ; aucun contrôle n'est effectué pour voir si les ressources identifiées par ces URIs existent réellement.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
relative	xs:string	L'URI relatif à résoudre par rapport à la base.
base	xs:string	L'URI de base.

### Exemple

Dans le mappage illustré ci-dessous, le premier argument fournit l'URI relative MyFile.html, et le second argument fournit l'URI de base file:///C:/Dir/. L'URI résolu sera une concaténation des deux, donc file:///C:/Dir/MyFile.html.



# 6.7.21 xpath2 | boolean functions

Les fonctions booléennes **true** et**false** ne prennent pas d'argument et retournent les valeurs de constante booléennes, **true** et **false**, respectivement. Elles peuvent être utilisées là où une valeur booléennen constante est requise.

## 6.7.21.1 false

Retourne la valeur booléenne false.



Langages XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.21.2 true

Retourne la valeur booléenne true.



### Langages

XQuery, XSLT 2.0, XSLT 3.0.

# 6.7.22 xpath2 | constructors

Les fonctions dans la partie "constructors" de la sous-bibliothèque de la bibliothèque XPath 2.0 construisent des types de données spécifiques depuis le texte d'entrée. La table suivante liste les fonctions de constructor disponibles.

xs:ENTITY	xs:double	xs:nonPositiveInteger
xs:ID	xs:duration	xs:normalizedString
xs:IDREF	xs:float	xs:positiveInteger
xs:NCName	xs:gDay	xs:short
xs:NMTOKEN	xs:gMonth	xs:string
xs:Name	xs:gMonthDay	xs:time
xs:QName	xs:gYear	xs:token
xs:anyURI	xs:gYearMonth	xs:unsignedByte
xs:base64Binary	xs:hexBinary	xs:unsignedInt
xs:boolean	xs:int	xs:unsignedLong
xs:byte	xs:integer	xs:unsignedShort
xs:date	xs:language	xs:untypedAtomic
xs:dateTime	xs:long	xs:yearMonthDuration
xs:dayTimeDuration	xs:negativeInteger	
xs:decimal	xs:nonNegativeInteger	

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Exemple

Généralement, le format lexical du texte d'entrée doit être celui prévu par le type de données à construire. Sinon, la transformation ne sera pas réussie Par exemple, si vous souhaitez construire une valeur xs:dateTime en utilisant la fonction de constructor xs:dateTime, le texte d'entrée doit avoir le format lexical du type de données xs:dateTime, qui est YYYY-MM-DDTHH:mm:ss.



Dans le mappage illustré ci-dessus, une constante de string ("2020-04-28T00:00:00") a été utilisée pour fournit l'argument d'entrée de la fonction. L'entrée peut aussi avoir été obtenue depuis un item dans le document source. La fonction xs:dateTime retourne la valeur 2020-04-28T00:00:00 de type xs:dateTime.

Pour consulter le type de données attendu d'un item de mappage (y compris le type de données des arguments de fonction), déplacer le curseur de la souris sur le connecteur d'entrée ou de sortie respectif.

# 6.7.23 xpath2 | context functions

Les fonctions context provenant de la bibliothèque **xpath2** fournissent des information diverses à propos de la date et de l'heure actuels, la collation par défaut utilisée par le processeur, la taille de la séquence actuelle et la position du nœud actuel.

# 6.7.23.1 current-date

Retourne la date actuelle (xs:date) depuis l'horloge du système.

f<sub>x</sub> current-date result D

Langages XQuery, XSLT 2.0, XSLT 3.0.

# 6.7.23.2 current-dateTime

Retourne la date et l'heure actuels (xs:dateTime) depuis l'horloge du système.

f<sub>x</sub> current-dateTime result D

Langages XQuery, XSLT 2.0, XSLT 3.0.

### 6.7.23.3 current-time

Retourne l'heure actuelle (xs:time) depuis l'horloge du système.



Langages XQuery, XSLT 2.0, XSLT 3.0.

# 6.7.23.4 default-collation

La fonction **default-collation** ne prend pas d'argument et retourne la collation par défaut, c'est à dire la collation qui est utilisée lorsqu'aucune collation n'est spécifiée pour une fonction où une peut être spécifiée.

Des comparaisons, y compris pour les fonctions max-string et min-string sont basées sur cette collation.



Langages XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.23.5 implicit-timezone

Retourne la valeur de la propriété "implicit timezone" depuis le contexte d'évaluation.



Langages XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.23.6 last

Retourne la nombre d'items dans le cadre de la séquence des items actuellement en cours de traitement. Chose importante, la séquence des items est déterminée par le <u>contexte de mappage</u><sup>374</sup> actuel, tel que décrit dans l'exemple ci-dessous.



### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Exemple

Supposons que vous disposez du fichier XML de source suivant :

```
<Articles>

<Articles
<pre>
<Article>

<Article>

</Article>

<Article>

<Article>

</Article>

</Article>

</Article>

</Article>

</Article>

</Article>

</Article>

</Article>

</Article>
```

Votre objectif est de copier des données dans un fichier XML avec un schéma différent. De plus, le décompte de tous les items doit être enregistré dans le fichier XML de cible. Cela peut être obtenu par un mappage comme celui ci-dessous :



Dans l'exemple ci-dessus, la fonction **last** retourne la position du dernier nœud dans le contexte de parent actuel et remplit l'attribut **count** avec la valeur **3**.



<item>Jacket</item>
</items>

Veuillez noter que la valeur **3** est la position du dernier item (et donc le décompte de tous les items) dans le contexte de mappage créé par la connexion entre **Article** et **items**. Si cette connexion n'existait pas, les items seraient toujours copiés vers la cible, mais la fonction **last** retournerait la valeur **1** de manière incorrecte, parce qu'il n'y aurait pas de <u>parent context</u><sup>(BO)</sup> à itérer. (Plus précisément, elle utiliserait le contexte implicité par défaut créé entre les items racine des deux composants, ce qui produit une séquence de 1 item, et pas de 3 comme attendu).

Il est généralement conseillé d'utiliser la fonction <u>count</u> <sup>665</sup> depuis la bibliothèque **core** au lieu de la fonction **last**, parce que cette première a un argument **parent-context**, ce qui vous permet de modifier le contexte de mappage explicitement.

# 6.7.24 xpath2 | durations, date and time functions

Les fonctions date and time de la bibliothèque **xpath2** vous permettent d'ajuster le fuseau horaire dans valeurs de dates et d'heures, d'extraire des composants particuliers depuis les valeurs date, time et duration, et de soustraire des valeurs date and time.

### Ajuster le fuseau horaire

Pour ajuster le fuseau horaire dans des valeurs date and time, les fonctions suivantes sont disponibles :

- adjust-date-to-timezone
- adjust-date-to-timezone (avec argument timezone)
- adjust-dateTime-to-timezone
- adjust-dateTime-to-timezone (avec argument timezone)
- adjust-time-to-timezone
- **adjust-time-to-timezone** (avec argument timezone)

Chacune de ces fonctions liées prennent une valeur xs:date, xs:time ou xs:dateTime en tant que le premier argument et ajuste l'entrée en ajoutant, supprimant ou en modifiant le composant de fuseau horaire selon la valeur du second argument (le cas échéant).

Les situations suivantes sont possibles lorsque le premier argument ne continent aucun fuseau horaire (par exemple, la date 2020-01 ou l'heure 14:00:00).

- Si l'argument **timezone** est présent, le résultat contiendra le fuseau horaire spécifié dans le second argument. Le fuseau horaire dans le second argument est ajouté.
- Si l'argument **timezone** est absent, le résultat contiendra le fuseau horaire implicite qui est le fuseau horaire du système. Le fuseau horaire du système est ajouté.
- Si l'argument timezone est vide, le résultat ne contiendra aucun fuseau horaire.

Les situations suivantes sont possibles lorsque le premier argument contient aucun fuseau horaire (par exemple, la date 2020-01-01+01:00 ou l'heure 14:00:00+01:00 ).

• Si l'argument **timezone** est présent, le résultat contiendra le fuseau horaire spécifié dans le second argument. Le fuseau horaire original est remplacé par le fuseau horaire dans le second argument.

- Si l'argument **timezone** est absent, le résultat contiendra le fuseau horaire implicite qui est le fuseau horaire du système. Le fuseau horaire original est remplacé par le fuseau horaire du système.
- Si l'argument timezone est vide, le résultat ne contiendra aucun fuseau horaire.

#### Extraire des composants de dates et d'heures

Pour extraire des valeurs numériques comme des heures, des minutes, des jours, des mois, etc. depuis des valeurs de date et d'heure, les fonctions suivantes sont disponibles :

- day-from-date
- day-from-dateTime
- hours-from-dateTime
- hours-from-time
- minutes-from-dateTime
- minutes-from-time
- month-from-date
- month-from-dateTime
- seconds-from-dateTime
- seconds-from-time
- timezone-from-date
- timezone-from-dateTime
- timezone-from-time
- year-from-date
- year-from-dateTime

Chacune de ces fonctions extrait un composant particulier des valeurs xs:date, xs:time, xs:dateTime et xs:duration. Le résultat sera soit xs:integer soit xs:decimal.

### Extraire des composants de durations

Pour extraire des composants time depuis des durées, les fonctions suivantes sont disponibles :

- days-from-duration
- hours-from-duration
- minutes-from-duration
- months-from-duration
- seconds-from-duration
- years-from-duration

La durée doit être spécifiée soit en tant que xs:yearMonthDuration (pour extraire les années et les mois) ou xs:dayTimeDuration (fpour extraire des jours, des heures, des minutes et des secondes). Toutes les fonctions retournent un résultat de type xs:integer, avec l'exception de la fonction seconds-from-duration, qui retourne xs:decimal.

#### Soustraire des valeurs de date et d'heure

Pour soustraire des valeurs de date et d'heure, les fonctions suivantes sont disponibles :

- subtract-dateTimes
- subtract-dates
- subtract-times

Chacune des fonctions de soustraction vous permet de soustraire une valeur d'heure d'une autre et de retourner une valeur de durée

## 6.7.24.1 adjust-date-to-timezone

Ajuste une valeur xs:date dans le fuseau horaire implicite dans le contexte d'évaluation (le fuseau horaire du système).

∫<sub>8</sub> adjust-date-to-timezone ▷ date result ▷

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
date	xs:date	La valeur d'entrée de type xs:date.

### Exemple

Le mappage suivant construit un xs:date depuis un string et le fournit en tant qu'argument à la fonction adjust-date-to-timezone.



Mappage XSLT 2.0

Si le mappage est exécuté sur un ordinateur situé dans un fuseau horaire de +02:00, la fonction adjuste la valeur de date pour inclure le fuseau horaire du système. Par conséquent, la sortie de mappage est 2020-04-30+02:00.

## 6.7.24.2 adjust-date-to-timezone

Ajuste une valeur xs:date dans un fuseau horaire spécifique ou aucun fuseau horaire. Si l'argument **timezone** est une séquence vide, la fonction retourne un xs:date sans fuseau horaire. Sinon, il retourne un xs:date avec un fuseau horaire.

$f_{\rm x}$ adjust-date-to-timezone	
date	
timezone	result

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
date	xs:date	La valeur d'entrée de type xs:date.
timezone	xs:dayTimeDuration	Le fuseau horaire exprimé en tant que valeur xs:dayTimeDuration. La valeur peut être négative. Par exemple, une valeur de fuseau horaire de -5 heures peut être exprimée en tant que <b>_PT5H</b> .

### Exemple

Le mappage suivant construit les deux paramètres dans la fonction <u>adjust-date-to-timezone</u> depuis des strings, en utilisant les fonction XPath 2 <u>constructor</u> correspondantes. L'objectif du mappage est d'ajuster le fuseau horaire à -5 heures. Ce fuseau horaire peut être exprimé en tant que <u>-PT5H</u>.



Mappage XSLT 2.0

La fonction ajuste la valeur de date au fuseau horaire fourni en tant qu'argument. Par conséquent, la sortie de mappage est 2020-04-30-05:00.

# 6.7.24.3 adjust-dateTime-to-timezone

Ajuste une valeur xs:dateTime dans le fuseau horaire implicite dans le contexte d'évaluation (le fuseau horaire du système).

	J <sub>8</sub> adjust-dateTime-to-timezone		
dateTime		ateTime	result 🗘

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type xs:dateTime.

## 6.7.24.4 adjust-dateTime-to-timezone

Ajuste une valeur xs:dateTime dans un fuseau horaire spécifique ou aucun fuseau horaire. Si l'argument **timezone** est une séquence vide, la fonction retourne un xs:dateTime sans fuseau horaire. Sinon, il retourne un xs:dateTime avec un fuseau horaire.

f <sub>8</sub> adjust-dateTim	e-to-timezone
dateTime	
timezone	result

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type xs:dateTime.
timezone	xs:dayTimeDuration	Le fuseau horaire exprimé en tant que valeur xs:dayTimeDuration. La valeur peut être négative. Par exemple, une valeur de fuseau horaire de -5 heures peut être exprimée en tant que <b>-PT5H</b> .

## 6.7.24.5 adjust-time-to-timezone

Ajuste une valeur xs:time dans le fuseau horaire implicite dans le contexte d'évaluation (le fuseau horaire du système).

f <sub>*</sub> adjust-time-to-timezone	
¢ time	result 🗘

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
time	xs:time	La valeur d'entrée de type xs:time.

## 6.7.24.6 adjust-time-to-timezone

Ajuste une valeur xs:time dans un fuseau horaire spécifique ou aucun fuseau horaire. Si l'argument **timezone** est une séquence vide, la fonction retourne un xs:time sans fuseau horaire. Sinon, il retourne un xs:time avec un fuseau horaire.

$f_{\rm R}$ adjust-time-to-timezone	
¢ time	
timezone	result

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
time	xs:time	La valeur d'entrée de type xs:time.
timezone	xs:dayTimeDuration	Le fuseau horaire exprimé en tant que valeur xs:dayTimeDuration. La valeur peut être négative. Par exemple, une valeur de fuseau

Nom	Туре	Description
		horaire de -5 heures peut être exprimée en tant que -рт5н.

## 6.7.24.7 day-from-date

Retourne un xs:integer représentant la partie day de la valeur xs:date fournie en tant qu'argument.



#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
date	xs:date	La valeur d'entrée de type xs:date.

#### Exemple

Le mappage suivant convertit un string en xs:date en utilisant la fonction constructor xs:date. Les fonctions day-from-date, month-from-date et year-from-date extraient chacune la partie respective de la date et l'écrivent dans un item séparé dans le fichier XML cible.



Mappage XQuery 1.0

Le sortie de mappage est la suivante :



```
<col2>4</col2>
<col3>2020</col3>
</row>
</rows>
```

# 6.7.24.8 day-from-dateTime

Retourne un xs:integer représentant la partie day de la valeur xs:dateTime fournie en tant qu'argument.

∫<mark>x day-from-dateTime</mark> ▷ dateTime result ▷

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type

## 6.7.24.9 days-from-duration

Retourne un xs:integer représentant le composant "days" de la représentation canonique de la valeur de durée fournie en tant qu'argument.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
duration	xs:duration	La valeur d'entrée de type xs:duration.

### Exemple

Le mappage illustré ci-dessous construit le xs:dayTimeDuration de **P2DT1H** (2 jours et 1 heure) et le fournit en tant qu'entrée de la fonction days-from-duration. Le résultat est **2**.



Mappage XSLT 2.0

Note : si la durée est P1DT24H (1 jour et 24 heures), la fonction retourne 2, pas 1. Cela est dû au fait que la représentation canonique de P1DT24H est en réalité P2D (2 jours).

# 6.7.24.10 hours-from-dateTime

Retourne un xs: integer représentant la partie hours de la valeur xs: dateTime fournie en tant qu'argument.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type

## 6.7.24.11 hours-from-duration

Retourne un xs:integer représentant le composant "hours" de la représentation canonique de la valeur de durée fournie en tant qu'argument.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
duration	xs:duration	La valeur d'entrée de type xs:duration.

### Exemple

Si la durée est **PT1H60M** (1 heure et 60 minutes), la fonction retourne **2**, pas **1**. Cela est dû au fait que la représentation canonique de **PT1H60M** est en réalité **PT2H** (2 heures).
# 6.7.24.12 hours-from-time

Retourne un xs:integer représentant la partie hours de la valeur xs:time fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
time	xs:time	La valeur d'entrée de type xs:time.

# 6.7.24.13 minutes-from-dateTime

Retourne un xs:integer représentant la partie minutes de xs:dateTime fourni en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type xs:dateTime.

# 6.7.24.14 minutes-from-duration

Retourne un xs:integer représentant le composant "minutes" de la représentation canonique de la durée fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
duration	xs:duration	La valeur d'entrée de type xs:duration.

### Exemple

Si la durée est **PT1M60s** (1 minute et 60 secondes), la fonction retourne **2**, pas **1**. Cela est dû au fait que la représentation canonique de **PT1M60s** est en réalité **PT2M** (2 minutes).

## 6.7.24.15 minutes-from-time

Retourne un xs:integer représentant la partie minutes de la valeur xs:time fournie en tant qu'argument.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
time	xs:time	La valeur d'entrée de type xs:time.

## 6.7.24.16 month-from-date

Retourne un xs:integer représentant la partie month de la valeur xs:date fournie en tant qu'argument.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
date	xs:date	La valeur d'entrée de type xs:date.

# 6.7.24.17 month-from-dateTime

Retourne un xs:integer représentant la partie month de la valeur xs:dateTime fournie en tant qu'argument.

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type xs:dateTime.

# 6.7.24.18 months-from-duration

Retourne un xs:integer représentant le composant "months" de la représentation canonique de la valeur de durée fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
duration	xs:duration	La valeur d'entrée de type xs:duration.

## 6.7.24.19 seconds-from-dateTime

Retourne un xs:integer représentant le composant seconds dans la valeur localisée de dateTime.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	

# 6.7.24.20 seconds-from-duration

Retourne un xs:integer représentant le composant "seconds" de la représentation canonique de la valeur de durée fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
duration	xs:duration	La valeur d'entrée de type xs:duration.

# 6.7.24.21 seconds-from-time

Retourne un xs:integer représentant la partie seconds de la valeur xs:time fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
time	xs:time	La valeur d'entrée de type xs:time.

# 6.7.24.22 subtract-dateTimes

Retourne le xs:dayTimeDuration qui correspond à la différence entre la valeur normalisée de **dateTime1** et la valeur normalisée de **dateTime2**.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
dateTime1	xs:dateTime	La première valeur d'entrée.
dateTime2	xs:dateTime	La deuxième valeur d'entrée.

# 6.7.24.23 subtract-dates

Retourne le xs:dayTimeDuration qui correspond à la différence entre la valeur normalisée de **date1** et la valeur normalisée de **date2**.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
date1	xs:date	La première valeur d'entrée.
date2	xs:date	La deuxième valeur d'entrée.

### Exemple

Le mappage illustré ci-dessous soustrait deux dates (2020-10-22 moins 2020-09-22). Le résultat est la valeur p30D de type xs:dayTimeDuration, qui représente une durée de 30 jours.



# 6.7.24.24 subtract-times

Retourne le xs:dayTimeDuration qui correspond à la différence entre la valeur normalisée de **time1** et la valeur normalisée de **time2**.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
time1	xs:time	La première valeur d'entrée.
time2	xs:time	La deuxième valeur d'entrée.

# 6.7.24.25 timezone-from-date

Retourne le composant timezone de la date fournie en tant qu'argument. Le résultat est un xs:dayTimeDuration qui indique une déviation depuis UTC; sa valeur peut s'étendre de +14:00 à -14:00 heures, inclus tous deux.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
date	xs:date	La valeur d'entrée de type xs:date.

# 6.7.24.26 timezone-from-dateTime

Retourne le composant timezone de la valeur xs:dateTime fournie en tant qu'argument. Le résultat est un xs:dayTimeDuration qui indique une déviation depuis UTC; sa valeur peut s'étendre de +14:00 à -14:00 heures, inclus tous deux.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type xs:dateTime.

# 6.7.24.27 timezone-from-time

Retourne le composant timezone de la valeur xs:time fournie en tant qu'argument. Le résultat est un xs:dayTimeDuration qui indique une déviation depuis UTC; sa valeur peut s'étendre de +14:00 à -14:00 heures, inclus tous deux.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
time	xs:time	La valeur d'entrée de type xs:time.

# 6.7.24.28 year-from-date

Retourne un xs:integer représentant la partie year de la valeur xs:date fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
date	xs:date	La valeur d'entrée de type xs:date.

# 6.7.24.29 year-from-dateTime

Retourne un xs:integer représentant la partie year de la valeur xs:dateTime fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
dateTime	xs:dateTime	La valeur d'entrée de type xs:dateTime.

# 6.7.24.30 years-from-duration

Retourne un xs:integer représentant le composant "years" de la représentation canonique de la valeur de durée fournie en tant qu'argument.

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
duration	xs:duration	La valeur d'entrée de type xs:duration.

# 6.7.25 xpath2 | node functions

Les fonctions de nœud provenant de la bibliothèque **xpath2** fournissent des informations concernant les nœuds (items) dans un composant de mappage.

La fonction lang prend un argument string qui identifie un code de langage (comme par exemple "en"). La fonction retourne **true** ou **false** selon le fait que le nœud contextuel a un attribut xml:lang avec une valeur qui correspond à l'argument de la fonction.

Les fonctions local-name, name et namespace-uri, retournent, respectivement, le nom local, le nom et l'URI d'espace de nom du nœud d'entrée. Par exemple, pour le nœud **altova:Products**, le nom local est **products**, le nom est **altova:Products** et l'URI de l'espace de noms est l'URI de l'espace de noms auquel le préfixe altova: est lié (voir l'exemple donné pour la fonction <u>local-name</u><sup>(302)</sup>). Chacune de ces trois fonctions a deux variantes :

- Avec aucun argument : la fonction est ensuite appliquée au nœud contextuel (pour un exemple d'un nœud contextuel, voir l'exemple donné pour la fonction lang<sup>800</sup>).
- Avec un argument qui doit être un nœud : la fonction est appliquée au nœud contextuel.

La fonction number prend un nœud en tant qu'entrée, atomise le nœud (c'est à dire extrait son contenu) et convertit la valeur en une décimale puis retourne la valeur convertie. Il existe deux variantes de la fonction number :

- Avec aucun argument : la fonction est ensuite appliquée au nœud contextuel (pour un exemple d'un nœud contextuel, voir l'exemple donné pour la fonction lang<sup>(300)</sup>).
- Avec un argument qui doit être un nœud : la fonction est appliquée au nœud contextuel.

## 6.7.25.1 lang

Retourne **true** si le nœud contextuel a un attribut xml:lang avec une valeur soit qui correspond exactement à l'argument **testlang**, ou en est un sous-ensemble. Sinon, la fonction retourne **false**.



### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
testlang	xs:string	Le code de langage à vérifier, par exemple, "en".

### Exemple

Le XML suivant contient des éléments para avec des valeurs différentes pour l'attribut xml:lang.

```
<page>
<para xml:lang="en">Good day!</para>
<para xml:lang="fr">Bonjour!</para>
<para xml:lang="fr">Bonjour!</para>
<para xml:lang="de-AT">Grüss Gott!</para>
<para xml:lang="de-DE">Guten Tag!</para>
<para xml:lang="de-CH">Grüezi!</para>
</page>
```

Le mappage illustré ci-dessous filtre uniquement les paragraphes en allemand, quelle que soit la variante du pays, avec l'aide de la fonction lang.



Mappage XSLT 2.0

Dans le mappage ci-dessus, pour chaque **para** dans la source, un **item** est créé dans la cible, de manière conditionnelle. La condition est fournie par un filtre qui transfère à la cible uniquement les nœuds dans lesquels la fonction **lang** retourne **true**. Ainsi, seul les nœuds qui ont l'attribut xml:lang définis sur "de" (ou un sousensemble de "de") satisferont la condition du filtre. Par conséquent, la sortie de mappage est le suivant :



Veuillez noter que la fonction lang opère dans le contexte de chaque **para**, à cause de la connexion parent entre **para** et **item**, voir aussi <u>Le contexte de mappage</u><sup>374</sup>.

# 6.7.25.2 local-name

Retourne la partie locale du nom du nœud contextuel en tant qu'un xs:string. Il s'agit d'une variante sans paramètres de la fonction local-name où le nœud contextuel est déterminé par les connexions dans votre

mappage. Pour spécifier un nœud explicitement, utiliser la fonction <u>local-name</u><sup>802</sup> qui prend un nœud d'entrée en tant que paramètre.



Langages XQuery, XSLT 2.0, XSLT 3.0.

# 6.7.25.3 local-name

Retourne la partie locale du nom du nœud contextuel en tant qu'un xs:string.

∫<sub>8</sub> local-name ⊳node result ⊳

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
node	node()	Le nœud d'entrée.

### Exemple

Dans le fichier XML suivant, le nom de l'élément p:product est un nom qualifié préfixé (QName). Le préfixe "p" est mappé dans l'espace de noms "http://mycompany.com".

Le mappage suivant extrait le nom local, le nom et l'URI d'espace de noms du nœud et écrit ces valeurs dans un fichier cible :



#### Mappage XSLT 2.0

La sortie de mappage est affichée ci-dessous. Chaque item **col** liste le résultat des fonctions **local-name**, **name** et **namespace-uri**, respectivement.

```
<rows>
        <row>
            <coll>product</coll>
            <col2>p:product</col2>
            <col3>http://mycompany.com</col3>
            </row>
</rows>
```

## 6.7.25.4 name

Retourne le nom du nœud contextuel. Il s'agit d'une variante sans paramètres de la fonction name où le nœud contextuel est déterminé par les connexions dans votre mappage. Pour spécifier un nœud explicitement, utiliser la fonction name <sup>804</sup> qui prend un nœud d'entrée en tant que paramètre.



Langages XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.25.5 name

Retourne le nom d'un nœud.

∫ <sub>8</sub> nar	ne
⊳node	result 🕻

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
node	node()	Le nœud d'entrée.

### Exemple

Voir l'exemple donné pour la fonction <u>local-name</u><sup>802</sup>.

## 6.7.25.6 namespace-uri

Retourne l'espace de nom URI du QName du nœud contextuel en tant xs:string. Il s'agit d'une variante sans paramètres de la fonction namespace-uri où le nœud contextuel est déterminé par les connexions dans votre mappage. Pour spécifier un nœud explicitement, utiliser la fonction <u>namespace-uri</u> <sup>604</sup> qui prend un nœud d'entrée en tant que paramètre.



Langages XQuery, XSLT 2.0, XSLT 3.0.

## 6.7.25.7 namespace-uri

Retourne l'espace de nom URI du QName du nœud, en tant que xs:string.

	$f_8$ names	space-uri
Ę	>node	result 🗘

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
node	node()	Le nœud d'entrée.

### Exemple

Voir l'exemple donné pour la fonction <u>local-name</u><sup>802</sup>.

## 6.7.25.8 number

Retourne la valeur du nœud contextuel, converti dans un xs:double. Il s'agit d'une variante sans paramètres de la fonction number où le nœud contextuel est déterminé par les connexions dans votre mappage. Pour spécifier un nœud explicitement, utiliser la fonction number au prend un nœud d'entrée en tant que paramètre.

Les seuls types qui peuvent être convertis dans des nombres sont des booléennes, des strings numériques et d'autres types numériques. Les valeurs d'entrée non-numériques (comme un string non-numérique) résultent en NaN (Not a Number - Pas un Nombre).



Langages XQuery, XSLT 2.0, XSLT 3.0.

# 6.7.25.9 number

Retourne la valeur du **nœud**, converti dans un xs:double. Les seuls types qui peuvent être convertis dans des nombres sont des booléennes, des strings numériques et d'autres types numériques. Les valeurs d'entrée non-numériques (comme un string non-numérique) résultent en NaN (Not a Number - Pas un Nombre).



Langages XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
node	mf:atomic	Le nœud d'entrée.

#### Exemple

Le XML suivant contient des items de type string:



Le mappage illustré ci-dessous tente de convertir tous ces strings en des valeurs numériques et les écrit dans un fichier XML cible. Veuillez noter que le type de données de **item** dans le composant XML cible est xs:integer alors que l'**item** de source est de type de données xs:string. Si la conversion échoue, l'item doit être sauté et ne sera pas copié dans le fichier cible.



#### Mappage XSLT 2.0

Un filtre a été utilisé pour parvenir à l'objectif de mappage. La fonction equal vérifie si le résultat de la conversion est "NaN". Si cela est faux, cela indique une conversion réussie, l'item est donc copié dans la cible. La sortie du mappage est comme suit :



# 6.7.26 xpath2 | numeric functions

Les fonctions numériques de la bibliothèque xpath2 comprennent les fonctions abs et round-half-to-even.

## 6.7.26.1 abs

Retourne la valeur absolue de l'argument. Par exemple, si l'argument d'entrée est -2 ou 2, la fonction retourne 2.



## Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
value	xs:decimal	La valeur d'entrée.

# 6.7.26.2 round-half-to-even

La fonction **round-half-to-even** arrondit le nombre fourni (premier argument) à la précision décimale (nombre de chiffres après la virgule) fourni dans le second argument optionnel. Par exemple, si le premier argument est **2.141567** et que le second argument est **3**, alors le premier argument (le nombre) est arrondi à trois chiffres après la virgule, le résultat sera donc **2.142**. Si aucune précision décimale (second argument) n'est fournie, le nombre sera arrondi à zéro places décimales, c'est à dire donc à un entier.

Le terme "even" dans le nom de la fonction réfère à l'arrondissement à un nombre pair lorsqu'un chiffre dans le nombre fourni se trouve entre deux valeurs. Par exemple, round-half-to-even(3.475, 2) retournerait **3.48**.

$f_{\!\!8}$ round-half-to-even	
₽ value	
precision	result

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
valeur	xs:decimal	Argument obligatoire qui fournit la valeur d'entrée à arrondir.

Nom	Туре	Description
precision	xs:integer	Argument optionnel qui spécifie le nombre de décimales à arrondir. La valeur par défaut est de <b>0</b> .

# 6.7.27 xpath2 | string functions

Les fonctions string de la bibliothèque **xpath2** vous permettent de traiter des strings (cela inclut la comparaison de strings, la conversion de strings en casse majuscule ou minuscule, l'extraction de sousstrings depuis des strings, etc.).

# 6.7.27.1 codepoints-to-string

Crée un string depuis une séquence de points de code Unicode. Cette fonction est le contraire de la fonction string-to-codepoints<sup>816</sup>.

 f<sub>\*</sub>
 codepoints-to-string

 Codepoints
 result P

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
codepoints	ZeroOrMore xs:integer	Cette entrée doit être connectée à une séquence d'items de type integer, où chaque entier spécifie un point de code Unicode.

### Exemple

L'XML suivant contient plusieurs éléments item qui stockent des valeurs de point de code chacun Unicode.

```
<items>
    <items>77</item>
    <item>97</item>
    <item>97</item>
    <item>112</item>
    <item>70</item>
    <item>111</item>
    <item>111</item>
    <item>99</item>
```

```
<item>101</item> </items>
```

Le mappage illustré ci-dessous fournit la séquence des items en tant qu'argument dans la fonction codepointto-string.





La sortie de mappage est MapForce.

# 6.7.27.2 compare

La fonction compare prend deux strings en tant qu'arguments et les compare en terme d'égalité et alphabétiquement. Si **string1** est alphabétiquement inférieur à **string2** (par exemple les deux strings sont "A" et "B"), la fonction retourne **-1**. Si les deux strings sont égaux (par exemple, "A" et "A"), la fonction retourne **0**. Si **string1** est supérieur à **string2** (par exemple, "B" et "A"), alors la fonction retourne **1**.

Cette variante de la fonction utilise la collation par défaut, qui est Unicode. Une autre variante de cette fonction existe là où vous pouvez fournir la collation en tant qu'argument.



Langages XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
string1	xs:string	Le premier string d'entrée.
string2	xs:string	Le second string d'entrée.

# 6.7.27.3 compare

La fonction compare prend deux strings en tant qu'arguments et les compare en terme d'égalité et alphabétiquement, en utilisant la collation fournie en tant qu'argument. Si **string1** est alphabétiquement inférieur à **string2** (par exemple les deux strings sont "A" et "B"), la fonction retourne **-1**. Si les deux strings sont égaux (par exemple, "A" et "A"), la fonction retourne **0**. Si **string1** est supérieur à **string2** (par exemple, "B" et "A"), alors la fonction retourne **1**.

f <sub>8</sub> compare	
⊳string1	
⊳string2	result 🛱
Collation	

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
string1	xs:string	Le premier string d'entrée.
string2	xs:string	Le second string d'entrée.
collation	xs:string	Spécifie la collation à utiliser pour la comparaison de string. Cette entrée peut provenir de la sortie de la fonction <u>default-</u> <u>collation</u> <sup>(72)</sup> ou il peut s'agir d'une collation comme http://www.w3.org/2005/xpath -functions/collation/html- ascii-case-insensitive.

### Exemple

Le mappage suivant compare les strings "A" et "a" en utilisant la collation insensible à la casse http://www.w3.org/2005/xpath-functions/collation/html-ascii-case-insensitive, qui est fournie par une constante.



Mappage XSLT 2.0

Le résultat du mappage ci-dessus est **0**, signifiant que les deux strings sont traités à égalité. Néanmoins, si vous remplacez la collation avec celle fournie par la fonction <u>default-collation</u>, la collation change pour passer à la collation de point de code Unicode par défaut, et le résultat de mappage devient **-1** ("A" est alphabétiquement inférieur à "a").



# 6.7.27.4 ends-with

Retourne true si string se termine avec substr; false sinon. La valeur retournée est de type xs:boolean.

Cette variante de la fonction utilise la collation par défaut, qui est Unicode. Une autre variante <sup>(312)</sup> de cette fonction existe là où vous pouvez fournir la collation en tant qu'argument.



### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
string	xs:string	Le string d'entrée (c'est à dire la "pile de foin").
substr	xs:string	Le sous-string (c'est à dire l'"aiguille").

## 6.7.27.5 ends-with

Retourne true si string se termine avec substr; false sinon. La valeur retournée est de type xs:boolean.

$f_{s}$ ends-with	
string	
⊳substr	result 🕻
collation	

#### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
string	xs:string	Le string d'entrée (c'est à dire la "pile de foin").
substr	xs:string	Le sous-string (c'est à dire l'"aiguille").
collation	xs:string	Spécifie la collation à utiliser pour la comparaison de string. Cette entrée peut provenir de la sortie de la fonction <u>default-</u> <u>collation</u> <sup>(72)</sup> ou il peut s'agir d'une collation comme http://www.w3.org/2005/xpath -functions/collation/html- ascii-case-insensitive.

## 6.7.27.6 lower-case

Retourne la valeur de string après avoir traduit chaque caractère dans son correspondant en minuscule.



### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
string	xs:string	La valeur d'entrée.

# 6.7.27.7 matches

La fonction **matches** teste si un string fourni (le premier argument) correspond à une expression régulière (le second argument). La syntaxe des expressions régulières doit être celui défini pour la facette pattern du Schéma XML. La fonction retourne **true** si le string correspond à l'expression régulière, **false** sinon.

$f_{\rm 8}$ matches	
l>input	
¢ pattern	result 🕻
flags	

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description	
input	xs:string	Туре	
pattern	xs:string	L'expression régulière à faire correspondre, voir <u>Expressions</u>	
flags	xs:string	<ul> <li>Argument optionnel influe sur la correspondance. Cet argument peut fournir n'importe quelle combinaison des flags suivants : i, m, s, x. Plusieurs flags peuvent être utilisés, par exemple, imx. Si aucun flag n'est utilisé, les valeurs par défaut des quatre flags seront utilisées. Les quatre flags sont les suivants :</li> <li>i Utiliser le mode insensible à la casse. Le défaut est sensible à la casse.</li> <li>m Utiliser le mode multiligne, dans lequel le string d'entrée est considéré avoir plusieurs lignes, chacune séparée par un caractère newline (x0a). Les caractères meta  et s indiquent le début et la fin de chaque ligne. Le mode par défaut est le mode string, dans lequel le string commence et termine par les caractères méta  et s.</li> </ul>	

Nom	Туре	Description	
		<ul> <li>Utiliser le mode dot-all. Le mode par défaut est le mode not- dot-all, dans lequel le caractére méta . Correspond à tous les caractères sauf le caractère newline (x0a). Dans le mode dot-all, le point correspond aussi au caractère newline.</li> </ul>	
		<ul> <li>Ignorer l'espace blanc. Par défaut, les caractères d'espace blanc ne sont pas ignorés.</li> </ul>	

# 6.7.27.8 normalize-unicode

Retourne la valeur de **string** normalisée conformément aux règles du formulaire de normalisation spécifié (le second argument). Pour plus d'informations concernant la normalisation Unicode, voir §2.2 de <u>https://www.w3.org/TR/charmod-norm/</u>.

f <sub>8</sub> normalize-unicode		
string		
( <mark>n</mark>	ormalizationForm	result

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
string	xs:string	La valeur string à être normalisé.
normalizationForm	xs:string	L'argument optionnel fournit le formulaire de normalisation. Le défaut est Unicode Normalization Form C (NFC).
		Les formulaires de normalisation NFC, NFD, NFKC et NFKD sont pris en charge.

# 6.7.27.9 replace

Cette fonction prend un string d'entrée, une expression régulière et un string de remplacement en tant qu'arguments. Elle remplace toutes les correspondances de l'expression régulière dans le string d'entrée avec le string de remplacement.. Si l'expression régulière correspond à deux strings se chevauchant dans le string d'entrée, seule la première correspondance est remplacée.

$f_{\rm s}$ replace	
P <mark>input</mark>	
▷ pattern	
replacement	result
flags	

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
input	xs:string	Туре
pattern	xs:string	L'expression régulière à faire correspondre, voir <u>Expressions</u> régulière <sup>627</sup> .
replacement	xs:string	Le string de remplacement.
flags	xs:string	Argument optionnel influe sur la correspondance. Cet argument est utilisé de la même manière que l'argument <b>flags</b> de la fonction <b>matches</b>

# 6.7.27.10 starts-with

Retourne **true** si **string** commence avec **substr**; **false** sinon. La valeur retournée est de type xs:boolean. La comparaison de string prend place conformément à la collation spécifiée.

f <sub>8</sub> starts-with	
⊳ string	
⊳substr	result 🖡
collation	

Langages XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
string	xs:string	Le string d'entrée (c'est à dire la "pile de foin").
substr	xs:string	Le sous-string (c'est à dire l'"aiguille").
collation	xs:string	Spécifie la collation à utiliser pour la comparaison de string. Cette entrée peut provenir de la sortie de la fonction default- collation <sup>(782)</sup> ou il peut s'agir d'une collation comme http://www.w3.org/2005/xpath -functions/collation/html- ascii-case-insensitive.

## Exemple

Le mappage suivant retourne la valeur true, parce que le string d'entrée "MapForce" commence avec le sousstring "Map", en partant du principe que la collation Unicode par défaut est utilisée.



# 6.7.27.11 string-to-codepoints

Retourne la séquence des points de code Unicode (valeur d'entier) qui constitue le string fourni en tant qu'argument. Cette fonction est le contraire de la fonction <u>codepoints-to-string</u><sup>808</sup>.

f <sub>8</sub> string-to-codepoints			
⊳input		put	result 🕻

Langages XQuery, XSLT 2.0, XSLT 3.0.

Nom	Туре	Description
input	xs:string	Le string d'entrée

# 6.7.27.12 substring-after

Retourne la partie du string **arg1** qui se produit après le string **arg2**.

$f_{\rm s}$ substring-after	
>arg1	
¢arg2	result 🛱
Collation	

## Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
arg1	xs:string	Le string d'entrée (c'est à dire la "pile de foin").
arg2	xs:string	Le sous-string (c'est à dire l'"aiguille").
collation	xs:string	Spécifie la collation à utiliser pour la comparaison de string. Cette entrée peut provenir de la sortie de la fonction <u>default-</u> <u>collation</u> <sup>(72)</sup> ou il peut s'agir d'une collation comme http://www.w3.org/2005/xpath -functions/collation/html- ascii-case-insensitive.

### Exemple

Si **arg1** est "MapForce", **arg2** est "Map", et **collation** est <u>default-collation</u><sup>762</sup>, la fonction retourne "Force".

# 6.7.27.13 substring-before

Retourne la partie du string arg1 qui se produit avant le string arg2.

f <sub>8</sub> substring-before	
>arg1	
⊳arg2	] result 🛱
collation	1

### Langages

XQuery, XSLT 2.0, XSLT 3.0.

### Paramètres

Nom	Туре	Description
arg1	xs:string	Le string d'entrée (c'est à dire la "pile de foin").
arg2	xs:string	Le sous-string (c'est à dire l'"aiguille").
collation	xs:string	Spécifie la collation à utiliser pour la comparaison de string. Cette entrée peut provenir de la sortie de la fonction <u>default-</u> <u>collation</u> <sup>722</sup> ou il peut s'agir d'une collation comme http://www.w3.org/2005/xpath -functions/collation/html- ascii-case-insensitive.

### Exemple

Si **arg1** est "MapForce", **arg2** est "Force", et **collation** est <u>default-collation</u><sup>762</sup>, la fonction retourne "Map".

## 6.7.27.14 upper-case

Retourne la valeur de string après avoir traduit chaque caractère dans son correspondant en majuscule.



### Langages

XQuery, XSLT 2.0, XSLT 3.0.

Paramètres

Nom	Туре	Description
string	xs:string	Туре

# 6.7.28 xpath3 | external information functions

Les fonctions d'information externe de la librairie **xpath3** vous permettent d'obtenir l'information sur l'environnement d'exécution XSLT ou d'extraire des données de ressources externes.

## 6.7.28.1 available-environment-variables

Retourne une liste de noms de variables d'environnement qui conviennent pour passer à la fonction **environment-variable**, comme séquence de strings (éventuellement vide).

∫<sub>8</sub> available-environment-variables names ⊳

Langages XSLT 3.0.

## 6.7.28.2 environment-variable

Retourne la valeur d'une variable d'environnement système, si elle existe. The type de retour est xs:string.

	f <sub>x</sub> environment-variable		
Ç	>na	ame	result Þ

Langages XSLT 3.0.

Nom	Туре	Description
nom	xs:string	Le nom de la varaible d'environnement.

## 6.7.28.3 unparsed-text

Lit une ressource externe (par exemple, un fichier) et retourne une représentation string de la ressource.

f <sub>x</sub> unparsed-text	
l> href	
encoding	result

#### Langages

XSLT 3.0.

### Paramètres

Nom	Туре	Description
href	xs:string	Un string sous forme d'une référence URI.
encoding	xs:string	Argument optionnel. Spécifie le nom de l'encodage, par exemple "UTF-8", "UTF-16". Si l'encodage ne peut pas être déterminé automatiquement, alors UTF-8 est présumé.

## 6.7.28.4 unparsed-text-available

Détermine si un appel vers unparsed-text avec des arguments particuliers réussirait. Le type de retour est xs:boolean.

1	$f_{\rm 8}$ unparsed-text-available	
Þ١	href	
5	encoding	result

#### Langages

XSLT 3.0.

### Paramètres

Nom	Туре	Description
href	xs:string	Un string sous forme d'une référence URI.
encoding	xs:string	Argument optionnel. Spécifie le nom de l'encodage, par exemple "UTF-8", "UTF-16". Si l'encodage ne peut pas être déterminé automatiquement, alors UTF-8 est présumé.

# 6.7.28.5 unparsed-text-lines

Lit une ressource externe (par exemple, un fichier) et retourne ses contenus comme une séquence de strings, une par ligne de texte dans la représentation string de la ressource.

	$\mathbf{f}_{\mathbf{x}}$	unparsed-t	ext-lines
Ç	>hr	ef	lines
ţ	> en	coding	lines

### Langages

XSLT 3.0.

Nom	Туре	Description
href	xs:string	Un string sous forme d'une référence URI.
encoding	xs:string	Argument optionnel. Spécifie le nom de l'encodage, par exemple "UTF-8", "UTF-16". Si l'encodage ne peut pas être déterminé automatiquement, alors UTF-8 est présumé.

# 6.7.29 xpath3 | formatting functions

Les fonctions de formatage disponibles de la bibliothèque **xpath3** sont utilisées pour formater la date, l'heure et les valeurs d'entier.

# 6.7.29.1 format-date

Retourne un string contenant une xs: date valeur formatée pour l'affichage.

$f_8$ format-	f <sub>8</sub> format-date	
¢ value		
▷ picture		
language	result 🕻	
calendar		
place		

### Langages

XSLT 3.0.

Nom	Туре	Description
value	xs:date	Fournit la valeur xs: date à formater. Paramètre obligatoire.
image	xs:string	Paramètre obligatoire. Voir section 9.8.4.1 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » (https://www.w3.org/TR/xpat h-functions-31).
language	xs:string	Paramètre optionnel. Voir section 9.8.4.8 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » ( <u>https://www.w3.org/TR/xpat</u> <u>h-functions-31</u> ).
calendrier	xs:string	Comme ci-dessus.

Nom	Туре	Description
place	xs:string	Comme ci-dessus.

# 6.7.29.2 format-dateTime

Retourne un string contenant une xs: dateTime valeur formatée pour l'affichage.

$f_{\!\scriptscriptstyle R}$ format-dateTime	
¢ value	
¢ picture	]
language	result 🛱
calendar	]
place	

### Langages

XSLT 3.0.

Nom	Туре	Description
value	xs:dateTime	Fournit la valeur xs: dateTime à formater.
image	xs:string	Paramètre obligatoire. Voir section 9.8.4.1 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » ( <u>https://www.w3.org/TR/xpat</u> <u>h-functions-31</u> ).
language	xs:string	Paramètre optionnel. Voir section 9.8.4.8 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » ( <u>https://www.w3.org/TR/xpat</u> <u>h-functions-31</u> ).
calendrier	xs:string	Comme ci-dessus.
place	xs:string	Comme ci-dessus.

# 6.7.29.3 format-integer

Formate un entier conformément à un string d'image donné, utilisant les conventions d'un langage naturel donné, si spécifié.

f <sub>8</sub> format-integer	
⊳value	
picture	result 🛱
language	

## Langages

XSLT 3.0.

Nom	Туре	Description
value	xs:integer	La valeur de l'entier de l'entrée à formater.
image	xs:string	Paramètre obligatoire. Voir la section 4.6.1 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » (https://www.w3.org/TR/xpat
		<u>h-functions-31</u> ).
language	xs:string	Paramètre optionnel. Spécifie le langage naturel selon la manière avec laquelle la valeur doit être formatée. Si spécifié, la valeur doit soit être un string vide ou toute valeur autorisée pour l'attribut xml:lang conformément à la Recommandation W3C « Extensible Markup Language (XML) 1.0 » (https://www.w3.org/TR/xml).

# 6.7.29.4 format-time

Retourne un string contenant une xs:time valeur formatée pour l'affichage.

f <sub>8</sub> format-time	
Value	
▷ picture	
language	result 🕻
calendar	
place	

### Langages

XSLT 3.0.

Nom	Туре	Description
value	xs:time	Fournit la valeur xs:time à formater.
image	xs:string	Paramètre obligatoire. Voir section 9.8.4.1 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » ( <u>https://www.w3.org/TR/xpat</u> <u>h-functions-31</u> ).
language	xs:string	Paramètre optionnel. Voir section 9.8.4.8 de la Recommandation W3C « XPath and XQuery Functions and Operators 3.1 » ( <u>https://www.w3.org/TR/xpat</u> <u>h-functions-31</u> ).
calendrier	xs:string	Comme ci-dessus.
place	xs:string	Comme ci-dessus.

# 6.7.30 xpath3 | math functions

Les fonctions mathématiques de la bibliothèque **xpath3** sont utilisées pour effectuer des calculs trigonométriques ou autres calculs mathématiques.

## 6.7.30.1 acos

Retourne l'arc cosinus d'un angle, allant de 0 à pi.



### Langages

XSLT 3.0.

### Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

## 6.7.30.2 asin

Retourne l'arc sinus d'un angle, allant de -pi/2 à pi/2.



### Langages

XSLT 3.0.

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

## 6.7.30.3 atan

Retourne l'arc tangente d'un angle, allant de -pi/2 à pi/2.



## Langages

XSLT 3.0.

### Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

# 6.7.30.4 atan2

Retourne une liste de noms de variables d'environnement qui conviennent pour passer à la fonction **environment-variable**, comme séquence de strings (éventuellement vide).

```
∫<sub>8</sub> available-environment-variables
names ₽
```

### Langages

XSLT 3.0.

## 6.7.30.5 cos

Retourne le cosinus trigonométrique de l'angle indiqué par la valeur. L'unité de la valeur est radian.



### Langages XSLT 3.0.

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

# 6.7.30.6 exp

Retourne le nombre d'Euler *e* à la puissance de la valeur.



### Langages

XSLT 3.0.

#### Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

# 6.7.30.7 exp10

Retourne 10 à la puissance de la valeur.



Langages

XSLT 3.0.

Nom	Туре	Description
value	xs:double	La valeur d'entrée.
# 6.7.30.8 log

Retourne le logarithme naturel (base e) d'une valeur.



## Langages

XSLT 3.0.

## Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

## 6.7.30.9 log10

Retourne le logarithme décimal (base 10) d'une valeur.



## Langages

XSLT 3.0.

## Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

# 6.7.30.10 pi

Retourne une approximation vers la constante mathématique pi.



#### Langages

XSLT 3.0.

## 6.7.30.11 pow

Retourne la valeur de **a** élevé à la puissance de **b**.



## Langages

XSLT 3.0.

## Paramètres

Nom	Туре	Description
а	xs:double	La valeur d'entrée <b>a</b> .
b	xs:double	La valeur d'entrée <b>b</b> .

## 6.7.30.12 sin

Retourne le sinus trigonométrique d'un angle indiqué par la valeur. L'unité de valeur est en radian.



## Langages

XSLT 3.0.

## Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

## 6.7.30.13 sqrt

Retourne la racine carrée non négative de l'argument.



## Langages

XSLT 3.0.

## Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

# 6.7.30.14 tan

Retourne la tangente trigonométrique de l'angle indiquée par la valeur. L'unité de valeur est en radian.



## Langages

XSLT 3.0.

## Paramètres

Nom	Туре	Description
value	xs:double	La valeur d'entrée.

# 6.7.31 xpath3 | URI functions

Les fonctions URI de la bibliothèque **xpath3** réalisent l'encodage, l'échappement et la conversion de valeurs pour l'utilisation dans les URI.

# 6.7.31.1 encode-for-uri

Encode les caractères réservés dans un string qui est identifié à être utilisé dans le segment de chemin d'un URI. Pour plus d'information concernant cette fonction, voir la section 6.2 de la Recommandation W3C de « XPath and XQuery Functions and Operators 3.1 » (<u>https://www.w3.org/TR/xpath-functions-31</u>).



#### Langages

XSLT 3.0.

#### Paramètres

Nom	Туре	Description
uri-part	xs:string	La valeur d'entrée de l'URI à encoder.

## 6.7.31.2 escape-html-uri

Échappe un URI de la même manière que les agents utilisateurs d'HTML gèrent des valeurs d'attribut prévues contenir des URI. Pour plus d'information concernant cette fonction, voir la section 6.4 de la Recommandation W3C de « XPath and XQuery Functions and Operators 3.1 » (<u>https://www.w3.org/TR/xpath-functions-31</u>).



## Langages

XSLT 3.0.

#### Paramètres

Nom	Туре	Description
uri	xs:string	La valeur d'entrée de l'URI à échapper.

# 6.7.31.3 iri-to-uri

Convertit un string contenant l'IRI (Internationalized Resource Identifier) en un URI (Uniform Resource Identifier). Pour plus d'information concernant cette fonction, voir la section 6.3 de la Recommandation W3C de « XPath and XQuery Functions and Operators 3.1 » (<u>https://www.w3.org/TR/xpath-functions-31</u>).



## Langages

XSLT 3.0.

#### Paramètres

Nom	Туре	Description
iri	xs:string	La valeur d'entrée de l'IRI.

# 6.7.32 xslt | xpath functions

Les fonctions dans ce sous-groupe sont des fonctions XPath 1.0 qui extraient des informations concernant des items de mappage (ou des nœuds). La plupart de ces fonctions prennent un nœud en tant qu'argument et retournent des informations concernant ce nœud. Les fonctions last et position fonctionnent dans le <u>contexte de mappage</u> actuel qui est déterminé par les connexions dans votre mappage.

**Note :** vous trouverez des fonctions XPath 1.0 supplémentaires dans la bibliothèque **core**.

## 6.7.32.1 lang

Retourne **true** si le nœud contextuel a un attribut xml:lang avec une valeur soit qui correspond exactement à l'argument **string**, ou en est un sous-ensemble. Sinon, la fonction retourne **false**.



Langages XSLT 1.0.

## Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Le code de langage à vérifier, par exemple, "en".

## Exemple

Voir l'exemple donné pour la fonction <u>lang</u><sup>600</sup> de la bibliothèque **xpath2**.

## 6.7.32.2 last

Retourne le numéro de la position du dernier nœud dans la liste de nœud traitée.



## Langages

XSLT 1.0.

## Exemple

Voir l'exemple donné pour la fonction <u>last</u> 22 de la bibliothèque **xpath2**.

## 6.7.32.3 local-name

Retourne la partie locale du nom du nœud fourni en tant qu'argument.



Langages XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Nom	Туре	Description
node	node()	Le nœud d'entrée.

## Exemple

Voir l'exemple donné pour la fonction <u>local-name</u> <sup>802</sup> de la bibliothèque **xpath2**.

## 6.7.32.4 name

Retourne le nom du nœud fourni en tant qu'argument.

	∫ <sub>8</sub> name		ne	
ç	>no	de	result 🕻	Ş

## Langages XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
node	node()	Le nœud d'entrée.

## Exemple

Voir l'exemple donné pour la fonction local-name <sup>802</sup> de la bibliothèque **xpath2**.

## 6.7.32.5 namespace-uri

Retourne l'espace de nom URI du nœud fourni en tant qu'argument.

<mark>∫<sub>8</sub> namespace-uri</mark> ⊳node result ⊳

## Langages

XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Nom	Туре	Description
node	node()	Le nœud d'entrée.

#### Exemple

Voir l'exemple donné pour la fonction <u>local-name</u><sup>802</sup> de la bibliothèque **xpath2**.

## 6.7.32.6 position

Retourne la position du nœud actuel dans le nœud qui est traité actuellement.



Langages

XSLT 1.0.

## 6.7.33 xslt | xslt functions

Les fonctions dans ce groupe sont des fonctions XSLT 1.0 diverses.

## 6.7.33.1 current

La fonction **current** ne prend aucun argument et retourne le nœud actuel.



Langages XSLT 1.0.

## 6.7.33.2 document

Accède aux nœuds depuis un document XML externe. Le résultat est sorti dans un nœud dans le document de sortie.

$f_{\rm R}$	f <sub>8</sub> document	
l∳uri		
>no	deset	result

#### Langages

XSLT 1.0.

## Paramètres

Nom	Туре	echo Hello, World!
uri	xs:string	Obligatoire. Spécifie le chemin vers le document XML. Le document XML doit être valide et parsable.
nodeset	node()	Optionnel Spécifie un nœud, l'URI de base qui est utilisé pour résoudre l'URI fournie en tant que le premier argument s'll est relatif.

## 6.7.33.3 element-available

La fonction **element-available** teste si un élément, saisi en tant que le seul argument de string de la fonction, est pris en charge par le processeur XSLT. Le string d'argument est évalué en tant qu'un QName. C'est pourquoi, les éléments XSLT doivent avoir un préfixe xs1: et les éléments de Schéma XML doivent avoir un préfixe xs: —puisqu'ils s'agit des préfixes déclarés pour ces espaces de noms dans le XSLT sous-jacent qui sera généré pour le mappage.. La fonction retourne une booléenne.

	$f_8$ element	-available
C	element	result 🗘

Langages XSLT 1.0.

## Paramètres

Nom	Туре	echo Hello, World!
element	xs:string	Le nom d'élément.

## 6.7.33.4 function-available

La fonction **function-available** est semblable à la fonction **element-available** et teste si le nom de fonction est fourni en tant que l'argument de la fonction est pris en charge par le processeur XSLT. Le string d'entrée est évalué en tant qu'un QName. La fonction retourne une booléenne.

$f_8$ function-	available
function	result 🖓

#### Langages

XSLT 1.0.

#### Paramètres

Nom	Туре	echo Hello, World!
fonction	xs:string	Le nom de la fonction.

## 6.7.33.5 generate-id

La fonction generate-id génère un string unique qui identifie le premier nœud dans le nodeset identifié par l'argument d'entrée optionnel. Si aucun argument n'est fourni, l'ID est généré dans le nœud contextuel. Le résultat peut être dirigé à tout nœud dans le document de sortie.



## Langages

XSLT 1.0, XSLT 2.0, XSLT 3.0.

#### Paramètres

Nom	Туре	Description
nodeset	node()	Argument optionnel qui fournit le nœud d'entrée.

## 6.7.33.6 system-property

La fonction **system-property** retourne des propriétés du processeur XSLT (le système). Trois propriétés de système, toutes dans l'espace de noms XSLT, sont obligatoires pour les processeurs XSLT. Il s'agit de xsl:version, xsl:vendor et xsl:vendor-url. Le string d'entrée est évalué en tant que QName et doit donc avoir le préfixe xsl: puisque le préfixe associé avec l'espace de noms XSLT dans la feuille de style XSLT sousjacente.

∫<sub>x</sub> system-property ⊳string result ⊳

## Langages

XSLT 1.0, XSLT 2.0, XSLT 3.0.

## Paramètres

Nom	Туре	Description
string	xs:string	Specifie le nom de propriété qui peut être un des suivants : xsl:version, xsl:vendor, xsl:vendor-url.

# 6.7.33.7 unparsed-entity-uri

Si vous utilisez un DTD, vous pouvez y déclarer une entité non parsée. Cette entité non parsée (par exemple une image) aura une URI qui situe l'entité non parsée. Le string d'entrée de la fonction doit correspondre au nom de l'entité non parsée qui a été déclarée dans le DTD. La fonction retourne ensuite l'URI de l'entité non parsée, qui peut ensuite être dirigée dans un nœud dans le document de sortie, par exemple dans un nœud href.

	f <sub>8</sub> unparsed-entity-uri		
ç	>st	ring	result 🗘

## Langages

XSLT 1.0.

#### Paramètres

Nom	Туре	echo Hello, World!
Paramètres	xs:string	Le nom de l'entité non parsée dont l'URI doit être extraite.

#### Scénarios de mappage avancé 7

Site web d'Altova : & Outil d'intégration des données

Cette section décrit les scénarios de mappage avancé et inclut les rubriques suivantes :

- Mappages en chaîne<sup>841</sup> ٠
- Mapper les noms de nœud<sup>851</sup> •
- Règles et stratégies de mappage<sup>872</sup> •
- <u>Traiter de multiples fichiers d'entrée ou de sortie</u><sup>883</sup>
   <u>Parser et sérialiser des strings</u><sup>904</sup>
- Volets de Sortie StyleVision<sup>912</sup> ٠
- Générer une documentation de mappage 916 •

# 7.1 Mappages en chaîne

MapForce prend en charge les mappages qui consistent en plusieurs composants contenus dans une chaîne de mappage. Les mappages en chaîne sont des mappages dont au moins un composant agit en tant que source et cible. Un tel composant crée une sortie qui sera utilisée ultérieurement en tant qu'entrée pour une étape de mappage suivante dans la chaîne. Un tel composant est appelé un composant "intermédiaire".

Par exemple, le mappage illustré ci-dessous montre une note de frais (en format XML) qui sera traitée en deux phases. La partie du mappage de A à B filtre uniquement les dépenses libellées en tant que "Travel". Le mappage de B à C filtre uniquement les dépenses "Travel" dont le coût est de moins de 1500. Le composant B est le composant "intermédiaire", étant donné qu'il a aussi bien des connexions d'entrée et de sortie. Ce mappage est disponible sous le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\ChainedReports.mfd.



#### ChainedReports.mfd

Les mappages en chaîne introduisent une fonction appelée "passage". "Passage" est un aperçu vous permettant de consulter la sortie produite à chaque étape d'un mappage en chaîne dans la fenêtre Sortie. Par exemple, dans le mappage ci-dessus, vous pouvez consulter et enregistrer la sortie XML résultant de A à B, ainsi que la sortie XML résultant de B à C.

**Note :** La fonction "passage" est uniquement disponible pour les composants basés sur fichier (par exemple, XML, CSV, et texte). Les composants de base de données peuvent être intermédiaires, mais la touche de passage n'est pas indiquée. Le composant intermédiaire est toujours régénéré à partir de zéro lors de la consultation ou de la génération du code. Cela ne serait pas faisable avec une base de données étant donné qu'elle devrait être supprimée avant chaque régénération.

Si le mappage est exécuté par MapForce Server, ou par le code généré, la chaîne de mappage complète sera exécutée. Le mappage génère les fichiers de sortie nécessaires à chaque étape de la chaîne, et la sortie d'une étape d'un mappage en chaîne est transmise en tant qu'entrée vers l'étape de mappage suivante.

Il est également possible de générer des noms de fichier dynamiques pour les composants intermédiaires. C'est à dire qu'ils peuvent accepter des connexions vers l'item "File:" depuis le mappage, si le composant est configuré dans ce sens. Pour plus d'informations, voir <u>Traiter plusieurs fichier d'entrée ou de sortie</u> <u>dynamiquement</u><sup>(83)</sup>.

## Touche Aperçu

Les composants B et C ont tout deux des touches d'aperçu. Cela vous permet de consulter dans MapForce le résultat de mappage intermédiaire de B, ainsi que le résultat final du mappage en chaîne. Cliquer sur la touche Aperçu du composant respectif, puis cliquer sur Sortie pour voir le résultat de mappage.

Les composants "Intermédiaires" dont la touche Passage est activée ne peuvent pas être consultés. Leur touche d'aperçu est automatiquement désactivée car il n'est pas approprié de consulter les données et de les laisser passer en même temps. Pour consulter la sortie d'un tel composant, cliquer tout d'abord sur la touche "passage" pour le désactiver, puis cliquer sur la touche d'aperçu.

## Touche Passage

Le composant intermédiaire B a une touche supplémentaire dans la barre de titre du composant appelée "passage".

Si la touche de passage est **active** 🗃 , MapForce mappera toutes les données dans la fenêtre d'aperçu en une seule fois ; du composant A au composant B, puis au composant C. Deux fichiers de résultat seront créés :

- le résultat du composant de mappage A vers le composant intermédiaire B
- le résultat du mappage du composant intermédiaire B, vers le composant cible C.

Si la touche passage est **inactive** 🔿 , MapForce exécutera uniquement des parties de la chaîne complète de mappage. Les données sont générées selon les touches d'aperçu qui ont été activées :

- Si la touche d'aperçu du composant B est active, le résultat du composant de mappage A vers le composant B est généré. La chaîne de mappage peut effectivement s'arrêter au composant B. Le composant C n'est pas impliqué dans l'aperçu.
- Si la touche d'aperçu du composant C est active, le résultat du composant de mappage intermédiaire B vers le composant C est généré. Étant donné que le passage n'est pas actif, la chaîne automatique a été interrompue pour le composant B. Seule la partie droite de la chaîne de mappage est exécutée. Le composant A n'est pas utilisé.

Lorsque la touche "passage" est désactivée, il est important que le composant intermédiaire ait des noms de fichier identiques dans les champs "Fichier XML d'entrée" et "Fichier XML de sortie". Cela permet de garantir que le fichier généré en tant que sortie lorsque vous consultez la portion du mappage située entre A et B est utilisé en tant qu'entrée lorsque vous consultez la portion du mappage située entre B et C. De plus, dans le code généré, ou dans l'exécution de MapForce Server, cela permet de garantir que la chaîne de mappage n'est pas rompue.

Comme mentionné précédemment, si le mappage est effectué par MapForce Server, ou par le code généré, la sortie de tous les composants est générée. Dans ce cas, les paramètres de la touche de passage du

composant B, et le composant d'aperçu actuellement sélectionné sont ignorés. Si vous prenez le mappage cidessus en tant qu'exemple, les deux fichiers de résultat seront générés comme suit :

- 1. Le fichier de sortie résultant du composant de mappage A à B
- 2. Le fichier de sortie résultant du composant de mappage B à C.

Les sections suivantes, <u>Exemple : Passage actif</u><sup>[843]</sup> et <u>Exemple : Passage non-actif</u><sup>[847]</sup>, illustrent de manière plus détaillée comment les données de source sont transférées différemment lorsque la touche de passage est active ou inactive.

## 7.1.1 Exemple : Passage actif

Le mappage utilisé dans cet exemple (ChainedReports.mfd) est disponible dans le dossier <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\. Ce mappage traite un fichier XML appelé ReportA.xml qui contient des notes de frais et ressemble à l'exemple affiché ci-dessous. Pour une plus grande simplicité, la déclaration d'espace de noms et quelques éléments expense-item ont été omis :

```
<?xml version="1.0" encoding="UTF-8"?>
<expense-report currency="USD" detailed="true">
  <Person>
      <First>Fred</First>
      <Last>Landis</Last>
      <Title>Project Manager</Title>
      <Phone>123-456-78</Phone>
      <Email>f.landis@nanonull.com</Email>
  </Person>
   <expense-item type="Travel" expto="Development">
      <Date>2003-01-02</Date>
      <Travel Trav-cost="337.88">
         <Destination/>
      </Travel>
      <description>Biz jet</description>
  </expense-item>
   <expense-item type="Lodging" expto="Sales">
      <Date>2003-01-01</Date>
      <Lodging Lodge-cost="121.2">
         <Location/>
      </Lodging>
      <description>Motel mania</description>
  </expense-item>
   <expense-item type="Travel" expto="Marketing">
      <Date>2003-02-02</Date>
      <Travel Trav-cost="2000">
         <Destination/>
      </Travel>
      <description>Hong Kong</description>
   </expense-item>
</expense-report>
```

ReportA.xml

L'objectif du mappage est de produire, sur la base du fichier ci-dessus, deux rapports supplémentaires :

- ReportB.xml ce rapport devrait uniquement contenir les notes de frais de type "Travel".
- ReportC.xml ce rapport devrait uniquement contenir les notes de frais de type "Travel" sans dépasser 1500.

Afin d'atteindre cet objectif, le composant intermédiaire du mappage (composant B) présente une touche passage 🗃 active, comme indiqué ci-dessous. Cela entraîne l'exécution du mappage en plusieurs étapes : de A à B, puis de B à C. La sortie créée par le composant intermédiaire sera utilisée en tant qu'entrée pour le mappage situé entre B et C.



Les noms des fichiers de sortie générés à chaque étape de la chaîne de mappage sont déterminés par les paramètres de chaque composant. (Pour ouvrir les paramètres de composant, cliquer dessus avec la touche de droite et sélectionner **Propriétés** depuis le menu contextuel). Concrètement, le premier composant est configuré pour lire des données depuis un fichier XML appelé **ReportA.xml**. Étant donné qu'il s'agit d'un composant de source, le champ **Fichier XML de sortie** n'est pas pertinent et n'a pas été rempli.

Nom de composant: ReportA		
Fichier de schéma		
ExpenseReport.xsd	Parcourir	Éditer
Fichier XML d'entrée		
ReportA.xml	Parcourir	Édition
Fichier XML de sortie		
	Parcourir	Éditer

Paramètres du composant de source

Comme indiqué ci-dessous, le deuxième composant (**ReportB**) est configuré pour créer un fichier de sortie appelé **ReportB.xml**. Veuillez noter que le champ **Fichier XML d'entrée** est grisé. Lorsque le Passage est actif (comme dans cet exemple), le champ **Fichier XML d'entrée** du composant intermédiaire est désactivé automatiquement. Le mappage ne nécessite pas de nom de fichier d'entrée pour s'exécuter, étant donné que la sortie créée à cette étape du mappage est stockée dans un fichier temporaire et est réutilisée plus loin dans le mappage. De même, si un **Fichier XML de sortie** est défini (comme illustré ci-dessous), il est utilisé pour le nom de fichier du fichier de sortie intermédiaire. Si aucun **Fichier XML de sortie** n'est défini, un nom de fichier par défaut sera utilisé automatiquement.

Nom de composant:	ReportB	]	
Fichier de schéma			
ExpenseReport.xsd		Parcourir	Éditer
Fichier XML d'entrée			
ReportB.xml		Parcourir	Édition
Fichier XML de sortie			
ReportB.xml		Parcourir	Éditer

Paramètres du composant intermédiaire

Enfin, le troisième composant est configuré pour produire un fichier de sortie appelé **ReportC.xml**. Le champ **Fichier XML d'entrée** n'est pas pertinent ici, parce qu'il s'agit d'un composant cible.

Nom de composant: ReportC		
Fichier de schéma		
ExpenseReport.xsd	Parcourir	Éditer
Fichier XML d'entrée	Parcourir	Édition
Fichier XML de sortie		
ReportC.xml	Parcourir	Éditer

Paramètres du composant cible

Si vous prévisualisez le mappage en cliquant sur l'onglet **Sortie** dans la fenêtre de mappage, deux fichiers sont montrés dans la sortie, comme prévu :

- 1. ReportB.xml, qui représente le résultat du mappage A à B
- 2. ReportC.xml, qui représente le résultat du mappage B à C.

Afin de choisir lequel des deux fichiers de sortie générés doit être affiché dans la fenêtre Sortie, cliquer soit sur les touches fléchées, soit choisir l'entrée désirée de la liste déroulante.

•	Preview 1 of 1 (2)     C:\Samples\ReportC.xml
1	xml version="1.0" er 📶 C:\Samples\ReportC.xml</td
2	expense-report xsi:n & Intermediate from component ReportB
2	http://www.w3.org/20
3	Person>
5	<  ast>  andis   ast
6	<title>Project Manager</title>
7	<phone>123-456-78</phone>
8	<email>f.landis@nanonull.com</email>
9	
10	<expense-item expto="Development" type="Travel"></expense-item>
11	<date>2003-01-02</date>
12	Travel Trav-cost="337.88">
13	<destination></destination>
14	
15	<description>Biz jet</description>
10	
10	expense-item type= Traver expto= Accounting > 
10	Content and the second seco
20	<pre></pre>
21	<t< td=""></t<>
22	<description>Ambassador class</description>
23	
24	

Fichiers de sortie générés

Lorsque le mappage est exécuté par MapForce, le paramètre "Écrire directement sur le fichier de sortie final" (configuré depuis **Outils | Options | Généralités**) détermine si les fichiers intermédiaires sont enregistrés en tant que fichiers temporaires ou en tant que fichiers physiques. Veuillez noter que cela est uniquement valide si le mappage est prévisualisé directement dans MapForce. Si ce mappage a été exécuté par MapForce Server ou par un code généré, des fichiers seraient produits à chaque étape dans la chaîne de mappage.

Si vous avez installé StyleVision, et si un fichier StyleVision Power Stylesheet (SPS) a été attribué au composant cible (comme dans cet exemple), alors la sortie de mappage finale peut être consultée et enregistrée en tant que fichier HTML, RTF. Pour générer et consulter cette sortie dans MapForce, cliquer sur l'onglet portant le nom correspondant.



Person	Personal Expense Report				Irrenc Deta	y: 💿 iled rej	Dollars O Euro port	s () 1	Yen Currency \$
Employe	ee Info	ormation							
	Fred First Nam	Lan ne Last	ndis Name		F	rojec	t Manager		
	f.landi: E-Mail	s@nanonull.	com		1 P	23-4: hone	56-78		
Expense	e List								
Туре	•	Expense T	0	Date (yyyy-mm- dd)		Expe	nses \$		Description
Travel	~	Developmen	t 🗸	2003-01- 02	<b>Tra</b> 337	avel 7.88	Lodging	Biz	jet
Travel	~	Accounting	~	2003-07- 07	<b>Tra</b> 101	avel .4.22	Lodging	Am	bassador class
Mapping	DB Quer	y Output	<b>⊡</b> H	TML ORTF	: 0	) PDF	Word 20	07+	

Sortie HTML générée

Veuillez noter que seule la sortie du composant de la cible finale dans la chaîne de mappage est affichée. Pour afficher une sortie StyleVision des composants intermédiaires, vous devez désactiver la touche de passage et prévisualiser le composant intermédiaire (comme affiché dans <u>Exemple : Passage inactif</u><sup>647</sup>).

## 7.1.2 Exemple : Passage inactif

Le mappage utilisé dans cet exemple (**ChainedReports.mfd**) est disponible dans le dossier **Documents-\Altova\MapForce2023\MapForceExamples\Tutorial\**. Cet exemple illustre les différentes manières dont la sortie est générée lorsque la touche de passage est désactivée dans le composant intermédiaire.



Comme indiqué dans <u>Exemple : Passage Actif</u> <sup>[433]</sup>, l'objectif du mappage est de produire deux rapports séparés. Dans l'exemple précédent, la touche de passage était activée , et les deux rapports ont été générés comme prévu et ont pu être consultés dans l'onglet **Sortie**. Néanmoins, si vous souhaitez ne consulter que l'un des rapports (soit **ReportB.xml** soit **ReportC.xml**), la touche de passage doit alors être désactivée (

- Ne consulter que la sortie générée de A à B, et ignorer la portion du mappage de B à C
- Ne consulter que la sortie générée de B à C, et ignorer la portion du mappage de A à B.

Lorsque vous désactivez la touche de passage comme indiqué ci-dessus, vous pouvez choisir de consulter soit **ReportB** soit **ReportC** (veuillez noter que les deux ont des touches d'aperçu <sup>(1)</sup>).

La désactivation de la touche de passage vous permet aussi de choisir quel fichier d'entrée doit être lu par le composant intermédiaire. Dans la plupart des cas, cela devrait être le même fichier que celui défini dans le champ **Fichier XML de sortie** (comme dans cet exemple).

Nom de composant:	ReportB		
Fichier de schéma			
ExpenseReport.xsd		Parcourir	Éditer
Fichier XML d'entrée			
ReportB.xml		Parcourir	Édition
Fichier XML de sortie			
ReportB.xml		Parcourir	Éditer

Paramètres du composant intermédiaire

Le fait d'avoir le même fichier d'entrée et de sortie sur le composant intermédiaire est particulièrement important si vous avez l'intention de générer du code depuis le mappage, et d'exécuter le mappage avec MapForce Server. Comme mentionné précédemment, dans ces environnements, toutes les sorties créées par chaque composant dans la chaîne de mappage sont générées. Il est donc plutôt logique que le composant intermédiaire reçoive un fichier pour le traitement (dans ce cas **ReportB.xml**) et envoie le même fichier au mappage suivant plutôt que de chercher un nom de fichier différent. Sachez que, si les noms du fichier d'entrée et de sortie ne sont pas les mêmes au niveau du composant intermédiaire (si la touche de passage est inactive) peut entraîner des erreurs comme "Le système ne peut pas trouver le fichier spécifié" dans du code généré ou dans l'exécution MapForce Server.

Si vous cliquez sur la touche d'aperçu 🔯 dans le troisième composant (**ReportC**), et que vous tentez de consulter le mappage dans MapForce, vous constaterez qu'une erreur d'exécution se produit. Cela est

prévisible puisque, conformément aux paramètres ci-dessus, un fichier appelé **ReportB.xml** est attendu en tant qu'entrée. Néanmoins, le mappage n'a pas encore produit un tel fichier (parce que la touche de passage n'est pas active et que seule la portion du mappage de B à C est exécutée). Vous pouvez aisément résoudre ce problème comme suit :

- 1. Cliquer sur la touche précédente sur le composant intermédiaire.
- 2. Cliquer sur l'onglet **Sortie** pour consulter le mappage.
- 3. Enregistrer le fichier de sortie résultant en tant que **ReportB.xml**, dans le même dossier que le mappage (cocuments>\Altova\MapForce2023\MapForceExamples\Tutorial\).

Maintenant, si vous cliquez à nouveau sur la touche d'aperçu sur le troisième composant (**ReportC**), l'erreur n'est plus affichée.

Lorsque la touche de passage est inactive, vous pouvez aussi consulter la sortie générée par StyleVision pour chaque composant présentant un fichier StyleVision Power StyleSheet (SPS) associé. En particulier, vous pouvez également consulter la version HTML du rapport intermédiaire (en plus de celle du rapport final) :



Persor	Personal Expense Report				Irrency:	Dollars O Euro report	s OYen Currency \$
Employe	ee Info	ormatio	n				
	Fred First Nan	ne	Landis Last Name		Proj Title	ect Manager	
	f.landi E-Mail	s@nanor	null.com		123 Phone	-456-78 •	
Expense	e List						
Тур	e	Expe	nse To	Date (yyyy-mm- dd)	E	xpenses \$	Description
Travel	~	Develop	ment	2003-01- 02	<b>Trave</b> 337.88	l Lodging	Biz jet
Travel	~	Account	ing 🔽	2003-07- 07	<b>Trave</b> 1014.2	Lodging	Ambassador class
Travel	~	Marketir	ng 🔽	2003-02- 02	<b>Trave</b> 2000	l Lodging	Hong Kong
Manaina	DR Ourse				0.00	NE O Ward ON	7.
mapping	DB Quer	y Out	pur join		OPL	or ovord 200	JI+

Sortie HTML du composant intermédiaire

# 7.2 Mapper noms de nœud

La plupart du temps, lorsque vous créez un mappage avec MapForce, l'objectif est de lire des *valeurs* depuis une source et d'écrire des *valeurs* dans une cible. Néanmoins, il peut y avoir des cas dans lesquels vous souhaitez accéder non seulement aux *valeurs* de nœud depuis la source, mais aussi aux noms de nœud. Par exemple, vous pouvez souhaiter créer un mappage qui écrit les noms d'élément ou d'attributs (pas des valeurs) depuis un XML de source et les convertit en des valeurs d'élément ou d'attribut (pas de noms) dans un XML cible.

Prenons l'exemple suivant : vous avez un fichier XML qui contient une liste de produits. Chaque produit possède le format suivant :

```
<product>
    <id>l</id>
    <color>red</color>
      <size>10</size>
</product>
```

Votre objectif est de convertir l'information concernant chaque produit dans des paires nom-valeur, par exemple

```
<product>
    <attribute name="id" value="1" />
    <attribute name="color" value="red" />
    <attribute name="size" value="10" />
</product>
```

Pour ce type de scénario, vous souhaitez accéder au nom de nœud depuis le mappage. Avec l'accès dynamique aux noms de nœud, vous pouvez effectuer des conversions de données comme celle présentées cidessus.

**Note :** Vous pouvez aussi effectuer la transformation ci-dessus en utilisant les fonctions de bibliothèque principale <u>node-name</u> et <u>static-node-name</u> 3. Néanmoins, dans ce cas, vous devrez savoir exactement quels noms d'élément vous attendez de la source, et vous devrez connecter chaque élément manuellement vers la cible. Néanmoins, ces fonctions peuvent ne pas être suffisantes, par exemple si vous souhaitez filtrer ou regrouper des nœuds par leur nom, ou si vous souhaitez manipuler le type de données du nœud depuis le mappage.

L'accès à des noms de nœud dynamiquement est possible non seulement lorsque vous souhaitez lire des noms de nœud, mais aussi lorsque vous souhaitez les écrire. Dans un mappage standard, le nom des attributs ou des éléments dans une cible est toujours connu avant que le mappage soit exécuté ; il provient du schéma sous-jacent du composant. Avec des noms de nœud dynamiques, toutefois, vous pouvez créer de nouveaux attributs ou éléments dont le nom n'est pas connu avant l'exécution du mappage. En particulier, le nom de l'attribut ou de l'élément est fourni par le mappage lui-même, depuis toute source prise en charge par MapForce.

Pour que l'accès dynamique aux éléments ou attributs enfants d'un nœud soit possible, le nœud doit effectivement avoir des éléments ou des attributs enfant et ce ne doit pas être le nœud racine XML.

Les noms de nœud dynamiques sont pris en charge lorsque vous mappez depuis ou vers les types de composant suivants :

- XML
- CSV/FLF\*

\* Nécessite l'édition MapForce Professional ou Enterprise.

**Note :** Dans le cas de CSV/FLF, l'accès dynamique implique l'accès à "fields" au lieu de "nodes", puisque les structures CSV/FLF n'ont pas de "nœuds".

Lorsque la cible de mappage est un fichier CSV ou FLF (champ de longueur fixe), les champs doivent être définis dans les paramètres de composant (et il n'est pas possible de changer le nom, l'ordre ou le nombre des champs de cible). Contrairement à XML, le format des fichiers de texte est fixe, donc seule la valeur de champ peut être manipulée, et pas le nom de champ, le nombre ou l'ordre.

Les noms de nœud dynamiques sont pris en charge dans un des langages de mappage suivants : Built-In\*, XSLT2,XSLT3, XQuery\*, C#\*, C++\*, Java\*.

\* Ces langages nécessitent l'édition MapForce Professional ou Enterprise.

Pour plus d'information concernant le fonctionnement des noms de nœud dynamiques, voir <u>Obtenir l'accès aux</u> <u>noms de nœud</u><sup>652</sup>. Pour un exemple de mappage étape par étape, voir <u>Exemple : Mapper des noms d'élément</u> <u>vers les valeur d'attribut</u><sup>664</sup>.

## 7.2.1 Obtenir l'accès aux noms de nœud

Lorsqu'un nœud dans un composant XML (ou un champ dans un composant CSV/FLF) a des nœuds enfants, vous pouvez obtenir le nom et la valeur de chaque nœud enfant directement dans le mappage. Cette technique est appelée "noms de nœud dynamiques". "Dynamique" se réfère au fait que le traitement a lieu "on the fly" (immédiatement), pendant l'exécution du mappage et n'est pas basé sur l'information de schéma statique qui est connu avant l'exécution du mappage. Cette rubrique présente des détails pour permettre l'accès dynamique aux noms de nœud et montre ce que vous pouvez en faire.

Lorsque vous lisez des données provenant d'une source, "noms de nœud dynamiques" signifie que vous pouvez faire les opérations suivantes :

- Obtenir une liste de tous les nœuds enfants (ou attributs) d'un nœud, en tant que séquence. Dans MapForce, "séquence" est une liste de zéro items ou plus que vous pouvez connecter à une cible et créer autant d'items dans la cible qu'il y a d'items dans la source. Ainsi, par exemple, si un nœud a cinq attributs dans la source, vous pouvez créer cinq nouveaux éléments dans la cible, chacun correspondant à un attribut.
- Lire non seulement les valeurs de nœud enfants (comme un mappage standard le fait), mais aussi leurs noms.

Lorsque vous écrivez des données dans une cible, "noms de nœud dynamiques" signifie que vous pouvez effectuer les opérations suivantes :

• Créer de nouveaux nœuds en utilisant des noms fournis par le mappage (des noms soit-disant "dynamiques"), par opposition aux noms fournis par les paramètres de composant (des noms soitdisant "static").

Afin d'illustrer les noms de nœud dynamiques, cette rubrique utilise le schéma XML suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Products.xsd. Ce schéma est accompagné par un document d'instance échantillon, Products.xml. Pour ajouter le schéma et le fichier d'instance à la zone de mappage, choisir la commande de menu Insérer | Schéma XML /Fichier et chercher <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Products.xml. Lorsque vous serez invité à choisir un élément racine, choisir products.

Pour activer les noms de nœud dynamiques pour le nœud product, cliquer dessus avec la touche de droite et choisir parmi une des commandes du menu contextuel suivantes :

- Afficher les attributs avec un Nom Dynamique, si vous souhaitez obtenir l'accès aux attributs du nœud
- Afficher les éléments enfants avec un Nom Dynamique, si vous souhaitez obtenir l'accès aux éléments enfants du nœud

😫 Produc	ts.	
📮 🗋 Fichi	ier: P	roducts.xml Fichier/Chaîne D
¢ ⊡<>pr	oduct	is D
	<b>*</b>	Ajouter double entrée avant Ajouter doublon après
╞└┈╲	₿₽ ₽	Supprimer doublon
		Déplacer vers le haut
		Déplacer <u>v</u> ers le bas
		<u>C</u> ommentaire/Instruction de traitement
	1	Ajouter espace de <u>n</u> oms
		Afficher Attributs avec nom dynamique
		Afficher Éléments enfant générigue avec nom dynamique
		Écr <u>i</u> re le contenu en tant que section CDATA
		Supprimer les co <u>n</u> nexions
		Créer composant varia <u>b</u> le
		Fonctions nœud et défauts
		Co <u>m</u> posant

Fig. 1 Activer des noms de nœud dynamiques (pour les éléments enfant)

**Note :** Les commandes ci-dessus sont disponibles uniquement pour les nœuds qui ont des nœuds enfants. De même, les commandes ne sont pas disponibles pour les nœuds de racine.

Lorsque vous faites passer un nœud dans le mode dynamique, un dialogue comme celui ci-dessous apparaît. Définir les options comme indiqué ci-dessous ; ces options seront présentées plus en détail dans <u>Accéder les</u> <u>nœuds de type spécifique</u><sup>(80)</sup>.

😢 Paramètres enfants nommés dynamiquement	×
Utiliser ces paramètres pour configurer l'accès aux éléments enfants en utilisant une	
subcure generique. Sélectiones los trans pour poséder ou contonu de l'item pour é duponiquement :	
Selectionner les types pour acceder au contenu de intem nomme dynamiquement :	- 1
L 마음 xs:anyType	$\sim$
✓ □봄 text()	
	$\sim$
La sélection d'un type à cet endroit rendra sa structure disponible uniquement pour le	
mappage. La trop actuel plant pas coché lors de la marche	
Afficher les nœuds de test pour filtrer ou créer des éléments par nom de nœud fixé	
Afficher les éléments enfants du schéma de l'élément parent	
OK Annule	r

Fig. 2 Dialogue "Paramètres d'Enfants nommés dynamiquement"

Fig. 3 Montre à quoi ressemble le composant lorsque les noms de nœud dynamiques sont activés pour le nœud product. Veuillez noter que l'apparence du composant a considérablement changé.



Fig.3 Noms de nœud dynamiques activés (pour les éléments)

Pour faire retourner le composant dans le mode standard, cliquer avec la touche de droite sur le nœud product, et désactiver l'option Afficher les Éléments enfants avec le nom dynamique depuis le menu contextuel.

L'image ci-dessous montre à quoi ressemble le même composant lorsque l'accès dynamique aux attributs d'un nœud est activé. Le composant a été obtenu en cliquant avec la touche de droite sur l'élément product, et en sélectionnant Afficher les attributs avec un Nom dynamique depuis le menu contextuel.



Fig. 4 Noms de nœud dynamiques activés (pour les attributs)

Pour replacer le composant dans le mode standard, cliquer avec la touche de droite dans le mode product, et désactiver l'option Afficher les attributs avec un Nom dynamique depuis le menu contextuel.

Comme illustré dans les Fig. 3 et Fig. 4, l'apparence du composant change lorsqu'un nœud (dans ce cas, product) est placé en mode "nom de nœud dynamique". La nouvelle apparence ouvre des possibilités pour les actions suivantes :

- Lire ou écrire une liste de tous les éléments ou attributs enfants d'un nœud. Ils sont fournis par l'item element() ou attribute(), respectivement.
- Lire ou écrire le nom de chaque élément ou attribut enfants. Le nom est fourni par les items nodename() et local-name().
- Dans le cas des éléments, lire ou écrire la valeur de chaque élément enfant, en tant que type de données spécifique. Cette valeur est fournie par le nœud de type cast (dans ce cas, l'item text()).
   Veuillez noter que seuls des éléments peuvent avoir des nœuds type cast. Les attributs sont toujours traités en tant que type "string".
- Regrouper ou filtrer les éléments enfants par nom. Par exemple, voir <u>Exemple : Regrouper et filtrer les</u> nœuds par nom<sup>867</sup>.

Les types de nœud avec lesquels vous pouvez travailler dans le mode "nom de nœud dynamique" sont décrits ci-dessous.

## element()

Ce nœud a un comportement différent dans un composant de source comparé à un composant cible. Dans un composant de source, il fournit les éléments enfant du nœud en tant que séquence. Dans la Fig.3, element() fournit une liste (séquence) de tous les éléments enfants de product. Par exemple, la séquence créée depuis le XML suivant contiendra trois items (puisqu'il y a trois éléments enfant de product) :

```
cproduct>
    <id>l</id>
    <color>red</color>
    <size>10</size>
</product>
```

Veuillez noter que le nom et le type de chaque item dans la séquence est fourni par le nœud node-name() et le nœud type cast, respectivement (discuté ci-dessous). Pour comprendre cela, imaginez que vous souhaitez transformer des données depuis un XML de source dans un XML cible comme suit :



Fig. 6 Mapper des noms d'élément XML dans les valeurs d'attribut (exigence)

Le mappage qui atteindrait cet objectif comme suit :



Fig. 7 Mapper des noms d'élément XML dans les valeurs d'attribut (dans MapForce)

Le rôle d'element() ici est de fournir la séquence des éléments enfants de product, alors que node-name() et text() fournissent le nom et la valeur de chaque item dans la séquence. Ce mappage est accompagné par un échantillon de tutoriel et est discuté plus en détail dans <u>Exemple : Mapper noms d'élément dans les valeurs</u> d'attributs<sup>864</sup>.

Dans un composant cible, element() ne crée rien par lui-même, ce qui est une exception à la règle de base du mappage "pour chaque item dans la source, créer un item cible". Les éléments sont créés par les nœuds type cast (en utilisant la valeur de node-name()) et par les nœuds de test de nom (utilisant leur propre nom).

## attribute()

Comme indiqué dans la Fig. 4, cet item permet l'accès à tous les attributs du nœud, à l'exécution du mappage. Dans un composant de source, il fournit les attributs du nœud de source connecté, en tant que séquence. Par exemple, dans le XML suivant, la séquence contiendrait deux items (puisque product a deux attributs) :

product id="1" color="red" />

Veuillez noter que le nœud attribute() fournit uniquement la valeur de chaque attribut dans la séquence, toujours en tant que type string. Le nom de chaque attribut est fourni par le nœud node-name().

Dans un composant cible, ce nœud traite une séquence connectée et crée une valeur d'attribut pour chaque item dans la séquence. Le nom d'attribut est fourni par node-name(). Par exemple, imaginer que vous souhaitez transformer des données depuis un XML de source dans un XML cible comme suit :

1 2 3 4 5 6	<pre><?xml version="1.0" encoding="UTF-8"?> </pre> <pre>cyroducts xsi:noNamespaceSchemaLocation="// </pre> <pre></pre> <pre><!--</th--><th></th><th></th><th></th></pre>			
8	<pre></pre>		1	xml version="1.0" encoding="UTF-8"?
9	<attribute name="id" value="2"></attribute>		2	<pre>products xsi:noNamespaceSchemaLocation="//\"</pre>
10	<attribute name="color" value="blue"></attribute>		3	<product color="red" id="1" size="10"></product>
11	<attribute name="size" value="20"></attribute>		4	<product color="blue" id="2" size="20"></product>
12	-		5	<product color="green" id="3" size="30"></product>
13		<b>9</b>	6	L
14	<attribute name="id" value="3"></attribute>			
15	<attribute name="color" value="green"></attribute>			
16	<attribute name="size" value="30"></attribute>			
17	-			
18	<pre>L </pre>			

Fig. 8 Mapper des valeurs d'attribut pour attribuer des noms (exigence)

Le mappage qui permet d'atteindre cet objectif ressemble à ce qui suit :



Fig. 9 Mapper des valeurs d'attribut pour attribuer des noms (dans MapForce)

**Note :** Cette transformation est également possible sans activer un accès dynamique dans des attributs d'un nœud. Ici, elle illustre comment attribute() fonctionne dans un composant cible.

Si vous souhaitez reconstruire ce mappage, il utilise les mêmes composants XML que le mappage **ConvertProducts.mfd** disponible dans le dossier

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\. La seule différence est que la cible est maintenant devenue la source, et que la source est devenue la cible. En guise de données d'entrée pour le composant source, vous nécessiterez une instance XML qui contient effectivement des valeurs d'attribut, par exemple :

</product> </products>

Veuillez noter que, dans l'extrait de code, l'espace de noms et la déclaration de schéma ont été omis, pour des raisons de simplicité.

#### node-name()

Dans un composant source, node-name() fournit le nom de chaque élément enfant de element(), ou le nom de chaque attribute(), respectivement. Par défaut, le nom fourni est de type xs:QName. Pour obtenir le nom en tant que string, utiliser le nœud local-name() (voir Fig. 3), ou utiliser la fonction <u>QName-as-</u> string<sup>759</sup>.

Dans un composant cible, node-name() écrit le nom de chaque élément ou attribut contenu dans element() ou attribute().

#### local-name()

Ce nœud fonctionne de la même manière que node-name(), à la différence que le type est xs:string au lieu de xs:QName.

#### Nœud type cast

Dans un composant source, le nœud type cast fournit la valeur de chaque élément enfant contenu dans element(). Le nom et la structure de ce nœud dépendent du type sélectionné depuis le dialogue "Paramètres Enfants nommés dynamiquement" (Fig. 2).

Pour changer le type du nœud, cliquer sur la touche **Changer sélection** ( ) et choisir un type depuis la liste des types disponibles, y compris un caractère générique de schéma (xs:any). Pour plus d'informations, voir <u>Accéder aux nœuds de type spécifique</u>.

Dans un composant cible, le nœud type cast écrit la valeur de chaque élément enfant contenu dans element(), en tant que type de données spécifique. De même, le type de données désiré peut être sélectionné en cliquant sur la touche **Changer sélection** ( ).

#### Nœuds de test de nom

Dans un composant de source, les nœuds de test de nom proposent un moyen pour regrouper ou filtrer les éléments enfants depuis une instance de source par nom. Vous pouvez souhaitez filtrer les éléments enfants par les noms pour vous assurer que le mappage accède aux données d'instance en utilisant le type correct (voir <u>Accéder aux nœuds de type spécifique</u><sup>(360)</sup>). Pour consulter un exemple, voir <u>Exemple : Regrouper et filtrer les nœuds par nom</u><sup>(867)</sup>.

En général, les nœuds de test de nom fonctionnent presque comme des nœuds d'élément normaux pour lire et écrire des valeurs et des structures sous-arborescentes. Néanmoins, étant donné que la sémantique de mappage est différente lorsque l'accès dynamique est activé, il existe des limites. Par exemple, vous ne pouvez pas concaténer la valeur de deux nœuds de test de nom.

Du côté cible, les nœuds de test de nom créent autant d'éléments dans la sortie qu'il y a d'items dans la séquence de source connectée. Leur nom contourne la valeur mappée dans node-name().

Si nécessaire, vous pouvez dissimuler les nœuds de test de nom depuis le composant. Pour ce faire, cliquer sur la touche **Changer la sélection** ( ) à côté du nœud element(). Ensuite, dans le dialogue "*Paramètres Enfants nommés dynamiquement*" (Fig. 2), décocher la case **Afficher les nœuds de test de nom...** 

# 7.2.2 Accéder aux nœuds de type spécifique

Comme mentionné dans la section précédente, <u>Obtenir l'accès aux noms de nœud</u><sup>(622)</sup>, vous pouvez accéder à tous les éléments enfants d'un nœud en cliquant avec la touche de droite et sélectionner la commande de menu contextuel **Afficher les éléments enfants avec un Nom Dynamique**. Au moment de l'exécution de mappage, cela entraîne une accessibilité du nom de chaque élément enfant par le biais du nœud nodename(), alors que la valeur est accessible par le biais d'un nœud type cast spécial. Dans l'image ci-dessous, le nœud type cast est le nœud text().



Attention, le type de données de chaque élément enfant n'est pas connu avant l'exécution de mappage. Du reste, il peut être différent pour chaque élément enfant. Par exemple, un nœud product dans le fichier d'instance XML peut avoir un élément enfant id de type xs:integer et un élément enfant size de type xs:string. Afin de vous permettre d'accéder au contenu de nœud d'un type spécifique, le dialogue affiché cidessous s'ouvre à chaque fois que vous activez l'accès dynamique dans des éléments enfant d'un nœud. Vous pouvez aussi ouvrir ce dialogue à tout moment ultérieurement, en cliquant sur la touche **Changer sélection** (

😤 Paramètres enfants nommés dynamiquement	×
Utiliser ces paramètres pour configurer l'accès aux éléments enfants en utilisant une structure générique.	
Sélectionner les types pour accéder au contenu de l'item nommé dynamiquement :	
<ul> <li>□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □</li></ul>	^
	~
, La sélection d'un type à cet endroit rendra sa structure disponible uniquement pour le mappage. Le type actuel n'est pas coché lors de la marche.	
Afficher les nœuds de test pour filtrer ou créer des éléments par nom de nœud fixé	
✓ Afficher les éléments enfants du schéma de l'élément parent	
OK Annule	r

Dialogue "Paramètres d'Enfants nommés dynamiquement"

Pour accéder au contenu de chaque élément enfant au moment d'exécution de mappage, vous disposez de plusieurs options :

- Accéder au contenu en tant que string. Pour ce faire, sélectionner la case à cocher text() dans le dialogue ci-dessus. Dans ce cas, un nœud text() est créé dans le composant lorsque vous refermez le dialogue. Cette option est appropriée si le contenu est de type simple (xs:int, xs:string, etc.) et est illustré dans l'<u>Exemple</u>: Mapper les noms d'élément dans les valeurs d'attribut<sup>664</sup>. Veuillez noter qu'un nœud text() est uniquement affiché si un nœud enfant du nœud actuel peut contenir du texte.
- 2. Accéder au contenu en tant qu'un type complexe particulier autorisé par le schéma. Lorsque des types complexes personnalisés définis globalement sont autorisés par le schéma pour le nœud sélectionné, ils sont aussi disponibles dans le dialogue ci-dessus, et vous pouvez cocher la case adjacente. Dans l'image ci-dessus, il n'y a pas de types complexes définis globalement par le schéma, pour qu'aucun ne soit disponible pour la sélection.
- Accéder au contenu en tant que type any. Cela peut être utile dans des scénarios dans le mappage avancé (voir "Accéder aux structures profondes" ci-dessous). Pour ce faire, sélectionner la case à cocher située à côté de xs:anyType.

Veuillez noter que lors de l'exécution de mappage, MapForce (par le biais du nœud type cast) n'a pas d'informations par rapport à ce qu'est le type réel du nœud d'instance. C'est pourquoi, votre mappage doit accéder au contenu de nœud en utilisant le type correct. Par exemple, si vous prévoyez que le nœud

d'une instance XML de source peut avoir des nœuds enfants de plusieurs types complexes, procéder comme suit :

a) Définir le nœud type cast pour être le type complexe que vous souhaitez faire correspondre (voir item 2 dans la liste ci-dessus).

b) Ajouter un filtre à lire depuis l'instance uniquement par le type complexe que vous souhaitez faire correspondre. Cette technique est illustrée dans <u>Exemple : Regrouper et filtrer les nœuds par nom</u><sup>867</sup>.

#### Accéder aux structures plus profondes

Il est possible d'accéder à des nœuds à des niveaux plus profonds dans le schéma qu'au niveau des enfants immédiats d'un nœud. Cela est utile dans les scénarios de mappage avancés. Dans des mappages simples comme Exemple : Mapper les noms d'élément sur des valeurs d'attribut <sup>(BCI)</sup>, vous n'aurez pas besoin de cette technique car le mappage accède uniquement aux enfants immédiats d'un nœud XML. Néanmoins, si vous souhaitez accéder dynamiquement aux structures plus profondes comme les "petits-enfants", les "arrière-petits-enfants", etc. suivez les instructions ci-dessous :

- 1. Créer un nouveau mappage.
- Dans le menu Insérer, cliquer sur Insérer Schéma XML/Fichier et chercher le fichier d'instance XML (dans cet exemple, le fichier Articles.xml provenant du dossier <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\).
- 3. Cliquer avec la touche de droite sur le nœud Articles et choisir la commande contextuelle Afficher les éléments enfant avec un nom dynamique.
- 4. Choisir xs:anyType depuis le dialogue "Paramètres Enfants nommés dynamiquement".
- 5. Cliquer avec la touche de droite sur le nœud xs:anyType et choisir à nouveau la commande contextuelle Afficher les éléments enfant avec un nom dynamique.
- 6. Choisir text() depuis le dialogue "Paramètres Enfants nommés dynamiquement".



Dans le composant ci-dessus, veuillez noter qu'il y a deux nœuds <code>element()</code>. Le second nœud <code>element()</code> propose un accès dynamique aux petits-enfants de l'instance <Articles> Articles> Articles.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xsi:noNamespaceSchemaLocation="Articles.xsd">
   <Article>
      <Number>1</Number>
      <Name>T-Shirt</Name>
      <SinglePrice>25</SinglePrice>
   </Article>
   <Article>
      <Number>2</Number>
      <Name>Socks</Name>
      <SinglePrice>2.30</SinglePrice>
   </Article>
   <Article>
      <Number>3</Number>
      <Name>Pants</Name>
      <SinglePrice>34</SinglePrice>
   </Article>
   <Article>
      <Number>4</Number>
      <Name>Jacket</Name>
      <SinglePrice>57.50</SinglePrice>
   </Article>
</Articles>
```

#### Articles.xml

Par exemple, pour obtenir les noms d'élément "petits-enfants" (Number, Name, SinglePrice), vous pouvez établir une connexion depuis le nœud local-name()sous le deuxième nœud element() vers un item cible. De même, pour obtenir des valeurs d'élément "petits-enfants" (1, T-Shirt, 25), vous pouvez établir une connexion depuis le nœud text().

Bien que cela ne s'applique pas à cet exemple, dans des situations réelles, vous pouvez aussi activer des noms de nœud dynamiques pour tout nœud xs:anyType ultérieur, pour atteindre des niveaux plus profonds.

Veuillez noter les points suivants :

- La touche <u>TYPE</u> vous permet de choisir tout type dérivé depuis le schéma actuel et de l'afficher dans un nœud séparé. Cela peut uniquement être utile si vous souhaitez mapper vers ou depuis des types de schéma dérivés (voir <u>Types de Schéma XML dérivé</u><sup>[28]</sup>).
- La touche Changer sélection ( ) située à côté d'un nœud element() ouvre le dialogue "Paramètres Enfants nommés dynamiquement" discuté dans cette rubrique.
- La touche Changer sélection ( ) située à côté de xs:anyAttribute vous permet de choisir tout attribut défini globalement dans le schéma. De même, la touche Changer sélection ( ) située à côté de xs:any permet de choisir tout élément défini globalement dans le schéma. Cela fonctionne de la même manière que le fait de mapper depuis ou vers des caractères génériques de schéma (voir aussi <u>Caractères génériques xs:any / xs:anyAttribute</u> ). Si vous utilisez cette option, assurez-vous que l'attribut ou l'élément sélectionné peut réellement exister à ce niveau spécifique conformément au schéma.

## 7.2.3 Exemple : Mapper les noms d'élément dans les valeurs d'attribut

Cet exemple vous montre comment mapper des noms d'élément depuis un document XML pour attribuer des valeurs dans un document XML cible. L'exemple s'accompagne d'un mappage d'échantillon, qui est disponible dans le chemin suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\ConvertProducts.mfd.

Afin de comprendre ce que fait l'exemple, partons du principe que vous avez un fichier XML qui contient une liste de produits. Chaque produit comporte le format suivant :

```
<product>
<id>l</id>
<color>red</color>
<size>10</size>
</product>
```

Votre objectif est de convertir l'information concernant chaque produit en des paires nom-valeur, par exemple :

```
cattribute name="id" value="1" />
        <attribute name="color" value="red" />
        <attribute name="size" value="10" />
</product>
```

Pour effectuer un mappage de données comme celui présenté ci-dessus avec un effort minimum, cet exemple utilise une fonction MapForce appelée "Accès dynamique aux noms de nœud". "Dynamique" signifie ici que, lorsque le mappage est exécuté, il peut lire les noms de nœud (pas uniquement les valeurs) et utilise ces noms en tant que valeurs. Vous pouvez créer le mappage requis en quelques étapes, voir ci-dessous.

## Étape 1: Ajouter le composant XML source au mappage

 Dans le menu Insérer, cliquer sur Schéma XML/Fichier, et chercher le fichier suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Products.xml. Ce fichier XML pointe vers le schéma Products.xsd situé dans le même dossier.

## Étape 2: Ajouter le composant XML cible au mappage

 Dans le menu Insérer, cliquer sur Schéma XML/Fichier, et chercher le fichier suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\ProductValuePairs.xsd.
 Lorsque vous êtes invité à fournir un fichier d'instance, cliquer sur Sauter. Lorsque vous êtes invité à choisir un élément racine, choisir products en tant qu'élément racine.

Á ce niveau, le mappage devrait ressembler à l'exemple suivant :
	products	
Θĺ	File: Products.xml File/String	Ę
¢ E	Products	ĥ
¢ !	🕀 🔇 product	ĥ
¢	() id	Ę
¢	() color	ĥ
¢	() size	Ę

E ProductValuePairs	
📮 🗍 File: ProductValuePairs.xml	File/String [
products	C
🔁 🔄 🔿 product	C
🔁 🕞 🔿 attribute	C
aname	C
value	C

### Étape 3: Activer l'accès dynamique aux nœuds enfants

- 1. Cliquer avec la touche de droite sur le nœud product dans le composant de source, et choisir Afficher les éléments enfants avec un Nom Dynamique depuis le menu contextuel.
- 2. Dans le dialogue qui s'ouvre, choisir text() en tant que type. Ne pas toucher aux autres options.

🙄 Paramètres enfants nommés dynamiquement	×	
Utiliser ces paramètres pour configurer l'accès aux éléments enfants en utilisant une structure générique.		
Sélectionner les types pour accéder au contenu de l'item nommé dynamiquement :		
🗆 唱 xs:anyType	~	
▼ □-E text()	<	
La sélection d'un type à cet endroit rendra sa structure disponible uniquement pour le mappage. Le type actuel n'est pas coché lors de la marche.		
Afficher les nœuds de test pour filtrer ou créer des éléments par nom de nœud fixé		
Afficher les éléments enfants du schéma de l'élément parent		
OK Annule	r	

Veuillez noter qu'un nœud text() a été ajouté au composant de source. Ce nœud fournira le contenu de chaque item enfant au mappage (dans ce cas, la valeur de "id", "color", et "size").



### Étape 4: Tracer les connexions de mappage

Enfin, tracer les connexions de mappage A, B, C, D comme illustré ci-dessous. En option, double-cliquer sur chaque connexion, en commençant par la connexion supérieure et en saisissant le texte "A", "B", "C", et "D", respectivement, dans le champ Description.



#### ConvertProducts.mfd

Dans le mappage illustré ci-dessus, la connexion A crée, pour chaque produit dans la source, un produit dans la cible. Jusqu'à présent, il s'agit d'une connexion MapForce standard qui n'adresse les noms de nœud d'aucune manière. La connexion B, néanmoins, crée pour chaque élément enfant de product rencontré, un nouvel élément dans la cible, appelé attribute.

La connexion B est une connexion cruciale dans le mappage. Nous rappelons ici l'objectif de cette connexion : elle porte une *séquence* des éléments enfants de product depuis la source vers la cible. Elle ne porte pas les *noms* ou les *valeurs*. Ainsi : si l'**element()** source a N éléments enfant, créer N instances de cet item dans la cible. Dans ce cas particulier, product dans la source a trois éléments enfant (id, color et size). Cela signifie que chaque product dans la cible aura trois éléments enfants portant le nom attribute.

Bien que non illustré dans cet exemple, la même règle est utilisée pour mapper les éléments enfant de **attribute()**: si l'item de source **attribute()** a N attributs enfants, créer N instances de cet item dans la cible.

Ensuite, la connexion C copie le nom de chaque élément enfant de product vers la cible (littéralement, "id", "color", et "size").

Enfin, la connexion D copie la valeur de chaque élément enfant du produit, en tant que type de string, dans la cible.

Pour consulter la sortie de mappage, cliquer sur l'onglet **Sortie** et observer le XML généré. Comme prévu, la sortie contient plusieurs produits dont les données sont stockées en tant que paires nom-valeur, ce qui était l'objectif de ce mappage.

```
<?xml version="1.0" encoding="UTF-8"?>
products xsi:noNamespaceSchemaLocation="ProductValuePairs.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <product>
      <attribute name="id" value="1"/>
      <attribute name="color" value="red"/>
      <attribute name="size" value="10"/>
   </product>
   <product>
      <attribute name="id" value="2"/>
      <attribute name="color" value="blue"/>
      <attribute name="size" value="20"/>
   </product>
   <product>
      <attribute name="id" value="3"/>
      <attribute name="color" value="green"/>
      <attribute name="size" value="30"/>
   </product>
</products>
```

Sortie de mappage générée

### 7.2.4 Exemple: Regrouper et filtrer les nœuds par nom

Cet exemple vous montre comment concevoir un mappage qui lit des paires clé-valeur depuis une liste de propriété XML (ou plist XML) et les écrit dans un fichier CSV. (Les listes de propriété XML représentent un moyen de stocker des informations d'objet macOS et iOS dans un format XML, voir <a href="https://developer.apple.com/library/mac/documentation/Coccoa/Conceptual/PropertyLists/UnderstandXMLPlist/U">https://developer.apple.com/library/mac/documentation/Coccoa/Conceptual/PropertyLists/UnderstandXMLPlist/U</a> <a href="https://developer.apple.com/library/mac/documentation/coccoa/Conceptual/PropertyLists/UnderstandXMLPlist/U">https://developer.apple.com/library/mac/documentation/Coccoa/Conceptual/PropertyLists/UnderstandXMLPlist/U</a> <a href="https://developer.apple.com/library/mac/documentation/coccoa/conceptual/PropertyLists/UnderstandXMLPlist/U">https://developer.apple.com/library/mac/documentation/Coccoa/Conceptual/PropertyLists/UnderstandXMLPlist/U</a> <a href="https://developer.apple.com/library/mac/documentation/coccoa/conceptual/PropertyLists/UnderstandXMLPlist/U">https://developer.apple.com/library/mac/documentation/coccoa/conceptual/PropertyLists/UnderstandXMLPlist/U</a> <a href="https://developer.apple.com/library/mac/documentation/coccoa/conceptual/PropertyLists/UnderstandXMLPlist/U">https://developer.apple.com/library/mac/documentation/coccoa/conceptual/PropertyLists/UnderstandXMLPlist/U</a> <a href="https://documentation.coccoa/conceptual/PropertyLists/U">https://documentation/coccoa/conceptual/PropertyLists/U</a> <a href="https://documentation.coccoa/conceptual/P">https://documentation/coccoa/conceptual/P</a> <a href="https://documentation.coccoa/conceptual/P">https://documentation/coccoa/conceptual/P</a> <a href="https://documentation.coccoa/conceptual/P">https://documentation/coccoa/conceptual/P</a> <a href="https://documentation.coccoa/conceptual/P">https://documentation/coccoa/conceptual/P</a> <a href="https://documentation.coccoa/conceptual/P">https://documentation/coccoa/conceptual/P</a>

#### <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\ReadPropertyList.mfd.

Le listage de code ci-dessous représente le fichier XML de source.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist SYSTEM "https://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
      <key>First Name</key>
      <string>William</string>
      <key>Last Name</key>
      <string>Shakespeare</string>
      <key>Birthdate</key>
      <integer>1564</integer>
      <key>Profession</key>
      <string>Playwright</string>
      <key>Lines</key>
      <array>
         <string>It is a tale told by an idiot,</string>
         <string>Full of sound and fury, signifying nothing.</string>
      </arrav>
   </dict>
</plist>
```

L'objectif du mappage est de créer une nouvelle ligne dans le fichier CSV depuis certaines paires clé-valeur trouvée sous le nœud <dict> dans le fichier de liste de propriété. En particulier, le mappage doit uniquement filtrer des paires <key> - <string>. Les autres paires clé-valeurs (par exemple, <key> - <integer>) doivent être ignorées. Dans le fichier CSV, la ligne doit stocker le nom de la propriété, séparée de la valeur de la propriété par une virgule. Autrement dit, la sortie doit ressembler à l'exemple suivant :

Prénom,William Nom de famille,Shakespeare Profession,Playwright

Pour atteindre cet objectif, le mappage utilise un accès dynamique à tous les nœuds enfants du nœud dict. Ensuite, le mappage utilise la fonction <u>group-starting-with</u> pour regrouper les paires clé-valeur extraites du fichier XML. Enfin, le mappage utilise un filtre pour ne filtrer que les nœuds dont le nom de nœud est "string".

Les étapes suivantes montrent comment le mappage requis peut être créé.

#### Étape 1 : Ajouter le composant XML source au mappage

- 1. Le langage de transformation de mappage est configuré sur <u>BUILT-IN</u><sup>26</sup>.
- Dans le menu Insérer, cliquez sur Schéma XML/Fichier, et chercher le fichier suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\plist.xml. Ce fichier XML pointe vers le schéma plist.xml situé dans le même dossier.

#### Étape 2 : Ajouter le composant CSV cible au mappage

- 1. Dans le menu Insérer, cliquez sur Fichier texte. Sur invitation, choisir l'option Utiliser un traitement simple pour les fichiers CSV standard....
- 2. Ajouter un champ CSV dans le composant, en cliquant sur Ajouter champ.
- Double-cliquez sur le nom de chaque champ et saisir "Key" en tant que nom du premier champ, et "Value" en tant que nom du second champ. Le champ "Key" stockera le nom de la propriété alors que

le champ "Value" stockera la valeur de la propriété. Pour plus d'informations concernant les composants CSV, voir <u>Fichiers CSV et texte</u><sup>446</sup>.

Champ1		Valeur	
string	<b>•</b>	string	•
<			2
Ajouter champ Ins <u>é</u> rer champ	Supprimer champ << >>>		

#### Étape 3 : Ajouter le filtre et les fonctions

- 1. Glisser les fonctions <u>equal</u><sup>633</sup>, <u>exists</u><sup>633</sup> et <u>group-starting-with</u><sup>635</sup> depuis la fenêtre de bibliothèque dans le mappage. Pour des informations générales concernant les fonctions, voir <u>Fonctions</u><sup>651</sup>.
- 2. Pour ajouter le filtre, cliquez dans le menu **Insérer**, puis cliquez sur **Filtre : Nœuds/Lignes**. Pour des informations générales concernant les filtres, voir <u>Filtres et conditions</u><sup>523</sup>.
- 3. Dans le menu Insérer, cliquez sur Constante, puis saisissez le texte "chaîne".
- 4. Cliquez avec la touche de droite sur le nœud dict dans le composant de source et choisissez Afficher les éléments enfants avec un Nom Dynamique depuis le menu contextuel. Dans le dialogue "Paramètres des Enfants nommés dynamiquement", veuillez vous assurer que la case à cocher Afficher les nœuds de test pour filtrer ou créer les élément par nom de nœud fixé est bien sélectionnée.

😤 Paramètres enfants nommés dynamiquement	$\times$	
Utiliser ces paramètres pour configurer l'accès aux éléments enfants en utilisant une structure générique.		
Sélectionner les types pour accéder au contenu de l'item nommé dynamiquement :		
□ B array	$\sim$	
🗆 唱 data		
□ B date		
□ B dict		
□ ¤ł false		
□ B integer		
□ 唱 key		
□ 中君 real		
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		
□ B true		
□ ¤ł text()		
La sélection d'un type à cet endroit rendra sa structure disponible uniquement pour le mappage. Le type actuel n'est pas coché lors de la marche. Afficher les nœuds de test pour filtrer ou créer des éléments par nom de nœud fixé		
OK Annule	•	

5. Tracer les connexions comme affiché ci-dessous.



ReadPropertyList.mfd

#### Explication du mappage

L'élément element() dans le composant source fournit à tous les enfants du le nœud dict, comme séquence à la fonction group-starting-with. La fonction group-starting-with crée un nouveau groupe à chaque fois qu'un nœud portant le nom key est rencontré. La fonction exists vérifie cette condition et retourne le résultat en tant que valeur booléenne vrai/faux de la fonction de regroupement.

Pour chaque groupe, le filtre vérifie si le nom du nœud actuel est égal à "string", avec l'aide de la fonction equal. Le nom lui-même est lu depuis le local-name(), qui fournit le nom du nœud en tant que string.

Les connexions vers le composant cible ont le rôle suivant :

- Une nouvelle ligne est créée dans le CSV cible uniquement si la condition de filtre est vraie.
- Key (nom de propriété) est prélevé depuis la valeur de l'élément key dans la source.
- Value (valeur de de propriété) est prélevé depuis le nœud de test de nom string name.

## 7.3 Règles et stratégies de mappage

En général, MapForce mappe des données de manière intuitive, mais il peut tomber sur des situations dans lesquelles la sortie contient trop ou pas assez d'items. Ce chapitre a pour but de vous aider à éviter des situations dans lesquelles le mappage produit un résultat non désiré dû à des connections incorrectes ou un contexte de mappage.

### Règles de mappage

Pour être valide, un mappage doit inclure au moins une source et au moins un composant de cible. Un composant de source est un composant qui lit des données, généralement depuis un fichier ou une base de données. Un composant de cible est un composant qui écrit des données, généralement vers un fichier ou une base de données. Si vous tentez d'enregistrer un mappage où les points cités ci-dessus ne sont pas vrais, une erreur apparaît dans la fenêtre Message : "Un mappage exige au moins deux structures connectées, pour un exemple, un schéma ou une structure de base de données".

Pour créer un mappage de données, vous dessinez des connexions de mappage entre des items dans les composants de source et de cible.

Toutes les connexions de mappage que vous tracez forment, ensemble, un algorithme de mappage. Au moment de l'exécution du mappage, MapForce évalue l'algorithme et traite les données basées sur celui-ci. L'intégrité et l'efficacité de l'algorithme de mappage dépend principalement des connexions. Vous pouvez modifier certains paramètres au niveau du mappage<sup>112</sup>, au niveau du composant<sup>60</sup>, ou même au niveau de la connexion<sup>90</sup>, mais, essentiellement, les *connexions* de mappage déterminent comment sont traitées vos données.

Gardez en tête les règles suivantes lorsque vous créez des connexions :

 Lorsque vous tracez une connexion *depuis* un item de source, le mappage lit les données associées avec cet item provenant du fichier ou de la base de données source. Les données peuvent avoir zéro, une, ou plusieurs occurrences (en d'autres termes, il peut s'agir d'une séquence, comme décrit cidessous). Par exemple, si le mappage lit des données depuis un fichier XML contenant des livres, le fichiers XML source peut contenir zéro, un ou plusieurs éléments **book**. Dans le mappage ci-dessous, veuillez noter que l'item **book** apparaît une seule fois dans le composant de mappage, bien que le fichier de source (instance) peut contenir plusieurs éléments **book**, ou aucun.



- 2. Lorsque vous tracez une connexion vers un item de cible, le mappage génère des données d'instance de ce type. Si l'item de source contient un contenu simple (par exemple, string ou entier) et si l'item cible accepte un contenu simple, MapForce copie le contenu vers l'item de cible et, si nécessaire, convertir le type de données. Zéro, une ou plusieurs valeurs peuvent être générées, selon les données de source entrantes, voir le point suivant.
- 3. Pour chaque (instance) item dans la source, un (instance) item est créé dans la cible. Il s'agit là de la règle générale de mappage dans MapForce. En prenant le mappage ci-dessus en guise d'exemple, si le XML source contient trois éléments book, alors trois éléments publication seront créés du côté cible. Veuillez noter qu'il existe aussi des cas spéciaux, voir <u>Séquences</u><sup>673</sup>.
- 4. Chaque connexion crée un contexte de mappage actuel. Le contexte détermine quelles données sont disponibles actuellement, pour le nœud de cible actuel. Ainsi, le contexte détermine quels items de source sont réellement copiés depuis la source vers le composant de cible. En traçant ou en omettant une connexion, vous pouvez modifier par inadvertance le contexte actuel et influer le résultat du mappage. Par exemple votre mappage peut appeler une base de données ou un service Web plusieurs fois sans que cela soit nécessaire au cours d'une seule exécution de mappage. Ce concept est décrit plus en détail ci-dessous, voir Le contexte de mappage

## 7.3.1 Séquences

Comme mentionné auparavant, la règle de mappage générale est : "pour chaque item se trouvant dans la source, en créer un dans la cible". Ici, "item" signifie une des choses suivantes :

- un nœud unique d'instance du fichier ou de la base de données d'entrée
- une séquence de zéro à plusieurs nœuds d'instance du fichier ou de la base de données d'entrée

Pendant l'exécution de mappage, si une séquence atteint un item de cible, cela crée une boucle qui génère autant de nœuds cible qu'il y a de nœuds de source. Néanmoins, il y a quelques exceptions à cette règle :

- Si l'item de cible est un élément root XML, il est créé une fois (et une seule fois). Si vous y connectez une séquence, le résultat peut ne pas être valide pour le schéma. Si les attributs de l'élément root sont aussi connectés, la sérialisation XML échouera au moment de l'exécution du mappage. C'est pourquoi, il est nécessaire de connecter une séquence à l'élément XML root.
- Si l'item de cible accepte uniquement une valeur, elle est créée une seule fois. Des exemples d'items qui acceptent uniquement une seule valeur : Attributs XML, champs de base de données, composants de sortie simples. Par exemple, le mappage ci-dessous génère une séquence de trois entiers (1, 2, 3) avec l'aide de la fonction generate-sequence. Néanmoins, le résultat contiendra uniquement un seul entier, cas la cible est un seul composant de résultat qui accepte une seule valeur. Les deux autres valeurs sont ignorées.



• Si le schéma de source précise qu'un item spécifique ne se produit qu'une seule fois, mais que le fichier d'instance en contient de nombreux, MapForce peut extraire le premier item depuis la source (qui doit exister conformément au schéma) et créer un seul item dans la cible. Pour désactiver ce comportement, décocher la case activer les optimisation de traitement d'entrée sur la base de min/maxOccurs depuis les paramètres de composant, voir aussi Paramètres de composant XML<sup>[12]</sup>.

Si la séquence est vide, rien n'est généré du côté cible. Par exemple, si la cible est un document XML et que la séquence de source est vide, aucun élément XML ne sera créé dans la cible.

Les fonctions agissent de manière similaire : si elles obtiennent une séquence en tant qu'entrée, alors elles seront appelées (et produire autant de résultats que) autant de fois qu'il y a d'items dans la séquence.

Si une fonction obtient une séquence vide en tant qu'entrée, elle retourne un résultat vide également, par conséquent, elle ne produit aucune sortie.

Néanmoins, il existe certaines catégories de fonctions qui, en raison de leur design, retournent une valeur même si elle obtiennent une séquence vide en tant qu'entrée :

- exists, not-exists, substitute-missing
- is-null, is-not-null, substitute-null (ces trois fonctions sont des alias des trois précédents)
- fonctions aggregate (sum, count, etc.)
- Fonctions définies par l'utilisateur qui acceptent des séquences et sont des fonctions régulières (pas inlined)

Si vous souhaitez remplacer une valeur vide, ajouter la fonction **substitute-missing** au mappage et remplacer la valeur vide par une valeur de substitution de votre choix. En alternative, vous pouvez obtenir le même résultat en utilisant <u>Fonctions Défauts et Nœud</u><sup>660</sup>.

Les fonctions peuvent avoir plusieurs entrées. Si une séquence est connectée à chaque entrée, cela résulte en un produit cartésien de toutes les entrées, ce qui n'est généralement pas le résultat souhaité. Afin d'éviter cela, ne connecter qu'une seule séquence à une fonction avec plusieurs paramètres ; tous les autres paramètres doivent être connectés aux items "singular" provenant de parents ou d'autres composants.

## 7.3.2 Le contexte de mappage

Les composants de mappage sont des structures hiérarchiques qui peuvent contenir plusieurs niveaux de profondeur. D'un autre côté, un mappage peut avoir plusieurs sources et composants, plus des composants intermédiaires comme des fonctions, des filtres, des value-maps, etc. Cela rajoute de la complexité dans l'algorithme de mappage, en particulier si plusieurs composants non-liés sont connectés. Afin de pouvoir exécuter le mappage en petites parties, une étape à la fois, un contexte actuel doit être établi poru chaque connexion.

On pourrait donc dire que plusieurs "contextes actuels" sont établis pour la durée de l'exécution de mappage, depuis les modifications de contexte actuels à chaque connexion traitée.

MapForce établit toujours le contexte actuel en commençant depuis l'*item de racine cible (nœud)*. C'est là que l'exécution du mappage commence réellement. La connexion à l'item de racine cible est retracée à tous les items de source qui y sont connectés directement ou indirectement, y compris par le biais de fonctions ou d'autres composants intermédiaires. Tous les items de source et les résultats produits par les fonctions sont ajoutés au contexte actuel.

Une fois que le processus dans le nœud cible est terminé, MapForce se déplace vers le bas de la hiérarchie. Concrètement, il traite tous les *items mappés* du composant cible du haut en bas. Pour chaque item, un nouveau contexte est établi qui contient au début tous les items du contexte parent. Ainsi, tous les items frères mappés dans un composant cible sont indépendants l'un de l'autre, mais ont accès à toutes les données de source de leurs items parent.

Nous allons voir comment les éléments sus-mentionnés s'appliquent en pratique, sur la base d'un mappage d'exemple, **PersonListByBranchOffice.mfd**. Vous trouverez ce mappage dans le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\**.



Dans le mappage ci-dessus, les composants de source et cible sont XML. Le fichier de source XML contient deux éléments **Office**.

Comme mentionné précédemment, l'exécution de mappage commence toujours depuis le nœud de racine cible (**PersonList**, dans cet exemple). En retraçant la connexion (par le biais du filtre et de la fonction) vers un item de source, vous pouvez conclure que l'item de source est **Office**. (L'autre chemin de connexion mène à un paramètres d'entrée et son objectif est expliqué ci-dessous).

S'il y avait une connexion directe entre **Office** et **PersonList**, alors, conformément à la règle de mappage générale, le mappage aurait créé autant d'items d'instance **PersonList** qu'il y a d'items **Office** dans le fichier source. Néanmoins, cela ne se produira pas ici, étant donné qu'il y a un filtre entre les deux. Le filtre n'apporte au composant de cible que les données qui satisfont à la condition Booléenne connectée à l'entrée **bool** du filtre. La fonction **equal** retourne **true** si le nom de bureau est égal à "Nanonull, Inc.". Cette condition n'est satisfaite qu'une seule fois, car il n'y a qu'un seul nom de bureau dans le fichier XML de source.

Par conséquent, la connexion entre **Office** et **PersonList** définit un seul bureau en tant que le contexte pour l'ensemble du document cible. Cela signifie que tous les descendants de l'item **PersonList** ont accès aux données du bureau "Nanonull, Inc.", et qu'aucun autre bureau n'existe dans le contexte actuel.

La connexion suivante est entre **Contact** et **Person**. Conformément à la règle générale de mappage, elle créera une cible **Person** pour chaque source **Contact**. À chaque itération, cette connexion établit un nouveau contexte actuel ; c'est pourquoi les connexions enfant (**first** à **First**, **last** à **Last**) fournissent des données depuis l'item de source vers la cible dans le contexte de chaque **Person**.

Si vous avez ignoré la connexion entre **Contact** et **Person**, alors le mappage créera uniquement une **Person** avec plusieurs nœuds **First**, **Last**, et **Details**. Dans ces cas, MapForce émet un avertissement et une suggestion dans la fenêtre Messages, par exemple :

Messages X	
▼ ▼ ▲ □ □ □ ● ♀ ♀ X	
Reproved the second sec	1
🕋 🗇 🛕 🗅 Person n'a aucune connexion d'entrée et sera généré une seule fois.	1
Vous pouvez essayer de connecter D <u>Contact</u> avec D <u>Person</u> pour la résolution.	1
💦 🛄 🕕 🕕 Le composant de sortie 🤹 PersonList n'a pas de nom de fichier de sortie défini. Un nom de fichier par défaut sera utilisé.	1
Conception of the second secon	1
Vue d'ensemble Messages	-

Enfin, le mappage comprend une fonction définie par l'utilisateur, LookupPerson. La fonction définie par l'utilisateur est aussi exécutée dans le contexte de chaque **Person**, étant donné que la connexion parent entre **Contact** et **Person**. Spécifiquement, à chaque fois qu'un nouvel item Person est créé du côté cible, la fonction est appelée pour remplir l'élément **Details** de la personne. Cette fonction prend trois paramètres d'entrée. Le premier (**OfficeName**) est défini pour lire des données provenant du paramètres d'entrée du mappage. Les données source pour ce paramètre peuvent aussi bien être fournies par l'item de source **Name**, sans aucunement changer la sortie de mappage. Dans tous les cas, la valeur de source est la même et est prise depuis un contexte de parent. En interne, la fonction de consultation concatène les valeurs reçues en tant qu'arguments et produit une seule valeur. Pour plus d'informations concernant le fonctionnement de la fonction **LookupPerson**, voir l'<u>Exemple: Consultation et Concaténation</u>

## 7.3.2.1 Base de données

Afin d'améliorer l'efficacité et de réduire l'usage des ressources en matériel et en réseau, il est recommandé d'éviter d'appeler la même base de données plusieurs fois dans le même mappage. Vous pouvez être confronté à des situations dans lesquelles vous ne pouvez pas éviter d'appeler une base de données plusieurs fois en raison de la nature du mappage, mais voici quelques recommandations :

- Si vous ne nécessitez qu'un seul appel de base de données, évitez de placer le composant de base de données dans un contexte parent qui exigerait d'appeler la base de données plusieurs fois. Cela pourrait se produire par exemple, si vous ajoutez un composant de base de données dans une fonction définie par l'utilisateur qui reçoit une séquence de valeurs en tant qu'entrée et qui reçoit donc un appel pour chaque item dans la séquence, voir aussi <u>Fonctions définies par l'utilisateur</u><sup>®77</sup> ci-dessous. Des variables sont généralement utiles pour rassembler des données dans le même contexte avant le les passer dans le composant cible.
- Si vous souhaitez rassembler des valeurs depuis une base de données (par exemple, pour compter le nombre d'enregistrements en utilisant la fonction count), il est recommandé de connecter la sortie de la fonction aggregate dans une variable où compute-when=once. Cela évite les appels répétitifs à la base de données, comme décrit dans l'Exemple: Compter les lignes de table de base de données.
- Essayer d'extraire toutes les données de base de données en un appel (par exemple, une instruction SQL-SELECT, ou une procédure stockée), contrairement à l'ajout du même composant de base de données plusieurs fois dans le mappage.
- Si vous souhaitez extraire des données depuis plusieurs tables ou modes depuis la même base de données, nous vous conseillons d'utiliser soit un composant Join (dans le mode SQL), soit une instruction SQL-SELECT. Ce dernier est plus pratique si vous préférez écrire l'instruction SQL SELECT vous-même. Si vous devez joindre des données de base de données à des données non-base de données, ou des données provenant de base de données différentes, utiliser des joins non-SQL. Pour optimiser l'exécution de joins non-SQL dans des mappages intenses en données, exécuter les mappages avec MapForce Server Advanced Edition.

 Si vous nécessitez de filtrer des données depuis une base de données, il est plus efficace d'utiliser un composant SQL-WHERE au lieu d'un filtre standard, puisque le composant précédent a été optimisé spécifiquement pour le travail avec des bases de données, dans la grammaire de la base de données respective.

## 7.3.2.2 Fonctions définies par l'utilisateur

Les fonctions définies par l'utilisateur (FDUs) sont des fonctions personnalisées intégrées dans le mappage, où vous définissez les entrées, les sorties et la logique de traitement. Chaque fonction définie par l'utilisateur peut contenir le même type de composant en tant que mappage principal, y compris les services Web et les bases de données.

Par défaut, si une FDU contient une base de données ou un composant de service Web et si les données d'entrée du FDU est une séquence de plusieurs valeurs, chaque entrée de valeur appellera la FDU et par conséquent entraînera dans une base de données ou un appel de service Web.

Ce comportement mentionné peut être acceptable pour le type de mappage dans lequel vous avez vraiment besoin que la FDU soit appelée *autant de fois qu'il y a de valeurs d'entrée* et que vous n'avez pas d'alternative.

Si vous ne souhaitez pas que cette situation se produise, vous pouvez configurer la FDU de manière à ce qu'elle soit appelée une seule fois même si elle reçoit une séquence de valeurs en tant qu'entrée. Généralement, c'est la technique que vous adopterez pour les FDU qui fonctionnement sur un ensemble de valeurs avant qu'elles puissent être retournées (comme des fonctions qui calculent les moyennes ou les totaux).

Configurer une FDU pour accepter plusieurs valeurs d'entrée dans le même appel de fonction est possible si la FDU est de type "regular", pas "inlined". (Pour plus de détails, voir le chapitre <u>Fonctions définies par</u> <u>l'utilisateur</u>.) Avec des fonctions régulières, vous pouvez spécifier que le paramètres d'entrée est une séquence en cochant la case **Entrée est une séquence**. Cette case à cocher est visible dans les paramètres de composant, après avoir double-cliqué dans la barre de titre d'un paramètre d'entrée. La case à cocher influe sur le nombre de fois que la fonction est appelée :

- Lorsque les données d'entrée sont connectées à un paramètre sequence, la fonction définie par l'utilisateur est appelée uniquement une fois et la séquence complète est passée dans la fonction définie par l'utilisateur.
- Lorsque les données d'entrée sont connectées à un paramètre **non-sequence**, la fonction définie par l'utilisateur est appelée *une seule fois dans chaque item unique dans la séquence*.

Pour un exemple, ouvrir le mappage de démonstration suivant : <Documents>\Altova\MapForce2023\MapForceExamples\InputIsSequence.mfd.



Le mappage ci-dessus illustre un cas typique d'une FDU qui fonctionne sur un ensemble de valeurs et nécessite donc toutes les valeurs d'entrée en un appel. Spécifiquement, la fonction définie par l'utilisateur **Calculate** retourne les températures minimum, maximum et moyennes, en prenant en tant que données d'entrée provenant d'un fichier XML de source. La sortie de mappage attendue est comme suit :



Comme d'habitude, l'exécution de mappage commence avec l'item supérieur du composant cible (**YearlyStats**, dans cette exemple). Pour remplir ce nœud, le mappage tente d'obtenir des données de source provenant de la FDU, qui, en revanche, déclenche le filtre. Le rôle du filtre dans ce mappage est de passer dans la FDU uniquement les températures depuis l'année 2008.

La case à cocher **Entrée est séquence** a été cochée pour le paramètre d'entrée de la FDU (pour voir cette case à cocher, double-cliquer sur la barre de titre de la fonction **Calculate** pour saisir le mappage de la fonction ; ensuite double-cliquer sur la barre de titre du paramètre d'entrée). Comme évoqué plus haut, l'option **Entrée est séquence** option entraîne l'approvisionnement de la séquence de valeurs complète en tant qu'entrée dans la fonction et la fonction est appelée une seule fois.

😢 Éditer Entrée		×
Nom: Tempera	atures	
Туре		
○ Type simple typ	pe (entier, chaîne, etc.)	
Type de donnée	s : 🗸	
Type complexe	(structure arborescente)	
Structure :	Temperatures.xsd Choisir Éditer	
Racine :	Temperatures/data Choisir	
	Enregistrer le chemin d'accès de structure relatif au fichier MFD	
Entrée requise		
🗹 Entrée est une sé	iquence	
	OK Annuler	

Si vous n'aviez pas coché la case à cocher **Entrée est séquence**, la FDU aurait été appelée pour chaque valeur dans la source. Par conséquence, le minimum, le maximum et la moyenne serait calculée pour chaque valeur unique individuelle et une sortie incorrecte serait produite.

En appliquant la même logique dans des FDU plus complexes qui comprennent une base de données ou des appels de service Web, il peut être possible d'optimiser l'exécution et d'éviter des appels inutiles à la base de données ou au service Web. Néanmoins, sachez que la case à cocher **Entrée est séquence** ne contrôle pas ce qui arrive à la séquence de valeur *après* qu'elle saisit la fonction. En d'autres termes, rien ne vous empêche de connecter la séquence des valeurs entrantes dans l'entrée d'un service Web et donc de l'appeler plusieurs fois. Considérez l'exemple suivant :



La FDU illustrée ci-dessus reçoit une séquence des valeurs depuis le mappage externe. En particulier, les données fournies au paramètre d'entrée provient d'une base de données. L'option paramètre d'entrée **Entrée est séquence** est sélectionnée, donc toute la séquence est fournie à la fonction en un appel. La fonction est censée accumuler plusieurs valeurs **quantity** et publier le résultat dans un service Web. Exactement un appel de service Web est attendu. Néanmoins, le service Web sera appelé plusieurs fois de manière incorrecte

lorsque le mappage est exécuté. La raison est que l'entrée **Request** du service Web reçoit *une séquence de valeurs*, pas une seule valeur.

Pour régler ce problème, connecter l'entrée **Request** du service Web dans le résultat de la fonction sum. La fonction produit une seule valeur, donc le service Web sera aussi appelé une fois :



Normalement, les fonctions aggregate comme sum, count, etc produisent une seule valeur. Néanmoins, s'il existe une connexion parent qui le permet, elles peuvent également produire une séquence de valeurs, comme décrit plus loin dans l'<u>Exemple : Changer le contexte Parent</u><sup>(80)</sup>.

## 7.3.2.3 Exemple : Changer le contexte de parent

Certains composants de mappage ont un item optionnel parent-context.

parent-context est un argument optionnel dans certaines fonctions d'agrégation core MapForce (comme dans min, max, avg, count). Dans un composant de source qui possède plusieurs séquences hiérarchiques, le contexte parent détermine l'ensemble de nœuds dans lequel la fonction doit fonctionner.

À l'aide de cet item, vous pouvez influer sur le mappage dans lequel ce composant devrait opérer et par conséquent changer le résultat du mappage. Les composants qui ont un **parent-context** optionnel sont : fonctions aggregate, variables et composants Join.



Pour voir comment la modification du contexte de parent peut être utile, consulter une démo en ouvrant le mappage suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\ParentContext.mfd.



Dans le XML source du mappage ci-dessus, il y a un nœud **Company** unique qui contient deux nœud **Office**. Chaque nœud **Office** contient plusieurs nœuds **Department**, et chaque **Department** contient plusieurs nœud **Person**. Si vous ouvrez le fichier XML dans un éditeur XML, vous pouvez voir que la distribution des personnes par bureau et par département est la suivante :

Bureau	Département	Nombre d'employés
Nanonull, Inc.	Administration	3
	Marketing	2
	Engineering	6
	IT & Technical Support	4
Nanonull Partners, Inc.	Administration	2
	Marketing	1
	IT & Technical Support	3

Le mappage compte toutes les personnes dans tous les départements. À cet effet, il utilise la fonction count depuis la bibliothèque core. Si vous cliquez sur l'onglet **Output** pour consulter le mappage, vous remarquerez qu'il produit une seule valeur, **21**, qui correspond au nombre total de personnes contenues dans le fichier XML source.

Le mappage fonctionne comme suit:

- Comme d'habitude, l'exécution de mappage débute avec le nœud supérieur du composant cible (rows, dans cet exemple). Il n'y a pas de connexion entrante pour rows. En résultat, un contexte de mappage implicite est établi entre Company (item supérieur du composant de source) et rows (item supérieur du composant cible).
- Le résultat de la fonction est une seule valeur, puisqu'il n'y a qu'une seule entreprise dans le fichier source.
- Afin de remplir l'item cible **col1**, MapForce exécute la fonction **count** dans le *contexte de parent implicite* mentionné ci-dessus, afin qu'il compte tous les nœuds **Person** depuis tous les bureaux et tous les départements.

L'argument **parent-context** de la fonction vous permet de modifier le contexte de mappage. Cela vous permet, par exemple, de compter le nombre de personnes dans chaque département. Pour ce faire, tracer deux (ou plus) connexions comme indiqué ci-dessous :



Dans le mappage ci-dessus, la connexion A change le contexte de parent de la fonction **count** dans le **Department**. En résultat, la fonction comptera le nombre de personnes se trouvant dans chaque département. Chose importante, la fonction retournera maintenant une *séquence* de résultats au lieu d'un seul résultat, étant donné que plusieurs département existent dans la source. C'est la raison pour laquelle la connexion B existe: pour chaque item dans la séquence résultante, elle crée une nouvelle ligne dans le fichier cible. Le résultat de mappage change maintenant (veuillez noter que les nombres correspondent exactement au décompte des personnes dans chaque département) :

```
<rows>
        <row>
            <coll>3</coll>
            </row>
            <row>
            <coll>2</coll>
```

```
</row>
    <row>
        <col1>6</col1>
    </row>
    <row>
        <col1>4</col1>
    </row>
    <row>
        <col1>2</col1>
    </row>
    <row>
        <col1>1</col1>
    </row>
    <row>
        <col1>3</col1>
    </row>
</rows>
```

Étant donné que le mappage actuel crée une ligne dans chaque département, vous pouvez copier en option le nom du bureau et du département aussi dans le fichier cible, en traçant des connexions C et D :



Ainsi, le résultat affichera non seulement le nombre de personnes mais aussi le nom correspondant du bureau et du département.

Si vous souhaitez compter le nombre de personnes dans chaque bureau, connectez le contexte de parent de la fonction **count** à l'item **Office** dans la source.



Avec les connexions affichées ci-dessus, la fonction **count** retourne un résultat pour chaque bureau. Il y a deux bureaux dans le fichier source, la fonction retournera maintenant deux séquences. Par conséquent, il y aura deux lignes dans le résultat, chaque ligne contiendra le nombre de personnes contenues dans ce bureau :

```
<rows>
<rows>
<row>
<coll>15</coll>
<coll>Nanonull, Inc.</coll>
</row>
<row>
<coll>6</coll>
<coll>nanonull Partners, Inc.</coll>
</row>
</row>
```

## 7.3.3 Contexte de priorité

Le contexte de priorité est une méthode d'influencer l'ordre dans lequel les paramètres d'entrée d'une fonction sont évalués. La configuration d'un contexte de priorité peut être nécessaire si votre mappage rejoint des données provenant de deux sources non liées.

Afin de comprendre comment le contexte de priorité fonctionne, souvenez-vous que, lorsqu'un mappage est exécuté, la connexion à un item d'entrée peut porter une *séquence* de plusieurs valeurs. En ce qui concerne les fonctions avec deux paramètres d'entrée, cela signifie que MapForce doit créer deux boucles, dont une doit être traitée en premier. Cette boucle est la boucle "extérieure". Par exemple, la fonction equal reçoit deux

paramètres : *a* et *b*. si *a* et *b* obtiennent tous deux une séquence de valeurs, alors MapForce traite comme suit :

- Pour chaque occurrence de a
  - Pour chaque occurrence de *b* 
    - a est-il égal à b?

Comme vous pouvez le voir dans l'exemple ci-dessus, chaque *b* est évalué dans le contexte de chaque *a*. Le contexte de priorité vous permet de modifier la logique de traitement de manière à ce que chaque *a* est évalué dans le contexte de chaque *b*. En d'autres termes, il vous permet d'échanger la boucle intérieure avec la boucle extérieure, par exemple :

- Pour chaque occurrence de *b* 
  - Pour chaque occurrence de a
    - a est-il égal à b?

Examinons à présent un mappage dans lequel le contexte de priorité touche la sortie du mappage. Dans le mappage ci-dessous, la fonction concat a deux paramètres d'entrée. Chaque paramètre d'entrée est une séquence qui a été générée avec l'aide de la fonction generate-sequence. La première séquence est "1,2" et la seconde séquence est "3,4".



Tout d'abord, examinons le mappage sans déterminer de contexte de priorité. La fonction concat commence à évaluer la première séquence en premier, donc elle combine des valeurs dans l'ordre suivant :

- 1 avec 3
- 1 avec 4
- 2 avec 3
- 2 avec 4

Cela est également reflété dans le résultat de mappage :

```
<data>

<value>13</value>

<value>14</value>

<value>23</value>

<value>24</value>

</data>
```

Si vous cliquez avec la touche de droite dans le deuxième paramètre d'entrée et que vous choisissez **Contexte** de priorité depuis le menu contextuel, celui-ci deviendra le contexte de priorité. Comme illustré ci-dessous, l'entrée du contexte de priorité est encerclé.



Cette fois, le deuxième paramètre d'entrée sera évalué en premier. La fonction `concat` concatènera toujours les mêmes valeurs, mais cette fois elle traitera la séquence `3,4` en premier. Par conséquent, le résultat donne :



Jusqu'à présent, vous n'avez vu que la partie théorique derrière le contexte de priorité. Pour consulter un scénario plus pratique, voir <u>Exemple: Filtrer avec un contexte de priorité</u><sup>886</sup>.

## 7.3.3.1 Exemple: Filtrer avec un contexte de priorité

Lorsqu'une fonction est connectée à un filtre, le contexte de priorité n'affecte non seulement la fonction ellemême, mais aussi l'évaluation du filtre. Le mappage ci-dessous illustre un cas typique lorsque vous nécessitez de définir un contexte de priorité pour pouvoir obtenir le résultat correct. Vous trouverez ce mappage sous le chemin suivant :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\FilterWithPriority.mfd.

**Note :** Ce mappage utilise des composants XML, mais la même logique que celle décrite ci-dessous s'applique pour tous les autres types de composants dans MapForce, y compris EDI, JSON, etc. Pour les bases de données, il est conseillé d'effectuer des filtres en utilisant des composants <u>SQL</u> <u>WHERE</u> <sup>(35)</sup> au lieu des filtres standard.



L'objectif du mappage ci-dessus est de copier des données de personnes depuis **Articles.xml** dans un nouveau fichier XML avec un schéma différent, **articledata.xml**. Dans le même temps, le mappage doit consulter les détails de chaque article dans le fichier **Products.xml** et les joindre à l'enregistrement d'article respectif. Veuillez noter que chaque enregistrement dans **Articles.xml** ait un **Number** et que chaque enregistrement dans **Articles.xml** ait un **Number** et que chaque enregistrement dans **Products.xml** ait un **id**. Si ces deux valeurs sont égales, alors toutes les autres valeurs (**Name**, **SinglePrice**, **color**, **size**) doivent être copiées dans la même **ligne** dans la cible.

Cet objectif a été accompli en ajoutant un filtre. Chaque filtre nécessite une condition Booléenne en tant qu'entrée ; seuls les nœud/lignes qui satisfont la condition seront copiés dans la cible. À cet effet, il existe une fonction equal dans le mappage. La fonction equal vérifie si le nombre d'article et l'ID de produit sont égaux dans les deux sources. Le résultat est ensuite fourni en tant qu'entrée dans le filtre. Si **true**, alors l'item **Article** est copié dans la cible.

Veuillez noter qu'un contexte de priorité a été défini dans le deuxième paramètres d'entrée de la second fonction **equal**. Dans ce mappage, le contexte de priorité fait la grande différence, et le fait de ne pas le configurer entraînera un résultat de mappage incorrect.

#### Mappage initial : Aucun contexte de priorité

Voici la logique de mappage sans contexte de priorité :

- Conformément à la règle de mappage général, pour chaque **Article** qui satisfait la condition de filtre, une nouvelle **row** est créée dans la cible. La connexion entre **Article** et **row** (par le biais de la fonction et du filtre) se charge de cette partie.
- Le filtre vérifie la condition pour chaque article. Pour ce faire, il itère à travers tous les produits, et apporte plusieurs produits dans le contexte actuel.
- Pour remplir l'id du côté cible, MapForce suit la règle générale (pour chaque item dans la source, crée un item dans la cible). Néanmoins, comme expliqué ci-dessus, tous les produits provenant de Products.xml se trouvent dans le contexte actuel. Il n'y a pas de connexion entre les product vers un autre endroit dans la cible afin de lire l'id d'un produit spécifique uniquement. En conséquence,

plusieurs éléments id seront créés pour chaque Article dans la cible. La même chose se produit avec color et size.

Pour résumer : des items provenant de **Products.xml** possèdent le contexte du filtre (qui doit itérer à travers chaque produit) ; c'est pourquoi, les valeurs **id**, **color** et **size** seront copiées dans chaque cible **row** autant de fois qu'il y a des produits dans le fichier source, et généreront un résultat incorrect comme celui ci-dessous :

```
<rows>
        <id>1</id>
        <id>1</id>
        <id>2</id>
        <id>1</id>
        <id>2</id>
        <id>2</id>
        <id>2</id>
        1
        2</id>
        1
        2</id>
        1
        2</id>
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        1
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
        2
```

#### Solution A : Utiliser le contexte de priorité

Le problème ci-dessus a été résolu en ajoutant un contexte de priorité dans la fonction qui calcule la condition Booléenne du filtre.

En particulier, si le second paramètres d'entrée de la fonction equal est désigné en tant que contexte de priorité, la séquence provenant de **Products.xml** est mis en priorité. Cela se traduit par la logique de mappage suivante :

- Pour chaque produit, remplir l'entrée b de la fonction equal (en d'autres termes, mettre la priorité sur b). À ce niveau, les détails du produit actuel se trouvent dans le contexte.
- Pour chaque article, remplir l'entrée **a** de la fonction **equal** et vérifier si la condition de filtre est vraie. Si oui, placer les détails de l'article également dans le contexte actuel.
- Ensuite, copier les détails de l'article et du produit depuis le contexte actuel dans les items respectifs dans la cible.

La logique de mappage ci-dessus produit le résultat correct, par exemple :

```
<rows>
<rows>
<row>
<id>l</id>
</row>
<id>l</id>
</row>
<row>
<row>
</row>
</rows>
```

### Solution B : Utiliser une variable

En tant qu'une solution alternative, vous pouvez apporter chaque article et produit qui correspond la condition du filtre dans le même contexte avec l'aide d'une variable intermédiaire. Les variables sont pertinentes pour des scénarios comme celui-ci parce qu'elles vous permettent de stocker des données temporairement dans le mappage, et vous aident donc à modifier le contexte selon vos besoins.

Pour des scénarios comme celui-ci, vous pouvez ajouter au mappage une variable qui a le même schéma que le composant de cible. Dans le menu **Insertion**, cliquer sur **Variable**, et fournir le schéma **articledata.xsd** en tant que structure lorsque vous y êtes invité.



Dans le mappage ci-dessus, les choses suivantes se produisent :

- Le contexte de priorité n'est plus utilisé. À la place de cela il y a une variable, qui a la même structure que le composant de cible.
- Comme d'habitude, l'exécution de mappage commence à partir du nœud racine de cible. Avant de remplir la cible, le mappage collectionne des données dans la variable.
- La variable est calculée dans le contexte de chaque produit. Cela se produit parce qu'il y a une connexion provenant de **product** vers l'entrée **compute-when** de la variable.
- La condition de filtre est donc vérifiée dans le contexte de chaque produit. Il faut que la condition soit vraie, la structure de la variable sera remplie et transmise vers la cible.

## 7.3.4 Composants de cible multiple

Un mappage peut avoir plusieurs composants de source et de cible. Lorsqu'il y a plusieurs composants de cible, vous pouvez prévisualiser uniquement une sortie de composant à la fois dans MapForce, celui que vous indiquez en cliquant sur la touche **Preview**. Dans d'autres environnement d'exécution (MapForce Server ou code généré), tous les composants de cible seront exécutés séquentiellement, et la sortie respective de chaque composant sera produite.

Par défaut, les composants de cible sont traités du haut en bas et de gauche à droite. Si nécessaire, vous pouvez influer sur cet ordre en changeant la position des composants de cible dans la fenêtre de mappage. Le point de référence est le coin gauche supérieur de chaque composant. Veuillez noter les points suivants :

- Si deux composants ont la même position verticale, alors celui placé à l'extrême gauche prend précédence.
- Si deux composants ont la même position horizontale, alors celui placé à le plus haut prend précédence.
- Dans la situation improbable que les composants aient exactement la même position, une ID de composant interne unique sera utilisée automatiquement, ce qui garantit un ordre bien défini mais qui ne peut pas être changé.

Pour voir un exemple de son fonctionnement, ouvrir la démo de mappage suivante :

<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\GroupingFunctions.mfd. Ce mappage consiste en plusieurs sources et plusieurs composants de cible ; seul un fragment est affiché ci-dessous.



Conformément aux règles, l'ordre de traitement par défaut de ce mappage dans MapForce Server et dans le code généré va du haut en bas. Vous pouvez vérifier que cela est bien le cas en générant un code XSLT 2.0, par exemple.

- 1. Dans le menu Fichier, cliquer sur Générer du code dans | XSLT 2.0.
- 2. Lorsque vous êtes invité à le faire, sélectionner un répertoire de cible pour le code généré.

Après la génération, le répertoire de cible inclut plusieurs fichiers XSLT et un fichier **DoTransform.bat**. Ce dernier peut être exécuté par RaptorXML Server (nécessite une licence séparée). Le fichier **DoTransform.bat** traite des composants dans le même ordre qu'ils ont été définis dans le mappage, du haut en bas. Cela peut être vérifié en regardant le paramètre --output de chaque transformation.

```
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-by.xml" --xml-
validation-error-as-warning=true %* "MappingMapTogroups.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-adjacent.xml" --
xml-validation-error-as-warning=true %* "MappingMapTogroups2.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
```

= key = value

```
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-into-blocks.xml" --
xml-validation-error-as-warning=true %* "MappingMapTogroups3.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v2.xml" --output="group-starting-
with.xml" --xml-validation-error-as-warning=true %* "MappingMapTogroups4.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v3.xml" --output="group_ending_with.xml"
--xml-validation-error-as-warning=true %* "MappingMapTogroups5.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
```

La dernière transformation produit un fichier de sortie appelée **group-ending-with.xml**. Déplaçons ce composant de cible sur le mappage tout en haut :



Si vous générez à nouveau le code XSLT 2.0, l'ordre de traitement change en conséquence :

```
RaptorXML xslt --xslt-version=2 --input="records-v3.xml" --output="group_ending_with.xml"
--xml-validation-error-as-warning=true %* "MappingMapTogroups.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-by.xml" --xml-
validation-error-as-warning=true %* "MappingMapTogroups2.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-adjacent.xml" --
xml-validation-error-as-warning=true %* "MappingMapTogroups3.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-into-blocks.xml" --
xml-validation-error-as-warning=true %* "MappingMapTogroups4.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v2.xml" --output="group-starting-
with.xml" --xml-validation-error-as-warning=true %* "MappingMapTogroups4.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v2.xml" --output="group-starting-
with.xml" --xml-validation-error-as-warning=true %* "MappingMapTogroups4.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v2.xml" --output="group-starting-
with.xml" --xml-validation-error-as-warning=true %* "MappingMapTogroups5.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
```

Dans l'extrait de code ci-dessus, le premier appel produit maintenant group-ending-with.xml.

Vous pouvez changer l'ordre de traitement d'une manière semblable que dans d'autres langages de code et dans les fichiers d'exécution compilés de MapForceServer (.mfx).

#### Mappages enchaînés

La même séquence de traitement telle que décrite ci-dessus, est suivie pour les mappages enchaînés. Néanmoins, le groupe de mappage enchaîné est pris en tant qu'une unité. Le repositionnement du composant intermédiaire ou de cible final d'un seul mappage enchaîné n'a pas d'effet sur la séquence de traitement. Ce n'est que si plusieurs "chaînes" ou plusieurs composants de cible existent dans un mappage que la position des composants de cible finaux de chaque groupe détermine lequel est traité en premier.

- Si deux composants finaux de cible ont la même position verticale, alors celui placé à l'extrême gauche prend précédence.
- Si deux composants finaux de cible ont la même position horizontale, alors celui placé à le plus haut prend précédence.
- Dans la situation improbable que les composants aient exactement la même position, une ID de composant interne unique sera utilisée automatiquement, ce qui garantit un ordre bien défini mais qui ne peut pas être changé.

## 7.4 Traiter plusieurs fichiers d'entrée ou de sortie dynamiquement

Vous pouvez configurer MapForce pour qu'il traite plusieurs fichiers (par exemple, tous les fichiers dans un répertoire) lorsque le mappage est en cours. L'utilisation de cette fonction vous permettra de résoudre les tâches suivantes :

- Fournir dans le mappage une liste des fichiers d'entrée à traiter
- Générer en tant que sortie de mappage une liste des fichiers au lieu du fichier de sortie unique
- Générer une application de mappage dans laquelle les noms du fichier d'entrée et de sortie sont tous les deux définis au moment du runtime
- Convertir un ensemble de fichiers dans un autre format
- Partager un fichier volumineux (ou une base de données) dans des parties plus petites
- Fusionner plusieurs fichiers en un grand fichier (ou les charger dans une base de données)

Vous pouvez configurer un composant MapForce pour traiter plusieurs fichiers d'une des manières suivantes :

- Fournir le chemin vers le(s) fichier(s) d'entrée ou de sortie en utilisant des caractères génériques au lieu d'un nom de fichier fixe, dans les paramètres de composant (voir <u>Changer les paramètres du composants</u>). Concrètement, vous pouvez saisir les caractères génériques \* et ? dans le dialogue Paramètres de composant, afin que MapForce résolve le chemin correspondant lorsque le mappage est exécuté.
- Connecter une séquence au nœud racine d'un composant. Celle-ci fournit le chemin dynamiquement (par exemple, le résultat de la fonction replace-fileext). Lorsque le mappage est exécuté, MapForce lira dynamiquement tous les fichiers d'entrée ou générer dynamiquement tous les fichiers de sortie.

Selon les objectifs que vous souhaitez atteindre, vous pouvez utiliser soit une de ces approches ou les deux sur le même mappage. Néanmoins, il n'est pas efficace d'utiliser les deux approches en même temps sur le même composant. Pour indiquer à MapForce quelle approche vous souhaitez utiliser pour un composant particulier, cliquer sur la touche **Fichier** (<u>File</u>) ou **Fichier/String** (<u>File/String</u>) disponible à côté du nœud racine d'un composant. Cette touche vous permet de spécifier le comportement suivant :

<i>Utiliser les noms de fichier depuis les Paramètres de composant</i>	Si le composant doit traiter un ou plusieurs fichiers d'instance, cette option indique à MapForce de traiter le(s) nom(s) de fichier définis dans le dialogue Paramètres de composant.
	Si vous choisissez cette option, le nœud de racine ne nécessite pas de connecteur d'entrée, étant donné que cela n'est pas utile.

	Si vous n'avez pas spécifié de fichiers d'entrée ou de sortie dans le dialogue Paramètres de composant, le nom du nœud racine est <b>Fichier : (défaut)</b> . Sinon, le nœud racine affiche le nom du fichier d'entrée, suivi par un point-virgule (;), suivi par le nom du fichier de sortie. Si le nom de l'entrée est le même qu'avec celui du fichier de sortie, il sera affiché en tant que nom du nœud racine.
	Image: Second system       Image: Second system         Image: Second
Utiliser les noms de fichier dynamiques fournis par le mappage	Cette option indique à MapForce de traiter le(s) nom(s) de fichier que vous définissez dans la zone de mappage, en connectent les valeurs au nœud de racine du composant. Si vous choisissez cette option, le nœud de racine reçoit un connecteur d'entrée auquel vous pouvez connecter des valeurs qui fournissent dynamiquement les noms de fichier à traiter pendant l'exécution de mappage. Si vous avez aussi défini les noms de fichier dans le dialogue Paramètres de composant, ces valeurs sont ignorées. Lorsque cette option est sélectionnée, le nom du nœud de racine est affiché en tant que <b>Fichier : <dynamic></dynamic></b> .

	Image: Cette option est exclusive mutuellement avec l'option Utiliser les noms de fichier depuis les Paramètres de composant.
Parser strings vers XML, Parser strings vers JSON, Parser strings vers CSV, Parser Strings vers FLF, Parse strings vers EDI	Lorsqu'elle est activée, cette option permet au composant d'accepter une valeur de string en tant qu'entrée dans le nœud de racine et de la convertir dans une structure XML, JSON, CSV, FLF ou EDI, respectivement. Pour plus d'informations, voir <u>Parser et sérialiser les strings</u> <sup>904</sup> .
Sérialiser XML vers Strings, Sérialiser JSON vers Strings, Sérialiser CSV vers Strings, Sérialiser FLF vers Strings, Sérialiser EDI vers Strings	Lorsqu'elle est activée, cette option permet au composant d'accepter une structure en tant qu'entrée, et de la convertir en un string. La structure d'entrée peut être XML, JSON, CSV, Champ de longueur fixe, ou EDI, respectivement. Pour plus d'informations, voir <u>Parser et</u> <u>sérialiser les strings</u> <sup>604</sup> .

Plusieurs fichiers d'entrée ou de sortie peuvent être définis pour les composants suivants :

- Fichiers XML
- Fichiers texte (CSV\*, fichiers FLF\* et fichiers FlexText\*\*)
- Documents EDI\*\*
- Feuilles de calcul Excel\*\*
- Documents XBRL\*\*
- Fichiers JSON\*\*
- Fichiers Protocol Buffers\*\*

\* Nécessite MapForce Professional Edition

\*\* Nécessite MapForce Enterprise Edition

La table suivante illustre la prise en charge pour les fichiers d'entrée et de sortie dynamiques et les caractères génériques dans les langages MapForce.

Langage cible	Nom de fichier d'entrée dynamique	Prise en charge de caractère générique pour le nom de fichier d'entrée	nom de fichier de sortie dynamique
XSLT 1.0	*	N'est pas pris en charge par XSLT 1.0	N'est pas pris en charge par XSLT 1.0
XSLT 2.0	*	*(1)	*

XQuery	*	*(1)	N'est pas pris en charge par XQuery
C++	*	*	*
C#	*	*	*
Java	*	*	*
BUILT-IN	*	*	*

Légende :

*	Pris en charge
(1)	XSLT 2.0, XSLT 3.0 et XQuery utilisent la fonction fn:collection. La mise en place dans les moteurs Altova XSLT 2.0, XSLT 3.0 et XQuery résolvent des caractères génériques. D'autres moteurs peuvent se comporter différemment.

# 7.4.1 Mapper plusieurs fichiers d'entrée dans un seul fichier de sortie

Pour traiter plusieurs fichiers d'entrée, procéder comme suit :

Saisir un chemin de fichier avec des caractères génériques (\* ou ?) en tant que fichier d'entrée dans le dialogue Paramètres de composant. Tous les fichiers correspondants seront traités. L'exemple cidessous utilise le caractère générique \* dans le fichier XML d'entrée pour fournir tous les fichiers en tant qu'entrée de mappage dont le nom commence avec "Nanonull-". Plusieurs fichiers d'entrée sont fusionnés dans un seul fichier de sortie parce qu'il n'y a pas de connecteur dynamique vers le composant de cible, alors que le composant de source accède à plusieurs fichiers avec le caractère générique \*. Veuillez noter que le nom du nœud de racine dans le composant cible est Fichier:
 <default>, indiquant qu'aucun chemin de fichier de sortie n'a été défini dans le dialogue de Paramètres de composant. Les fichiers de source sont donc ajoutés dans le document cible.



MergeMultipleFiles.mfd (MapForce Basic Edition)

Mapper une séquence de strings dans le nœud File du composant de source. Chaque string dans la séquence représente un nom de fichier. Les strings peuvent aussi contenir des caractères génériques qui sont résolus automatiquement. Une séquence de noms de fichier peut être fournie par des composants comme un fichier XML, des champs de saisie de base de données.



MergeMultipleFiles\_List.mfd (MapForce Basic Edition)

# 7.4.2 Mapper plusieurs fichiers d'entrée dans plusieurs fichiers de sortie

Pour mapper plusieurs fichiers dans plusieurs fichiers cible, vous devez générer des noms de fichiers de sortie uniques. Dans certains cas, les noms de fichier de sortie peuvent être dérivés depuis de strings dans les données d'entrée, et dans d'autres cas, il est utile de dériver le nom de fichier de sortie du nom de fichier d'entrée, par ex. en changeant l'extension de fichier.

Dans le mappage suivant, le nom de fichier de sortie est dérivé du nom de fichier d'entrée, en ajoutant le préfixe "Persons-" à l'aide de la fonction concat.



MultipleInputToMultipleOutputFiles.mfd (MapForce Basic Edition)

**Note :** Éviter simplement de connecter les nœuds de racine d'entrée et de sortie, sans utiliser des fonctions de traitement. Sinon, vous écraserez vos fichiers d'entrée lorsque vous exécutez le mappage. Vous pouvez changer les noms de fichier de sortie en utilisant des fonctions comme la fonction concat, comme indiqué ci-dessus.

L'option de menu **Fichier | Paramètres de mappage** vous permet de définir globalement les paramètres de chemin de fichier utilisés par le mappage (voir <u>Changer les paramètres de mappage</u><sup>112</sup>).

# 7.4.3 Fournir les noms de fichier en tant que Paramètres de mappage

Pour fournir les noms de fichier personnalisés en tant que paramètres d'entrée dans le mappage, procéder comme suit :

- Ajouter un composant d'entrée simple dans le mappage (dans le menu Fonction, cliquer sur Insérer entrée). Pour plus d'informations concernant de tels composants, voir <u>Fournir des Paramètres dans le</u> <u>mappage</u><sup>477</sup>.
- 2. Cliquer sur la touche Fichier (<u>File</u>) ou Fichier/String (<u>File/String</u>) du composant de source et sélectionner Utiliser les noms de fichiers dynamiques fournis par les mappages.
- 3. Connecter le composant d'entrée simple dans le nœud de racine du composant qui agit en tant que source de mappage.

Pour un exemple, voir Exemple: Utiliser les noms de fichier en tant que paramètres de mappage<sup>482</sup>.

## 7.4.4 Consulter plusieurs fichiers de sortie

Cliquer sur l'onglet Sortie pour afficher le résultat de mappage dans une fenêtre d'aperçu. Si le mappage produit plusieurs fichiers de sortie, chaque fichier possède son propre volet numéroté dans l'onglet Sortie. Cliquer sur la touche fléchée pour voir les fichiers de sortie individuels.

	×	Aperçu 1 de 1 (2) C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\Persons-Nanonull-Branch.xml 🗸
1	E	
2 3 4		<first>Steve</first> <last>Meier</last> -
5 6 7	¢	<pre>     <person role="Accounts Receivable">         Theo<!--/First-->         Bone  </person></pre>
8 9	¢	<person role="PR &amp; Marketing Manager US"></person>
10 11 12		<pre></pre>
13 14 15	¢	<pre>&gt; <person role="IT Manager">      <first>Valentin</first>      <li>ast&gt;Bass</li> </person></pre>
16 17	¢	
18 19 20		<pre></pre>
21 22 23	¢	<person role="Support Engineer"> Mark   Reduceed</person>
24 25		

MultipleInputToMultipleOutputFiles.mfd

Pour enregistrer les fichiers de sortie générés, suivre une des étapes suivantes :

- Dans le menu Sortie, cliquer sur Enregistrer tous les fichiers de sortie ( 🗐 ).
- Cliquer sur la touche de barre d'outils Enregistrer toutes les sorties générées ( 🗐 ).

## 7.4.5 Exemple : Séparer un fichier XML en plusieurs fichiers

Cet exemple vous montre comment générer dynamiquement plusieurs fichiers XML depuis un seul fichier XML source. Le mappage d'accompagnement pour cet exemple est disponible sous le chemin suivant : <br/>
<Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\Tut-ExpReport-dyn.mfd.

Le fichier XML de source (disponible dans le même dossier que le mappage) contient les notes de frais d'une personne appelée "Fred Landis" et contient cinq postes de dépenses de types différents. L'objectif de cet exemple est de générer un fichier XML séparé pour chacun des postes de dépense recensé ci-dessous.

A Person									
First			Fred						
() Last			Landis						
() Title			Project Manager						
() Phone 123-			123-456-78	23-456-78					
Email f.landis@nanonull.com									
A expense-item (5)									
		= type	= expto	() Date	Travel	Lodging			
	1	Travel	Development	2003-01-02	🗹 Travel Trav-cost=337.88				
	2	Lodging	Sales	2003-01-01		🔳 Lodging l			
	3	Travel	Accounting	2003-07-07	Travel Trav-cost=1014.22				
	4	Travel	Marketing	2003-02-02	🔳 Travel Trav-cost=2000				
	5	Meal	Sales	2003-03-03					

mf-ExpReport.xml (as shown in XMLSpy Grid view)

Étant donné que l'attribut type défini le type d'item de dépense spécifique, voici l'item que nous allons utiliser pour séparer le fichier de source. Pour atteindre l'objectif de cet exemple, procéder comme suit :

- Insérer une fonction concat (vous pouvez la glisser depuis la bibliothèque core | fonctions string du volet Bibliothèques).
- 2. Insérer une constante (dans le menu **Insérer**, cliquer sur **Constante**) et saisir ".xml" en tant que sa valeur.
- Insérer la fonction <u>auto-number</u> (vous pouvez la glisser depuis la bibliothèque core | fonctions générateur du volet Bibliothèques).
- 4. Cliquer sur la touche Fichier ( File ) ou Fichier/String ( File/String ) du composant cible et sélectionner Utiliser les noms de fichier dynamique fournis par le mappage.
- 5. Créer les connexions tel qu'affiché ci-dessous puis cliquer sur l'onglet **Sortie** pour voir le résultat du mappage.


Tut-ExpReport-dyn.mfd (MapForce Basic Edition)

Veuillez noter que les fichiers de sortie résultants sont nommés dynamiquement comme suit :

- L'attribut type fournit la première partie du nom de fichier (par exemple, "Travel").
- La fonction <u>auto-number</u> fournit le numéro séquentiel du fichier (par exemple, "Travel1", "Travel2", etc.).
- La constante fournit l'extension de fichier, qui est ".xml", donc "Travel1.xml" est le nom de fichier du premier fichier.

# 7.4.6 Exemple : Séparer une table de base de données dans plusieurs fichiers XML

Cet exemple vous montre comment générer dynamiquement plusieurs fichiers XML, un pour chaque enregistrement d'une table de base de données. Le mappage d'accompagnement pour cet exemple est

disponible sous le chemin suivant :

#### <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\PersonDB-dyn.mfd.

Le fichier de base de données de source (disponible dans le même dossier que le mappage) inclut une table Person qui contient 21 enregistrements. L'objectif de cet exemple est de générer un fichier XML séparé pour chaque enregistrement dans la table Person.

Person					
PrimaryKey 👻	ForeignKey 👻	EMail 👻	First 🔹	Last 👻	
1	1	v.callaby@nanonu	Vernon	Callaby	
2	1	f.further@nanonu	Frank	Further	
3	և	l.matise@nanonu	Loby	Matise	
4	2	j.firstbread@nanc	Joe	Firstbread	
5	2	s.sanna@nanonul	Susi	Sanna	
6	3	f.landis@nanonul	Fred	Landis	
7	3	m.landis@nanonu	Michelle	Butler	
8	3	t.little@nanonull.	Ted	Little	

Étant donné que le champ "PrimaryKey" identifie uniquement chaque personne dans la table, il s'agit de l'item que nous allons utiliser pour séparer la base de données de source dans des fichiers séparés. Pour atteindre l'objectif de cet exemple, procéder comme suit :

- Insérer une fonction concat (vous pouvez la glisser depuis la bibliothèque core | fonctions string du volet Bibliothèques)
- 2. Insérer une constante (dans le menu **Insérer**, cliquer sur **Constante**) et saisir ".xml" en tant que sa valeur.
- 3. Cliquer sur la touche Fichier ( File ) ou Fichier/String ( File/String ) du composant cible et sélectionner Utiliser les noms de fichier dynamique fournis par le mappage.
- 4. Créer les connexions tel qu'affiché ci-dessous puis cliquer sur l'onglet **Sortie** pour voir le résultat du mappage.



PersonDB-dyn.mfd (MapForce Professional Edition)

Veuillez noter que les fichiers de sortie résultants sont nommés dynamiquement comme suit :

- Le champ **PrimaryKey** fournit la première partie du nom de fichier (par exemple, "1").
- La constante fournit l'extension de fichier (".xml"), donc "1.xml" est le nom de fichier du premier fichier.

# 7.5 Parser et sérialiser des strings

Le parsage et la sérialisation de string est une technique avancée de mappage qui vous permet de configurer le composant soit pour parser les données depuis un string, soit pour sérialiser les données vers un string. Cette technique peut être considérée comme une alternative à la lecture des données depuis des fichiers (ou d'écrire des données dans des fichiers). Les composants MapForce qui parsent les strings ou sérialisent des données dans des strings peuvent être utiles dans un grand nombre de situations, par exemple si :

- vous souhaitez insérer des structures comme des fichiers XML dans des champs de base de données.
- vous souhaitez convertir des fragments XML stockés dans des champs de base de données dans des fichiers XML autonomes.
- vous avez des données legacy stockées en tant que texte (par exemple, un contenu de longueur fixe dans un seul champ de base de données), et que vous souhaitez convertir ces données dans une structure entièrement triable et basée sur champs

Le parsage de string et la sérialisation est disponible pour les types de composant MapForce suivants :

- Texte (CSV, texte à champ de longueur fixe)
- Fichier de schéma XML

Le parsage de string et la sérialisation sont pris en charge pour les types de composant MapForce suivants :

- Texte (CSV, champ de saisie de longueur fixe)
- Fichiers de schéma XML

Pour tous les types de composants ci-dessus, le parsage de string et la sérialisation sont pris en charge dans le langage cible BUILT-IN. De plus, le parsage de strings dans JSON ou la sérialisation de JSON depuis des strings est prise en charge dans BUILT-IN, C# et Java.

# 7.5.1 À propos du composant Parser/Sérialiser

Un composant Parser/Sérialiser dans MapForce est un composant hybride qui n'est ni un composant source ni un composant cible. Selon le rôle qu'ils jouent dans la conception de mappage, ce genre de composants doivent être placés entre d'autres composants source et cible.

Vous pouvez utiliser le composant "Parser/Sérialiser String" pour le parsage du string lorsque, pour une raison ou une autre, vous devez convertir un string qui a une structure (par exemple, quelque XML stocké en tant que string dans une base de données) en un format différent. Le parsage de données depuis le string de source dans le composant "Parser/Sérialiser" signifie que le string de source est transformé en une structure MapForce ; ainsi, vous obtenez l'accès à tout élément ou attribut du XML de source stocké en tant que string.



Composant générique "Parser String"

Le diagramme ci-dessus illustre la structure typique d'un composant MapForce qui parse un string. Veuillez noter que le composant "Parser/Sérialiser String" est placé entre la source et la cible du mappage. Le fonctionnement de ce composant est d'accepter certaines structures de string en tant qu'entrée, par le biais d'un seul connecteur MapForce connecté à son nœud **String** supérieur. La structure de sortie peut être une des cibles de données prises en charge par MapForce.

Lorsque vous sérialisez des données depuis un composant vers un string, c'est l'inverse qui se produit. L'ensemble de la structure du composant MapForce devient une structure de string que vous pouvez manipuler ultérieurement, si vous le souhaitez. Par exemple, ceci permet d'écrire un fichier XML (ou fragment XML) dans un champ de base de données ou dans une seule cellule d'une feuilles de calcul Excel.



Composant générique "Sérialiser vers String"

Le diagramme ci-dessus illustre un composant générique MapForce "Sérialiser vers String". Ce composant en question accepte toute source de données comme entrée prise en charge par MapForce (par le biais de

connecteurs MapForce standard). La structure de sortie est un string que vous pouvez passer par la suite par le biais d'un connecteur MapForce unique appartenant au nœud **String** supérieur du composant à l'item de composant cible (par exemple, une cellule de feuille de calcul). Pour consulter un exemple, voir <u>Exemple :</u> <u>Sérialiser vers un String (XML à base de données)</u><sup>907</sup>.

Vous pouvez désigner un composant pour le parsage ou la sérialisation de string à tout moment depuis la fenêtre de mappage. Pour ce faire, cliquez sur la touche **Fichier/String** (<u>File/String</u>) adjacente au nœud de racine, puis sélectionnez l'option désirée.



Modifier le mode de composant

**Note :** un composant "Parser/Sérialiser String" ne peut pas lire des données depuis un string et l'écrire dans un string simultanément. C'est pourquoi, le nœud de racine peut soit avoir un connecteur entrant, soit un connecteur sortant (pas les deux). Une erreur sera générée si vous tentez d'utiliser le même composant pour les deux opérations.

Lorsque vous désignez un composant pour le parsage ou la sérialisation de string, l'apparence du composant change comme suit :

- Le composant obtient le préfixe parse ou serialize dans le titre.
- La barre de titre présente une couleur d'arrière-plan jaune, semblable aux composants de fonction.
- Le nœud supérieur débute avec le préfixe String: et est identifié par l'icône 🗐 .
- Si le composant parse un string, le connecteur de sortie depuis le nœud de racine n'est pas significatif et n'est donc pas disponible.
- Si le composant est sérialisé vers un string, le connecteur d'entrée dans le nœud de racine n'est pas significatif et n'est donc pas disponible.

Lorsqu'un composant se trouve dans le mode "Parser/Sérialiser String", vous pouvez modifier ses paramètres comme s'il se trouvait en mode basé sur fichier (voir <u>Changer les paramètres du composants</u><sup>(80)</sup>). Veuillez noter que tous les paramètres de composant ne sont pas disponibles lorsqu'un composant se trouve soit en mode "Parser" ou "Sérialiser".

# 7.5.2 Exemple : sérialiser vers un string (XML vers base de données)

Cet exemple vous guide dans les étapes nécessaires pour créer un design de mappage qui sérialise les données dans un string. L'exemple est accompagné d'un fichier d'échantillon. Si vous souhaitez consulter le fichier échantillon avant de vous lancer dans cet exemple, vous pouvez l'ouvrir depuis le chemin suivant : 

<a href="mailto:consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-consulta-con

Partons d'un exemple dans lequel vous avez un fichier XML (et son schéma lié) qui consiste en plusieurs éléments <Person>. Chaque élément <Person> réunit le prénom d'une personne, le nom de famille, le titre professionnel, l'extension téléphonique et l'adresse e-mail, comme suit :

```
<Person>

<First>Joe</First>

<Last>Firstbread</Last>

<Title>Marketing Manager Europe</Title>

<PhoneExt>621</PhoneExt>
```

Votre objectif est d'extraire chaque élément <Person> depuis le fichier XML et de l'insérer littéralement (y compris les balises XML) en tant que nouvel enregistrement de base de données dans la table PEOPLE d'une base de données SQLite. La table PEOPLE contient uniquement deux colonnes : ID et PERSON. Sa définition complète est la suivante :

CREATE TABLE PEOPLE (ID INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, PERSON TEXT);

Une fois que le mappage a été exécuté, le résultat attendu est que la table **PEOPLE** aura le même nombre de lignes que celui des éléments <Person> dans le fichier XML.

Pour atteindre cet objectif, procéder comme suit :

- Ajouter à la zone de mappage le composant XML de source (utiliser la commande de menu Insérer | Schéma XML/Fichier). Le fichier échantillon est disponible sous : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\MFCompany.xml.
- 2. Dupliquer (copier-coller) le composant XML.
- 3. Sur le composant XML dupliqué, cliquer sur File/String, puis sélectionner Sérialiser XML sur Strings.



4. Cliquer avec la touche de droite sur le composant dupliqué et choisir **Changer l'élément racine** depuis le menu contextuel. Puis changer l'élément de racine sur <Person>.

😫 Sélectionner élément racine		×
Nom	Espace de noms URI	~
Company	http://my-company.com/namespace	
Person	http://my-company.com/namespace	
		~
	OK Annuler	

En général, vous pouvez changer l'élément de racine à tout élément qui a une déclaration globale (pas locale) dans le schéma XML. Tout élément qui n'est pas défini globalement dans votre schéma n'est pas listé dans le dialogue "Sélectionner l'élément racine".

5. Double-cliquer le composant et décocher la case Écrire déclaration XML. Cela évite l'écriture de la déclaration XML pour chaque élément <Person>.

😫 Paramètres de co	omposant		×				
Nom de composant:	MFCompany						
MFCompany.xsd		Parcourir	Éditer				
		1 di codini	Luiter				
MECompany.xml		Parcourin	Édition				
		Parcourii	Luidon				
Fichier XML de sortie			<u> </u>				
MFCompany.xmi		Parcourir	Editer				
Préfixe pour l'espace	de noms de cible :						
Ajouter référence	schéma/DTD (laisser le champ vide j	pour un accès ab	solu du schéma):				
Écrire la déclaratio	on XML						
Convertir les vale	urs en types cible (désactiver pour c	onserver le forma	atage des				
Valeurs numerique	es ou de date au risque d'ecrire une :	sortie invalide)					
Créer signature n	umérique (exécution intégrée)	Paramètres o	de signature				
En cas d'échec d	le la création : O Arrêter le traite	ment					
	Continuer sans	signature					
Encodage de sortie	2						
Nom d'encodage:	Unicode UTF-8		$\sim$				
Ordre des octets:	Little Endian 🛛 🗸	Inclure indica	teur				
Fichier de Feuille de s	Fichier de Feuille de style StyleVision Power						
		Parcourir	Créer				
Activer les optimis	Activer les optimisations de traitement d'entrée sur la base des min/maxOccurs						
Enregistrer tous le	es chemins de fichier relatifs au fichie	er MFD					
		ОК	Annuler				

 Ajouter à la zone de mappage le composant de base de données SQLite cible, depuis le champ suivant : <Documents>\Altova\MapForce2023\MapForceExamples\Tutorial\\dbserialize.db. (Pour ajouter le composant de base de données, utiliser la commande de menu Insérer | Base de données, voir aussi Connexion à une base de données<sup>(13)</sup>). Lorsque vous serez invité à insérer un objet de base de données, choisir la table PEOPLE.

🕩 Insérer objets de base de données	
Veuillez choisir les Tables, Aperçus, Procédures stockées ou cr SELECT définies par l'utilisateur sont stockées localement dans	éez vos propres instructi le fichier mfd.
Source	
▼ 🗸 🕈 Ajouter/Modifier l'instruction SELECT	Ajouter/Modifier rela
dbserialize2     main     Tables     PEOPLE     Instructions SELECT	
Afficher aperçu >> Utiliser des noms d'objet par rapport	au schéma par défaut

7. Lier les composants comme indiqué ci-dessous. À gauche du mappage, l'élément <Person> mappe dans un composant de sérialisation. À droite du mappage, la valeur de string sérialisée est insérée dans la colonne PERSON de la table de la base de données PEOPLE. Enfin, le connecteur dessiné depuis <Person> vers la table PEOPLE indique à MapForce de créer un nouvel enregistrement pour chaque élément <Person> rencontré.



- 8. Cliquer sur la touche A: In dans le composant de base de données, et procéder comme suit :
  - a. Sélectionner l'option **Supprimer tous les enregistrements**. Au moment du mappage, tous les enregistrements existants seront supprimés de la base de données avant d'en insérer de nouveaux
  - b. Sélectionner l'option généré en BD à côté de la colonne ID. Cela garantit que l'ID de l'enregistrement sera généré par la base de données. Noter que l'option généré en BD apparaît

aussi si la colonne prend en charge cette option. Pour les colonnes qui ne sont pas une identité ou un champ auto-incrémenté, l'option **max+1** est disponible à la place, cette option analysera quelle est la valeur ID maximum existant dans la base de données et d'insérer l'entier disponible suivant, incrémenté de 1.

😤 Actions de table de base de données - P	EOPLE		×
Instruction SQL à exécuter avant le premier e Aucune SUPPRIMER tous les enregistrements SQL personnalisé :	nregistrement ]supprimer aussi les enregistrements da	ans les tables enfants	
Actions à exécuter pour chaque enregistreme Les données d'entrée sont comparées aux dor Cocher la case "NULL equal" pour traiter les va Si toutes les comparaisons sont vraies, l'action	nt Inées de la table de BD, avec les opérat aleurs NULL en tant qu'égales. I spécifique est exécutée.	teurs définis ici.	
Action sur enregistrement	NULL égale Insérer tout	▼ ∧	Apposer action
D ID	max() + 1	<b>•</b>	
D PERSON	valeur mappée	• 💌	Insérer action

Vous venez de créer un design de mappage qui sérialise les données en string. Si vous cliquez sur l'onglet **Sortie**, la requête SQL d'aperçu indique que des enregistrements séparés seront insérés dans la base de données pour chaque élément <Person> dans le fichier XML, ce qui était l'objectif de ce mappage.

# 7.6 Volets de Sortie StyleVision

Dans les mappages où le composant cible est XML, il est possible de consulter et enregistrer la sortie de mappage en tant qu'HTML, RTF, PDF, Word 2007+, et les documents texte si <u>Altova StyleVision</u> est installée sur votre ordinateur. Si vous utilisez l'édition Enterprise de StyleVision, les graphiques seront aussi rendus dans ces aperçus. Lorsqu'un mappage prend en charge la consultation dans l'un de ces formats, des volets supplémentaires deviennent disponibles à côté du volet **Sortie** (*voir la capture d'écran ci-dessous*).

#### Important

- Si StyleVision Professional est installé, il est possible de voir des sorties HTML, RTF et Texte.
- Avec StyleVision Enterprise, il est possible d'apercevoir la sortie HTML, RTF, PDF et Word 2007+.
- La consultation de la sortie de mappage en tant que PDF exige Java, Acrobat Reader, et FOP (Formatting Objects Processor) version 0.93 ou 1.0. FOP est installé avec StyleVision, à moins que vous ayez choisi de ne pas l'installer pendant l'installation de StyleVision.
- Dans l'édition 64-bit de MapForce, les aperçus Word 2007+ et RTF sont ouverts en tant qu'applications non intégrées.
- Si votre mappage contient des composants qui agissent à la fois en tant que sources et cibles (composants pass-through), la consultation StyleVision sera possible uniquement pour les composants où le bouton **Preview** 
   du composant a été activé. Pour plus d'informations concernant ce type de mappages, voir <u>Mappages en chaîne</u><sup>641</sup>.

Pour consulter vos données de mappage dans les volets de sortie StyleVision, l'étape suivante est requise :

- Altova StyleVision doit être installée sur votre ordinateur soit en tant qu'installation standalone ou faisant partie de Altova MissionKit.
- Le composant cible doit avoir un fichier StyleVision Power Stylesheet (SPS) qui y est associé. Le fichier de feuille de style peut être créé ou édité avec StyleVision. Vous ne pouvez pas éditer ou modifier la feuille de style directement dans MapForce, mais vous pouvez l'ouvrir par le biais de MapForce dans StyleVision. Une fois que la feuille de style est prête, vous pouvez l'assigner à un composant MapForce cible, comme indiqué ci-dessous.

### Configuration des volets de sortie StyleVision

Les instructions ci-dessous vous aideront à définir les volets de sortie StyleVision.

#### Attribuer une feuille de style StyleVision Power à un composant cible

Pour attribuer un fichier SPS à un composant cible, suivez les étapes suivantes :

- 1. Dans StyleVision, créez le fichier de feuille de style requis. Assurez-vous d'utiliser le même schéma XML en tant que source que pour le composant MapForce.
- 2. Dans MapForce, cliquez avec la touche de droite sur le composant XML et sélectionnez Propriétés.
- Dans le dialogue Paramètres de composant, situé à côté du fichier StyleVision Power Stylesheet, cherchez le fichier de feuille de style créé précédemment (voir la capture d'écran cidessous).

Fichier de Feuille de style StyleVision Power		
CompletePO.sps	Parcourir	Éditer

Note :le chemin vers le fichier StyleVision Power Stylesheet peut être absolu ou relatif. Pour des détails, voir aussi <u>Chemins relatifs et absolus</u><sup>83</sup>.

#### Enregistrer la sortie générée par StyleVision

Vous pouvez enregistrer la sortie générée par StyleVision sur un fichier de la même manière que pour enregistrer le résultat de tout autre mappage. Cliquez sur le bouton de la barre d'outils (Enregistrer sortie générée) ou allez au menu de Sortie et cliquez sur Enregistrer fichier de sortie.

#### Automatiser la génération de différents formats avec des produits Altova

Si vous voulez que votre mappage génère HTML, PDF, RTF, Word 2007+ et des fichiers de texte automatiquement (soit sur le même ordinateur, soit sur un autre ordinateur ou même une plateforme), vous pouvez utiliser <u>Serveur MapForce</u> ou <u>StyleVision Server</u>. Il s'agit de produits de serveur séparément sous licence qui élargissent respectivement la fonctionnalité de MapForce et StyleVision. Dans ce scénario, chaque application joue son propre rôle spécifique :

- MapForce vous permet de concevoir un mappage (.mfd) qui définit les entrées et sorties de transformation de données (par exemple, base de données à XML)
- Le serveur MapForce exécute le mappage exécutable (.mfx) depuis la ligne de commande ou depuis un API (soit dans le même système d'exploitation, soit dans un autre).
- StyleVision vous permet de concevoir la feuille de style (.sps) requise pour transformer la sortie de mappage en HTML, PDF, RTF, Word 2007+ et les fichiers de texte.
- Le serveur de StyleVision Server exécute la feuille de style .sps qui transforme la sortie de mappage en un format désiré. Ceci est réalisé dans la ligne de commande ou depuis un API (sur le même système d'exploitation ou différent).
- Les deux serveurs StyleVision Server et MapForce peuvent optionnellement être exécutés sous la gestion du <u>Serveur FlowForce</u> (sous licence séparément). Dans ce scénario, les mappages MapForce et transformations StyleVision peuvent être exécutés comme prévus, déclenchés ou des tâches sur demande. Ceci signifie que les mappages MapForce et transformations StyleVision peuvent être entièrement automatisés.

### Exemples

L'exemple ci-dessous (MapForceExamples\CompletePo.mfd) affiche la sortie dans le volet de sortie de StyleVision appelé HTML. Ce mappage produit une commande d'achat en format XML. Cliquez avec la touche de droite sur le composant cible, sélectionnez **Propriétés**, et constatez qu'un fichier .sps lui a été attribué.

Si vous cliquez sur le volet HTML, vous verrez la sortie suivante :

					Fax +1 (32. office ww	1) 555 5155 - 9 @nanonull.com w.nanonull.com
Purch	ase Orde	er Numbe	r: PO			
	то:		Mr	s./Mr.	Ted Little	
		Lo	ng Way			
		Lo	s-Angeles			
		CA	34424			
		Ou	r Customer	Identifi	er: ID-3	
Order	r Date:					
Shipp	ing Date:					
	<b>a</b>				1	
Item	Quantity	Name	Unit Price (\$)	10tal (\$)		
	5	Pants	34	1/0		
	17	T-Shirt	25	425		
				595	]	
Other	C					
Other	Commen	ls:				
Authorized Signature Data						
Authori	zeu signature		Date			
If you h	ave any ques	tions about th	is purchase order, p	lease contact	us.	

Un autre exemple est Tutorial\Yearlysales.mfd. La feuille de style attribuée à ce mappage a été conçue dans StyleVision de manière à ce qu'il soit possible de contrôler le type de graphique en changeant la valeur de l'élément ChartType. Cela permet de changer le type de graphique directement depuis le mappage : Vous pouvez changer la valeur par défaut de la constante sur n'importe quelle valeur allant de 1 à 7. Si vous placez le

curseur de la souris sur le composant value-map, vous verrez les valeurs possibles (voir la capture d'écran cidessous).



La valeur par défaut de la constante est 2, ce qui génère un graphique camembert 3D dans la sortie. Pour afficher d'autres types de graphiques, changez cette valeur en une autre valeur autorisée et cliquez sur l'onglet **Sortie** pour voir les modifications.

# 7.7 Générer la documentation de mappage

Vous pouvez générer une documentation détaillée concernant des mappages en format HTML, Microsoft Word (.doc), ou RTF format. Si StyleVision est installé, vous pouvez en outre générer une documentation en format PDF.

#### Exigences préalables

- Microsoft Word 2000 ou plus doit être installé si vous souhaitez générer une documentation en format Microsoft Word.
- StyleVision doit être installé si vous souhaitez générer une documentation en format PDF ou personnaliser le design de la documentation générée.

Par défaut, la documentation est générée avec un design fixe, dans lequel vous pouvez configurer des options de base comme les commandes que vous souhaitez y inclure, la profondeur des chemins affichés et autres réglages. Si StyleVision est installé, vous pouvez également profiter de plusieurs fichiers StyleVision Power Stylesheets (SPS) inclus ou même créer votre propre design dans StyleVision.

#### Pour générer une documentation de mappage :

1. Dans le menu **Fichier**, cliquer sur **Générer documentation**. Cela ouvre le dialogue "Générer documentation".

Générer Documentation	×
<ul> <li>Design documentation</li> <li>Utiliser un design fixe pour la documentation en format</li> <li>Utiliser un design défini par l'utilisateur pour les formats</li> <li>Choisir design SPS : %AltovaMapForceDoc%\Function</li> </ul>	t HTML, Word ou RTF. s HTML, Word, RTF ou PDF. Nécessite StyleVision. onsCallGraph.sps v Parcourir Édition
Format sortie O HTML Microsoft Word O RTF O PDF (voir ci-dessus) Répartir résultat en plusieurs fichiers	Limite de longueur de chemin Items d'entrée Connexions Items de sortie 3 Items de sortie 3 Items de sortie
Afficher le fichier des résultats après l'avoir généré Inclure     Composants d'entrée     Composants variable     Composants de sortie     Composants restant     Fonctions définies par l'utilis.     Tout cocher     Tout décocher	Détails es Annotations Nœuds connectés tes Types de données Nœuds déconnectés Noms de bibliothèque Tout cocher Tout décocher

2. Choisir les paramètres requis et cliquer sur OK.

Les paramètres que vous pouvez configurer sont décrits ci-dessous.

### Design de documentation

- Choisir "Utiliser design fixe..." pour utiliser le modèle de documentation intégré.
- Choisir "Utiliser design défini par l'utilisateur..." pour utiliser un StyleVision Power Stylesheet prédéfinie créée dans StyleVision. Les fichiers SPS sont disponibles dans le dossier ...
   \Documents\Altova\MapForce2023\Documentation\MapForce\. Pour plus de détails, voir Feuilles de style StyleVision Power prédéfinies<sup>918</sup>.
- Cliquer sur **Parcourir** pour chercher un fichier SPS prédéfini.
- Cliquer sur Éditer pour lancer StyleVision et ouvrir la SPS sélectionnée dans une fenêtre StyleVision.

### Format de sortie

 Le format de sortie est spécifié ici : soit HTML, Microsoft Word, RTF ou PDF. Les documents Microsoft Word sont créés avec l'extension de fichier .doc lorsqu'ils sont générés en utilisant un design fixe, et avec une extension de fichier .docx lors d'une génération en utilisant une SPS StyleVision. Le format de sortie PDF exige l'utilisation de StyleVision et est uniquement disponible si vous avez choisi une SPS StyleVision.

- Choisir "Répartir la sortie sur plusieurs fichiers" si vous souhaitez générer plusieurs fichiers de documentation, un fichier pour chaque composant individuel comme le composant d'entrée ou de sortie. Dans les designs fixes, les liens entre plusieurs documents sont créés automatiquement.
- Si l'option "Afficher le fichier de résultat après génération" est activée, MapForce ouvrira les fichiers générés dans le navigateur ou l'application par défaut, le cas échéant

### Limite de longueur de chemin

Utiliser ces options pour définir la longueur maximum du "chemin" à afficher pour les items d'entrée ou de sortie ou les connexions. Par exemple, avec une longueur par défaut de **3**, un chemin d'item serait affiché comme **.../ShortPO/LineItems/LineItem**.

### Inclure

Choisir ici les composants spécifiques qui doivent être inclus dans la documentation générée.

### Détails

Utiliser ces options pour personnaliser le niveau de détail dans la documentation générée. L'option **Noms de bibliothèque** insère le préfixe "core" pour les fonctions.

# 7.7.1 Feuilles de style Power StyleVision prédéfinies

Si StyleVision est installé sur votre ordinateur, vous pouvez générer une documentation de mappage en choisissant un des fichiers StyleVision Power Stylesheet (SPS) predéfinis en tant que modèle, au lieu du design fixe intégré. Les feuilles de style SPS prédéfinies sont disponibles :

- **FunctionCallGraph.sps** montre le graphique d'appel du mappage principal et toute fonction définie par l'utilisateur.
- FunctionsUsedBy.sps montre quelles fonctions sont utilisées directement ou indirectement dans le mappage.
- ImpactAnalysis.sps recense tout nœud de source et de cible, et la route prise via les différentes fonctions vers le nœud de cible.
- **OverallDocumentation.sps** montre toutes les connexions, fonctions et nœuds de cible. Ce modèle produit le maximum de détails et est identique à la sortie "fixed design" intégrée.

Vous pouvez sélectionner la feuille de style requise à chaque fois avant de générer une documentation, comme indiqué ci-dessous. Les fichiers se trouvent dans le dossier ...\MapForce2023\Documentation\MapForce.

Générer Documentation				×
Design documentation O Utiliser un design fixe O Utiliser un design déf	pour la documentation en forma ini par l'utilisateur pour les format	t HTML, Word ou RTF. s HTML, Word, RTF ou l	PDF. Nécessite StyleVision.	
Choisir design SPS :	%AltovaMapForceDoc%\Functi	ionsCallGraph.sps	<ul> <li>Parcourir Édition</li> </ul>	
Format sortie ● HTML	%AltovaMapForceDoc%\Function %AltovaMapForceDoc%\Function %AltovaMapForceDoc%\Impact %AltovaMapForceDoc%\Overall	onsCallGraph.sps onsUsedBy.sps tAnalysis.sps IIDocumentation.sps		
		Connexions	3	
O PDF (voir ci-dessus)		✓ Items de sortie	3	

Les exemples ci-dessous illustrent la sortie produite par chacune de ces feuilles de style. Les exemples ont été générés depuis un des mappages de démonstration installés avec MapForce,

**PersonListByBranchOffice.mfd**. Bien que ces exemples illustrent spécifiquement la sortie HTML, la mise en page est semblable à d'autres formats. Pour plus d'informations concernant la création ou la personnalisation de fichiers SPS, voir <u>Feuilles de style personnalisées</u><sup>(623)</sup>.

# Feuille de style "FunctionCallGraph.sps"

PersonListByBranchOffice.mfd × +	- 0	×
$\leftarrow \rightarrow$ C $\textcircled{a}$ (i) file:///C:/Users/alto $\bigtriangledown$ $\swarrow$	III\ >>	≡
This report shows call graphs of the main mapping a defined functions.	nd all user-	^
Main mapping  core.equal  core.filter   <u>user.LookupPerson</u>    core.filter   <u>user.EqualAnd</u>		
<pre>     core.equal      core.logical-and    <u>user.Person2Details</u>      core.concat user.EqualAnd  core.equal  core.logical-and</pre>		
		~

# Feuille de style "FunctionsUsedBy.sps"

PersonListByBranch	Office.mfd × +	- 🗆	×
$\leftarrow$ $\rightarrow$ C $rac{1}{2}$	i file:///C:/Users/alto	⊍ ☆ III\ ≫	≡
This report lists all another functions. changes in user-de functions can be af Library core	functions and their din This is especially impo fined functions in order fected.	rect and indirect use in rtant for planning r to see what other	^
Function	Directly used by	Indirectly used by	
core.concat	user.Person2Details	Main mapping user.LookupPerson	
core.equal	Main mapping user.EqualAnd	user.LookupPerson	
core.filter	Main mapping user.LookupPerson		
core.logical-and	user.EqualAnd	Main mapping user.LookupPerson	
			~

# Feuille de style "ImpactAnalysis.sps"

PersonListByBranchOffice.mf	а × +	- 0	×
$\leftrightarrow$ $\rightarrow$ C $\textcircled{0}$ $\textcircled{0}$	ile:///C:/Users/alto	⊍ ☆ II\ ≫	≡
This report lists every in	nut and output no	de connection	
independently and is perfect for further impact analysis with modelling tools.			
Input Node	Functions	Output Node	
OfficeName	core.equal, core.filter	PersonList	
OfficeName	user.LookupPerson	PersonList/Person/Details	
BranchOffices/Office	core.filter	PersonList	
BranchOffices/Office/Name	core.equal, core.filter	PersonList	
BranchOffices/Office/Contact		PersonList/Person	
/Office/Contact/first		PersonList/Person/First	
/Office/Contact/first	user.LookupPerson	PersonList/Person/Details	
/Office/Contact/last		PersonList/Person/Last	
/Office/Contact/last	user.LookupPerson	PersonList/Person/Details	

### Feuille de style "OverallDocumentation.sps"

PersonListByBranchOt	ffice.mfd × +	- 0	×
$\leftarrow$ $\rightarrow$ C $rac{1}{2}$	i file:///C:/Users/altova	~ ··· ⊠ ☆ >>	≡
Mapping PersonListByBranchOffice.mfd			^
Input OfficeName			
Nodes	Connections		
OfficeName Type: string Default: <u>core.constant("Nanonull, Inc.")</u>	<u>core.equal =&gt; a   result =&gt;</u> <u>core.filter =&gt; bool   on-true =&gt;</u>	PersonList Type: restriction of xs:anyType [01] Annotation: List of Persons	
	user.LookupPerson => Office Name   result =>	PersonList/Person/Details Type: xs:string [01]	
Input BranchOffices (Branc	hOffices.xsd)		
File: BranchOffices yml	connections		_
Type: string			
BranchOffices Type: restriction of xs:anyType [01]			
BranchOffices/Name Type: restriction of xs:string			~

# 7.7.2 Feuilles de style personnalisées

Outre le design fixe intégré, vous pouvez créer des feuilles de style personnalisées pour la documentation de mappages générés avec StyleVision (<u>https://www.altova.com/stylevision</u>). Vous pouvez aussi changer une des <u>feuilles de style prédéfinies</u><sup>(918)</sup>, par exemple, en ajustant les polices d'écriture et les autres styles.

Les StyleVision Power StyleSheet (SPS) vous permettent de réaliser un design personnalisé. L'avantage d'utiliser une SPS pour générer une documentation de mappage est que vous disposez du contrôle sur le design de la documentation.

Pour créer un fichier SPS personnalisé, les conditions suivantes doivent être remplies :

 Le Schéma XML qui apporte la structure de la documentation MapForce générée. Ce schéma est appelé MapForceDocumentation.xsd et est fourni avec votre pack d'installation MapForce. Il est stocké dans le dossier ...\Documents\Altova\MapForce2023\Documentation\MapForce. Veuillez noter que MapForceDocumentation.xsd inclut le fichier Documentation.xsd situé dans le dossier ci-dessus.

#### 924 Scénarios de mappage avancé

 Certaines données d'échantillon pour tester et apercevoir le design personnalisé. Vous pouvez utiliser le fichier XML suivant en tant que données d'échantillon : ...
 \Documents\Altova\MapForce2023\Documentation\MapForce\SampleData\PersonListByBranch Office.xml.

Les fichiers mentionnés ci-dessus doivent être référencés dans la fenêtre Aperçu de design dans StyleVision, par exemple :

Ape	Aperçu de design >		
$\nabla$	Sources		
6	\$XML (princip	oal)	0
	Schéma	MapForceDocumentation.xsd	0
	XML de travail	PersonListByBranchOffice.xml	0
	Modèle XML		0
	Signature XML	Désactivé	0
	ajouter nouvelle	source	
$\nabla$	Modules		
	ajouter nouvea	u Module	
$\nabla$	Fichiers CSS	3	
	ajouter nouvea	u fichier CSS	
$\nabla$	Paramètres	;	
	ajouter nouvea	u paramètre	
$\nabla$	Fichier XSLT	Г	
	ajouter nouvea	u fichier XSLT	

Dans StyleVision, vous créez un design en glissant des nœuds depuis la fenêtre Arborescence de Schéma dans la zone de design puis y attribuez des styles et des propriétés.



Vous pouvez aussi ajouter des composants supplémentaires comme des liens et des images dans le design SPS. Pour consulter le design dans un format spécifique, cliquez sur un des onglets suivants : **HTML**, **RTF**,

**PDF**, ou **Word 2007+**. Pour plus de détails, veuillez vous référer à la StyleVision documentation (<u>https://www.altova.com/fr/documentation</u>).

# 8 Débogueur

MapForce comprend un débogueur de mappage disponible pour le langage de transformation BUILT-IN de MapForce. Le débogueur de mappage vous aide à obtenir les objectifs suivants :

- Consulter et analyser les valeurs produites par le mappage à tout niveau de connecteur individuel.
- Marquer dans le mappage le contexte (jeu de nœuds) responsables pour la production d'une valeur particulière.
- Exécuter un mappage étape par étape, afin de voir comment MapForce traite ou calcule chaque valeur en temps réel et consulter la sortie de mappage telle qu'elle est générée.
- Définit des points d'étape (points d'arrêt) auxquels l'exécution de mappage doit s'arrêter et afficher les valeurs actuellement en cours de traitement.
- Afficher l'historique des valeurs traitées par un connecteur depuis le début de l'exécution de mappage jusqu'à la position d'exécution actuelle.

Le débogage d'un mappage est disponible lorsque le langage de transformation est en BUILT-IN. Si vous commencez à déboguer un mappage conçu pour un langage différent, vous serez invité à changer le langage de mappage à BUILT-IN. Vous pouvez aussi convertir un mappage en BUILT-IN en sélectionnant la commande de menu **Sortie | Moteur d'exécution Built-in**. Dans les deux cas, la conversion en BUILT-IN sera réussie si le mappage ne doit pas inclure les composants qui ne sont pas disponibles dans le langage BUILT-IN (par exemple, les fonctions XSLT).

Le débogueur de MapForce, contrairement à un débogueur traditionnel, est un débogueur qui ne traverse pas la ligne de code du programme ligne par ligne (puisque vous n'écrivez aucun code avec MapForce). Au lieu de cela, le débogueur expose les résultats du code généré par MapForce produits depuis les mappages que vous concevez. Plus spécifiquement, le débogueur archive les valeurs qui sont transmises de et vers les composants de mappage par le biais de leurs connecteurs entrée et de sortie. Les valeurs archivées sont ensuite disponibles pour votre analyse directement dans le mappage ou par le biais de fenêtres dédiées.

Les sections suivantes marquent les différents moyens vous permettant d'utiliser le débogueur de mappage.

Les paramètres de débogueur sont disponibles dans la boîte de dialogue <u>Options</u><sup>1049</sup>. La liste de commandes de débogage disponibles est disponible dans <u>Déboger</u><sup>1040</sup>.

## Déboguer avec des points d'arrêt

Lorsque vous devez stopper l'exécution de débogage à un endroit particulier du mappage, vous pouvez définir des points d'arrêt, comme vous le feriez dans un environnement de développement traditionnel. La différence est que les points d'arrêt sont ajoutés non pas à une ligne de code mais à un connecteur d'entrée ou de sortie d'un composant de mappage. Vous pouvez aussi ajouter des conditions aux points d'arrêt (ceci peut être utile si vous voulez interrompre l'exécution uniquement si la condition définie est satisfaite).



Vous pouvez définir des points d'arrêt sur les connecteurs désirés et exécutez le mappage jusqu'au premier point d'arrêt rencontré, ensuite passez au prochain, etc. De cette manière, vous pouvez analyser le contexte de mappage et les valeurs associées avec les connecteurs choisis. Vous pouvez aussi accélérer ou ralentir l'exécution avec les commandes **Pas à pas détaillé**, **Pas à pas sortant**, **Pas à pas principal**, et **Étape Minimale** fournies par le débogueur. Ces commandes vous permettent de sauter des portions du mappage, ou, au contraire, d'exécuter des portions du mappage d'une manière plus granulaire, si nécessaire.

### Déboguer étape par étape

Vous pouvez débogueur un mappage étape par étape et analyser le contexte et les valeurs de mappage associés avec chaque étape. Ce scénario est semblable au scénario précédent, car vous pouvez accélérer ou ralentir l'exécution avec les commandes **Step Into**, **Step Out**, **Step Over** et **Minimal Step** commands.



## Analyser le journal des valeurs

Vous pouvez configurer MapForce pour se souvenir du journal de toutes les valeurs (historique de traçage) qui ont été traitées par tous les connecteurs alors que vous déboguez un mappage. Il n'est pas forcément pertinent de garder l'historique complet de traçage pour les mappages qui demandent beaucoup de données, cette option peut donc être désactivée, le cas échéant. Lorsque l'option est activée, vous pouvez analyser le journal complet des valeurs traitées par chaque connecteur jusqu'à la position d'exécution actuelle. Vous pouvez aussi demander à MapForce de recréer le contexte de mappage associé avec une valeur particulière, ce qui peut vous aider à comprendre pourquoi cette valeur a été produite.



## Définir le contexte à une valeur liée à la position d'exécution actuelle

Lorsque le débogueur se trouve dans une position d'exécution particulière dans le mappage, il est possible d'analyser le contexte d'une valeur passée relativement à la position d'exécution actuelle (cela est comparable à un petit saut en arrière dans le temps) :



Un contexte a pour but d'expliquer pourquoi une valeur est calculée ; en d'autres termes, il décrit comment une valeur particulière dans le mappage a pu être générée. Le contexte est généralement la position d'exécution actuelle, bien que MapForce peut aussi vous permettre de définir un contexte dans le passé récent. Lorsque le contexte est défini sur une valeur particulière, MapForce marque directement sur le mappage les nœuds pertinents, propose des astuces à côté des connecteurs de mappage, et expose des informations supplémentaires dans les fenêtres ayant trait au débogueur (les fenêtres Valeurs, Contexte, et Points d'arrêt).

Une fois avoir inspecté un contexte de mappage qui n'est pas le même que celui de la position d'exécution actuelle, vous pouvez réinitialiser le contexte à la position d'exécution actuelle :



# Limitations

- Lorsque MapForce exécute un mappage, il peut optimiser du code en interne (par exemple, en cachant des données ou en calculant des résultats intermédiaires au niveau de points arbitraires). Cela peut provoquer l'inaccessibilité de certains connecteurs (et donc de points d'arrêt) pour le débogage, dans quel cas, MapForce affiche une notification. Veuillez noter que les optimisations de code MapForce (et, par conséquent, le comportement exposé par le débogueur) peut différer d'une release MapForce à une autre, même si l'outil de mappage est le même pour un mappage donné.
- Le débogueur peut déboguer la génération de sortie pour un composant cible à la fois. S'il existe plusieurs composants cible dans le mappage, vous devrez choisir lequel doit être exécuté par le débogueur.
- Actuellement, le débogage n'est pas pris en charge pour les actions de table de la base de données (telles que "Insert All", "Update If", etc.) des composants de base de données.
- Les points d'arrêt ne peuvent pas être ajoutés sur une des entités suivantes : constantes, la fonction core | position, les items descendants des connexions "Copier-tout", les paramètres des fonctions définies par l'utilisateur "inline".

# 8.1 Préparation de débogage

La préparation de débogage est nécessaire principalement pour les mappages de données volumineuses qui nécessitent généralement beaucoup de mémoire de système pour pouvoir être exécutés. C'est le cas des mappages qui traitent soit des fichiers d'entrée ou de sortie très volumineux, soit itèrent à répétition à travers une collection importante de données.

Pour accélérer le débogage et réduire les exigences de mémoire, il est recommandé de procéder aux étapes suivantes avant de commencer le débogage :

- Si le mappage est complexe, supprimer ou déconnecter des parties du mappage qui ne nécessitent pas de débogage.
- Si le mappage utilise des fichiers d'entrée volumineux, les remplacer avec des fichiers de taille plus réduite.
- Veuillez vous assurer que l'option Garder historique du traçage complet est désactivée (voir Paramètres de débogage<sup>1049</sup>)

De même, pour vous assurer que vous déboguez la sortie correcte, contrôlez les points suivants, le cas échéant :

- Si le mappage a plusieurs composants cible, choisir le composant cible à déboguer en cliquant sur la touche Aperçu (
- Si le mappage est un mappage enchaîné (voir <u>Mappages en chaîne</u><sup>841</sup>), relâchez la touche **Passage** (
   ) dans le composant intermédiaire. Le débogage de composants de Passage n'est pas prise en charge actuellement.

En option, si vous souhaitez que le débogueur s'arrête au niveau de connecteurs importants dont vous souhaitez analyser la valeur, ajoutez les points d'arrêt à ces connecteurs (voir <u>Ajouter et supprimer des points</u> <u>d'arrêt</u><sup>(35)</sup>).

# 8.2 À propos du mode Déboguer

Lorsque vous lancez le débogage (en appuyant sur F5, ou F11, ou Ctrl + F11), MapForce exécute le mappage dans le mode de débogage.

Tant que MapForce se trouve dans le mode débogage, le mappage est en lecture seule. Bien que vous puissiez déplacer des composants dans la zone de mappage, la plupart des commandes n'est pas disponible. Cela inclut des commandes comme la validation et le déploiement de mappage, la génération de code, les mappages de documentation, l'ajout de nouveaux composants dans la zone de mappage ou le rechargement de mappages existants, etc.

Le mode de débogage vous permet d'analyser le contexte responsable pour produire une valeur particulière. Cette information est disponible directement sur le mappage, et dans les fenêtres Valeurs, Contexte, et Points d'arrêt. Par défaut, ces fenêtres sont affichées lorsque vous commencez le débogage et sont dissimulées lorsque vous arrêtez le débogage.

MapForce se trouve en mode débogage (et le mappage est en lecture seule) jusqu'à ce que vous stoppiez le débogage, en appuyant sur **Shift + F5** (ou en cliquant sur la touche de la barre d'outils **Arrêter le débogage**).

L'image suivante illustre un échantillon de mappage (**SimpleTotal.mfd**, depuis le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\**) qui est débogué en plusieurs étapes (en appuyant sur **F11** pour avancer d'une étape).



L'environnement de développement MapForce dans le mode de débogage

Les astuces visuelles et les autres informations fournies par MapForce lorsque vous vous trouvez en mode de débogage sont décrites ci-dessous.

### Le volet de mappage

Pendant le débogage, le volet de mappage affiche des informations supplémentaires :

- Des superpositions de données (voir ci-dessous) montrent la valeur actuelle et les valeurs liées près de leurs connecteurs.
- Le contexte actuel (montré en tant que structure dans la fenêtre Contexte) est marqué comme suit :
  - $_{\odot}$  Les connecteurs dans le contexte sont rayés en magenta (  $^{|\!|\!\!|\!|\!\!|}$  ).
  - o Les connecteurs dans le contexte ambigu sont pointillés en magenta ( 👘 ).
  - o Les connexions dans le contexte sont rayées en magenta.
  - o Les connexions dans le contexte ambigu sont rayées en magenta plus clair.
- L'emplacement d'exécution actuel est affiché avec une icône de connecteur verte ( ).

### Superpositions de données

Les valeurs traitées par chaque connecteur sont affichées en tant que superpositions de données (petits rectangles) près de leur connecteur correspondant. Une superposition de données actuellement sélectionnée est marquée par un gros cadre rouge. Les valeurs modifiées depuis la dernière étape sont affichées en rouge foncé. Pour des nœuds avec contenu simple, la superposition de données combine les deux valeurs - le nom de nœud et la valeur. Si le nom de nœud a été itéré plusieurs fois avant la

position d'exécution actuelle, l'index de l'itération actuelle est indiqué par le nombre contenu dans des crochets.

Les superpositions de données présentent le comportement suivant :

- Si vous pointez la souris sur une superposition de données, elle est amenée temporairement vers l'avant, si vous cliquez dessus, vous l'amenez définitivement vers l'avant. Un clic permet aussi de sélectionner le connecteur correspondant.
- Les superpositions de données peuvent être déplacées par glissement.
- Les superpositions de données se déplacent lorsqu'un composant est déplacé. Si les superpositions de données apparaissent superposées c'est parce que les composants sont trop proches l'un de l'autre. Faire glisser les composants autour de la zone de mappage pour avoir plus d'espace, et les superpositions de données se déplaceront en bloc avec le composant.
- Cliquer sur une superposition de données pour montrer sa valeur dans la fenêtre Valeurs.
- Cliquer sur un connecteur permet aussi de sélectionner sa superposition de données.

### Points d'arrêt

Les points d'arrêt sont des jalons, lorsque le mappage rencontre un point d'arrêt, il doit stopper au cours de l'exécution dans le mode de débogage. Ce terme vous est peut-être familier par analogie avec d'autres environnements de développement intégrés. Contrairement à d'autres environnements de développement dans lesquels vous ajoutez des points d'arrêt dans une ligne de code, un point d'arrêt dans MapForce peut être ajouté à un connecteur d'entrée ou de sortie (petit triangle à la gauche ou à la droite de la connexion). Dans le volet de mappage, les points d'arrêt sont représentés en tant que cercles rouges. Tout point d'arrêt défini est aussi affiché dans la fenêtre Points d'arrêt. Voir aussi Ajouter et supprimer des points d'arrêt

### Position de débogueur actuelle

Le triangle vert ( ) indique la position du débogueur. Cette position est soit un connecteur d'entrée ou de sortie de tout composant donné.

La valeur actuellement traitée est aussi affichée dans la fenêtre Valeur, dans l'onglet Contexte.

L'ensemble des connexions et/ou des connecteurs colorés en magenta hachuré indiquent le contexte de mappage actuel. La même information est aussi affichée en tant que structure hiérarchique dans la fenêtre Contexte (voir <u>Utiliser la fenêtre de Contexte</u><sup>(639)</sup>).

Lorsque vous définissez manuellement le contexte d'une valeur, la position actuelle du débogueur se trouve dans une position dans le passé relative à la position d'exécution la plus récente. Pour vous aider à distinguer entre la position d'exécution la plus actuelle et celle se trouvant dans le passé, le connecteur "position actuelle" peut apparaître avec les couleurs suivantes dans l'interface du débogueur.

۵	Vert représente "le présent"; il indique la position d'exécution actuelle (voir <u>Consulter la valeur</u> <u>actuelle d'un connecteur</u> <sup>944</sup> ).
	Jaune représente "le passé"; il indique que vous êtes en train de consulter un connecteur dans le passé, relatif à la position d'exécution actuelle. Cela peut se produire après avoir défini un contexte manuellement (voir <u>Définir le contexte dans une valeur</u> <sup>947</sup> ).

### Fenêtre Valeurs

La fenêtre Valeurs fournit des informations concernant les valeurs traitées par le mappage. Elle vous permet de voir ce que le mappage traite à l'emplacement actuel de l'exécution, ou dans un contexte particulier que vous pouvez définir vous-même. Voir aussi <u>Utiliser la fenêtre Valeurs</u><sup>(837)</sup>.

## Fenêtre Contexte

La fenêtre Contexte offre un aperçu hiérarchique de l'ensemble de nœuds et fonctions qui sont pertinentes pour la position de débogueur actuelle. Voir aussi <u>Utiliser la fenêtre Contexte</u>

### Fenêtre Points d'arrêt

La fenêtre Points d'arrêt affiche la liste des points d'arrêt de débogage créés depuis que MapForce a été lancé. Si vous avez défini des points d'arrêt sur plusieurs mappages, tous apparaîtront dans la fenêtre Points d'arrêt. Voir aussi <u>Utiliser la fenêtre Points d'arrêt</u><sup>941</sup>.

# 8.3 Ajouter et supprimer des points d'arrêt

Les points d'arrêt sont des jalons désignés au niveau desquels le mappage doit s'interrompre pendant l'exécution dans le mode de débogage. Tous les points d'arrêt que vous créez sont stockés globalement pour tous les mappages et sont affichés dans la fenêtre Points d'arrêt. Les points d'arrêt sont valides jusqu'à ce que vous les supprimiez explicitement ou fermez MapForce.

**Note :** Les points d'arrêt ne peuvent pas être ajoutés aux entités suivantes : constantes, fonction core | position, items descendants des connexions "Copy-all", paramètres des fonctions définies par l'utilisateur de "inline".

Les points d'arrêt peuvent être simples ou conditionnels. Les points d'arrêt simples arrêtent l'exécution de mappage de manière inconditionnelle. Les points d'arrêt conditionnels arrêtent l'exécution de mappage uniquement lorsque la condition qui leur est attribuée est satisfaite. Les conditions prennent la forme des fonctions de bibliothèques intégrées MapForce auxquelles vous fournissez des valeurs personnalisées. Autrement dit, si la condition retourne true, le point d'arrêt arrêtera l'exécution de mappage.

#### Pour créer un point d'arrêt simple, suivre une des étapes suivantes :

- Cliquer avec la touche de droite sur un connecteur d'entrée ou de sortie (les petits triangles à gauche ou à droite d'un composant), et sélectionner **Point d'arrêt du débogueur**.
- Cliquer sur un connecteur d'entrée ou de sortie, et appuyer sur F9.

#### Pour créer un point d'arrêt conditionnel :

1. Cliquer avec la touche de droite sur un connecteur, et choisir Propriétés de point d'arrêt.

😢 Propriétés de point de rupture	×
Emplacement Composant : $f_x$ core.multiply Connecteur : $\triangleright$ result	
Points d'arrêt	~
Valeur : 2	OK Annuler

2. Cliquer pour sélectionner les deux cases à cocher **Point d'arrêt** et **Condition**.

 Choisir les fonctions requises depuis la liste et saisir la valeur de fonction (le cas échéant). Par exemple, dans l'exemple ci-dessus, le point d'arrêt arrêtera l'exécution de mappage si la valeur passant à travers est supérieure à 2.

Si le type de données du connecteur dans lequel vous ajoutez le point d'arrêt conditionnel ne correspond pas au(x) type(s) attendu(s) par la fonction, MapForce tentera de convertir automatiquement le type de données. Si la conversion automatique n'est pas possible, l'exécution de mappage échouera. Pour éviter cela, s'assurer d'utiliser des types de données. Par exemple, la fonction **core.starts-with** attend une valeur de string pour que le connecteur du point d'arrêt ait le même type.

## Supprimer les points d'arrêt

Pour supprimer un point d'arrêt, cliquer avec la touche de droite sur le connecteur sur lequel le point d'arrêt existe, et sélectionner **Point d'arrêt de débogueur**. En alternative, cliquer sur le connecteur d'entrée ou de sortie sur lequel le point d'arrêt existe, puis appuyer sur **F9**.

Vous pouvez aussi supprimer des points d'arrêt depuis la fenêtre Points d'arrêt (voir <u>Utiliser la fenêtre Points</u> <u>d'arrêt</u><sup>[641</sup>).

### Points d'arrêt non atteignables

Il peut arriver que MapForce affiche un message "Les points d'arrêt ne peuvent pas être atteints" :

MapForce		
1	Quelques points de rupture ne peuvent pas être atteints. L'exécution de mappage ne passera pas par là.	
	Ne plus afficher ce message.	
	Continuer Étape Arrêter débogueur	

Cela indique que les points d'arrêt ne peuvent pas être atteints par le débogueur, pour une des raisons suivantes :

- Un point d'arrêt a été défini sur un connecteur qui ne participe pas au mappage.
- Le point d'arrêt ne peut pas être atteint par MapForce à cause des optimisations d'exécution (voir <u>Limites</u><sup>(22)</sup>).

Cliquer sur **Continuer** pour avancer au prochain point d'arrêt défini (ou se rendre à la fin de l'exécution de débogage). Cliquer sur **Étape** pour lancer le débogage en étapes.

Vous pouvez désactiver les notifications à propos du point d'arrêt non atteignable rencontré par le débogueur, soit en cliquant sur **Ne pas afficher ce message à nouveau**, soit comme suit :

- 1. Dans le menu Outils, cliquer sur Options.
- 2. Cliquer sur Messages.
- 3. Cliquer pour supprimer la case à cocher Informations concernant les points d'arrêt non atteignables.
### 8.4 Utiliser la fenêtre Valeurs

La fenêtre Valeurs affiche des informations concernant les valeurs traitées par le mappage lorsqu'il se trouve en mode débogage. L'information affichée dans la fenêtre Valeurs dépend de la position actuelle du débogueur et des éléments d'interface utilisateur que vous avez cliqué. La fenêtre Valeurs contient les onglets suivants :

#### L'onglet "Contexte"

L'onglet **Contexte** affiche la valeur actuellement traitée (la même valeur dont le contexte est affiché dans la fenêtre Contexte). Il s'agit soit de la valeur au niveau de la position d'exécution actuelle du débogueur, soit de la valeur d'un connecteur traité dans le passé. MapForce vous aide à distinguer entre les deux couleurs utilisées :

۵	Vert représente "le présent"; il indique la position d'exécution actuelle (voir <u>Consulter la valeur</u> <u>actuelle d'un connecteur</u> ).
	Jaune représente "le passé"; il indique que vous êtes en train de consulter un connecteur dans le passé, relatif à la position d'exécution actuelle. Cela peut se produire après avoir défini un contexte manuellement (voir <u>Définir le contexte dans une valeur</u> <sup>947</sup> ).

#### L'onglet "Lié"

L'onglet **Lié** affiche des valeurs qui sont liées à (ou représentent le "passé proche" de) la valeur traitée actuellement. Normalement, vous n'avez pas besoin de cliquer expressément sur cet onglet ; MapForce y passe automatiquement lorsque vous cliquez sur la superposition de données d'un connecteur qui est lié à la position d'exécution actuelle du débogueur. Voir <u>Retourner dans le passé récent</u><sup>945</sup>.

#### L'onglet "Séquence"

Si présent, l'onglet **Séquence** vous permet d'obtenir l'accès aux valeurs d'un connecteur qui traite une séquence. Cet onglet est uniquement visible si un connecteur a traité une séquence d'items (par exemple, avec une fonction d'agrégat comme sum ou count). lorsque vous cliquez sur la superposition de données d'un connecteur qui a traité une séquence d'items, la fenêtre Valeurs affiche une entrée dans le format "**n items**", où **n** est le nombre d'items traités par le connecteur. Pour obtenir l'accès à chaque valeur, double-cliquer sur cette entrée (ou cliquer avec la touche de droite et choisir **Agrandir la séquence** depuis le menu contextuel).

Values			<b>•</b> 4	×
Context	Related	History		
values				_
2 item(s)	Expand	Sequence		

Les valeurs sont ensuite affichées dans l'onglet Séquence.

Values		<b>→</b> # ×
Context	Related	Sequence + >
▷ values		
170		
425		

#### L'onglet "Historique"

L'onglet **Historique** affiche des valeurs qui ont été traitées par un nœud particulier depuis le démarrage du débogage et jusqu'à la position d'exécution actuelle. Voir <u>Consulter l'historique des valeurs traitées par un</u> <u>connecteur</u><sup>946</sup>.

### 8.5 Utiliser la fenêtre Contexte

Pendant que MapForce se trouve en mode débogage, la fenêtre Contexte affiche une structure de connecteurs qui sont pertinents pour la position actuelle du débogueur. En d'autres termes, elle fournit le contexte de mappage responsable pour produire la valeur de mappage actuelle.

MapForce établit le contexte actuel comme suit :

- 1. Commencer avec le nœud racine de la structure de cible.
- 2. Descendre sur le nœud cible actuel.
- À partir du nœud de cible actuel, se déplacer à gauche dans le mappage à travers tout composant qui mène à la position actuelle. Ces composants peuvent être des composants filtre ou tri, des fonctions intégrée ou définies par l'utilisateur, les variables, etc.

La fenêtre Contexte sert aussi bien en tant qu'aide informationnelle et de navigation. Pour choisir un nœud particulier dans le mappage directement depuis le contexte actuel, cliquer avec la touche de droite sur le nœud dans la fenêtre Contexte, et cliquer sur **Sélectionner dans mappage**. Cela peut être particulièrement utile lorsque le mappage est volumineux, afin d'éviter les défilements intensifs.

La fenêtre Contexte peut afficher les icônes spéciales suivantes et les notations :

Icône	Description
89	Représente le mappage auquel le contexte appartient. Il peut s'agir soit du mappage principal soit du mappage d'une fonction définie par l'utilisateur.
	Contexte 🗸 🗸
	<ul> <li>Bain Mapping</li> <li>Image: Directory</li> <li>Directory</li> <li>Directory[1]</li> <li>Image: BuildFileHierarchy</li> <li>Directory</li> <li>Directory</li> <li>Directory</li> <li>Directory</li> <li>Directory</li> </ul>
⊳	Représente un connecteur. Les nœuds cibles traitées jusqu'à présent présentent leur position qui s'affiche dans des crochets.



Outre les icônes ci-dessus, la fenêtre Contexte contient les icônes standard de tout type de composant qui est présent dans le mappage.

#### Fenêtre Contexte et fonctions définies par l'utilisateur

Si le contexte actuel contient des fonctions définies par l'utilisateur, elles sont également affichées dans la fenêtre Contexte. Veuillez noter que si le contexte actuel sert à calculer une valeur d'entrée d'une fonction définie par l'utilisateur, le contexte est déterminé comme suit :

- 1. Depuis la cible vers le connecteur de sortie de la fonction définie par l'utilisateur vers le connecteur d'entrée de la fonction définie par l'utilisateur.
- 2. Depuis ici vers la gauche.
- **Note :** Une fonction définie par l'utilisateur peut se produire plusieurs fois dans le contexte. Cela a lieu soit parce que plusieurs appels de fonction sont enchaînés soit parce que la fonction définie par l'utilisateur est définie comme récursive.

### 8.6 Utiliser la fenêtre Points d'arrêt

La fenêtre Points d'arrêt vous permet de consulter et de gérer les points d'arrêt globalement. Par défaut, la fenêtre Points d'arrêt est affichée lorsque MapForce se trouve en mode de débogage. Si vous souhaitez que la fenêtre Points d'arrêt soit visible en permanence, choisissez la commande de menu **Affichage | Fenêtre Déboguer | Points d'arrêt** lorsque MapForce n'est pas en mode de débogage.

La fenêtre Points d'arrêt affiche tous les points d'arrêt créés depuis que vous avez lancé MapForce, regroupés par le fichier de mappage auquel ils appartiennent. Alors que MapForce est ouvert, tout point d'arrêt associé avec un mappage est "mémorisé" par MapForce et est affiché dans la fenêtre Points d'arrêt, même si vous avez fermé le fichier de mappage entre-temps. Le mappage en cours de débogage actuellement est représenté avec une couleur de texte standard dans la fenêtre Points d'arrêt, alors que les autres mappages (ceux qui sont fermés ou actifs) sont grisés.



Vous pouvez rapidement ouvrir tout mappage en le double-cliquant (ou un de ses points d'arrêt) dans la fenêtre Points d'arrêt.

Note : Une fois que vous fermez ou redémarrez MapForce, tous les points d'arrêt seront supprimés.

Les informations concernant les points d'arrêt sont affichées sous forme de grille avec les colonnes suivantes :

Colonne	Description
Nom	Le nom du nœud dont fait partie le point d'arrêt.
Parent	Le nom du composant de mappage dont fait partie le point d'arrêt.
Valeur de trace	La valeur qui passe à travers le connecteur sur lequel se trouve le point d'arrêt. La valeur de trace est affichée pendant l'exécution du débogage.
Condition	Si le point d'arrêt est conditionnel, cette colonne affiche la condition du point d'arrêt.

Les points d'arrêt peuvent être associés à une des icônes suivantes.

Icon	Description
•	Point d'arrêt actif. Dénote un point d'arrêt depuis un mappage actuellement en cours de débogage.
0	Point d'arrêt inactif. Dénote un point d'arrêt depuis un mappage ouvert, mais qui n'est pas actuellement en cours de débogage.
Q	Point d'arrêt inaccessible. Dénote un point d'arrêt qui ne peut pas être atteint par le débogueur.
0	Point d'arrêt conditionnel. Dénote un point d'arrêt avec une condition qui lui est attachée.

#### Pour consulter ou modifier les propriétés d'un point d'arrêt :

• Cliquer dessus avec la touche de droite, et sélectionner **Propriétés de point d'arrêt** depuis le menu contextuel.

#### Pour supprimer un point d'arrêt :

- Cliquer avec la touche de droite sur le point d'arrêt que vous souhaitez supprimer, puis sélectionner **Supprimer point d'arrêt** depuis le menu contextuel.
- Cliquer sur un point d'arrêt, puis appuyer sur Supprimer.

La commande de contexte **Supprimer tous les points d'arrêt** supprime tous les points d'arrêt affichés dans la fenêtre des points d'arrêt, quelque soit le mappage auquel ils appartiennent.

Voir aussi : Ajouter et supprimer des points d'arrêt<sup>935</sup>

### 8.7 Visualiser une sortie partiellement générée

Lorsque vous déboguez en étapes ou en utilisant des points d'arrêt, vous pouvez consulter la sortie de mappage générée jusqu'à la position actuelle du débogueur. La visualisation de la sortie partiellement générée est prise en charge par XML, texte plat, et les composants de cible EDI.

Par défaut, lorsque vous appuyez sur **F5** (sans avoir défini de points d'arrêt), MapForce exécute l'ensemble du mappage dans le mode de débogage, puis passe à l'onglet **Sortie**, en affichant la sortie finale générée. Néanmoins, si vous avez défini des points d'arrêt ou si vous déboguez par étapes (**F11**, ou **Ctrl + F11**), l'exécution du débogueur stoppe alors que la sortie de mappage est en cours de génération. Même si la sortie de mappage est partiellement rédigée à ce niveau, vous pouvez toujours cliquer sur l'onglet **Sortie**, et la visualiser.

Gen	eratin	g result C:\Users\altova\Documents\Altova\MapForce2016\MapForceExamples\CompletePO.xml
1	<	<pre>?xml version="1.0" encoding="UTF-8"?&gt;</pre>
2	- <	CompletePO xsi:noNamespaceSchemaLocation="
	fi	e:///C:/Users/altova/Documents/Altova/MapForce2016/MapForceExamples/CompletePO.xsd" xmlns:xsi="
	h	tp://www.w3.org/2001/XMLSchema-instance">
3	•	<customer></customer>
4		<number>3</number>
5		<firstname>Ted</firstname>
6		<lastname>Little</lastname>
7	÷.	<address></address>
8		<street>Long Way</street>
9		<city>Los-Angeles</city>
10		<zip>34424</zip>
11		<state>CA</state>
12		
13		
14	e	<linettems></linettems>
15	¢	<lineltem< th=""></lineltem<>

#### Limites

- Le nœud de cible actuellement calculé n'est pas toujours affiché dans l'onglet Sortie. Par exemple, les attributs XML sont rassemblés en interne et rédigés immédiatement.
- Si la sortie produit plusieurs fichiers, seul le fichier rédigé actuellement peut être affiché ; le passage à un autre fichier de sortie est désactivé.

### 8.8 Consulter la valeur actuelle d'un connecteur

Lorsque la position d'exécution actuelle du débogueur ( ) se trouve sur un connecteur particulier (soit parce que vous effectuez un débogage en étapes ou parce qu'il y a un point d'arrêt défini dans le connecteur), la valeur actuelle traitée par le connecteur est affichée dans l'onglet **Contexte** de la fenêtre Valeurs. Il s'agit de la valeur qui va être rédigée dans la sortie, c'est à dire, "le présent". Il s'agit aussi de la valeur dont le contexte est affichée dans la fenêtre Contexte (voir <u>Utiliser la fenêtre Contexte</u>).

Pour comprendre cet exemple, ouvrir l'échantillon **PreserveFormatting.mfd** dans le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\**. Cliquer dans le connecteur d'entrée du nœud Number dans le composant de cible, et appuyer sur **F9** pour y ajouter un point d'arrêt.



Ensuite, appuyer sur F5 pour commencer le débogage et observer les résultats.



Comme indiqué dans l'image, la position de débogueur actuelle (et le point d'arrêt) se trouve sur le nœud Number du composant de cible. La fenêtre Valeurs indique que ce nœud traite la valeur "1" (cette valeur est aussi marquée avec un bord rouge épais dans le mappage).

### 8.9 Retourner dans le passé récent

Lorsque vous cliquez sur une superposition de données (petite case rectangulaire) à côté d'un connecteur de mappage, la fenêtre **Valeurs** affiche le nom et, en option, la valeur associée avec le connecteur sélectionné. Le centre d'attention n'est plus sur la position actuelle du débogueur, mais sur la superposition de données sélectionnée. Vous pouvez considérer cette vue comme un petit saut en arrière dans le temps dans l'historique de débogage. Il s'agit du passé "récent", puisque le mappage affiche uniquement des superpositions de données pour les derniers connecteurs liés à la position du débogueur actuelle. Lorsque vous cliquez sur une telle superposition de données "liées", la fenêtre Valeurs passe automatiquement à l'onglet **Lié**.

# Pour une illustration de ce scénario, ouvrir le mappage **PreserveFormatting.mfd** du répertoire <br/> **Cocuments>\Altova\MapForce2023\MapForceExamples\**.

Une fois avoir ouvert le mappage, cliquer sur le connecteur situé à côté du nœud Number dans le composant de cible, et appuyer sur **F9** pour y ajouter un point d'arrêt. Appuyer sur **F5** pour commencer le débogage, puis cliquer sur la superposition de données (petite case rectangulaire) à côté du nœudNumber du composant de source.



Étant donné qu'un connecteur est généralement itéré plusieurs fois pendant la durée de vie d'un mappage, l'index actuel de l'itération est affiché entre des crochets : **<Number>[1]**. De même, étant donné que le connecteur porte une valeur, celle-ci est aussi représentée après le signe égal : **<Number>[1]=1**. La même valeur est affichée sur une nouvelle ligne dans la fenêtre Valeurs, comme indiqué ci-dessous.

Si vous nécessitez des informations supplémentaires concernant une valeur particulière, n'oubliez pas que vous pouvez recréer le contexte qui l'a produit (voir <u>Configurer le Contexte en une Valeur</u><sup>947</sup>).

# 8.10 Consulter l'historique des valeurs traitées par un connecteur

Si l'option **Garder historique du traçage complet** est activée (voir <u>Paramètres du débogueur</u><sup>(009)</sup>), vous pouvez consulter l'historique de toutes les valeurs qui ont été traitées par ce connecteur (jusqu'à la position d'exécution actuelle).

L'historique est affiché lorsque vous cliquez sur un connecteur, puis cliquez sur l'onglet **Historique** de la fenêtre Valeurs. Veuillez noter que cette opération est uniquement utile pour les connecteurs qui ont des valeurs traitées depuis le début de l'exécution du mappage jusqu'à la position du débogueur actuel.

Pour illustrer cet exemple, procédons à un débogage d'un mappage du début à la fin sans utiliser de points d'arrêt, puis observons l'historique de toutes les valeurs qui ont été traitées par un connecteur particulier. En premier lieu, ouvrez le mappage **PreserveFormatting.mfd** depuis le répertoire **<Documents>\Altova\MapForce2023\MapForceExamples\**. S'il est déjà ouvert, assurez-vous de procéder comme suit :

- Supprimer tous les points d'arrêt, le cas échéant, (voir <u>Ajouter et supprimer des points d'arrêt</u><sup>935</sup>)
- Arrêter le débogage s'il est actuellement en cours d'exécution, en appuyant sur Shift + F5.

Lorsque vous êtes prêt, appuyez sur **F5** pour lancer une nouvelle opération de débogage. Lorsque vous appuyez sur **F5**, MapForce exécute le mappage en mode débogage et passe à l'onglet **Sortie**. Cliquez sur l'onglet **Mappage** pour retourner à la fenêtre de mappage principal, puis cliquez sur le nœud result de la fonction format-number (marquée en rouge dans l'image ci-dessous). Enfin, cliquez sur l'onglet **Historique** de la fenêtre Valeurs, et notez les valeurs affichées.

	4	Valeur	S			•	ųх
<article> <number> <name> <singleprice> f<sub>x</sub> format-number</singleprice></name></number></article>	Image: Second state sta	Cont D resu "0002 "0003 "0003 "0005"	texte JIt 5.00" 2.30" 4.00" 7.50"	Lié/e	Historique		
→ format		Conte	xte			•	4 ×
decimal-point-character							
sprouping-separator							

Comme indiqué dans l'image ci-dessus, ce nœud particulier (result) a traité quatre valeurs au total. Si vous nécessitez plus d'informations concernant une valeur particulière, n'oubliez pas que vous pouvez recréer le contexte qui l'a produit (voir <u>Configurer le Contexte en une Valeur</u><sup>647</sup>).

### 8.11 Configurer le Contexte en une Valeur

Configurer le contexte en une valeur est une action qui peut être comparée à un saut dans le passé, afin de consulter plus de détails à propos du contexte de mappage qui a produit cette valeur. Vous pouvez définir le contexte sur toute valeur affichée dans la fenêtre Valeurs (dans l'onglet Lié, l'onglet Séquence ou l'onglet Historique). Si vous avez activé l'option Garder historique du traçage complet (voir Paramètres du débogueur), l'onglet Historique affiche toutes les valeurs traitées par le connecteur sélectionnées actuellement ; c'est pourquoi, dans ce cas, vous pouvez en plus définir le contexte sur toute valeur dans le passé pour ce connecteur.

Valeurs				-	д	×
Contexte Lié/e Historiqu			Historique			
▷ result	▷ result					
*00 *00034.00 *00057.50	<u>D</u> éf	inir le con	texte			

Pour définir le contexte en une valeur, suivre une des étapes suivantes :

- Cliquez avec la touche de droite sur la valeur, et choisissez **Définir Contexte** depuis le menu contextuel.
- Double-cliquer sur la valeur.

Lorsque vous définissez le contexte à une valeur, MapForce met la zone de mappage en surbrillance afin de recréer la situation qui a produit la valeur, et peuple la fenêtre **Valeurs** et la fenêtre **Contexte** conformément au contexte sélectionné. Pour consulter une légende des indications visuelles utilisées dans la zone de mappage tout en se trouvant dans un contexte, voir <u>À propos du mode Déboguer</u><sup>(G31)</sup>. Pour plus d'informations concernant le contexte lui-même, voir <u>Utiliser la fenêtre Contexte</u><sup>(G32)</sup>.

Le connecteur d'un contexte défini manuellement est jaune ( ), qui indique que vous ne vous trouvez plus dans la position d'exécution la plus récente. Pour retourner à la position d'exécution la plus récente (le cas échéant), cliquer sur la touche **Réinitialiser sur actuel** de l'onglet **Contexte** dans la fenêtre Valeurs.

Values			-	ф,	×
Context	Related	History			
🜔 result					
"00025.00"					
	Reset to C	urrent			

### 9 Automatiser des mappages et MapForce

Les mappages conçus avec MapForce peuvent être exécutés dans un environnement de serveur (y compris les serveurs Linux et macOS) et avec des performances de niveau de serveur, par les moteurs de transformation Altova suivants (mis sous licence séparément) :

- RaptorXML Server. L'exécution d'un mappage avec ce moteur est pertinent si le langage de transformation du mappage est XSLT 1.0, XSLT 2.0, XSLT 3.0 ou XQuery. Voir <u>Automatisation avec</u> <u>RaptorXML Server</u><sup>[449]</sup>.
- MapForce Server (ou MapForce Server Advanced Edition). Ce moteur convient pour tout mappage lorsque le langage de transformation est BUILT-IN\*. Le langage BUILT-IN prend en charge la plupart des fonctions de mappage dans MapForce, alors que MapForce Server (et en particulier, MapForce Server Advanced Edition) fournit la meilleure performance pour exécuter un mappage. Voir <u>Automatisation avec MapForce Server</u><sup>(60)</sup>.

\* Le langage de transformation BUILT-IN nécessite l'édition MapForce Professional ou Enterprise.

De plus, MapForce permet d'automatiser la génération de code XSLT, XQuery, C#, C++ et Java depuis l'interface de ligne de commande. Cela inclut la capacité à compiler les fichiers d'exécution de serveur (.mfx) prévus pour l'exécution MapForce Server. Pour plus d'informations, voir <u>Interface de ligne de commande</u> <u>MapForce</u>.

### 9.1 Automatisation avec RaptorXML Server

RaptorXML Server (ci-après abrégé par RaptorXML) est le processeur XML et XBRL hyper-rapide de troisième génération d'Altova. Il est conçu pour être optimisé pour les tous derniers standards et des environnements 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 qui peuvent être téléchargées sur la page de téléchargement d'Altova (<u>https://www.altova.com/download-trial-server.html</u>):

- RaptorXML Server est un moteur de traitement XML très rapide qui prend en charge XML, Schéma XML, XSLT, XPath, XQuery, etc.
- RaptorXML+XBRL Server prend en charge toutes les fonctions de RaptorXML Server avec en outre des fonctions permettant de traiter et de valider la famille XBRL de standards.

Si vous générez du code dans XSLT, ou dans XQuery, MapForce crée un fichier batch appelé **DoTransform.bat** qui est placé dans le dossier de sortie que vous choisissez lors de la génération. L'exécution du fichier batch appelle RaptorXML Server et exécute la transformation XSLT (ou XQuery) sur le serveur.

Si vous avez l'intention d'exécuter ou d'automatiser les mappages MapForce pour d'autres sorties sur un serveur, voir <u>Automatisation avec MapForce Server</u><sup>(50)</sup>.

**Note :** Vous pouvez aussi <u>consulter le code XSLT</u><sup>105</sup> et XQuery à l'aide du moteur intégré.

### 9.2 Automatisation avec MapForce Server

MapForce Server est une solution logicielle de serveur pour entreprise conçue pour fonctionner avec les systèmes d'exploitation Windows, Linux et macOS. Le rôle de MapForce Server est d'exécuter des mappages dans un environnement de serveur (y compris sur des plateformes non-Windows) et avec une performance de niveau serveur. Tout mappage MapForce dont le langage d'exécution cible est BUILT-IN se qualifie pour une exécution de serveur (voir aussi <u>Choisir un langage de transformation</u><sup>(25)</sup>). MapForce Server peut fonctionner soit en autonomie (invoqué depuis la ligne de commande ou l'API), soit sous la gestion de FlowForce Server.

Si MapForce Server est utilisé en tant que produit individuel, le mappage MapForce doit être compilé et copié sur l'appareil sur lequel MapForce Server est exécuté. Le mappage est ensuite exécuté à l'aide de la commande de ligne de commande run. Vous pouvez aussi exécuter le mappage en invoquant la méthode run de l'API de MapForce Server. Pour plus d'informations, voir <u>Compiler des mappages sur des fichiers d'exécution</u> <u>MapForce Server</u>.

Si MapForce Server est exécuté sous la gestion de FlowForce Server, le mappage peut être déployé sur un appareil cible par le biais d'une connexion HTTP (ou SSL/HTTPS) directement depuis MapForce. Sur le serveur, le mappage peut ensuite être exécuté en tant que tâche déclenchée ou programmée, ou bien, par le biais d'un appel de service Web défini depuis l'interface d'administration de FlowForce Server. Pour plus d'informations, voir <u>Déployer des mappages sur FlowForce Server</u><sup>600</sup>.

Il existe deux éditions de MapForce Server:

- MapForce Server
- MapForce Server Advanced Edition

MapForce Server Advanced Edition fournit les mêmes fonctions que MapForce Server, et inclut en plus des fonctions d'optimisation pour les mappages qui se qualifient pour l'optimisation. C'est le cas avec les mappages qui joignent ou filtrent des quantités importantes de données et lorsqu'il est possible d'appliquer une optimisation join de manière à augmenter la vitesse d'exécution. Contrairement à MapForce Server, MapForce Server Advanced Edition peut exécuter des mappages lorsque des fonctions de nœud sont présentes, voir Défauts et fonctions de nœud <sup>660</sup>.

Limites :

- Les signatures numériques XML ne sont pas prises en charge
- Les connexions de base de données ADO, ADO.NET et ODBC sont uniquement prises en charge sur Windows. (Pour les autres systèmes d'exploitation, voir <u>Préparer des mappages pour l'exécution de</u> <u>serveur</u>.<sup>(951</sup>).

Pour plus d'informations concernant MapForce Server, veuillez vous référer à sa documentation (<u>https://www.altova.com/fr/documentation</u>).

### 9.3 **Préparer des mappages pour l'exécution de serveur**

Un mappage conçu et prévisionné avec MapForce peut référer à des ressources se trouvant en dehors de l'appareil et du système d'exploitation actuels (comme des bases de données). Outre ce fait, dans MapForce, tous les chemins de mappage suivent des conventions de style Windows par défaut. De plus, l'appareil sur lequel MapForce Server est exécuté pourrait ne pas prendre en charge les mêmes connexions de base de données que l'appareil sur lequel de mappage a été conçu. C'est pour cette raison que les mappage exécutés dans un environnement de serveur nécessitent généralement une certaine préparation, en particulier si l'appareil cible n'est pas le même que l'appareil de source.

**Note :** Le terme "appareil source" renvoie à l'ordinateur sur lequel MapForce est installé et le terme "appareil cible" réfère à l'ordinateur sur lequel MapForce Server ou FlowForce Server est installé. Dans le scénario le plus simple, il s'agit du même ordinateur. Dans un scénario plus complexe, MapForce est exécuté sur un appareil Windows alors que MapForce Server ou FlowForce Server est exécuté sur un appareil Linux ou macOS.

En règle générale, il est recommandé de s'assurer que le mappage soit validé avec succès dans MapForce avant de le déployer dans FlowForce Server ou de le compiler dans un fichier d'exécution MapForce Server (voir <u>Valider les mappages</u><sup>103</sup>).

Si MapForce Server est exécuté seul (sans FlowForce Server), les licences requises sont les suivantes :

- Sur l'appareil source, l'édition MapForce Enterprise ou Professional est exigée pour concevoir le mappage et le compiler dans un fichier d'exécution de serveur (.mfx), voir <u>Compiler des mappages sur</u> <u>des fichiers MapForce Server Execution</u><sup>(957)</sup>.
- Sur l'appareil cible, MapForce Server ou MapForce Server Advanced Edition est exigé pour exécuter le mappage.

Si MapForce Server est exécuté sous la gestion FlowForce Server, les exigences suivantes s'appliquent :

- Sur l'appareil source, l'édition MapForce Enterprise ou Professional est exigée pour concevoir le mappage et le déployer sur un appareil cible, voir <u>Déployer les mappages sur FlowForce Server</u><sup>339</sup>.
- MapForce Server et FlowForce Server doivent tous deux être sous licence sur l'appareil cible. Le rôle de MapForce Server est d'exécuter le mappage ; le rôle de FlowForce est de rendre le mappage disponible en tant que tâche qui profite des fonctions telles que exécutions programmées ou exécution sur demande, exécution en tant que service Web, gestion d'erreur, traitement conditionnel, notifications d'e-mail, etc.
- FlowForce Server doit être activé et exécuté sous l'adresse et le port de réseau. Le service "FlowForce Web Server" doit être lancé et configuré pour accepter les connexions provenant des clients HTTP (ou HTTPS si configuré) et ne doit pas être bloqué par le pare-feu. Le service "FlowForce Server" doit aussi être démarré et exécuté à l'adresse et le port désigné.
- Vous devez disposer d'un compte utilisateur FlowForce Server avec les permissions d'un des conteneurs (par défaut, le conteneur **/public** est accessible à n'importe quel utilisateur authentifié).

#### Considérations d'ordre général

 Si vous comptez exécuter le mappage sur un appareil cible avec un MapForce Server autonome, tous les fichiers d'entrée référencés par le mappage doivent également être copiés dans l'appareil cible. Si MapForce Server est exécuté sous la gestion FlowForce Server, aucun besoin de copier les fichiers manuellement. Dans ce cas, les fichiers d'instance et de schéma sont inclus dans le pack déployé sur l'appareil cible, voir <u>Déployer des mappages sur FlowForce Server</u><sup>989</sup>.

- Si le mappage comprend des composants de base de données qui nécessitent des pilotes de base de données spécifiques, ces pilotes doivent aussi être installés sur l'appareil cible. Par exemple, si votre mappage lit des données provenant d'une base de données Microsoft Access, Microsoft Access ou Microsoft Access Runtime (<u>https://www.microsoft.com/en-us/download/details.aspx?id=50040</u>) doit aussi être installés sur l'appareil cible.
- Lorsque vous déployez un mappage sur des plateformes non-Windows, les connexions de base de données ADO, ADO.NET et ODBC sont modifiées automatiquement en JDBC. Les connexions SQLite natives et PostgreSQL natives sont préservées et ne nécessitent aucune autre configuration. Voir "Connexions aux bases de données" ci-dessous.
- Si le mappage contient des appels de fonction personnalisés (par exemple vers des fichiers .dll ou .class), Ces dépendances ne sont pas déployées avec le mappage, puisqu'ils ne sont pas connus avant l'exécution. Dans ce cas, les copier manuellement dans l'appareil cible. Le chemin du fichier .dll ou .class sur le serveur doit être le même que dans la fenêtre "Gérer bibliothèques" de MapForce, par exemple.

Manage Libraries		
🔁 🛃 New Design1		
User-Defined Functions		
Own Library Imports Add		
🕀 😢 Global Library Imports 🛛 Add		
🖓 🔂 C:\Libraries\Converters.dll		
- Functions		
		Montrer tous les documents ouverts
f <sub>8</sub> Converters, Version=1.0.0.0,		Montrer les chemins de fichier
Own Library Imports		

- Certains mappages lisent plusieurs fichiers d'entrée en utilisant un chemin de caractère générique (voir <u>Traiter plusieurs fichiers d'entrée et de sortie dynamiquement</u><sup>(B33)</sup>). Dans ce cas, les noms de fichier d'entrée ne sont pas connus avant l'exécution et ils ne seront donc pas déployés. Pour que le mappage puisse être exécuté correctement, les fichiers d'entrée doivent exister sur l'appareil cible.
- Si le chemin de sortie de mappage contient des répertoires, ces répertoires doivent exister sur la machine cible. Sinon, une erreur sera générée lorsque vous exécuterez le mappage. Ce comportement est différent de celui sur MapForce, où des répertoires non existants sont générés automatiquement si l'option Générer une sortie dans les fichiers temporaires est activée (voir Modifier les options MapForce<sup>(MB)</sup>).
- Si le mappage appelle un service web qui nécessite une authentification HTTPS avec un certificat client, le certificat doit aussi être transféré à l'appareil cible, voir .
- Si le mappage se connecte à des bases de données basées sur des fichiers comme Microsoft Access et SQLite, le fichier de base de données doit être transféré manuellement à l'appareil cible ou enregistré sous un répertoire partagé qui est accessible aussi bien à l'appareil source que cible et référencé à partir de cet endroit, voir "Bases de données basées sur fichier" ci-dessous.

#### Rendre les chemins portables

Si vous comptez exécuter le mappage sur un serveur, veuillez vous assurer que le mappage suive les conventions de chemin applicables et utilise une connexion de base de données prise en charge.

Pour rendre les chemins portables pour les systèmes d'exploitation non-Windows, utiliser des chemins relatifs et non pas absolus lors de la conception du mappage dans MapForce:

- 1. Ouvrir le fichier de design de mappage désiré (.mfd) avec MapForce sur Windows.
- 2. Dans le menu Fichier, choisir Paramètre de mappage, et décocher la case Rendre les chemins absolus dans le code généré si elle était cochée.
- 3. Pour chaque composant de mappage, ouvrir le dialogue **Propriétés** (en double-cliquant la barre de titre du composant, par exemple), et modifier tous les chemins de fichier d'absolus en relatifs. Ensuite, cocher la case **Enregistrer tous les chemins de fichier relatifs au fichier MFD**. Pour un plus grand confort, vous pouvez copier tous les fichiers d'entrée et les schémas dans le même dossier que le mappage lui-même, et les référencer uniquement par le nom de fichier.

Pour plus d'informations concernant la gestion des chemins relatifs et absolus pendant la conception des mappage, , voir <u>Utiliser des chemins relatifs et absolus</u><sup>(32)</sup>.

Chose importante, MapForce Server et FlowForce Server prennent en charge tous les deux un soit-disant "répertoire de travail" par rapport auquel tous les chemins relatifs seront résolus, voir aussi <u>Chemins dans des</u> <u>environnements d'exécution variés</u><sup>85</sup>. Le répertoire de travail est spécifié au moment de l'exécution du mappage, comme suit :

- Dans FlowForce Server, en éditant le paramètre "Répertoire de travail" de toute tâche.
- Dans MapForce Server API, par le biais de la propriété WorkingDirectory de l'API COM et .NET, ou par le biais de la méthode setWorkingDirectory de l'API.
- Dans la ligne de commande MapForce Server, le répertoire de travail est le répertoire actuel du shell de commande.

#### Connexions de base de données

Sachez que les connexions ADO, ADO.NET et ODBC ne sont pas prises en charge sur les appareils Linux et macOS. Ainsi, si l'appareil cible est Linux ou macOS, ces connexions sont converties en JDBC lorsque vous déployez le mappage sur FlowForce ou lorsque vous compilez le mappage sur un fichier d'exécution MapForce Server. Dans ce cas, vous disposez des options suivantes avant de déployer le mappage ou de le compiler dans un fichier d'exécution de serveur :

- Dans MapForce, créer une connexion JDBC vers une base de données (voir <u>Configurer une connexion</u> <u>JDBC</u><sup>182</sup>).
- Dans MapForce, remplir les détails de connexion de la base de données JDBC dans la section "Paramètres spécifiques à JDBC" du composant de base de données (voir <u>Paramètres de composant</u> <u>de base de données</u><sup>(281)</sup>).

Si le mappage utilise une connexion native à une base de données PostgreSQL ou SQLite, la connexion native est préservée et aucune conversion JDBC n'aura lieu, voir <u>Mappages de base de données dans plusieurs</u> environnements d'exécution<sup>157</sup>. Si le mappage se connecte à une base de données basée sur fichier, comme Microsoft Access et SQLite, une configuration supplémentaire est nécessaire, voir "Bases de données basée sur fichier" ci-dessous.

L'exécution de mappage avec des connexions JDBC exige que le Java Runtime Environment ou le Java Development Kit soit installé sur la machine du serveur. Il peut s'agir soit d'un Oracle JDK soit d'un build open source comme Oracle OpenJDK.

- La variable d'environnement JAVA\_HOME doit pointer vers le répertoire d'installation JDK.
- Sur Windows, un chemin Java Virtual Machine trouvé dans le registre de Windows prendra la priorité sur la variable JAVA\_HOME.

 La plateforme JDK (64-bit, 32-bit) doit être la même que celle de MapForce Server. Sinon, vous risquez d'obtenir une erreur : "JVM est inaccessible".

#### Pour configurer une connexion JDBC sur Linux ou macOS:

- 1. Télécharger le pilote JDBC fournit par le revendeur de base de données et l'installer sur le système d'exploitation. Veillez à choisir la version 32-bit si votre système d'exploitation fonctionne sur 32-bit, et la version 64-bit si votre système d'exploitation fonctionne sur 64-bit.
- Définir les variables d'environnement à l'emplacement sur lequel le pilote JDBC est installé. Généralement, vous devrez définir la variable CLASSPATH, et éventuellement quelques autres variables. Afin de déterminer les variables d'environnement spécifiques que vous devez configurer, consulter la documentation fournie avec le pilote JDBC.
- **Note** : Sur macOS, le système prévoit que toute bibliothèque JDBC installée se trouve dans le répertoire /Library/Java/Extensions. C'est pourquoi nous recommandons de déballer le pilote JDBC à cet emplacement ; sinon, vous devrez configurer le système pour chercher la bibliothèque JDBC dans le chemin dans lequel vous avez installé le pilote JDBC.

#### Connexions Oracle Instant Client sur macOS

Ces instructions sont applicables si vous vous connectez à une base de données Oracle par le biais de Oracle Database Instant Client, sur macOS. Conditions préalables :

- Java 8.0 ou plus est installé. Si l'appareil Mac est exécuté avec une version Java précédente à Java 8, vous pouvez aussi vous connecter par le biais de la bibliothèque JDBC Thin for All Platforms et ignorer les instructions ci-dessous.
- Oracle Instant Client doit être installé. Vous pouvez télécharger l'Oracle Instant Client depuis la page de téléchargement officielle. Veuillez noter qu'il y a plusieurs packs Instant Client packages disponibles sur la page de téléchargement Oracle. Veuillez vous assurer de sélectionner un pack avec un prise en charge Oracle Call Interface (OCI), (par exemple, Instant Client Basic). De même, veillez à choisir la version 32-bit si votre système d'exploitation fonctionne sur 32-bit, et la version 64-bit si votre système d'exploitation fonctionne en 64-bit.

Une fois que vous avez téléchargé et déballé l'Oracle Instant Client, éditer le fichier de liste de propriété (.plist) envoyé avec le programme d'installation de manière à ce que les variables d'environnement suivantes pointent vers l'emplacement des chemins de pilote correspondants, par exemple :

Variable	Valeur échantillon
CLASSPATH	/opt/oracle/instantclient_11_2/ojdbc6.jar:/opt/oracle/instantclien t_11_2/ojdbc5.jar
TNS_ADMIN	/opt/oracle/NETWORK_ADMIN
ORACLE_HOME	/opt/oracle/instantclient_11_2
DYLD_LIBRARY_PATH	/opt/oracle/instantclient_11_2
PATH	<pre>\$PATH:/opt/oracle/instantclient_11_2</pre>

Note : Éditer les valeurs d'échantillon ci-dessus pour correspondre aux chemins sur lesquels les fichiers

Oracle Instant Client sont installés sur votre système d'exploitation.

#### Bases de données basées sur fichier

Les bases de données basées sur fichier comme Microsoft Access et SQLite ne sont pas incluses dans le paquet déployé sur FlowForce Server ou dans le fichier d'exécution MapForce Server compilé. Ainsi, si l'appareil source et cible ne sont pas identiques, suivez les étapes suivantes :

- 1. Dans MapForce, cliquer avec la touche de droite sur le mappage et décocher la case **Rendre** chemins absolus dans le code généré (voir <u>Changer les paramètres de mappage</u><sup>112</sup>).
- Cliquer avec la touche de droite sur le composant de base de données sur le mappage et ajouter une connexion au fichier de base de données en utilisant un chemin relatif, voir <u>Configurer le chemin vers</u> les Bases de données sur base de fichier<sup>12</sup>. Un moyen simple d'éviter les problème liés au chemin est d'enregistrer le design de mappage (fichier .mfd) dans le même répertoire que le fichier de base de données et de référer dans ce dernier depuis le mappage uniquement par le nom de fichier (en utilisant donc un chemin relatif).
- Copier le fichier de base de données dans un répertoire sur l'appareil cible (appelons-le "répertoire de travail"). Nous vous conseillons de garder ce répertoire en tête, il sera nécessaire pour exécuter le mappage sur le serveur, tel que montré ci-dessous.

Pour exécuter ce type de mappage sur le serveur, suivre une des étapes suivantes :

- Si le mappage est exécuté par MapForce Server sous le contrôle de FlowForce Server, configurer la tâche FlowForce Server pour pointer vers le répertoire de travail créé précédemment. Le fichier de base de données doit résider dans le répertoire de travail.
- Si le mappage est exécuté par le MapForce Server autonome dans la ligne de commande, changer le répertoire actuel en répertoire de travail (par exemple, cd path\to\working\directory) avant d'appeler la commande run de MapForce Server.
- Si le mappage est exécuté par l'API MapForce Server, définir le répertoire de travail par programme avant d'exécuter le mappage. Pour faciliter la tâche, la propriété WorkingDirectory est disponible pour l'objet MapForce Server dans l'API COM et .NET. Dans l'API Java, la méthode setWorkingDirectory est disponible.

Si les appareils source et cible sont des appareils Windows exécutés sur le réseau local, une autre approche est de configurer le mappage pour lire le fichier de la base de données depuis un répertoire partagé commun, comme suit :

- 1. Stocker le fichier de base de données dans un répertoire qui est accessible aussi bien par l'appareil source que cible.
- Cliquer avec la touche de droite sur le composant de base de données sur le mappage et ajouter une connexion au fichier de base de données en utilisant un chemin absolu (voir <u>Configurer le chemin vers</u> les Bases de données sur base de fichier<sup>(62)</sup>).

#### **Ressources globales**

Si un mappage inclut les références aux Ressources globales au lieu des chemins directs ou des connexions de base de données, vous pourrez également utiliser des Ressources globales du côté serveur. Lorsque vous compilez un mappage dans un fichier d'exécution MapForce Server (.mfx), les références aux Ressources globales resteront intactes afin que vous puissiez les fournir du côté serveur, lors de la marche du mappage. Lorsque vous déployez un mappage vers FlowForce Server, vous pouvez choisir en option s'il doit utiliser des ressources sur le serveur.

Pour que les mappages (ou des fonctions de mappage, dans le cas de FlowForce Server) puissent être exécutés avec succès, le fichier lui-même, le dossier ou les détails de connexion de base de données que vous fournissez en tant que Ressources globales doivent être compatibles avec le nouvel environnement de serveur. Par exemple, les chemins de fichier et de dossier doivent utiliser la convention Linux pour des chemins si le mappage sera exécuté sur un serveur Linux. De même, les Ressources globales définies en tant que connexions de base de données doivent être possibles sur l'appareil du serveur.

Pour plus d'informations, voir <u>Ressources globales dans MapForce Server</u><sup>(99)</sup> et <u>Ressources globales dans</u> <u>FlowForce Server</u><sup>(99)</sup>.

### 9.4 Compiler des mappages sur des fichiers d'exécution MapForce Server

Lorsque la langue cible d'un mappage créé dans MapForce est défini sur BUILT-IN, celui-ci peut être exécuté non seulement par MapForce, mais aussi par MapForce Server (voir <u>À propos de MapForce Server</u><sup>160</sup>). Il existe deux moyens d'exécuter un mappage avec MapForce Server:

- Si MapForce Server est exécuté en mode autonome (donc si aucun FlowForce Server n'est installé), le mappage doit être compilé dans un fichier d'exécution de serveur (.mfx), comme indiqué ci-dessous. Vous pouvez ensuite exécuter le fichier .mfx au niveau de la ligne de commande, en utilisant la commande run. Vous pouvez aussi exécuter le mappage en invoquant la méthode run de l'API MapForce Server. Pour plus d'informations, voir la documentation MapForce Server (https://www.altova.com/fr/documentation).
- Si MapForce Server est exécuté sous la gestion de FlowForce Server, le mappage peut être déployé sur un appareil sur lequel sont exécutés MapForce Server et FlowForce Server (voir <u>Déployer des</u> <u>mappages sur FlowForce Server</u><sup>(60)</sup>).

#### Conditions préalables

Voir <u>Préparer des mappages pour l'exécution de serveur</u><sup>951</sup>.

#### Pour compiler un mappage dans un fichier MapForce Server Execution (.mfx) :

- 1. Ouvrir un mappage dans MapForce (par exemple, myMapping.mfd).
- 2. Dans le menu Fichier, cliquer sur Compiler dans fichier MapForce Server Execution.
- 3. Choisir le dossier dans lequel vous souhaitez placer le fichier .mfx et changer le nom du fichier le cas échéant.
- 4. Cliquer sur **Enregistrer**. Le fichier MapForce Server Execution **myMapping.mfx** est généré dans le dossier choisi.

## Pour compiler un mappage dans un fichier MapForce Server Execution (.mfx), en utilisant la ligne de commande :

• Exécuter MapForce au niveau de la ligne de commande et spécifier le fichier de mappage et l'option de ligne de commande / COMPILE.

Par exemple, la commande suivante compile le mappage C: \Users\altova\Documents\Altova\MapForce2023\MapForceExamples\SimpleTotal.mfd dans un fichier d'exécution MapForce Server qui sera créé dans le répertoire de sortie cible C:\Users\altova\Desktop.

```
"C:\Program Files (x86)\Altova\MapForce2023\MapForce.exe" "C:
\Users\altova\Documents\Altova\MapForce2023\MapForceExamples\SimpleTotal.mfd" /COMPILE
"C:\Users\altova\Desktop"
```

Voir aussi l'Interface de ligne de commande MapForce<sup>965</sup>.

#### Qu'est-ce qui est inclus dans le fichier .mfx

Le fichier .mfx comprend les données suivantes :

- L'algorithme de mappage, qui comprend toutes les fonctions définies par l'utilisateur (FDUs) importées depuis les autres mappages.
- Les noms de fichier d'entrée et de sortie référencés depuis des composants. Les chemins sont absolus ou relatifs, selon les paramètres du mappage, voir <u>Chemins dans des environnement</u> <u>d'exécution différents</u><sup>(85)</sup>.
- Si le mappage contient des composants XML, les informations nécessaires concernant le schéma XML pour pouvoir exécuter le mappage sont encodées dans l'algorithme de mappage.
- Les détails de connexion de la base de données, si le mappage contient des connexions à la base de données. Les mots de passe sont chiffrés.

Les fichiers d'instance d'entrée (XML, CSV, Text) qui sont utilisés par le mappage ne sont pas inclus dans le fichier .mfx compilé. La même chose vaut pour les bases de données à base fichier comme Access ou SQLite. Pour plus de détails, voir <u>Préparer des mappages pour l'exécution de serveur</u><sup>(851)</sup>.

#### Compiler des mappages pour une version MapForce Server spécifique

Si votre MapForce Server est une version plus ancienne que celle de MapForce, elle risque de ne pas pouvoir exécuter les fichiers .mfx créés avec une version plus récente de MapForce, étant donné que des nouvelles fonctions auront probablement été ajoutées entre-temps. Dans ces cas, vous pouvez compiler le fichier .mfx pour une version spécifique de MapForce Server, comme suit :

- 1. Dans le menu Outils, cliquer sur Options, puis cliquer sur Génération.
- 2. Sous Fichier d'exécution de serveur, à côté de Générer pour la version MapForce, choisir la version MapForce Server requise depuis la liste déroulante.

Fichier d'exécution de serveur	DBC erver)
Générer pour la version MapForce Server : plus actuel V	

Si vous possédez une version de MapForce Server plus récente, n'oubliez pas de modifier cette option. Si vous n'avez pas de raison particulière pour compiler sur une version spécifique de MapForce Server, sélectionner l'option la "plus actuelle" (il s'agit de l'option par défaut). Lorsque cette option est choisie, le fichier .mfx est compilé pour la version la plus récente de MapForce Server et peut profiter des toutes dernières fonctions et améliorations qui pourraient ne pas être disponibles dans les versions plus anciennes.

Pour spécifier une version MapForce Server cible dans la ligne de commande, exécuter la commande /COMPILE avec /MFXVERSION, par exemple :

"C:\Program Files (x86)\Altova\MapForce2023\MapForce.exe" /COMPILE /MFXVERSION:2023

Voir aussi l'Interface de ligne de commande MapForce<sup>965</sup>.

#### Autres options

La compilation des Fichiers d'exécution de MapForce est aussi concernée par les options suivantes :

Convertir toutes les connexions de base de données ADO et ODBC sur JDBC	Si cette option est activée, les connexions de base de données ADO, ADO.NET et ODBC sont transformées en JDBC en utilisant le pilote JDBC et l'URL de base de données définie dans le dialogue Paramètres de composant de base de données (voir <u>Paramètres de composant de base de données</u> ). La connexion JDBC sera utilisée implicitement si l'appareil cible est un serveur Linux ou macOS.
Ignorer les signatures	Cette option est uniquement applicable pour MapForce Enterprise. Elle est
numériques (non prises en	activée par défaut. Si le mappage utilise des signatures numériques XML, il
charge par MapForce	saute l'information de signature numérique, puisque MapForce Server ne prend
Server)	pas en charge des signatures numériques XML.

Pour consulter ou modifier ces options :

• Dans le menu Outils, cliquer sur Options, puis cliquer sur Génération.

Ces options sont aussi disponibles depuis l'interface de ligne de commande. Voir aussi l'<u>Interface de ligne de</u> commande MapForce<sup>965</sup>.

### 9.5 Déployer des mappages sur FlowForce Server

Le fait de déployer un mappage sur FlowForce Servers signifie que MapForce organise les ressources utilisées par le mappage spécifique dans un objet et les passe par HTTP (ou HTTPS si configuré) sur l'appareil sur lequel FlowForce Server est exécuté. Les mappages MapForce sont généralement déployés sur FlowForce Server pour automatiser leur exécution par le biais des tâches FlowForce Server. Une fois qu'un mappage est déployé, vous pouvez créer une tâche FlowForce Server contenant toutes les fonctions que vous souhaitez, et profiter de toutes les fonctions spécifiques à la tâche (par exemple, définir des conditions de déclenchement personnalisées pour la tâche, l'exposer en tant que service Web, etc.).

Note : Le terme "appareil source" se réfère à l'ordinateur sur lequel MapForce est installé et le terme "appareil cible" se réfère à l'ordinateur sur lequel FlowForce Server est installé. Dans le scénario le plus simple, il s'agit du même ordinateur. Dans des scénarios plus avancés, MapForce est exécuté sur un appareil Windows alors que FlowForce Server est exécuté sur un appareil Linux ou macOS.

Le pack déployé sur FlowForce contient les éléments suivants :

 Le mappage lui-même. Après déploiement, le mappage devient disponible dans l'interface d'administration FlowForce Server en tant que fonction de mappage (.mapping), sous le chemin que vous spécifiez. Tout composant de source devient des arguments d'entrée, et tous les composants de cible deviennent des arguments de sortie de cette fonction.

4	Execute funct	tion /public/Comple	etePO.map	pin	9	▼ 🖉	
	Parameters:	Customers:	(input)	뎥	altova://packagedfile/C:/Users/	いたいあいさな ひょう おびらう ひろう ひなんり	/MapForceExamples/Customers.xm
		Articles:	(input)	뎚	altova://packagedfile/C:/Users/		/MapForceExamples/Articles.xml
		ShortPO:	(input)	욚	altova://packagedfile/C:/Users/		/MapForceExamples/ShortPO.xml
		CompletePO:	(output)	욚	CompletePO.xml		
		Working-directory	:		+		

• Toute sorte de fichiers d'instance d'entrée (XML, CSV, Text) qui sont utilisés par le mappage.

#### Prérequis

Voir <u>Préparer des mappages pour l'exécution de serveur</u><sup>951</sup>.

#### Déployer le mappage sur FlowForce Server

- Exécutez MapForce et s'assurer que le langage de transformation est défini dans BUILT-IN (soit cliquer sur la touche de la barre d'outils Built-in <sup>BUILT</sup>, soit sélectionner la commande de menu Sortie | Moteur d'exécution Built-in).
- 2. Dans le menu **Fichier**, cliquer sur Déployer sur **FlowForce Server**. Le dialogue Déployer mappage s'ouvre.

Deploy Mappir	ng	×	
Enter the host name mapping.	e and port of a FlowForce Administration Interface to deploy the curre	ent	
Server:	localhost $\checkmark$ Port: 8082		
<u>U</u> ser:	root Use SSL		
Pass <u>w</u> ord:	••••		
Login:	Directly		
-Global Resources	on the Server		
Use <u>R</u> esources			
Resource Path:	/public/		
	The path must start with a slash character,		
Deploy As			
<u>P</u> ath:	/public/CompletePO.mapping Browse		
The path must start with a slash character.			
S <u>a</u> ve mapping	before deploying		
Open web bro	owser to create new job		
	OK Cancel		

3. Saisissez vos paramètres de déploiement (comme décrit ci-dessous), et cliquez sur OK. Si vous avez coché la case **Ouvrir navigateur pour créer une nouvelle tâche**, l'interface d'administration FlowForce Server s'ouvre dans le navigateur, et vous pouvez commencer à créer une tâche FlowForce Server immédiatement.

La table suivante liste les paramètres de déploiement de mappage disponibles dans le dialogue Déployer mappage.

Paramètre	Description
Serveur, Port et Utiliser SSL	Saisir le nom d'hôte du serveur (ou adresse IP) et le port de FlowForce Server. Ceux-ci pourraient être <b>localhost</b> et <b>8082</b> si FlowForce Server est exécuté sur le même appareil au port par défaut. En cas de doute, se connecter sur l'interface d'administration FlowForce Server Web et contrôler l'adresse IP et le port affiché dans la barre d'adresse du navigateur Web. Si vous êtes confronté à des erreurs de connectivité, s'assurer que l'appareil sur lequel FlowForce Server est exécuté est configuré pour permettre des connexions entrantes sur l'adresse et le port désigné.

Paramètre	Description
	Pour déployer le mappage par le biais d'une connexion chiffrée par SSL, cocher la case <b>Utiliser SSL</b> . Cela part du principe que FlowForce Server est déjà configuré pour accepter les connexions SSL. Pour plus d'informations, veuillez vous référer à la documentation FlowForce Server ( <u>https://www.altova.com/fr/documentation</u> .
Utilisateur et mot de passe	Le nom d'utilisateur et le mot de passe à saisir dépendent de la valeur de la liste déroulante de Login (voir option suivante). Si la liste déroulante de Login est définie sur <b><default></default></b> ou sur <b>Directement</b> , saisir votre nom d'utilisateur et mot de passe FlowForce Server. Sinon, saisir votre nom d'utilisateur et votre mot de passe de domaine et choisir le nom de domaine depuis la liste déroulante Login.
Connexion	Si Directory Service est activé dans FlowForce Server, choisir le nom de domaine depuis cette liste déroulante, et saisir vos identifiants de domaine dans les champs Utilisateur et Mot de passe (voir des options précédentes).
Utiliser des Ressources, Chemin de Ressource	Sélectionnez la case à cocher <b>Utiliser Ressources</b> si la fonction du mappage doit utiliser <u>Ressources</u> après l'avoir déployé sur le serveur. Si vous cochez la case, vous devez aussi saisir le chemin de la ressource respective dans le champ de saisie <b>Chemin de ressource</b> . Pour sélectionner la ressource, cliquer sur la touche <b>Ellipse</b> .
	S'il n'y a pas encore de ressources à sélectionner sur le serveur, cliquer sur <b>Déployer Ressources Globales</b> , et déployez la Ressource Globale requise sur le serveur. Pour plus d'informations, voir <u>Déployer des</u> <u>Ressources Globales sur FlowForce Server</u> <sup>(33)</sup> .
	Si vous ne cochez pas la case <b>Utiliser des ressources</b> , toute Ressource Globale sera résolue, sur la base de la configuration actuellement choisie. Sur le serveur, la fonction de mappage ne nécessitera plus de Ressources Globales, mais utilisera la valeur résolue à la place.
Chemin	Cliquez sur <b>Navigateur</b> , et sélectionnez le chemin dans lequel la fonction de mappage doit être enregistrée dans la hiérarchie du conteneur de FlowForce Server. Par défaut, le chemin est défini sur le conteneur <b>/public</b> de FlowForce Server.
	Depuis le dialogue, vous pouvez aussi créer de nouveaux conteneurs ou supprimer les conteneurs et les mappages existants, pourvu que vous déteniez les permissions et les privilèges de FlowForce Server requis.
Enregistrer le mappage avant le déploiement	Cette option est disponible si vous déployez un mappage non enregistré. Cocher cette case pour enregistrer le mappage avant le déploiement.
Ouvrir le navigateur pour chercher une nouvelle tâche	Si vous cochez cette case, l'interface d'administration FlowForce Server Web s'ouvre dans le navigateur après le déploiement et vous pouvez commencer à créer une tâche FlowForce Server immédiatement.

#### Dépannage

Le tableau suivant regroupe les problèmes qui risque de survenir lors du déploiement d'un mappage, suivi des solutions.

Problème	Solution
Le déploiement du mappage retourne l'erreur suivante : I/O operation on file failed. I/O Error 28: Failed to connect to <server> port 8082. Timed out System error 10060: Un essai de connexion a échoué car la partie connectée n'a pas répondu après une période de temps ou à établi une connexion échouée car l'hôte connecté n'a pas répondu.</server>	Assurez-vous que sur l'appareil cible, le service <i>FlowForce Web Server</i> est en cours d'exécution et est configuré pour écouter les connexions sur le port spécifié ( <b>8082</b> , par défaut). De même, veuillez vous assurer que le pare-feu ne bloque pas des connexions entrantes par ce port. Le service <i>FlowForce Server</i> doit être exécuté pour que le déploiement soit possible.
Le déploiement du mappage retourne l'erreur suivante : I/O operation on file failed. I/O Error 413: Payload Too Large	Cette erreur peut apparaître si le fichier d'entrée du mappage déployé dépasse la limite de taille maximum des requêtes HTTP autorisées par FlowForce Server (environ 100 MB). Vous pouvez augmenter la limite en définissant l'option max_request_body_size (en octets) dans les fichiers <b>flowforceweb.ini</b> et <b>flowforce.ini</b> . Pour plus de détails, voir la documentation FlowForce Server.

#### Sélectionner la version de serveur (uniquement Windows)

Si le serveur sur lequel vous déployez le mappage a plusieurs versions de MapForce Server exécutées sous la gestion de FlowForce Server (uniquement applicable sur des serveurs Windows), alors vous serez également invité à spécifier la version de MapForce Server avec laquelle vous souhaitez que ce mappage soit exécuté.

Select MapForce	Server			×
Multiple versions of MapForce Server were found which can all execute this mapping.				
Select the most appropriate version automatically				
O Choose version <u>m</u> anually:				
Version:	2017r3	$\sim$		
			OK Cano	cel

**Note :** Le dialogue apparaît lorsque le répertoire d'installation de FlowForce Server contient des fichiers .tool pour chaque version MapForce Server exécutée sous gestion FlowForce Server. Par défaut, un fichier MapForce Server .tool est ajouté automatiquement dans ce répertoire lorsque vous installez MapForce Server sous forme d'installation FlowForce Server. Le chemin sous lequel les fichiers .tool sont stockés dans FlowForce est : **C:\Program Files\Altova\FlowForceServer2023\tools.** Si vous

utilisez des versions supplémentaires de MapForce Server que vous souhaitez faire fonctionner sous une gestion FlowForce Server, leurs fichiers .tool peuvent être copiés manuellement dans le répertoire ci-dessus. Le fichier .tool de MapForce Server peut être trouvé sous : C:\Program Files\Altova\MapForceServer2023\etc.

### 9.6 Interface de ligne de commande MapForce

La syntaxe générale d'une commande MapForce sous la ligne de commande :

```
MapForce.exe <filename> [/{target} [[<outputdir>] [/options]]]
```

#### Légende

Les annotations suivantes sont utilisées pour indiquer une syntaxe de ligne de commande :

Notation	Description
Texte sans crochets ou parenthèses	Items que vous devez saisir comme indiqué
<texte chevrons="" dans="" des=""></texte>	Caractère générique pour laquelle vous devez fournir une valeur
[Texte dans des crochets]	Items optionnels
{Texte dans des accolades}	Ensemble d'items requis ; en choisir un
Barre verticale ( )	Le séparateur pour des items mutuellement exclusifs ; en choisir un
Ellipse ()	ltems pouvant être répétés

#### <filename>

Le fichier de design de mappage (.mfd) ou le projet de mappage (.mfp) à partir duquel le code doit être généré. Pour générer le code pour tout le projet, définir la cible /GENERATE (voir description ci-dessous) et saisir le chemin de projet sous <filename>, par exemple, **MapForceExamples.mfp**.

#### /{target}

Spécifie le langage cible ou l'environnement pour lequel le code doit être généré. Les cibles de génération de code suivants sont pris en charge.

Cible	Description
/COMPILE[:compileoptions]	Compile un mappage vers un fichier d'exécution MapForce Server (.mfx). En option, les options suivantes peuvent être fournies, délimitées par une virgule :
	JDBC Transforme toutes les connexions de base de données sur JDBC en utilisant sur le pilote JDBC et l'URL de base de données défini dans le dialogue Paramètres de composant de base de données, voir aussi <u>Paramètres de composant de base de</u> <u>données</u> <sup>(281)</sup> .

Cible	Description	
	NOXMLSIGNATURES	Suppression de la génération des signatures numériques dans le fichier MapForce Server Execution (noter que les signatures numériques ne sont pas prises en charge par MapForce Server).
/GENERATE	Génère un code de pro projet en utilisant les p <u>dossiers de projet</u> <sup>(113)</sup> . de fournir un projet Map	ojet pour tous les mappages dans le fichier de paramètres de dossier actuel, voir <u>Gérer les</u> Si vous choisissez cette cible, assurez-vous pForce (fichier .mfp) en tant que <filename>.</filename>
/XSLT	Génère un code XSLT	1.0.
/XSLT2	Génère un code XSLT	2.0.
/XSLT3	Génère un code XSLT	3.0.
/XQuery	Génère un code XQuery.	
/JAVA	Génère un code Java.	
/CS	Génère un code C#. Cette commande permet aussi en option de définir des options spécifiques pour la génération de code, concrètement :	
	/CS[:{VS2010 VS2013	3 VS2015 VS2017 VS2019}]
	VS2010	Visual Studio 2010
	VS2013	Visual Studio 2013
	VS2015	Visual Studio 2015
	VS2017	Visual Studio 2017
	VS2019	Visual Studio 2019
	Si aucune version de Visual Studio n'est spécifiée, le code sera généré en utilisant la version Visual Studio définie dans les options de génération de code, voir <u>Options de générateur de code</u> .	
/CPP	Génère un code C++. Cette commande permet aussi en option de définir des options spécifiques pour la génération de code, concrètement :	
	/CPP[:{VS2010 VS201 DLL},{MFC NoMFC}]	L3 VS2015 VS2017},{MSXML XERCES3},{LIB
	Le premier groupe d'op Valeurs valides :	tion définit la version cible de Visual Studio.

Cible	Description	
	VS2010	Visual Studio 2010
	VS2013	Visual Studio 2013
	VS2015	Visual Studio 2015
	VS2017	Visual Studio 2017
	VS2019	Visual Studio 2019
	Le deuxième groupe code généré. Valeurs	d'option spécifie la bibliothèque XML ciblée par le valides :
	MSXML	Génère un code pour MSXML 6.0
	XERCES3	Génère un code pour Xerces 3
	Le troisième groupe o par rapport aux biblio Valeurs valides :	l'option spécifie si les bibliothèques statiques hèques dynamiques doivent être générées.
	LIB	Générer des bibliothèques statiques LIB
	DLL	Générer des bibliothèques DLL
	Le quatrième groupe ou sans prise en cha	d'option spécifie si le code doit être généré avec rge MFC. Valeurs valides :
	MFC	Active la prise en charge MFC
	NoMFC	Désactive la prise en charge MFC
	Si les options ci-dessous ne sont pas spécifiées, le code sera généré en utilisant la version Visual Studio définies dans les options de génération de code, voir <u>Options de générateur de code</u> <sup>[20]</sup> .	

#### <outputdir>

Paramètre optionnel qui spécifie le répertoire de sortie. Si un chemin de sortie n'est pas fourni, le répertoire de travail actuel sera utilisé. Veuillez noter que tout chemin de fichier relatif sera relatif par rapport au répertoire de travail actuel.

Lorsque la cible est /GENERATE et que le paramètre <outputdir> n'est pas défini, le langage de génération de code, ainsi que le chemin de sortie de chaque mappage sont fournis par les paramètres définis pour chaque dossier dans le projet, voir <u>Gérer les dossiers de projet</u><sup>(118)</sup>.

Lorsque la cible est /GENERATE et que le paramètre <outputdir> est défini, la valeur <outputdir> fournie au niveau de la ligne de commande prend précédence sur le directoire de sortie définie au niveau du projet root.

Néanmoins elle ne prend pas précédence sur les paramètres de génération de code définis dans chaque dossier dans le projet.

#### /options

/options ne sont pas exclusives mutuellement. Une ou plusieurs des options suivantes peuvent être définies.

Option	Description
[/MFXVERSION[: <version>]</version>	Cette option est applicable si la cible est /COMPILE. Elle compile le fichier MapForce Server Execution (.mfx) pour une version particulière de MapForce Server. Vous pouvez fournir en tant que valeur une version de MapForce Server, en commençant à partir de <b>2013r2</b> , jusqu'à la version actuelle de MapForce. Voir aussi <u>Compiler des mappages pour</u> <u>une version MapForce Server spécifique</u> <sup>633</sup> .
/GLOBALRESOURCEFILE <filename></filename>	Cette option est applicable si le mappage utilise des Ressources globales pour résoudre un fichier d'entrée ou de sortie ou des chemins de dossier ou des bases de données. Pour plus d'informations, voir <u>Ressources globale Altova</u> <sup>971</sup> . L'option /globalresourcefile spécifie le chemin vers un fichier .xml
	de Ressources globales. Veuillez noter que, si /globalResourceFile est défini, alors /globalResourceConFig doit aussi être défini.
/GLOBALRESOURCECONFIG <config></config>	Cette option spécifie le nom de la configuration de Ressource globale (voir aussi la version précédente). Veuillez noter que, si /globalresourcefile est défini, alors /globalresourceconfig doit aussi être défini.
/LIBRARY <libname> ()</libname>	Utiliser avec un langage cible de génération de code pour spécifier des bibliothèques de fonctions supplémentaires. Cette option peut être spécifiée plus d'une fois pour charger plusieurs bibliothèques. Voir aussi <u>aGérer des Bibliothèques de fonction</u> <sup>555</sup> .
/LOG <logfilename></logfilename>	Génère un fichier de journal sous le chemin spécifié. <logfilename> peut être un nom de chemin complet, par exemple, il peut inclure un nom de répertoire et de fichier. Néanmoins, si un chemin complet est fourni, le répertoire doit exister pour que le fichier de journal soit généré. Si vous spécifiez uniquement le nom de fichier, celui-ci sera placé dans le répertoire actuel de l'invite de commande Windows.</logfilename>

#### Remarques

- Les chemins relatifs sont relatifs par rapport au répertoire de travail, qui est le répertoire actuel de l'application appelant MapForce. Cela s'applique au chemin du nom de fichier .mfd, chemin du nom de fichier .mfp, au répertoire de sortie, au nom de fichier de journal, et au nom de fichier de ressource globale.
- Ne pas utiliser la barre oblique de la fin et les guillemets de fermeture au niveau de la ligne de commande (par exemple, "C:\My directory\"). Ces deux caractères sont interprétés par le parseur de ligne de commande en tant que double guillemet littéral. Utiliser les barres obliques renversées

doubles \\ si les espaces se produisent au niveau de la ligne de commande et que vous devez utiliser les guillemets ("c:\My Directory\\"), ou essayer d'éviter d'utiliser les espaces et donc les guillemets.

#### **Exemples**

1) Pour lancer MapForce et ouvrir le mappage <filename>.mfd, utiliser :

MapForce.exe <filename>.mfd

Pour générer un code XSLT 2.0 et créer un fichier log portant le nom <logfilename>, utiliser :

MapForce.exe <filename>.mfd /XSLT2 <outputdir> /LOG <logfilename>

3) Pour générer un code XSLT 2.0 en prenant en compte la configuration de ressource globale <grconfigname> depuis le fichier de ressource globale <grfilename>, utiliser :

Mapforce.exe <filename>.mfd /XSLT2 <outputdir> /GLOBALRESOURCEFILE
<grfilename> /GLOBALRESOURCECONFIG <grconfigname>

4) Pour générer une application C# pour Visual Studio 2015 et sortir un fichier log, utiliser :

MapForce.exe <filename>.mfd /CS:VS2015 <outputdir> /LOG <logfilename>

5) Pour générer une application C++ en utilisant les paramètres de génération de code définis dans **Outils** | **Options**, et sortir un fichier log, utiliser :

MapForce.exe <filename>.mfd /CPP <outputdir> /LOG <logfilename>

6) Pour générer une application C++ pour Visual Studio 2015, MSXML, avec des bibliothèques statiques, prise en charge de MFC et pas de fichier log, utiliser :

MapForce.exe <filename>.mfd /CPP:VS2015,MSXML,LIB,MFC

7) Pour générer une application C++ pour Visual Studio 2015, Xerces, avec des bibliothèques dynamiques, pas de prise en charge MFC et un fichier log, utiliser :

MapForce.exe <filename>.mfd /CPP:VS2015,XERCES,DLL,NoMFC <outputdir> /LOG <logfilename>

8) Pour générer une application Java et sortir un fichier log, utiliser :

MapForce.exe <filename>.mfd /JAVA <outputdir> /LOG <logfilename>

9) Pour générer un code pour tous les mappages dans le projet, en utilisant le langage et le répertoire de sortie définis dans les paramètres du dossier (de chaque dossier dans le projet), utiliser :

MapForce.exe <filename>.mfp /GENERATE /LOG <logfilename>

10) Pour générer un code Java pour tous les mappages dans le fichier de projet, utiliser :

MapForce.exe <filename>.mfp /JAVA /LOG <logfilename>

Veuillez noter que le langage de génération de code défini dans les paramètres de dossier sont ignorés et que Java est utilisé pour tous les mappages.

11) Pour fournir des fichiers d'entrée et de sortie dans la ligne de commande pour un mappage Java compilé précédemment, utiliser :

```
java -jar <mappingfile>.jar /InputFileName <inputfilename> /OutputFileName
<outputfilename>
```

Les paramètres /InputFileName et /OutputFileName sont les noms de composants d'entrée spéciaux dans le mappage MapForce qui vous permettent d'utiliser des paramètres dans l'exécution de la ligne de commande (voir Fournir des paramètres au mappage 477).

12) Pour compiler un mappage dans un fichier d'exécution MapForce Server, pour la version MapForce Server 2023, et réprimant les signatures XML :

MapForce.exe <filename>.mfd /COMPILE:NOXMLSIGNATURES <outputdir> /MFXVERSION:2023 /LOG
<logfilename>

### 10 Ressources globales Altova

Les Ressources globales d'Altova sont des alias pour des fichiers, dossiers et ressources de bases de données. Chaque alias peut avoir de multiples configurations et chaque configuration correspond à une seule ressource. Pour cette raison, quand vous utilisez une ressource globale, vous pouvez basculer entre ses configurations. Par exemple, vous pouvez créer une ressource "database" avec deux configurations : développement et production. Dépendant de vos objectifs, vous pouvez basculer entre ces configurations. Dans MapForce, vous pouvez aisément extraire des données soit de la base de données de développement ou de production, en choisissant la configuration souhaitée à partir d'une liste déroulante avant de consulter le mappage.

Les Ressources globales peuvent être utilisées dans toutes les différentes applications d'Altova (voir la soussection ci-dessous).

#### Ressources globales dans d'autres produits d'Altova

Lorsqu'ils sont stockés en tant que Ressources globales, les détails de connexion à la base de données deviennent réutilisables et disponibles dans plusieurs applications d'Altova. Par exemple, si vous avez souvent besoin d'ouvrir le même fichier dans plusieurs applications desktop Altova, vous pourriez le définir en tant que Ressources globales. Si vous devez modifier le chemin de fichier, vous allez devoir le changer à un endroit uniquement. Actuellement, les Ressources globales peuvent être définies et utilisées dans les produits Altova suivants :

- Altova Authentic
- DatabaseSpy
- MobileTogether Designer
- <u>MapForce</u>
- <u>StyleVision</u>
- <u>XMLSpy</u>
- FlowForce Server
- <u>MapForce Server</u>
- RaptorXML Server et RaptorXML+XBRL Server.

#### Dans cette section

Cette section explique comment créer et configurer différents types de ressources globales. La section est organisée en rubriques suivantes :

- <u>Configuration des RY</u><sup>972</sup>
- <u>Ressources globales, Partie 1<sup>972</sup></u>
- Configuration des Ressources globales, Partie 2<sup>974</sup>
- Fichiers XML en tant que Ressources globales<sup>978</sup>
- <u>Dossiers en tant que Ressources globales</u><sup>980</sup>
- Bases de données en tant que Ressources globales<sup>982</sup>
- Résultats de transformation comme Ressources globales<sup>984</sup>
- Ressources globales dans des environnements d'exécution<sup>988</sup>

### **10.1** Configuration des Ressources globales, Partie 1

La configuration des Ressources globales a lieu en deux parties : (i) créer une ressource globale dans la boîte de dialogue **Gérer les Ressources globales** (*voir ci-dessous*) et (ii) définir les propriétés de cette ressource globale dans le dialogue **Ressource globale**. La deuxième partie est discutée dans la <u>prochaine rubrique</u><sup>974</sup>.

Les Ressources globales d'Altova sont définies dans le dialogue **Gérer les Ressources globales**, qui peuvent être accédées de deux manières :

- Cliquez sur la commande de menu Outils | Ressources globales.
- Cliquez sur l'icône Gérer les Ressources globales dans la barre d'outils des Ressources globales (voir la capture d'écran ci-dessous).



#### Fichier de définition des ressources globales.

Les informations sur les ressources globales sont stockées dans le fichier XML appelé fichier de définition des Ressources globales. Ce fichier est créé lorsque la première ressource globale est définie dans la boîte de dialogue **Gérer les Ressources globales** (*voir la capture d'écran ci-dessous*) et enregistré.

Gérer les Ressources globales	×
Fichier de définitions C:\Users\j.cannas\Documents\Altova\GlobalResources.xml	<mark>⊯</mark> Parcourir
Déployer vers serveur	
E Fichiers	+ Ajouter
Cossiers / Base URLs	A Education
🕀 👘 Bases de données	
	Supprimer
OK	Annuler

Lorsque vous ouvrez la boîte de dialogue **Gérer les Ressources globales** pour la première fois, l'emplacement par défaut et le nom du fichier de définition des Ressources globales sont spécifiés dans la zone de texte *Définitions (voir la capture d'écran ci-dessus)*:

#### C:\Users\<username>\Documents\Altova\GlobalResources.xml

Ce fichier est défini comme fichier de définition des Ressources globales par défaut pour toutes les applications d'Altova. Une ressource globale peut être enregistrée depuis toute application d'Altova dans ce fichier et sera
immédiatement disponible dans toutes les autres applications en tant que ressource globale. Pour définir et enregistrer une ressource globale dans le fichier de définition des ressources globales, ajoutez une ressource globale dans le dialogue **Gérer les Ressources globales** et cliquez sur **OK** pour l'enregistrer.

Pour sélectionner un fichier de définition des ressources globales déjà existant afin d'en faire le fichier de définition des ressources globales d'une application d'Altova spécifique, recherchez-le par le biais du bouton **Naviguer** de la zone de texte *Définitions (voir la capture d'écran ci-dessus)*.

La boîte de dialogue **Gérer les Ressources globales** vous permet également d'éditer et de supprimer des ressources globales existantes.

#### Notes :

- Vous pouvez donner n'importe quel nom au fichier de définition des ressources globales et l'enregistrer à un emplacement accessible à vos applications Altova. Tout ce que vous devez faire dans chaque application est spécifier ce fichier comme fichier de définition des Ressources globales pour cette application (dans la zone de texte *Définitions*). Les ressources deviennent des produits à échelle globale d'Altova lorsque vous utilisez un seul fichier de définitions dans l'ensemble des produits globaux d'Altova.
- Vous pouvez aussi créer de multiples fichiers de définition des ressources globales. Toutefois, seulement un de ces fichiers peut être actif à tout moment dans une application donnée d'Altova, et uniquement les définitions contenues dans ce fichier seront disponibles pour l'application. La disponibilité des ressources peut être restreinte pour cette raison ou conçue pour se chevaucher sur l'ensemble des produits, tel que requis.

# **10.2** Configuration des Ressources globales, Partie 2

La deuxième partie de la configuration des ressources globales consiste en la définition de propriétés d'une ressource globale dans la boîte de dialogue **Ressource globale**. Les propriétés dépendent du type de ressource globale (*voir les sous-sections ci-dessous*). Vous pouvez accéder la boîte de dialogue **Ressource globale** en cliquant sur le bouton **Ajouter** dans la <u>boîte de dialogue Gérer les Ressource globales</u><sup>(972)</sup>.

Pour en savoir plus sur la configuration des différents types de ressources globales, voir les exemples suivants : <u>Fichiers XML en tant que Ressources globales</u><sup>978</sup>, <u>Dossiers en tant que Ressources globales</u><sup>980</sup>, <u>Bases de données en tant que Ressources globales</u><sup>982</sup>.

#### Fichiers

Les propriétés spécifiques au fichier sont affichées dans la boîte de dialogue **Ressource globale** ci-dessous. La configuration est divisée en trois parties majeures : (i) le nom du fichier, (ii) l'emplacement de ce fichier, et (iii) la liste de configurations définies dans l'alias du fichier.

😢 Ressource globale 🛛 🕹		
Nom alias <u>A</u> lias ressource : Rappo	rtsMatinaux	
Configurations          Image: second system         Image: second system	Paramètres pour configuration "Default"	

Les paramètres *Résultat de la transformation de MapForce* et *Résultat de la transformation de StyleVision* sont discutés dans <u>Résultats de transformation en tant que Ressources globales</u><sup>(84)</sup>.

#### Dossiers

Les propriétés spécifiques au dossier sont affichées dans la boîte de dialogue **Ressource globale** ci-dessous. La configuration est divisée en trois parties majeures : (i) le nom du dossier, (ii) l'emplacement de ce dossier, et (iii) la liste de configurations définies dans l'alias du fichier.

😢 Ressource globale		×
Nom alias <u>A</u> lias ressource : Rapp	portsMatinaux	
Configurations	Paramètres pour configuration "ProductionDirectory"	

### Bases de données

Lorsque vous ajoutez une connexion de base de données en tant que Ressource globale, un assisant à la connexion vous guide à travers les étapes requises pour configurer la connexion. Pour plus d'informations, voir <u>Démarrer l'assistant de connexion à la base de données</u><sup>(10)</sup>. Une fois que l'assistant a terminé, les paramètres de connexion de la base de données sont affichés dans la boîte de dialogue **Ressource globale** (*voir la capture d'écran ci-dessous*).

😫 Ressource globale				×
Nom alias				
Alias ressource : Sou	urceDatabase			
Configurations				
+ + *	Paramètres pour conf	iguration "Default"		
Default				
ReleaseDatabase	E Choisir la base	de données		
	Base de données :			
	⊽ General			^
	Connection String		Data Source=C:\Users\altova\Documents\Altova\MapForce2018\MapFor tova.mdb:Provider=Microsoft.jet.OLEDB.4.0	ceExample
	Root Object			
	Database Kind		MS Access	
	Import Kind		ADO	
	Database Version		4	
	∇ Connection			×
	<			>
	Paramètres d'exécut	ion spécifiques à MapForce		
	$\nabla$			
	DataSource	C:\Users\altova\Document	s\Altova\MapForce2018\MapForceExamples\altova.mdb	
	Catalog	altova		
	Provider	Microsoft.Jet.OLEDB.4.0		
	JDBCDatabaseURL	jdbc:odbc:;DRIVER=Micros	oft Access Driver (*.mdb);	
	JDBCDriver	sun.jdbc.odbc.JdbcOdbcDr	iver	
	UserID			
	Password			
	OtherProperties			
	1			
			ОК	Annuler

Dans le dialogue **Ressource Globale**, il est possible d'éditer certains paramètres de connexions de base de données. Les paramètres sont groupés en deux catégories : les paramètres de base de données et les paramètres d'exécution spécifiques à MapForce (*voir ci-dessous*).

#### Base de données

Ces paramètres sont partagés parmi les applications Altova. Dans MapForce, ils sont utilisés au moment de la conception, c'est-à-dire, lorsque le mappage est chargé, ou lorsque vous cliquez sur le volet **Sortie** dans MapForce pour consulter le mappage.

#### Paramètres d'exécution spécifiques à MapForce

Ces paramètres sont applicables lorsque vous générez du code de programmation ou compilez un mappage sur le fichier d'exécution MapForce Server (.mfx). Ils sont utilisés lors de l'exécution du mappage comme suit :

- Dans le code de programme C++, C# ou Java généré.
- Si vous compilez le mappage dans un fichier d'exécution MapForce Server, la conversion JDBC automatique a lieu. Pour plus d'informations sur la conversion JBCD automatique, voir <u>Mappages de base de données dans des environnements d'exécution divers</u><sup>157</sup>.

Si un mappage utilise une Ressource globale pour se connecter à une base de données, les détails de connexion de base de données dans le dialogue **Ressource globale** prend précédence sur ceux définis dans le mappage. La boîte de dialogue Paramètres de composant vous informe que les paramètres de connectivité

sont définis en tant que Ressource globale. Pour changer le composant de base de données pour se connecter à la base de données directement (sans utiliser de Ressources globales), cliquez sur **Changer**, et suivre les étapes de l'assistant pour se reconnecter à la base de données.

### Icônes du dialogue Ressource globale

÷	<i>Ajouter une configuration :</i> Affiche la boîte de dialogue Ajouter une configuration dans laquelle vous saisissez le nom de la configuration à ajouter.
<b>-</b> 10	<i>Ajouter une configuration en tant que copie :</i> Affiche la boîte de dialogue Ajouter une configuration dans laquelle vous pouvez saisir le nom de la configuration à créer en tant que une copie de la configuration choisie.
*	Supprimer : Supprime la configuration sélectionnée.
ß	Ouvrir : Rechercher le fichier à créer en tant que ressource globale.
	<i>Ouvrir :</i> Rechercher le dossier à créer en tant que ressource globale.

# Configuration de la Ressources globale : Procédures générales

La procédure générale de création et de configuration de ressources globales est décrite ci-dessous :

- 1. Cliquez sur le 🛃 bouton de la barre d'outils (Gérer les Ressources globales). En alternative, allez au menu Outils and cliquez sur Ressources globales.
- 2. Cliquez sur **Ajouter** et choisissez le type de ressource que vous souhaitez créer (fichier, dossier, base de données). La boîte de dialogue **Ressource globale** apparaîtra.
- 3. Saisir un nom descriptif dans la zone de texte Alias de ressource (par ex., InputFile).
- 4. Définir la configuration par défaut dépend du type de la ressource globale : (i) pour un fichier ou dossier, recherchez le fichier ou dossier vers lequel cette ressource devrait pointer par défaut ; (ii) pour une connexion de base de données, cliquez sur **Choisir base de données** et suivez l'assistant de connexion à la base de données pour se connecter à la base de données (voir <u>Se connecter à une base de données</u>). Cette connexion de base de données sera utilisée par défaut lorsque vous exécutez le mappage.
- 5. Si vous avez besoin d'une configuration supplémentaire (par ex., un dossier de sortie additionnel), cliquez sur le bouton 📩 dans la boîte de dialogue **Ressource globale**, saisissez le nom de cette configuration et spécifiez le chemin vers cette configuration.
- 6. Répétez l'étape précédente pour chaque configuration supplémentaire requise.
- **Note :** les connexions de base de données sont prises en charge comme ressources globales uniquement dans les éditions de MapForce Professional and Enterprise.

# **10.3** Fichiers XML en tant que Ressources globales

Cette rubrique explique comment utiliser les fichiers XML en tant que ressources globales. Il existe des situations où vous allez éventuellement devoir changer le fichier XML de nombreuses fois par jour. Par exemple, chaque matin, vous devez exécuter un mappage particulier et générer un rapport en utilisant un fichier XML en tant qu'entrée de mappage, et chaque soir le même rapport doit être généré depuis un autre fichier XML. Au lieu d'éditer le mappage plusieurs fois par jour (ou garder de multiples copies de celui-ci), vous pourriez configurer le mappage de telle manière qu'il lise d'un fichier défini comme ressource globale (un genre de *alias du fichier*). Dans cet exemple, vos alias de fichier auront deux configurations :

- 1. La configuration par défaut fournit un fichier XML « matinal » en tant qu'entrée de mappage.
- 2. La configuration RapportsDuSoir fournit un fichier XML « du soir » en tant qu'entrée de mappage.

Pour créer et configurer un alias de fichier, suivez les étapes ci-dessous.

### Étape 1 : Créer une ressource globale

D'abord, nous devons créer un alias de fichier. Suivez les instructions ci-dessous :

- 1. Cliquez sur le 🛃 bouton de la barre d'outils (Gérer les Ressources globales). En alternative, allez au menu Outils and cliquez sur Ressources globales.
- 2. Cliquez sur **Ajouter | Fichier** et saisissez le nom dans la zone de texte **alias de Ressource**. Dans cet exemple, nous appelons notre configuration par défaut RapportsMatinaux.
- 3. Cliquez sur le bouton **Parcourir** à côté du champ de texte **La Ressource pointera vers ce fichier** et sélectionnez Tutorial\mf-ExpReport.xml.
- 4. Cliquez sur 📩 dans la section **Configurations** et nommez cette deuxième configuration RapportsDuSoir.
- 5. Cliquez sur **Parcourir** et sélectionnez Tutorial\mf-ExpReport2.xml.

# Étape 2 : Utiliser la Ressource Globale dans le mappage

Maintenant, nous pouvons utilisez la ressource globale nouvellement créée dans notre mappage. Pour que le mappage lise de la ressource globale, suivez les étapes ci-dessous :

- 1. Ouvrez le mappage Tutorial\Tut-ExpReport.mfd.
- 2. Double-cliquez sur l'en-tête du composant source pour ouvrir une boîte de dialogue **Paramètres de** composant.
- 3. À côté du fichier XML d'entrée, cliquez sur Parcourir, puis cliquez sur Ressources globales et sélectionnez l'alias de fichier RapportsMatinaux. Cliquer sur Ouvrir.
- 4. Ouvrez une nouvelle fois la boîte de dialogue **Paramètres de composant** : Le chemin de fichier XML d'entrée est maintenant devenu altova://file\_resource/MorningReports, ce qui indique que le chemin utilise une Ressource Globale.

### Étape 3 : Exécuter le mappage avec la configuration désirée

Vous pouvez désormais basculer entre les fichiers XML d'entrée avant d'exécuter le mappage :

- Pour utiliser mf-ExpReport.xml comme entrée, sélectionnez l'élément de menu Outils | Configuration active | Par défaut.
- Pour utiliser mf-ExpReport2.xml comme entrée, sélectionnez l'item de menu Outils | Configuration active | RapportsDuSoir.

En alternative, sélectionnez la configuration requise depuis la liste déroulante **Ressources Globales** dans la barre d'outils (*voir la capture d'écran ci-dessous*).



Pour consulter le résultat de mappage directement dans MapForce, cliquez sur l'onglet Sortie.

# **10.4 Dossiers en tant que Ressources globales**

Cette rubrique explique comment utiliser les dossiers en tant que ressources globales. Il existe des situations dans lesquelles vous devrez générer la même sortie dans différents répertoires. À cette fin, nous devons créer un alias de dossier avec deux configurations :

- 1. La configuration Default générera la sortie dans C:\Test.
- 2. La configuration Production générera la sortie dans C:\ par.

Pour créer et configurer un alias de dossier, suivez les étapes ci-dessous.

# Étape 1 : Créer une ressource globale

D'abord, nous devons créer un alias de dossier. Suivez les instructions ci-dessous :

- 1. Cliquez sur le 🛃 bouton de la barre d'outils (Gérer les Ressources globales). En alternative, allez au menu Outils and cliquez sur Ressources globales.
- 2. Cliquez sur **Ajouter | Dossier** et saisissez un nom dans la zone de texte **alias de Ressource**. Dans cet exemple, nous appelons notre configuration par défaut OutputDirectory.
- Cliquez sur le bouton Parcourir à côté du champ de texte Paramètres pour une configuration "Par défaut" et sélectionnez C:\Test. Assurez-vous que ce dossier existe déjà dans votre système d'exploitation.
- 4. Cliquez sur 📩 et saisissez un nom pour la deuxième configuration. Dans cet exemple, nous appelons notre deuxième configuration ProductionDirectory.
- 5. Cliquez sur **Parcourir** et sélectionnez le dossier C: \Production. Assurez-vous que ce dossier existe déjà dans votre système d'exploitation.

# Étape 2 : Utiliser la Ressource Globale dans le mappage

La prochaine étape est de faire en sorte que le mappage utilise l'alias de dossier que nous venons de créé. Suivez les étapes ci-dessous :

- 1. Ouvrez le mappage Tutorial\Tut-ExpReport.mfd.
- 2. Double-cliquez sur l'en-tête du composant cible pour ouvrir la boîte de dialogue **Paramètres de composant**.
- 3. Cliquez sur **Ressources globales**, puis cliquez sur **Enregistrer**.
- 4. Enregistrez le fichier XML de sortie comme output.xml. Le chemin de fichier XML de sortie est maintenant devenu altova://folder\_resource/OutputDirectory/Output.xml, qui indique que le chemin est défini comme ressource globale.

# Étape 3 : Exécuter le mappage avec la configuration désirée

Vous pouvez désormais basculer entre les dossiers de sortie avant d'exécuter le mappage :

- Pour utiliser C: \Test en tant que configuration de sortie, sélectionnez l'item de menu Outils |
   Configuration active | Par défaut.
- Pour utiliser C: \Production comme répertoire de sortie, sélectionnez l'item de menu Outils | Configuration active | ProductionDirectory.

Par défaut, la sortie de mappage est écrite en tant que fichier temporaire, à moins que vous ayez configuré explicitement MapForce pour écrire les sorties vers les fichiers permanents. Pour configurer MapForce afin qu'il génère les fichiers générés, suivez les étapes suivantes :

- 1. Allez au menu Outils et cliquez sur Options.
- 2. Dans la section **Généralités**, choisissez l'option **Écrire directement dans les fichiers de sortie finaux**.

# **10.5** Bases de données en tant que Ressources globales

Cette rubrique explique comment utiliser les bases de données tant que ressources globales. Vous rencontrerez des situations dans lesquelles vous aurez besoin de mapper des données depuis des bases de données avec la même structure mais différentes données. Utiliser une ressource de base de données qui vous permet de basculer entre bases de données sans éditer votre mappage. À cette fin, nous devons créer un alias de base de données avec deux configurations :

- 1. La configuration Default pointera vers la DevelopmentDatabase : MapForceExamples\altova.mdb.
- 2. La configuration ReleaseDatabase pointera vers Tutorial\altova.mdb.

Pour créer et configurer un alias de base de données, suivez les étapes ci-dessous.

## Étape 1 : Créer une ressource globale

La première étape est de créer un alias de base de données. Suivez les étapes ci-dessous :

- 1. Cliquez sur le 🛃 bouton de la barre d'outils (Gérer les Ressources globales). En alternative, allez au menu Outils and cliquez sur Ressources globales.
- 2. Cliquez sur Ajouter | Base de données et saisissez un descriptif nom dans la zone de texte alias de Ressource. Dans cet exemple, nous appelons la configuration par défaut DevelopmentDatabase.
- 3. Cliquez sur **Choisir Base de données**, sélectionnez **Microsoft Access (ADO)** et cherchez MapForceExamples\altova.mdb.
- 4. Cliquez sur 📩 et nommez la deuxième configuration ReleaseDatabase.
- 5. Cliquez sur **Choisir Base de données**, sélectionnez **Microsoft Access (ADO)** et cherchez Tutoriel\altova.mdb.

### Étape 2 : Utiliser la Ressource Globale dans le mappage

Dans la prochaine étape, nous allons configurer le mappage pour que vous puissiez utiliser l'alias de base de données :

- 1. Ouvrez le mappage Tutorial\PersonDB.mfd.
- 2. Double-cliquez sur le composant de base de données pour ouvrir la boîte de dialogue **Paramètres de** composant. Cliquez sur **Changer**.
- 3. Sélectionnez **Ressources globales** dans la boîte de dialogue **Sélectionner une base de données** et sélectionnez l'alias DevelopmentDatabase. Cliquez sur Connexion.
- 4. Vous êtes invité à sélectionner les objets de base de données, laissez la sélection par défaut telle quelle et cliquez sur **OK**.

Les paramètres de connectivité peuvent être modifiés en cliquant sur le bouton de la barre d'outils

### Étape 3 : Exécuter le mappage avec la configuration désirée

Vous pouvez maintenant facilement passer entre les bases de données avant d'exécuter le mappage comme suit :

- Pour utiliser la configuration DevelopmentDatabase, sélectionnez l'item de menu Outils | Configuration active | Par défaut.
- Pour utiliser la configuration ReleaseDatabase, sélectionnez l'item de menu Outils | Configuration active |ReleaseDatabase.

En alternative, sélectionnez la configuration requise depuis la liste déroulante **Ressources Globales** (*voir la capture d'écran ci-dessous*).

Default	- 🕄 🚽
,	

Lorsque vous passez entre les configurations, la boîte de dialogue **Configuration switch** vous informe que la ressource a été modifiée. Cliquez sur **Recharger**.

**Note :** les bases de données utilisées dans cet exemple contiennent les mêmes données. C'est la raison pour laquelle il n'y a pas de différence dans les sorties générées.

# 10.6 Résultats de transformation MapForce et StyleVision comme Ressources globales

Vous pouvez utiliser le résultat d'un mappage de MapForce ou d'une transformation StyleVision comme ressource globale. Cette rubrique vous montre comment créer une ressource globale depuis le résultat de transformation et utilise cette ressource globale dans toutes les différentes applications d'Altova.

Afin de rendre une sortie de mappage disponible comme ressource globale, le langage de transformation du mappage doit être défini comme Built-In, ou le mappage doit contenir d'autres composants qui sont pris en charge par le langage Built-In.

#### Important :

- Les flux de travail mentionnés ci-dessus sont significatifs entre les applications desktop d'Altova installées sur le même ordinateur.
- Il n'est **pas** possible d'utiliser le résultat des transformations de MapForce et StyleVision comme ressources globales dans les produits du serveur Altova et dans MapForce Basic Edition.

L'exemple ci-dessous montre comment utiliser le résultat d'une transformation de MapForce comme ressource globale.

## Exemple : Résultat de la transformation de MapForce

Cet exemple illustre comment créer un flux de travail entre Altova MapForce et <u>Altova XMLSpy</u>, utilisant les Ressources globales. Plus spécifiquement, l'exemple montre comment créer une ressource globale depuis un mappage de MapForce, déclencher l'exécution de ce mappage dans XMLSpy, et visualiser les sorties dans XMLSpy, qui ont été générés par MapForce.

# Étape 1 : Créer une ressource globale

Vous pouvez prendre cette étape en utilisant MapForce ou XMLSpy.

- 1. Cliquez sur le 🛃 bouton de la barre d'outils (Gérer les Ressources globales). En alternative, allez au menu Outils and cliquez sur Ressources globales.
- 2. Cliquez sur Ajouter | Fichier et saisissez le descriptif nom dans la zone de texte alias de Ressource. Dans cet exemple, nous appelons notre configuration par défaut RésultatDeMappage.
- 3. Sélectionnez l'option Résultat de la transformation de MapForce.
- 4. Cliquez sur **Parcourir** et sélectionnez le mappage Tutorial\Tut-ExpReport-multi.mfd. Comme indiqué ci-dessous ce mappage à une entrée et deux sorties.

🙄 Ressource globale			×
Nom alias Alias ressource : Mappir	ngResult		
Configurations + 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Paramètres pour configura Fichier Résultat de la transfo Résultat de la transfo La Ressource pointera vi C: \Users\altova \Docum Saisies Nom mf-ExpReport	ation "Default" formation MapForce formation StyleVision ers la sortie générée de : ents\Altova\MapForce2018\MapForceExamples\Tutorial\Tut Chemin d'accès	42
			_
	Sorties (utiliser le bouton	n radio pour choisir)	_
	Nom	Chemin d'accès	^
	ExpReport-larget     ExpReport Target	C:\temp\file1.xml	
	S Expreport-larger		
	1		
		OK Annul	er

La capture d'écran ci-dessous illustre les deux sorties dans la boîte de dialogue de la **Ressource globale**. Nous générerons chaque fichier de sortie séparément dans le dossier C:\temp (voir l'étape 2 ci-dessous).



### Étape 2 : Générer des fichiers de sortie

À cette étape, nous aimerions générer chacun des deux fichiers de sortie dans le dossier (*voir la capture d'écran ci-dessus*) dans le dossier c:\temp et changer les noms de fichier. Pour ce faire, nous allons créer une configuration pour chaque sortie. Suivez les étapes ci-dessous :

- 1. Dans la section Sorties de la boîte de dialogue **Ressource globale**, cliquez sur **Parcourir** à côté de la première sortie et saisissez C:\temp\file1.xml comme nom de fichier de destination. Il s'agit de la configuration par défaut qui produira le premier fichier de sortie.
- 2. Cliquez sur 📩 sous les *Configurations* et saisissez un nom pour la nouvelle configuration (dans cet exemple, output2). Dans la section *Sorties*, cliquez sur la case d'option à côté du deuxième fichier (secondXML.xml).
- Dans la section Outputs, cliquez sur Parcourir à côté du deuxième sortie et saisissez c: \temp\file2.xml en tant que nom de fichier de destination. Il s'agit de la deuxième configuration qui produit le deuxième fichier de sortie.
- 4. Cliquez sur OK.

# Étape 3 : Utiliser la Ressource globale

Les instructions ci-dessous vous montre comment utiliser la ressource globale que nous avons créé à l'étape précédente.

#### Configuration par défaut

Pour utiliser la configuration par défaut dans XMLSpy, suivez les étapes ci-dessous.

- 1. Exécuter XMLSpy.
- 2. Allez au menu Outils et cliquez sur Ressources globales.
- 3. Dans la section *Fichiers*, cliquez sur la ressource globale MappingResult, puis cliquez sur **Aperçu** (*voir la capture d'écran ci-dessous*). Ceci permet d'exécuter le mappage, produit la sortie par défaut (file1.xml) et le charge dans le volet principal de XMLSpy. Le fichier est enregistré comme c: \temp\file1.xml.

🛃 Manage Global Resources		×
Definitions file: C:\Users\altova\Docun	nents\Altova	<mark>⊯ B</mark> rowse
📮 🚰 Files	~	+⇒ <u>A</u> dd
MappingResult		// Eda
Folders / Base URLs		✓ Eait
····· 🍙 Databases		🗱 <u>D</u> elete
		🔍 <u>V</u> iew
	$\sim$	
<	>	
E	OK	Cancel

#### Deuxième configuration

Pour déclencher l'exécution de mappage avec la deuxième configuration, procédez comme suit :

- 1. Allez au menu Outils dans XMLSpy et cliquez sur Configuration Active | Output2.
- 2. Cliquez sur Recharger lorsque vous y êtes invité.

Par conséquent, le deuxième fichier de sortie est chargé dans la fenêtre principale de XMLSpy. Le fichier est enregistré comme C:\temp\file2.xml.

# 10.7 Ressources globales dans des environnements d'exécution variés

Cette sous-section explique comment travailler avec les ressources globales dans différents environnements d'exécution. La sous-section est organisée en rubriques suivantes :

- <u>Ressources globales dans le code généré</u>
   <sup>988</sup>
- <u>Ressources globales dans MapForce Server</u><sup>989</sup>
- <u>Ressources globales dans FlowForce Server</u><sup>(88)</sup>

# **10.7.1** Ressources globales dans XSLT, XSLT2, XQuery

Cette rubrique explique comment es ressources globales sont utilisées dans le code généré. Pour plus d'information, veuillez voir les sous-sections ci-dessous.

#### Ressources globales dans XSLT, XSLT2, XQuery

Lorsque vous générez un code XSLT ou XSLT2 et que le mappage utilise des Ressources globales, cela n'affecte pas la feuille de style XSLT générée. Avec ou sans ressources globales, vous pouvez spécifier de manière flexible les fichiers d'entrée et de sortie quand vous exécutez la feuille de style XSLT dans votre processeur XSLT. La même chose s'applique au code XQuery généré.

Une exception est le fichier **DoTransform.bat** généré pour l'exécution RaptorXML. Les Ressources globales utilisées par le mappage seront résolues dans les chemins actuels dans **DoTransform.bat**. La configuration étant actuellement sélectionnée depuis la liste déroulante des ressources globales sera prise en compte. Pour plus d'informations concernant la livraison de Ressources globales dans RaptorXML, voir <u>la documentation de RaptorXML</u>.

#### Ressources globales dans C++, C#, Java

Lorsque vous générez un code de programme C#, C++, ou Java, Ressource globale utilisée par le mappage sera résolue. Par exemple, un alias de fichier ou de dossier défini en tant que Ressource globale sera converti dans le chemin de fichier ou de dossier actuel. Si une configuration de Ressource globale particulière est sélectionnée depuis la liste déroulante des Ressources globales, le code sera généré pour la configuration sélectionnée. La fenêtre **Messages** affiche des informations expliquant exactement comment une Ressource globale a été résolue (*voir capture d'écran ci-dessous*).

м	lessages
R	
	CompletePO.mfd: Mapping validation successful - 0 error(s), 0 warning(s)
	╦ 🗖 🤡 <u>CompletePO.mfd</u> : Code generation completed successfully 0 error(s), 0 warning(s)
2	🙀 🥼 😫 CompletePO: Resolved global resource 'altova://folder_resource/CodeGeneration/CompletePO.xml' to 'C:\codegen\CompletePO.xml'

Afin de générer le code pour une configuration de Ressource globale particulière, choisissez-le depuis la liste déroulante de la Ressource globale avant de générer le code. En alternative, si vous générez du code depuis la ligne de commande, fournissez les paramètres globalresourcefile et globalresourceconfig au niveau de la ligne de commande. Pour plus d'informations, voir Interface de ligne de commande MapForce<sup>953</sup>.

Il n'est pas possible de basculer vers les ou faire référence aux Ressources globales depuis le code généré. À la place, vous pouvez modifier le code pour changer le chemin de fichier d'entrée ou de sortie.

Note : dans C# ou Java, vous pouvez modifier le chemin et le type de données de l'entrée ou de la sortie<sup>100</sup>.

# **10.7.2** Ressources globales dans MapForce Server

Lorsque vous compilez un mappage dans un fichier d'exécution MapForce Server (.mfx), toutes les références de ressources globales utilisées par le mappage sont préservées, mais pas résolues. Cela signifie que vous devrez fournir ces références du côté du serveur pour pouvoir exécuter le mappage avec succès. Dans MapForce Server, l'étape suivante est requise pour exécuter un fichier .mfx qui utilise les ressources globales :

- Le fichier de définition des ressources globales. Sur l'appareil sur lequel MapForce est installé, le fichier est appelé GlobalResources.xml. Vous trouverez ce mappage dans le dossier Documents\Altova. Vous pouvez copier ce fichier sur l'appareil sur lequel MapForce est exécuté et créer plusieurs fichiers de ce type, le cas échéant. Voir aussi <u>Configuration des Ressources globales</u>. Partie 1<sup>(372)</sup>.
- Le nom de configuration de la Ressource globale. Chaque Ressource globale a une configuration par défaut. Vous pouvez aussi créer des configurations supplémentaires. Pour plus d'informations, voir <u>Configuration des Ressources globales, Partie 2</u><sup>974</sup>.

Dans MapForce, le fichier de définition des ressources globales et le nom de configuration de la Ressource globale sont définis ou modifiés depuis l'interface utilisateur graphique. Dans MapForce Server, elles sont spécifiées lors de l'exécution du mappage (*voir ci-dessous*).

 Si vous exécutez le mappage par l'interface de ligne de commande, définissez les options -globalresourceconfig et --globalresourcefile après la commande run, par exemple :

```
C:\Program Files (x86)\Altova\MapForceServer2023\bin\MapForceServer.exe run
SomeMapping.mfx --globalresourcefile="C:
\Users\me\Documents\Altova\GlobalResources.xml" --globalresourceconfig="Default"
```

 Si vous exécutez le mappage par l'API de MapForce Server, appelez la méthode setoptions deux fois avant d'appeler la méthode Run. Le premier appel est nécessaire pour fournir le chemin de fichier de définition de la Ressource Globale en tant qu'option, et le second appel est nécessaire pour fournir le nom de configuration de la Ressource Globale.

Pour plus d'informations, voir la documentation de MapForce Server.

# **10.7.3** Ressources globales dans FlowForce Server

Dans FlowForce Server, les ressources globales ne sont pas stockées dans un fichier XML comme dans les applications desktop. Dans FlowForce, chaque ressource est un objet réutilisable qui peut contenir des chemins de fichier ou de dossier ou des détails de connexion de base de données. Les ressources peuvent être copiées, exportées et importées, et sont sujettes au même mécanisme d'accès d'utilisateur que d'autres objets de FlowForce Server. Ceci signifie que tout utilisateur FlowForce peut utiliser toute ressource dans ses fonctions de mappage, s'il est doté des permissions exigées.

Une fois que vous avez créé un mappage avec des ressources globales dans MapForce, vous pouvez le déployer sur FlowForce Server. Au moment du déploiement, vous pouvez choisir si le mappage doit utiliser des ressources globales, cochant alors la case **Utiliser les ressources** dans le dialogue de déploiement. Si vous ne cochez pas la case, toutes les ressources globales utilisées par le mappage seront résolues, basées sur la configuration sélectionnée actuellement. Si vous avez coché la case, la fonction de mappage nécessitera aussi des ressources dans FlowForce Server. La capture d'écran ci-dessous est un exemple de fonction de mappage déployée vers FlowForce qui exige des ressources pour être exécutée. Veuillez noter que le premier paramètre obtient un chemin par défaut de la ressource.

Function ReadJSON.mapping in /public		
Function Input Parameters		
Name:         People         (input)         Image: String         Default:         altova://file_resource/SourceFile		
Name: Text file (output) Type: string Default: Text file.csv		
Name:         Working-directory         Type:         string as directory         Default:		
Resources		
Run function using resources: /public/GlobalResources_Default.resources	▼ 🖉	

Dans FlowForce Server, c'est la fonction de mappage qui utilise des ressources globales, pas la tâche. La fonction de mappage lit le chemin du premier fichier d'entrée depuis la ressource. Cela signifie que toutes les tâches employant cette fonction utiliseront le même chemin, à moins que vous contourniez le chemin depuis la page de configuration de la tâche.

Vous pouvez aussi déployer des Ressources globales dans FlowForce Server en tant qu'objets autonomes. Cela signifie il n'est pas nécessaire de déployer d'abord un mappage pour pouvoir déployer une ressource globale. Pour plus d'informations, voir *Déployer des Ressources sur FlowForce Server* ci-dessous.

Pour plus d'informations concernant la consommation des ressources dans FlowForce Server, voir (<u>the</u> <u>FlowForce Server documentation</u>).

#### Déployer des ressources sur FlowForce Server

Vous pouvez déployer des Ressources Globales créées avec MapForce sur FlowForce Server. Lors du déploiement, vous devez choisir la configuration avec laquelle la ressource doit être déployée vers le serveur. Si vous avez besoin de toutes les configurations de la même ressource globale sur le serveur, vous pouvez déployer cette ressource globale plusieurs fois et choisir la configuration souhaitée avant chaque déploiement. Vous pouvez également changer le nom de chaque ressource globale sur le serveur et choisir le conteneur de destination sur le serveur.

Vous pouvez déployer des Ressources Globales sur FlowForce Server soit au moment où vous déployez le mappage ou séparément. Pour déployer les ressources globales sur FlowForce Server, suivez les étapes suivantes :

- 1. Exécuter MapForce.
- 2. Cliquez sur la touche de la barre d'outils Gérer les Ressources globales . En alternative, allez au menu Outils and cliquez sur Ressources globales.
- 3. Cliquer sur **Déployer sur le serveur**. Ceci ouvre la boîte de dialogue **Déployer la configuration de la Ressource globale** (*voir la capture d'écran ci-dessous*).

😢 Déployer une 🤇	Configuration de la Ressource globale	×
Saisir le nom de l'hôt des Ressources glob	te et le port d'une Interface d'administration FlowForce pour déployer pales.	
Serveur:	localhost $\checkmark$ Port: 8082	
Utilisateur:	root	
Mot de passe:	••••	
Connexion:	<défaut> ~</défaut>	
Ressources global	es	
Configurations	Default $\checkmark$	
Déployer en tant d	que	
Chemin d'accès:	/public/GlobalResources_Default.resources Parcourir	
	Le chemin doit débuter avec une barre oblique.	
	OK Annuler	

- 4. Saisir les détails de connexion sur FlowForce Server (serveur, port, utilisateur, mot de passe, méthode de connexion). Ces paramètres sont les mêmes que ceux requis lors du déploiement d'un mappage, voir <u>Déployer des mappages sur FlowForce Server</u><sup>960</sup>.
- 5. Sélectionnez une configuration depuis la liste de **Configuration**. Cette liste inclut toutes les configurations depuis le <u>fichier de définition des ressources globales</u> actuel. Veuillez noter que seule une configuration de ressource globale peut être déployée à la fois. Vous pouvez déployer la même ressource plusieurs fois avec un nom différent, si vous avez besoin de toutes les configurations sur le serveur.
- 6. Choisir un chemin cible où la ressource doit être enregistrée sur le serveur. Cliquer sur **Parcourir** pour afficher un dialogue dans lequel vous pouvez sélectionner un conteneur FlowForce cible, ou en créer un nouveau, le cas échéant.
- 7. Cliquez sur OK.

Vous pouvez voir l'information sur le déploiement des ressources globales sur FlowForce Server dans la fenêtre des **Messages**.

**Note :** les Ressources globales qui exécutent d'autres applications d'Altova ne sont *pas* prises en charge dans un environnement de serveur. Pour plus d'informations, voir <u>Résultats de transformation en tant</u> <u>que Ressources globale</u><sup>034</sup>.

# 11 Catalogs in MapForce

MapForce prend en charge un sous-ensemble de mécanismes de catalogue XML OASIS. Le mécanisme du catalogue permet à MapForce d'extraire des schémas communément utilisés (y compris d'autres fichiers) des dossiers utilisateurs locaux. Cela augmente la vitesse de traitement générale, permet aux utilisateurs de travailler hors ligne (c'est-à-dire sans connexion à un réseau) et améliore la portabilité des documents (parce que les URI ne devraient être modifiés uniquement dans les fichiers catalogue.)

Le mécanisme du catalogue dans MapForce fonctionne comme suit dans la présente section :

- <u>Comment fonctionnent les catalogues</u>
   <sup>994</sup>
- <u>Structure du catalogue dans MapForce</u>998
- Personnaliser vos catalogues<sup>998</sup>
- Variables d'Environnement<sup>1000</sup>

Pour plus d'informations sur les catalogues, voir la spécification de catalogues XML.

# **11.1 Comment fonctionnent les catalogues**

Les catalogues peuvent être utilisés pour rediriger les Schémas DTD et XML. Alors que le concept derrière les mécanismes dans les deux cas est le même, les détails sont différents et expliqués ci-dessous.

# DTD

Les catalogues sont communément utilisés pour rediriger un appel vers un DTD ou un URI local. Pour ce faire, des identifiants publics ou système sont mappés dans le fichier catalogue vers l'URI local requis. Donc, si la déclaration **DOCTYPE** dans un fichier XML est lue, son identifiant public ou système localise la ressource locale requise par le biais du mappage du fichier catalogue.

Pour les schémas populaires, l'identifiant **PUBLIC** est normalement prédéfini, requérant uniquement que l'URI dans le fichier catalogue mappe l'identifiant **PUBLIC** à la copie locale correcte. Lorsque le document XML est parsé, l'identifiant **PUBLIC** qui le compose est lu. Si cet identifiant est trouvé dans un fichier catalogue, l'URL correspondant dans le catalogue fichier sera consulté et le schéma sera lu depuis cet emplacement. Donc, par exemple, si le fichier SVG est ouvert dans MapForce :

L'identifiant **PUBLIC** de ce fichier SVG file est recherché dans le catalogue. Disons que le fichier catalogue contient l'entrée suivante :

#### catalog>

```
...
cpublic publicId="-//W3C//DTD SVG 1.1//EN" uri="schemas/svg/svgll.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 MapForce, 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.

# **11.2** Structure du catalogue dans MapForce

Lorsque MapForce 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 MapForce et est utilisé pour localiser les schémas et les feuilles de style utilisés par des processus spécifiques de MapForce, tels que StyleVision Power Stylesheets qui sont des feuilles de style utilisées pour générer le mode Authentique des documents XML de Altova.

#### Variables d'emplacement

Les variables utilisées dans RootCatalog.xml (liste ci-dessus) ont les valeurs suivantes :

%PersonalFolder%	Dossier personnel de l'utilisateur actuel, par exemple C: \Users\ <name>\Documents</name>
%CommonSchemasFolder%	C:\ProgramData\Altova\Common2023\Schemas

8	
ApplicationWritableDataFolde	
r%	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 MapForce.
- CustomCatalog.xml eSt SitUé dans votre dossier MyDocuments\Altova\MapForce.
- Les fichiers catalog.xml sont chacun dans un dossier de schéma spécifique, ces dossiers de schéma étant dans le dossier des Schémas communs.

# **11.3 Personnaliser vos catalogues**

Lorsque vous créez des entrées dans customCatalog.xml (ou tout autre fichier catalogue qui doit être lu par MapForce), 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 <u>spécification des catalogues XML</u>. Notez que chaque élément peut prendre l'attribut xml:base, qui est utilisé pour spécifier la base URI de cet élément.

- <public publicId="PublicID of Resource" uri="URL of local file"/>
- <system systemId="SystemID of Resource" uri="URL of local file"/>
- <uri name="filename" uri="URL of file identified by filename"/>
- <rewriteURI uriStartString="StartString of URI to rewrite" rewritePrefix="String to replace StartString"/>
- <rewriteSystem systemIdStartString="StartString of SystemID" rewritePrefix="Replacement string to locate resource locally"/>

Veuillez noter les points suivants :

- Dans le cas où il n'y a pas d'identifiant public, comme pour toutes les feuilles de style, l'identifiant système peut être directement mappé avec un URL par le biais de l'élément système.
- Un URI peut être mappé avec un autre URI en utilisant l'élément uri.
- Les éléments rewriteURI et rewriteSystem permettent la réécriture respectivement de la partie initiale d'un URI ou d'un identifiant système. Ceci permet de lancer un chemin de fichier à remplacer et, par conséquent, permet de cibler un autre répertoire. Pour plus d'informations sur les éléments, voir la <u>spécification des catalogues XML</u>.

À partir de la version 2014, MapForce adhère étroitement à la spécification <u>spécification des catalogues XML</u> (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 MapForce 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 MapForce trouve un schéma référencé

Un schéma est référencé dans un document XML par le biais de l'attribut xsi:scemaLocation (*voir cidessous*). 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 MapForce. 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 MapForce 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 MapForce des informations sur les assistants de saisie à des fins d'édition si vous vouliez créer des documents conformément à ces recommandations antérieures.

# 11.4 Variables d'Environnement

Les variables d'environnement shell peuvent être utilisées dans l'élément nextCatalog pour spécifier le chemin menant aux différents emplacements de système (*voir RootCatalog.xml liste ci-dessus*). Les variables d'environnement shell suivantes sont prises en charge :

%PersonalFolder%	Chemin complet vers le dossier Personnel de l'utilisateur actuel, par exemple c: \Users\ <name>\Documents</name>
%CommonSchemasFolder %	C:\ProgramData\Altova\Common2023\Schemas
% ApplicationWritableD ataFolder%	C:\ProgramData\Altova
%AltovaCommonFolder%	C:\Program Files\Altova\Common2023
%DesktopFolder%	Chemin complet vers le dossier Bureau de l'utilisateur actuel.
%ProgramMenuFolder%	Chemin complet vers le dossier Menu Programme pour l'utilisateur actuel.
%StartMenuFolder%	Chemin complet vers le dossier Démarrage pour l'utilisateur actuel.
%StartUpFolder%	Chemin complet vers le dossier Démarrage pour l'utilisateur actuel.
%TemplateFolder%	Chemin complet vers le dossier Modèle pour l'utilisateur actuel.
%AdminToolsFolder%	Chemin complet vers le répertoire de système de fichier qui stocke des outils administratifs pour l'utilisateur actuel.
%AppDataFolder%	Chemin complet vers le dossier Données d'Application pour l'utilisateur actuel.
%CommonAppDataFolder %	Chemin complet vers le répertoire de fichier contenant les données d'application pour tous les utilisateurs.
%FavoritesFolder%	Chemin complet du dossier Favoris pour l'utilisateur actuel.
%PersonalFolder%	Chemin complet vers le dossier Personnel pour l'utilisateur actuel.
%SendToFolder%	Chemin complet vers le dossier EnvoyerÀ pour l'utilisateur actuel.
%FontsFolder%	Chemin complet vers le dossier Polices Système.
%ProgramFilesFolder%	Chemin complet vers le dossier Fichiers Programme pour l'utilisateur actuel.
%CommonFilesFolder%	Chemin complet vers le dossier Fichiers Communs pour l'utilisateur actuel.
%WindowsFolder%	Chemin complet vers le dossier Windows pour l'utilisateur actuel.
%SystemFolder%	Chemin complet vers le dossier Système pour l'utilisateur actuel.
%LocalAppDataFolder%	Chemin complet vers le répertoire de fichier système qui sert en tant qu'archivage de données pour les applications locales (nonroaming).
%MyPicturesFolder%	Chemin complet vers le dossier MesPhotos.

# 12 Plug-in MapForce pour Visual Studio

Vous pouvez intégrer MapForce 2023 dans les versions Microsoft Visual Studio 2012/2013/2015/2017/2019/2022. Cette intégration aide à combiner les capacités de mappage de MapForce avec l'environnement de développement de Visual Studio. Lorsque le plug-in de MapForce est activé, vous pouvez créer des mappages et des projets de mappage directement depuis Visual Studio (*voir la capture d'écran ci-dessous*).



### Installation

Pour installer le Plug-in MapForce pour Visual Studio, suivez les étapes suivantes :

- 1. Installer Microsoft Visual Studio 2012/2013/2015/2017/2019/2022. Veuillez noter qu'à partir de Visual Studio 2022, Visual Studio est mis à disposition uniquement comme application 64-bit.
- 2. Installez MapForce(Enterprise ou Professional Edition). Si vous avez installé Visual Studio 2022+, alors vous devez installer la version 64-bit de MapForce.
- Téléchargez et exécutez le pack d'intégration MapForce pour Microsoft Visual Studio. Ce pack est disponible sur la page de téléchargement de MapForce (Enterprise et Professional Editions) sur www.altova.com

Une fois que le pack d'intégration a été installé, vous pourrez utiliser MapForce dans l'environnement de Visual Studio.

#### Important

Vous devez utiliser le pack d'intégration correspondant à votre version de MapForce (la version actuelle est 2023). La package d'intégration n'est pas spécifique à l'édition et peut donc être utilisé pour les deux éditions Enterprise et Professional.

#### Information sur les menus et les fonctions

Lorsque le plug-in MapForce pour Visual Studio est activé, vous pouvez accéder aux menus et fonctions communs comme indiqué (*voir ci-dessous*). Vous pouvez personnaliser les menus et les barres d'outils MapForce depuis le menu **Tools | Customise** de Visual Studio.

- **Note :** dans Visual Studio 2019 et plus élevé, la fonctionnalité de MapForce peut être accédée dans le menu **Extensions** de Visual Studio. Dans des versions antérieures de Visual Studio, les fonctions de MapForce sont disponibles dans des menus de niveau supérieur de Visual Studio.
- Thèmes

Vous pouvez sélectionner les thèmes de MapForce dans le menu **MapForce** de Visual Studio. Les options sont les thèmes Classique, Clair et Sombre.

Créer et ouvrir des fichiers/projets

Si le plug-in de MapForce pour Visual Studio est activé, vous pouvez créer, ouvrir et travailler avec des mappages et des projets de mappage directement depuis Visual Studio. Pour créer un nouveau fichier design de mappage dans Visual Studio, utilisez la commande de menu **Fichier | Nouveau**. Pour créer un nouveau projet, utilisez la commande de menu **Fichier | Nouveau** projet. Pour ouvrir des fichiers de mappage existants ou des proejts, vous pouvez utiliser les menus suivants de Visual Studio : **Fichier | Ouvrir | Fichier** ou **Fichier | Ouvrir | Projet/Solution**. Puis, vous pouvez rechercher les types de fichiers liés à MapForce.

Ressources globales

Les <u>Ressources globales</u><sup>(271)</sup> MapForce sont disponibles dans le menu **MapForce | Gérer Ressources Globales** de Visual Studio. À partir de Visual Studio 2019, le menu correspondant est **Extensions | MapForce | Gérer les Ressources globales**.

Débogage

Après avoir ouvert un fichier de mappage, les commandes de débogage de mappage sont disponibles dans le <u>menu Déboguer</u><sup>(104)</sup> et dans la barre d'outils **Déboguer**. À partir de Visual Studio 2019, le menu correspondant est **Extensions | MapForce | Debug**.

Options MapForce

Les options MapForce sont disponibles dans le menu Tools | Options MapForce de Visual Studio.

Personnalisation du volet de mappage

Lorsqu'un mappage de MapForce est dans le volet principal de Visual Studio, le menu **View | MapForce** devient disponible. Il comprend les mêmes options que le menu **View** de la version autonome de MapForce.

Fenêtre de bibliothèques

Si la fenêtre **Bibliothèques** MapForce n'est pas visible dans Visual Studio, vous pouvez l'activer depuis le menu **View | MapForce | Libraries Window** (ce menu devient disponible dans Visual Studio lorsqu'un fichier de mappage est ouvert). Une fois que la fenêtre **Bibliothèques** est activée, vous pouvez l'ancrer dans une position particulière dans l'interface.

#### Aide et soutien

Les menus Aide, Centre de support, Vérifier les mises à jour et à propos de MapForce sont disponibles dans le menu Aide | Aide MapForce de Visual Studio.

# 13 Plug-in MapForce pour Eclipse

Eclipse est un framework open source qui intègre différents types d'applications fournies sous la forme de plugins. Vous pouvez intégrer les éditions MapForce Enterprise et Professional dans les versions d'Eclipse 2022-09, 2022-06, 2022-03, 2021-12 et accéder aux fonctions MapForce directement depuis Eclipse.



Plug-in MapForce Enterprise Edition pour Eclipse

Les rubriques suivantes vous soutiennent dans l'installation et l'utilisation du plug-in MapForce pour Eclipse.

- Installer le plug-in MapForce pour Eclipse<sup>1005</sup>
- La perspective MapForce<sup>1007</sup>
- <u>Accéder aux menus et fonctions communs</u><sup>1009</sup>
- Travailler avec des mappages et des projets<sup>(013)</sup>
- Étendre le plug-in MapForce pour Eclipse<sup>1023</sup>

# **13.1** Installer le plug-in MapForce pour Eclipse

### Prérequis

- Eclipse 2022-09, 2022-06, 2022-03, 2021-12 (<u>http://www.eclipse.org</u>) exploitation 64-bit.
- Java Runtime Environment (JRE) ou Java Development Kit (JDK) pour la plateforme 64-bit.
- MapForce Enterprise ou Professional Edition 64-bit.
- **Note :** tous les prérequis cités ci-dessus doivent être dotés de la plateforme 64-bit. L'intégration avec d'autres plateformes 32-bit n'est plus prise en charge, bien qu'elle puisse encore fonctionner.

Une fois que les pérequis ci-dessus sont en place, vous pouvez installer le Package d'intégration (64-bit) MapForce pour intégrer MapForce dans Eclipse. L'intégration peut être effectuée au cours de l'installation du Package d'intégration ou manuellement depuis Eclipse une fois que le Package d'intégration a été installé. Le Package d'intégration MapForce est disponible pour être téléchargé à l'adresse <u>https://www.altova.com/components/download</u>.

**Note :** Eclipse doit être fermé pendant que vous installez ou désinstallez le Package d'intégrationMapForce.

## Intégrez MapForce lors de l'installation du Package d'intégration

Vous pouvez intégrez MapForce dans Eclipse lors de l'installation du Package d'intégration MapForce. Pour ce faire, suivez les étapes suivantes :

- 1. Exécutez le Package d'intégration MapForce pour lancer l'assistant d'installation.
- 2. Suivez les étapes initiales de l'installation avec l'assistant d'installation.
- 3. Dans l'étape d'intégration, sélectionnez *Permettre à l'assistant d'intégrer Altova MapForce plug-in dans Eclipse*, et cherchez le répertoire dans lequel le fichier exécutable pour Eclipse (eclipse.exe) se trouve.
- 4. Cliquez sur Suivant et terminez l'installation.

La perspective MapForce et les menus seront disponibles dans Eclipse la prochaine fois que vous le démarrez.

### Intégrer MapForce dans Eclipse manuellement

Une fois que vous avez installé le Package d'intégration MapForce, vous pouvez intégrer manuellement MapForce dans Eclipse comme suit:

- 1. Dans Eclipse, sélectionnez la commande de menu Aide | Installer nouveau logiciel.
- 2. Dans la boîte de dialogue Installer, cliquez sur Ajouter.

🖨 Install —			
Available Software			
Select a site or enter the location of a site.			
Work with: type or select a site <a> <u>A</u>dd</a>	<u>M</u> anage		
type filter text	Select All		
Name	Deselect All		
(i) There is no site selected.			

3. Dans la boîte de dialogue Ajouter Référentiel, cliquez sur Local. Recherchez le dossier C:\Program Files\Altova\Common2023\eclipse\UpdateSite et sélectionnez-le. Donner un nom au site (comme « Altova »).

Add Repository				×
<u>N</u> ame:	Altova		L <u>o</u> cal	
Location:	file:/C:/Program Files/Altova/Common2021/eclipse/Upda		<u>A</u> rchiv	e
ОК				
?	A <u>d</u> d		Cance	I

- Répétez les étapes 2 à 3 ci-dessus, en sélectionnant cette fois-ci le dossier C:\Program Files\Altova\<% APPNAMESHORT%>\eclipse\UpdateSite et en donnant un nom comme « Altova » MapForce".
- 5. Dans la boîte de dialogue Installer, sélectionnez *Uniquement sites locaux*. Puis, sélectionnez le dossier « catégorie Altova » et cliquez sur **Suivant**.
- 6. Revérifiez les éléments à installer puis cliquez sur Suivant pour procéder.
- 7. Pour accepter le contrat de licence, sélectionnez la case à cocher respective.
- 8. Cliquez sur **Terminer** et terminez l'installation.
- **Note :** Si vous avez des problèmes avec le plug-in (icônes manquantes, par exemple), lancez Eclipse depuis la ligne de commande avec l'indicateur -clean.

# 13.2 La perspective MapForce

Dans Eclipse, une perspective est un affichage GUI qui est configuré avec la fonctionnalité d'une application spécifique. Lorsque MapForce est intégré dans Eclipse, une perspective MapForce par défaut est créée automatiquement. Cette perspective est une GUI qui inclut les éléments de GUI de MapForce : ses modes d'édition, ses menus, ses assistants à la saisie, et d'autres barres latérales.

Lorsqu'un fichier dont le type de fichier est associé avec MapForce est ouvert (.xml, par exemple), ce fichier peut être édité dans la perspective MapForce. De même, un fichier d'un autre type de fichier peut être ouvert dans une autre perspective dans Eclipse. En outre, pour tout fichier actif, vous pouvez sauter la perspective, vous permettant donc d'éditer ou de traiter ce fichier dans un autre environnement.

Deux avantages principaux se présentent donc pour les perspectives :

- 1. la possibilité de changer rapidement l'environnement de travail du fichier actif et
- 2. La possibilité de passer entre les fichiers sans devoir ouvrir un nouvel environnement de développement (l'environnement associé est disponible dans une perspective)

Le travail avec la perspective MapForce implique les éléments suivants :

- Passer à la perspective MapForce.
- Définir les préférences pour la perspective MapForce.
- Personnaliser la perspective MapForce.

### Passer à la perspective MapForce

Dans Eclipse, sélectionnez la commande **Window | Perspective | Open Perspective | Other**. Dans le dialogue qui s'ouvre (*capture d'écran ci-dessous*), sélectionnez **MapForce** et cliquez sur **Ouvrir**. La fenêtre vide ou le document actif aura maintenant la perspective MapForce. Voici comment l'utilisateur change de perspective par le menu. Pour accéder à une perspective plus rapidement depuis une autre perspective, la perspective requise peut être recensée dans le sous-menu **Open Perspective** au-dessus de l'item **Other**. Ce paramètre est dans le dialogue de personnalisation (*voir plus bas ci-dessous*).

Les perspectives peuvent aussi être changées lorsqu'un fichier est ouvert ou est rendu actif. La perspective de l'application associée avec un type de fichier d'un fichier sera ouvert automatiquement lorsque ce fichier est ouvert pour la première fois. Avant de changer de perspective, un dialogue apparaît vous demandant si vous souhaitez associer automatiquement la perspective par défaut avec ce type de fichier. Vérifiez l'option *Do Not Ask Again* si vous souhaitez associer la perspective avec un filetype sans devoir y être invité chaque fois qu'un fichier de ce filetype est ouvert, puis cliquez sur **OK**.

### Les préférences pour la perspective MapForce

Les préférences d'une perspective comprennent : (i) un paramètre pour changer automatiquement la perspective lorsqu'un fichier d'un type de fichier associé est ouvert (*voir ci-dessus*), et (ii) des options pour inclure ou exclure des barres d'outils MapForce individuelles et (iii) accéder aux options MapForce. Pour accéder au dialogue Préférences, sélectionnez la commande **Window | Preferences**. Dans la liste des perspectives du panneau de gauche, sélectionnez MapForce, puis sélectionnez les préférences requises. Terminez en cliquant sur **OK**.

### Personnalisez la perspective MapForce

Les options de personnalisation vous permettent de déterminer quels raccourcis et commandes sont inclus dans la perspective. Pour accéder au dialogue Personnaliser la Perspective, transformez la perspective en perspective active et sélectionnez la commande **Window | Perspective | Customize Perspective**.

- Dans les onglets *Toolbar Visibility* et *Menu Visibility*, vous pourrez spécifier quelles barres d'outils et menu doivent être affichés.
- Dans l'onglet *Action Set Availability*, vous pouvez ajouter des groupes de commandes à leurs menu parents et à la barre d'outils. Si vous souhaitez activer un groupe de commande, cochez sa case.
- Dans l'onglet Shortcuts du dialogue Customize Perspective, vous pouvez configurer des raccourcis pour les sous-menus. Sélectionnez le sous-menu dans la liste déroulante des sous-menus. Puis sélectionnez une catégorie de raccourcis, et cochez les raccourcis que vous souhaitez inclure à la perspective.

Cliquer sur Appliquer et Fermer pour terminer la personnalisation et pour que les modifications prennent effet.

# Aperçu des la perspective MapForce

Par défaut, la perspective MapForce dans Eclipse est organisée comme suit :

- La fenêtre de design de mappage est disponible en tant qu'éditeur Eclipse. Il dispose des mêmes onglets et fonctions dans l'édition autonome de MapForce.
- La fenêtre Bibliothèques (Libraries) est disponible en tant que mode Eclipse, à gauche de l'éditeur de mappage principal. Si cet affichage n'est pas visible, passer à la perspective MapForce, puis sélectionner la commande de menu Window | Show View | Libraries. Le mode Bibliothèques vous permet de travailler avec des fonctions prédéfinies ou définies par personnalisation et des bibliothèques de fonction.
- Le volet Messages est disponible en tant que mode Eclipse, sous l'éditeur de mappage principal. Si le mode Message n'est pas visible, passer à la perspective MapForce, puis choisir la commande de menu **Window | Show View | Messages**. Le mode Messages affiche les messages de validation, erreurs et avertissements.
- Le volet Aperçu est disponible en tant que mode Eclipse. Si le mode Aperçu n'est pas visible, passer à la perspective MapForce, puis sélectionner la commande de menu Window | Show View |
   Overview. Ce mode vous permet de naviguer rapidement dans une région particulière lorsque la zone de design de mappage est très grande.
## 13.3 Accéder aux menus et fonctions communs

Dans Eclipse, vous pouvez accéder la plupart des fonctions MapForce depuis les même menus que dans la version autonome, sauf pour certaines variations spécifique à Eclipse qui sont listées ci-dessous. Il s'agit de la configuration par défaut ; néanmoins, vous pouvez personnaliser les préférences d'interface depuis Eclipse, si vous le souhaitez (voir <u>La Perspective MapForce</u><sup>100</sup>).

**Note :** Dans Eclipse, certains groupes ou commandes de menu MapForce sont désactivées (ou non disponibles) si le contexte n'est pas pertinent. Par exemple, le menu **Insertion** devient disponible uniquement lorsqu'un fichier de design de mappage (.mfd) est activé dans Eclipse.

Pour plus d'informations concernant les menus de standard MapForce, voir <u>Référence menu</u><sup>(024)</sup>.

### Commandes MapForce générales

Dans l'édition autonome de MapForce, les commandes applicables aux fichiers de design de mappage (comme Validate, Deploy to FlowForce Server, Generate Code, et autres) sont disponibles dans le menu File. Dans Eclipse, ces commandes sont disponibles dans le menu MapForce, ou dans la barre d'outils MapForce. Veuillez noter que les commandes pour ouvrir ou enregistrer les fichiers (y compris les fichiers de projet MapForce) sont disponibles dans le menu File d'Eclipse.

Les thèmes de MapForce peuvent être sélectionnés dans le menu MapForce.



La barre d'outils MapForce dans Eclipse

La touche de barre d'outils 🖾 ouvre le fichier d'aide MapForce.

La touche de barre d'outils affiche des commandes spécifiques aux fichiers MapForce. Lorsque vous agrandissez cette touche, les commandes disponibles dépendent du type de fichier actif actuellement dans l'éditeur Eclipse. Par exemple, les commandes spécifiques au fichier de design de mappage (.mfd) sont disponibles lorsqu'un fichier de ce type est actif (dans focus) dans l'éditeur Eclipse.

#### Ressources globales

Pour accéder ou gérer les Ressources globales, suivre une des étapes suivantes :

- Cliquez pour agrandir la touche de barre d'outils MapForce, puis cliquez sur Ressources globales.
- Dans le menu MapForce, cliquez sur Ressources globales.

### Projets MapForce

Dans l'édition standard de MapForce, le menu **Projet** contient plusieurs commandes applicables aux fichiers de projet de mappage (.mfp). Dans Eclipse, ces commandes existent comme suit :

• Les commandes pour ouvrir ou enregistrer un projet sont disponibles depuis le menu Eclipse File.

• D'autres commandes spécifiques au projet sont disponibles en tant que commandes contextuelles. Pour afficher les commandes contextuelles, créez ou ouvrez un fichier de projet MapForce (.mfp) dans Eclipse, puis cliquez avec la touche de droite sur le projet.



Veuillez noter qu'outre les projets MapForce standard (.mfp), dans Eclipse vous pouvez aussi créer des projets de type "MapForce/Eclipse". Ce type de projet est de nature double et peut être configuré pour la création et la génération automatique du code MapForce. Voir <u>Travailler avec des mappages et des projets</u><sup>(013)</sup>.

### **Options MapForce**

Des options MapForce sont disponibles depuis le menu **Window | Preferences**. Dans le dialogue Préférences, choisissez **MapForce**, puis cliquer sur **Ouvrir le dialogue d'options MapForce**.

Preferences		- • •
<ul> <li>Preferences</li> <li>type filter text</li> <li>General</li> <li>Ant</li> <li>Code Recommenders</li> <li>Help</li> <li>Install/Update</li> <li>Java <ul> <li>MapForce</li> <li>Maven</li> <li>Mylyn</li> <li>Run/Debug</li> <li>Team <ul> <li>Validation</li> <li>WindowBuilder</li> <li>XML</li> </ul> </li> </ul></li></ul>	MapForce MapForce Perspective Automatically switch to MapForce perspective at fill Preferences shared with MapForce installation Open MapForce Options Dialog	
	Restore <u>D</u> efau	ults Apply
?	ОК	Cancel

Dialogue Préférences

### Fenêtre de bibliothèques

Dans Eclipse, la fenêtre MapForce Bibliothèques est disponible en tant qu'affichage. Cet affichage est situé par défaut à gauche de la fenêtre d'éditeur principal. (Tous les affichages liés à MapForce deviennent visibles dans l'interface Eclipse lorsque la perspective MapForce est activée, voir aussi La perspective MapForce<sup>1007</sup>).

### Version de plug-in MapForce

Pour consulter la version installée actuellement du plug-in MapForce pour Eclipse, sélectionner l'option de menu Eclipse **Aide | à propos d'Eclipse**. Puis, choisissez l'icône MapForce.

### Aide et soutien

Les menus MapForce Help, Support Center, Check for Updates et About sont disponibles dans le menu Aide | Aide MapForce d'Eclipse.

# **13.4** Travailler avec des mappages et des projets

Lorsque le plug-in MapForce pour Eclipse est installé, vous pouvez créer depuis Eclipse les mêmes mappages et les types de projet de mappage comme dans l'édition autonome de MapForce, depuis l'intérieur d'un projet Eclipse. Pour concevoir, tester, compiler et déployer des codes de mappages, vous pouvez soit créer un nouveau projet d'Eclipse ou utiliser un projet Eclipse existant (par exemple, un projet Java auquel vous pourrez souhaiter ajouter des mappages MapForce).

De plus, vous pouvez travailler avec tous vos mappages dans le cadre d'un type de projet spécifique qui devient disponible dans Eclipse après avoir installé le plug-in MapForce — le **projet MapForce/Eclipse**. À moins de choisir de la personnaliser, un Java Builder et un MapForce Code Generation Builder sont attribués par défaut au projet MapForce/Eclipse. De plus, il détient deux natures Eclipse : nature MapForce et la nature JDT (outils de développement Java). En résultat, un projet MapForce/Eclipse se comporte comme suit lorsque vous enregistrez ou modifiez une de ses ressources (comme le fichier de design de mappage) :

- Si l'option de menu Project > Build automatically est activée, le code de mappage est généré automatiquement. Lorsqu'un ou plusieurs fichiers de projet MapForce existent dans le projet MapForce/Eclipse, le langage de génération de code et les dossiers de cible de sortie sont déterminés par les paramètres dans chaque fichier de projet. Sinon, Eclipse vous demande de choisir un emplacement.
- Tous les messages d'erreur et de sortie sont affichés dans les affichages Messages et Problèmes.

Cette section contient les rubriques suivantes :

- Créer un projet MapForce/Eclipse<sup>1013</sup>
- Créer de nouveaux mappages
   <sup>1015</sup>
- Importer des mappages existants dans un projet Eclipse<sup>1017</sup>
- <u>Configurer un Build et une Génération automatique de code MapForce</u><sup>1020</sup>

### 13.4.1 Créer un projet MapForce/Eclipse

Pour créer un projet MapForce/Eclipse, suivez les étapes ci-dessous :

- 1. Dans le menu File, cliquer New | Other.
- 2. Choisir la catégorie MapForce/Eclipse Project.

New	- • •
Select a wizard	
<u>W</u> izards:	
type filter text	
🕟 🧽 General	
CVS	
👂 🗁 Git	
> 🦻 Java	
Description Provide Automatica Provide Automatic	
▲ ➢ MapForce/Eclipse Project	
MapForce/Eclipse Project	
> 🦢 Maven	
Figure 1 asks	
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

3. Cliquer sur Next.

New MapForce Project	- • ×
Specify Name and Code Language of Project	
Create new MapForce/Eclipse Project	
Project name: MapForceEclipseProject1	
Use <u>d</u> efault location	
Location: C:\eclipse44\workspace\MapForceEclipseProject1	B <u>r</u> owse
Builder for MapForce mappings and projects	
add MapForce builder to project	
Additional Builders for Code Generated by MapForce	
ouse no additional builders	
Ise JDT builder	
ouse following nature:	
org.eclipse.jdt.core.javanature	
? < <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

- Saisir un nom de projet et choisir un emplacement où sauvegarder le projet. Laisser les options ajouter un générateur MapForce au projet et utiliser générateur JDT comme elles sont.
- 5. Cliquer sur Finish.

### 13.4.2 Créer de nouveaux mappages

Vous pouvez créer les types de fichier MapForce dans le cadre d'un projet Eclipse :

- mappages MapForce
- fichiers de projet MapForce
- projets de Service Web MapForce (disponible dans l'Édition MapForce Enterprise)

#### Pour créer un de ces types de fichier dans le cadre d'un projet Eclipse :

- 1. Créer un nouveau projet Eclipse ou ouvrir un projet existant.
- 2. Dans le menu File, cliquer sur New, puis cliquer sur Other.

● New	
Select a wizard	
<u>W</u> izards:	
type filter text	
<ul> <li>General</li> <li>CVS</li> <li>Git</li> <li>Java</li> <li>MapForce Files</li> <li>MapForce Mapping</li> <li>MapForce Project File</li> <li>MapForce Web Service Project</li> <li>MapForce/Eclipse Project</li> <li>Maven</li> <li>Tasks</li> <li>WindowBuilder</li> </ul>	
? < <u>Back</u> <u>Next &gt;</u> <u>Finish</u>	Cancel

3. Choisir le type de fichier requis depuis le dialogue d'assistant, puis cliquer sur Next.

MapForceMessages.NewProjectFilePageText	
Specify Location and Name of the New MapForce Project File	
Create a New MapForce Project File	
Enter or select the parent folder:	
JavaProject1	
JavaProject1	
File name: MappingProject1.mfp	
<u>A</u> dvanced >>	
(?) < <u>Back</u> <u>N</u> ext > <u>Finish</u>	Cancel

4. Choisir un dossier parent dans votre projet existant, et puis cliquer sur Finish.

## 13.4.3 Importer des mappages existants dans un projet Eclipse

Pour importer des mappages MapForce et leurs fichiers dépendants dans un projet Eclipse existants :

- 1. Ouvrir le projet dans lequel vous souhaitez importer les fichiers.
- 2. Dans le menu File, cliquer sur Import.

€ Import	- • •
Select Import resources from the local file system into an existing project.	Ľ
Select an import source:	
type filter text	
<ul> <li>General</li> <li>Archive File</li> <li>Existing Projects into Workspace</li> <li>File System</li> <li>Preferences</li> <li>CVS</li> <li>Git</li> <li>Git</li> <li>Install</li> <li>Maven</li> <li>Run/Debug</li> <li>Tasks</li> <li>Team</li> <li>XML</li> </ul>	
(?) < <u>B</u> ack <u>Next</u> > <u>F</u> inish	Cancel

3. Sélectionner File System, puis cliquer sur Next.

⊖ Import	
File system Please specify folder	
From directory: C:\Users\User\Documents\AltovaMathbb{Altova\Mathbb{Mathbb{Mathbb{Altova\Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbb{Mathbbb{Mathbb{Mathbb{Mat	pForce2015\MapForceExamples  Browse Browse Browse Browse Browse
? < <u>B</u> ack	Next > Einish Cancel

- 4. À côté de **From directory**, chercher l'emplacement des fichiers que vous souhaitez importer, puis sélectionner les fichiers requis.
- 5. À côté de **Into folder**, cliquer sur **Browse**, puis sélectionner le projet dans lequel vous ajoutez les fichiers (dans cet exemple, *MapForceEclipseProject1*).



6. Cliquer sur OK, puis cliquer sur Finish.

### 13.4.4 Configurer un Build et une Génération automatique de code MapForce

La création et la génération de code MapForce automatique est activée par défaut dans tout projet MapForce/Eclipse (voir <u>Créer un projet MapForce/Eclipse</u><sup>1018</sup>). Si vous souhaitez activer la création et la génération automatique du code MapForce dans un projet existant qui n'est pas de type *MapForce/Eclipse*, vous pouvez y parvenir en ajoutant manuellement le builder *MapForce Code Generation* et la nature *MapForce*.

#### Pour ajouter le builder MapForce Code Generation à un projet :

• Ajouter au fichier .project Eclipse les lignes marquées ci-dessous :

```
<buildSpec>
<buildCommand>
```

<name>org.ec</name>	lipse.jdt.core.javabuilder
<arguments></arguments>	
<buildcommand></buildcommand>	
<name>com.alt</name>	cova.mapforceeclipseplugin.MapForceBuilder
<arguments></arguments>	

#### Pour ajouter la nature MapForce à un projet :

• Ajouter au fichier .project Eclipse les lignes marquées ci-dessous :

```
<natures>
    <nature>org.eclipse.jdt.core.javanature</nature>
        <nature>com.altova.mapforceeclipseplugin.MapForceNature</nature>
        </natures>
```

Astuce : Vous pouvez rapidement ouvrir le fichier .project depuis l'affichage Navigator d'Eclipse (Pour activer cet affichage, sélectionner la commande de menu Window | Show View | Navigator).

#### Pour passer automatiquement la génération de code MapForce allumé/éteint :

• Dans le menu **Project**, cliquer sur **Build automatically**.

#### Pour désactiver le MapForce Code Generation builder:

- 1. Dans le menu **Project**, cliquer sur **Properties**.
- 2. Cliquer sur **Builders**.

Properties for JavaProject1		- • •
type filter text	Builders	<b>◇ ・</b> ⇒ <b>・</b>
<ul> <li>Resource</li> <li>Builders         <ul> <li>Java Build Path</li> <li>Java Code Style</li> <li>Java Compiler</li> <li>Java Editor             <ul> <li>Java Compiler</li> <li>Java Editor</li> <li>Javadoc Location</li> <li>Project References</li> <li>Refactoring History</li> <li>Run/Debug Settings</li> <li>Task Repository</li> <li>Task Tags</li> <li>Validation</li> <li>WikiText</li> </ul> </li> </ul> </li> </ul>	Configure the builders for the project:	New         Import         Edit         Remove         Up         Down
?	ОК	Cancel

3. Cliquer pour décocher la case à cocher MapForce Code Generation.

# 13.5 Étendre le plug-in MapForce pour Eclipse

Le plug-in MapForce pour Eclipse fournit un point d'extension Eclipse avec l'ID **com.altova.mapforceeclipseplugin.MapForceAPI**. Vous pouvez utiliser cette extension pour adapter ou étendre les fonctions du plug-in MapForce. Le point d'extension vous donne accès à l'interface COM de la commande MapForce et de l'<u>API MapForce</u><sup>(122)</sup>.

Le pack d'installation Eclipse MapForce contient un exemple simple d'un plug-in qui utilise ce point extension. Il vérifie dans les fichiers ouverts la présence d'événements de nouveaux mappages MapForce et définit le niveau de zoom de l'affichage de mappage à 70%.

La documentation JavaDoc du point d'extension est disponible dans le répertoire d'installation du plug-in MapForce (généralement, C:\Program Files\Altova\MapForce2023\eclipse\docs\).

Avant d'installer et d'exécuter le plug-in MapForce échantillon, veuillez vous assurer que les exigences suivantes ont été remplies :

- Vous utilisez un pack d'intégration Java 64-bit, Eclipse 64-bit, MapForce 64-bit et MapForce 64-bit.
- Le plug-in JDT (Java Development Tools) est installé.
- Le PDE Eclipse (plug-in development environment) est installé.

#### Pour importer le projet de plug-in MapForce échantillon dans votre espace de travail :

- 1. Lancer Eclipse.
- 2. Dans le menu File, cliquer sur Import.
- 3. Choisir General | Existing projects into Workspace, et cliquer sur Next.
- 4. Cliquer sur la touche **Browse**... située à côté du champ "'Select root directory" et choisir le répertoire de projet d'échantillon, par ex. **C:\Program**

```
Files\Altova\MapForce2023\eclipse\workspace\MapForceExtension.
```

5. Choisir l'option **Copy projects into workspace**, et cliquer sur **Finish**. Un nouveau projet nommé "MapForceExtension" a été créé dans votre espace de travail.

#### Pour exécuter le plug-in d'extension échantillon :

- 1. Passer à la perspective Java.
- 2. Dans le menu Run, cliquer sur Run Configurations.
- Cliquer avec la touche de droite Eclipse Application et choisir New. (Si vous ne pouvez pas voir "Eclipse application" dans la liste, les Eclipse Plug-In Development Tools ne sont pas installés dans votre environnement Eclipse. Pour installer les Eclipse Plug-in Development Tools, cliquer sur Install New Software dans le menu Help et installer "Eclipse Plugin Development Tools" depuis le site de téléchargement "The Eclipse Project Updates".)
- 4. Saisir un nom pour votre nouvelle configuration (dans cet exemple, *SampleMapForcePlugin*), puis cliquer sur **Apply**.
- 5. Vérifier que le plug-in d'espace de travail MapForceClient est sélectionné dans l'onglet 'Plug-ins'.
- 6. Cliquer sur Run. Un nouveau Eclipse Workbench s'ouvre.
- 7. Ouvrir un mappage MapForce dans le nouveau Workbench. Il s'ouvrira avec un niveau de zoom par défaut de 70%.

#### Commandes de menu 14

Cette section de référence contient une description des commandes de menu MapForce. Les commandes de menu suivantes sont disponibles :

- Fichier<sup>1025</sup> Éditer<sup>1028</sup> Insérer<sup>1029</sup> •
- •
- •
- Projet<sup>1032</sup> •
- Composant<sup>1034</sup> •
- Composant Composant Fonction Sortie Composant Comp ٠
- ٠
- •
- Déboguer<sup>1040</sup> •
- Mode<sup>1041</sup> •
- Outils 1043
- •
- Fenêtre <sup>1059</sup> Aide <sup>1060</sup> •
- •

# 14.1 Fichier

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Fichier.

#### Nouveau

Crée un nouveau document de mappage. Dans les éditions Professional et Enterprise vous pouvez aussi créer un projet de mappage (.mfp). Pour plus de détails, voir <u>Projets</u><sup>115</sup>..

Ouvrir

Ouvre le design de mappage précédemment ouvert (.mfd). Dans les éditions Professional et Enterprise, vous pouvez aussi ouvrir un projet de mappage (.mfp). Pour plus de détails, voir <u>Projets</u><sup>115</sup>...

Enregistrer/Enregistrer sous/Enregistrer tout

L'option **Enregistrer** enregistre le mappage actuel actif sous son nom actuel. L'option **Enregistrer sous** vous permet d'enregistrer le mappage actuellement ouvert avec un différent nom. La commande **Enregistrer tout** enregistre tous les fichiers de mappage ouverts.

Recharger

Recharger le mappage actuel actif rétablit vos derniers changements.

■ Fermer/Fermer tout

La commande **Fermer** ferme le mappage actuel actif. La commande **Fermer tout** ferme tous les mappages actuels ouverts. Vous devez décider si vous souhaitez enregistrer les fichiers qui n'ont pas encore été enregistrés.

□ Imprimer/Aperçu d'impression/Paramètres d'impression

La commande **Imprimer** ouvre la boîte de dialogue **Imprimer** (*voir ci-dessous*) qui vous permet d'imprimer vos mappages. **Utiliser actuel** conserve le facteur de zoom actuellement défini pour le mappage. **Utiliser optimal** redimensionne le mappage pour qu'il s'adapte à la taille de la page. Vous pouvez aussi préciser le facteur de zoom numériquement. Les barres de déroulement du composant ne sont pas imprimées. Vous pouvez aussi préciser si vous voulez répartir les graphiques sur plusieurs pages ou non.



La commande **Aperçu** ouvre la même boîte de dialogue **Imprimer** avec les mêmes paramètres, tel que décrit ci-dessus. La commande **Paramètres d'impression** ouvre la boîte de dialogue **Paramètres d'impression** dans laquelle vous pouvez sélectionner une imprimante et configurer les paramètres papier.

■ Valider mappage

La commande **Valider mappage** vérifie si tous les mappages sont valides et affiche les messages d'information pertinents, les avertissements et les erreurs. Pour plus de détails, voir <u>Validation</u><sup>103</sup>.

Paramètres de mappage

Ouvre la <u>boîte de dialogue des paramètres de mappage</u><sup>(112)</sup> où vous pouvez définir les paramètres spécifiques au document.

□ Ouvrir gestionnaire d'identifiant (Enterprise Edition)

Ouvre le **Gestionnaire d'identifiant** qui vous permet de gérer les identifiants requis dans les mappages qui réalisent une authentification HTTP de base ou une autorisation OAuth 2.0.

Générer du code dans le langage sélectionné/Générer du code dans

Génère le code dans le langage sélectionné actuellement. Pour plus d'information sur les langages de transformation disponible, voir Langages de transformation<sup>25</sup>.

#### Générer le code dans | XSLT (XSLT2, XSLT3)

Cette commande génère le(s) fichier(s) depuis le(s) fichier(s) XSLT source. En sélectionnant cette option, le dialogue **Browse for Folder** s'ouvre dans lequel vous devez sélectionner l'emplacement du fichier XSLT. Le nom du/des fichier(s) XSLT est défini dans la <u>boîte de dialogue des Paramètres de mappage</u><sup>(112)</sup>

#### Générer le code dans | XQuery | Java | C# | C++ (éditions Professional et Enterprise)

Cette commande génère le code de source dans un des langages de transformation sélectionnés. En sélectionnant cette option, le dialogue **Browse for Folder** s'ouvre dans lequel vous devez sélectionner l'emplacement des fichiers générés. Les noms des fichiers d'application générés (ainsi que le fichier de solution \*.csproj fichier de projet C#, \*.sln et \*.vcproj visual C++ fichier de projet) sont définis dans la boîte de dialogue <u>Mapping Settings</u><sup>(112)</sup>.

Le nom de fichier créé par le code exécuté apparaît dans la fenêtre **Fichier XML de sortie** du dialogue <u>Paramètres de composant</u> is la cible est un document XML/Schéma.

Generation Server Compiler sur fichier d'exécution MapForce Server (éditions Professional et Enterprise)

Génère un fichier qui peut être exécuté par MapForce Server pour exécuter la transformation de mappage. Pour les détails, voir <u>Compiler un mappage de MapForce</u><sup>957</sup>.

Déployer sur FlowForce Server (éditions Professional et Enterprise)

Déployez le mappage actuellement actif sur FlowForce Server. Pour les détails, voir <u>Déployer un mappage</u> <u>de MapForce</u><sup>660</sup>.

Générer documentation (éditions Professional et Enterprise)

Génère la documentation de vos projets de mappage de manière très détaillée dans plusieurs formats de sortie. Pour plus d'informations, voir <u>Générer de la documentation de mappage <sup>916</sup></u>.

Fichiers récents

Affiche la liste des fichiers ouverts le plus récemment.

Quitter

Quitte l'application. Vous devrez décider si vous souhaitez enregistrer les fichiers qui ne le sont pas encore.

# 14.2 Édition

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Édition. La plupart des commandes dans ce menu deviennent actives lorsque vous regardez le résultat d'un mappage dans le volet **Sortie** ou dans le code aperçu, par exemple dans le volet **XSLT**.

Annuler

MapForce dispose d'un nombre illimité d'étapes "Annuler" que vous pouvez utiliser pour remonter vos

étapes de mappage. Vous pouvez aussi utiliser la commande de barre d'outils bour annuler les actions.

Rétablir

La commande rétablir vous permet de rétablir des commandes annulées précédemment. Vous pouvez retourner en arrière et en avant dans l'historique de rétablissement en utilisant ces deux commandes.

Vous pouvez aussi utiliser la commande de barre d'outils pour rétablir les actions.

Recherche

Vous permet de rechercher du texte spécifique dans tous les volets **XQuery** (*éditions Professional et Enterprise*), **XSLT**, **XSLT2**, **XSLT3**, et **Sortie**. Vous pouvez également effectuer la recherche en utilisant la commande de la barre d'outils

Trouver suivant

Cherche l'occurrence suivante du même string de recherche. Vous pouvez aussi rechercher la prochaine occurrence en utilisant is bouton de la barre d'outils.

Trouver précédent

Cherche l'occurrence précédente du même string de recherche. Rechercher la prochaine occurrence est également possible avec la commande de barre d'outils

□ Couper/ Copier/ Coller/ Supprimer

Les commandes Édition des fenêtres standard, qui vous permettent de couper, coller, et supprimer tout composant ou fonctions visibles dans la fenêtre de mappage.

Tout sélectionner

Sélectionne tous les composants dans le volet **Mappage** ou le texte/code dans les volets **XQuery** (*éditions Professional et Enterprise*), **XSLT**, **XSLT2**, **XSLT3**, et **Sortie**.

## 14.3 Insérer

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Insérer.

—	Schéma	XML	/Fichier
	oonomu		

Ajoute un fichier de schéma XML ou le fichier d'instance au mappage. Si vous sélectionnez un fichier XML sans référence de schéma, MapForce peut <u>générer un schéma XML correspondant</u><sup>121</sup>. Si vous choisissez un fichier de schéma XML, vous serez invité à inclure en un fichier d'instance XML qui fournit les données pour l'aperçu. Vous pouvez également ajouter un fichier XML/XSD via la commande de la

barre d'outils

Base de données (éditions Professional et Enterprise)

Ajoute un composant de base de données. voir <u>Bases de données</u><sup>156</sup>. Vous pouvez également ajouter un

composant de base de données via la commande de la barre d'outils Enterprise, vous pouvez aussi ajouter des bases de données NoSQL comme composants.

EDI (Enterprise Edition)

Ajoute un document EDI. Vous pouvez également ajouter un composant EDI via la commande de la barre d'outils .

Fichiers de texte (éditions Professional et Enterprise)

Ajoute un document de fichier plat comme un fichier de texte CSV ou de longueur fixe. Pour plus d'information, voir <u>Fichiers CSV et Texte</u><sup>449</sup>. Vous pouvez également ajouter un fichier texte via la

commande de la barre d'outils Le MapForce Enterprise Edition vous permet également de traiter les fichiers de texte avec FlexText.

■ Fonction de Services web (*Enterprise Edition*)

Ajouter un appel à un service Web générique. Vous pouvez également ajouter un service Web via le bouton de la barre d'outils .

■ Fichier Excel 2007+ (Enterprise Edition)

Ajoute un fichier Microsoft Excel 2007+ (.xlsx). Si Excel 2007+ n#a pas été installé sur votre appareil, vous pouvez toujours mapper de ou vers des fichiers Excel 2007+. Dans ce cas, vous ne pouvez pas voir le résultat dans le volet **Sortie**, mais vous pouvez toujours enregistrer le résultat. Vous pouvez également

ajouter un fichier Excel via la commande de la barre d'outils

■ Document XBRL (Enterprise Edition)

Ajoute une instance XBRL ou un document de taxonomie. Vous pouvez également ajouter un composant XBRL via la commande de la barre d'outils

Schéma JSON/Fichier (Enterprise Edition)

Ajoute au schéma JSON ou un fichier. Vous pouvez également ajouter a composant JSON via la commande de la barre d'outils

■ Fichier Protocol Buffers (*Enterprise Edition*)

Ajoute un fichier binaire encodé dans un format Protocol Buffers. Vous pouvez également ajouter a fichiers

binaires dans le format Protocol via la commande de la barre d'outils

Insérer entrée

Des composants Simple-Input peuvent être utilisés comme paramètres d'entrée qui sont pertinents à tout le mappage ou uniquement dans le contexte des fonctions définies par l'utilisateur. Pour plus d'informations, voir <u>Simple Input</u><sup>477</sup> et <u>Paramètres dans des UDF</u><sup>631</sup>. Vous pouvez également Insérer un simple composant d'entrée utilisant la commande de la barre d'outils **I**.

Insérer sortie

Les composants de sortie simples peuvent être utilisés comme composants de sortie dans les mappages et comme paramètres des fonctions définies par l'utilisateur. Pour plus d'informations, voir <u>Simple</u> <u>Output</u><sup>483</sup> et <u>Paramètres dans des UDE</u><sup>581</sup>. Vous pouvez également insérer un simple composant de sortie utilisant le commande de la barre d'outils

Constante

Insère une constante qui fournit les données fixes à un connecteur d'entrée. Vous pouvez sélectionner les types de données suivants : string, nombre et tous les autres. Vous pouvez également insérer une

constante en utilisant la commande de la barre d'outils

Variable

Insère <u>une variable</u><sup>(422)</sup>, qui est l'équivalent d'une fonction définie par l'utilisateur régulière (non-inline). Une variable est un type de composant spécial utilisé pour stocker un résultat de mappage intermédiaire pour un traitement ultérieur. Vous pouvez également ajouter une variable en utilisant la commande de la barre d'outils

■ Join (éditions Professional et Enterprise)

Le composant Join vous permet de joindre des données en modes SQL et non-SQL. Vous pouvez

également ajouter un composant Join utilisant la commande de la barre d'outils . Pour des détails, voir <u>Joindre des données</u>.

Trier: Nœuds/Lignes

Insère un composant qui vous permet de trier des nœuds (voir <u>Trier Nœuds/Lignes</u><sup>617</sup>). Vous pouvez également ajouter un composant Tri utilisant la commande de la barre d'outils.

■ Filtrer: Nœuds/Lignes

Insère un composant Filtre qui peut filtrer des données depuis toute autre structure de composant prise en charge par MapForce, y compris des bases de données. Pour plus d'informations, voir <u>Filtres et</u>

Conditions 23. Vous pouvez également ajouter une filtre en utilisant la commande de la barre d'outils

SQL/NoSQL-WHERE/ORDER (éditions Professional et Enterprise)

Insère un composant qui vous permet de filtrer des données de base de données de manière conditionnelle. Pour plus d'information, voir <u>(voir Composant SQL WHERE / ORDER)</u><sup>355</sup>. Vous pouvez également accéder un composant SQL/NoSQL-WHERE/ORDER via la commande de la barre d'outils



#### Value-Map

Insère un composant qui transforme une valeur d'entrée en une valeur de sortie en utilisant une table de consultation. Cela est utile lorsque vous devez mapper un ensemble de valeurs dans un autre ensemble de valeurs (par ex., des numéros de mois dans des noms de mois). Pour plus d'informations, voir <u>Value-Maps</u>. Vous pouvez également insérer une Value-Map en utilisant la commande de la barre d'outils

Condition IF-Else

Insère une Condition If-Else qui se prête pour des scénarios dans lesquels vous devez traiter une vale	eur
simple par condition. Pour plus d'informations, voir <u>Filtres et Conditions</u> <sup>523</sup> . Vous pouvez également	
ajouter une condition if-Eise utilisant la commande de la barre d'outils 1	

Exception (éditions Professional et Enterprise)

Le composant d'exception vous permet d'interrompre une procédure de mappage lorsqu'une condition spécifique est remplie. Vous pouvez également ajouter un composant d'Exception utilisant la commande

de la barre d'outils . Dans MapForce Enterprise Edition, ce composant vous permet de définir les messages Fault dans les projets de mappage WSDL. Pour plus d'information sur les composants d'Exception, voir <u>Exceptions</u> .

# 14.4 Projet

MapForce vous permet de grouper vos mappages dans les <u>projets de mappage</u><sup>(115)</sup>. Cette rubrique recense toutes les commandes de menu disponibles dans le menu **Projet**.

Recharger projet

Recharge le projet actuellement actif et passe à la fenêtre Projet.

- Fermer projet
   Ferme le projet actuellement actif.
- Enregistrer projet
   Enregistre le projet actuellement actif.
- Ajouter des fichiers au projet

Vous permet d'ajouter des mappages au projet actuel.

- Ajouter fichier actif au projet
   Ajoute le fichier actuellement actif au projet actuellement ouvert.
- Créer dossier
   Cette option ajoute un nouveau dossier au projet actuel. Voir <u>Dossiers de projet</u><sup>(118)</sup>.
- Ouvrir le mappage (*Enterprise Edition*)

Ouvre le mappage actuellement marqué/sélectionné dans l'onglet Projet

Créer le mappage pour l'opération (*Enterprise Edition*)

Crée un fichier de mappage pour l'opération actuellement sélectionnée du projet WSDL.

- Ajouter un fichier de mappage pour l'opération (*Enterprise Edition*)
   Vous permet d'ajouter un fichier de mappage précédemment enregistré à l'opération WSDL actuellement active.
- Insérer Service Web (*Enterprise Edition*)
   Vous permet d'insérer un service Web sur la base d'un fichier WSDL existant.
- Ouvrir dans XMLSpy

Ouvre le fichier WSDL sélectionné dans <u>Altova XMLSpy</u>.

Générer le Code pour tous le projet

Génère le code de projet pour tout le projet actuellement visible dans la fenêtre de **Projet**. Le code est généré dans le langage sélectionné pour tous les fichiers de mappage .mfa dans chacun des dossiers.

#### Générer code dans

Génère le code de projet dans le langage que vous avez sélectionné depuis le menu contextuel.

#### Propriétés

Ouvre un dialogue où vous pouvez définir les paramètres de projet<sup>118</sup>.

Projets récents

Affiche une liste des projets ouverts récemment.

## 14.5 Composant

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Composant.

Modifier Élément racine

Vous permet de changer l'élément racine du document d'instance XML.

Éditer une définition de Schéma dans XMLSpy

Afin de pouvoir éditer un schéma dans <u>Altova XMLSpy</u>, vous devez cliquer un composant XML, puis sélectionnez l'option Éditer définition de Schéma dans XMLSpy.

Cette commande vous permet d'éditer un fichier FlexText.

Ajouter/Supprimer/Éditer des objets de base de données (éditions Professional et Enterprise)

Vous permet d'ajouter, de supprimer et de changer des objets de base de données dans le composant de base de données. Voir <u>Bases de données</u><sup>156</sup>.

Créer un mappage pour EDI X12 997 (Enterprise Edition)

Le X12 997 Functional Acknowledgment rapporte le statut de l'interchange EDI. Toutes les erreurs rencontrées pendant le traitement du document y sont rapportées. MapForce peut générer automatiquement un document X12 997 que vous pourrez envoyer au destinataire.

Créer un mappage pour EDI X12 999 (Enterprise Edition)

Le X12 999 Implementation Acknowledgment Transaction Set rapporte la non-conformité du guide d'implémentation HIPAA ou les erreurs d'application. MapForce peut générer automatiquement le composant X12 999 dans le mappage et créer automatiquement les connexions de mappage nécessaires.

Actualiseréditions Professional et Enterprise)

Recharge la structure du composant de base de données actuellement active.

■ Ajouter double entrée Avant/Après

Insère une copie de l'item sélectionné avant/aprés l'item sélectionné. L'entrée dupliquée ne peut pas être utilisée comme source de données. Pour plus d'information, voir <u>Dupliquer l'entrée</u><sup>(81)</sup>.

Supprimer doublon

Supprime un item dupliqué.

■ Instructions de traitement/Commentaire

Cette option vous permet d'insérer des <u>commentaires et instructions de traitement</u><sup>(132)</sup> dans les composants XML.

Écrire du contenu en tant que section CDATA

Cette commande crée une <u>section CDATA</u><sup>(13)</sup> qui est utilisée pour représenter des parties d'un document comme données de caractère qui devraient normalement être interprétés comme balise.

Actions de Table de base de données (éditions Professional et Enterprise)

Vous permet de configurer l'insertion de base de données, mettre à jour et supprimer des actions et d'autres options d'enregistrements de base de données. Voir <u>Paramètres d'actions de table de base de données</u> <sup>(30)</sup> pour plus d'information.

Requête de base de données (éditions Professional et Enterprise)

Crée une instruction SELECT sur la base de la table/champ sur lequel vous avez cliqué dans le composant de la base de données. Cliquez sur une table/un champ pour rendre active cette commande, et l'instruction SELECT est placée automatiquement dans la **fenêtre** Select.

■ Aligner arborescence à gauche

Rend l'arborescence d'un composant left-justified.

■ Aligner arborescence à droite

Rend l'arborescence d'un composant right-justified.

Propriétés

Affiche les paramètres d'un composant actuellement sélectionné. Voir <u>Changer les paramètres de</u> <u>composant</u><sup>(80)</sup>.

## 14.6 Connexion

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Composant.

■ Auto connexion des enfants correspondants

Activer/désactiver l'option **Auto-connexion des enfants correspondants**. Pour plus d'information sur les connexions et leurs types, voir <u>Connexions</u><sup>(87)</sup>.

Paramètres pour connecter des enfants correspondants

Vous aide à définir des connexions d'enfants correspondants. Pour les détails, voir <u>Connexions d'enfants</u> correspondants<sup>®3</sup>.

■ Connecter les enfants correspondants

Cette commande vous permet de créer de multiples connexions pour des items avec les mêmes noms dans les composants source et cible. Les paramètres que vous définissez dans cette boîte de dialogue

s'appliquent si la commande de la barre d'outils 😫 (Auto-connexion des éléments enfants) a été activée. Pour les plus d'informations, voir <u>Connexions d'enfants correspondants</u>

■ Orienté vers la cible (Standard)

Change le type de connecteur sur un mappage standard. Pour plus d'information, voir <u>Connexions Target-</u> <u>driven vs. source-driven</u><sup>91</sup>.

■ Copier-tout (Copier les items enfants)

Crée des connexions popur tous les items enfant correspondants. Le principal avantage des connexions copier-tout est qu'elles simplifient visuellement l'espace de travail du mappage : Une connexion, représentée par une ligne épaisse, est créée à la place de connexions multiples. Pour plus de détails, voir <u>Connexions copier tout</u><sup>(95)</sup>.

■ Orienté vers la source (contenu mixte)

Change un type de connexion à une connexion orientée vers la source qui vous permet de mapper automatiquement le contenu mixte (nœuds texte et enfant) dans le même ordre que dans le fichier XML source. Pour plus d'informations, voir <u>Connexions orientées vers la source</u><sup>[91]</sup>.

Propriétés

Ouvre la boîte de dialogue **Paramètres de connexion** qui vous permet de définir des types de connexion et des paramètres d'annotation. Pour plus d'informations, voir <u>Paramètres de connexion</u><sup>97</sup>.

# 14.7 Fonction

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Fonction.

Gréer une fonction définie par l'utilisateur

Créer une (UDF) <u>fonction définie par l'utilisateur</u><sup>575</sup>. Vous pouvez également créer l'UDF en utilisant la commande de la barre d'outils **d**.

Créer une fonction définie par l'utilisateur depuis la sélection

Crée une fonction définie par l'utilisateur basée sur les éléments sélectionnés actuellement dans la fenêtre de mappage. Pour les détails, voir <u>Créer des UDF</u> **378**. Vous pouvez également créer un UDF de la sélection utilisant la commande de la barre d'outils **179**.

Paramètres de fonction

Ouvre la boîte de dialogue **Éditer la fonctions définie par l'utilisateur** qui vous permet de changer des paramètres de l'UDF. Pour les détails, voir <u>Éditer des UDF</u><sup>576</sup>.

Supprimer la fonction

Supprime la fonction définie par l'utilisateur actuellement active si vous travaillez dans un contexte qui permet ceci.

Insérer entrée

Des composants Simple-Input peuvent être utilisés comme paramètres d'entrée qui sont pertinents à tout le mappage ou uniquement dans le contexte des fonctions définies par l'utilisateur. Pour plus d'informations, voir <u>Simple Input</u> et <u>Paramètres dans des UDF</u> . Vous pouvez également Insérer un simple composant d'entrée utilisant la commande de la barre d'outils **Paramètres**.

Insérer sortie

Les composants de sortie simples peuvent être utilisés comme composants de sortie dans les mappages et comme paramètres des fonctions définies par l'utilisateur. Pour plus d'informations, voir <u>Simple</u> <u>Output</u><sup>(43)</sup> et <u>Paramètres dans des UDF</u><sup>(53)</sup>. Vous pouvez également Insérer un simple composant de sortie utilisant le bouton de la barre d'outils **i**.

## 14.8 Sortie

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Sortie.

SLT 1.0/XSLT 2.0/XSLT 3.0/XQuery/Java/C#/C++/Built-In

Définit le langage de transformation dans lequel le mappage doit être exécuté. La sélection du langage de transformation dépend de votre édition de MapForce. Pour des détails, voir <u>Langages de transformation</u><sup>25</sup>. Vous pouvez aussi sélectionner des langages de transformation dans la barre d'outils.

■ Valider le fichier Sortie

Valide le fichier XML de sortie par rapport à un schéma référencé. Voir Validation<sup>103</sup>.

Enregistrer le fichier de sortie

Enregistre les données dans le volet Sortie dans un fichier.

Enregistrer tous les fichiers de sortie

Enregistre tous les fichiers de sortie générés des mappages dynamiques<sup>883</sup>. Voir le <u>Tutoriel 4</u><sup>66</sup>.

Régénérer Sortie

Régénère les données visibles dans le volet Sortie.

Exécuter SQL/NoSQL-Script (éditions Professional et Enterprise)

Si un script SQL/NoSQL est actuellement visible dans le volet **Sortie**, le script exécute le mappage vers une base de données cible, prenant les actions de table définies en compte. Pour en savoir plus sur toutes les bases de données prises en charge, voir <u>Bases de données</u><sup>(156)</sup>.

Insérer/Supprimer signet

Insère/supprime un signet à la position du curseur dans le volet Sortie.

Signet suivant/précédent

Navigue vers le signet suivant/précédent dans le volet Sortie.

Supprimer tous les signets

Supprime tous les signets définis actuellement dans le volet Sortie.

■ Texte XML Pretty-Print

Reformate votre document XML dans le volet **Sortie** pour que le document ait un affichage structuré : Chaque nœud enfant est décalé de son parent par un seul caractère de tabulation. Dans le volet **Sortie**, les paramètres de taille de l'onglet définis dans le dialogue <u>Paramètres du Mode Texte</u><sup>100</sup> (groupe d'onglets) prennent effet.

Paramètres du Mode Texte

Affiche la boîte de dialogue **Paramètres du Mode Texte** qui vous permet de personnaliser les paramètres du Mode Texte dans le volet **XQuery** (*éditions Professional et Enterprise*), le volet **Sortie** et le volet **XSLT**. Le dialogue affiche également les raccorcis clavieractuellement définis. Pour plus d'informations, voir <u>Fonctions de Mode Texte</u><sup>103</sup>.

## 14.9 Déboguer

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Déboguer.

#### Commencer le débogage

Lance ou continue le débogage jusqu'à la survenue d'un point d'arrêt ou la fin d'un mappage. Vous pouvez également lancer le débogage en utilisant la commande de la barre d'outils . Pour plus d'information sur le mode de débogage, voir <u>Débogueur</u><sup>(920)</sup>.

#### Arrêter le débogage

Stoppe le débogage. Cette commande permet de quitter le mode de débogage et fait repasser MapForce dans le mode standard. Vous pouvez également arrêter le débogage en utilisant la commande de la barre d'outils . Pour plus d'information sur le mode de débogage, voir <u>Débogueur</u><sup>(926)</sup>.

Pas à pas détaillé

Exécute le mappage jusqu'à ce qu'une étape unique soit terminée dans un endroit quelconque du mappage. Dans le débogueur du mappage, un pas est un groupe logique de calculs dépendants qui produisent généralement un seul item de séquence. Dépendant du contexte de mappage, cette commande vous dit environ ceci : *aller à gauche/aller à enfant cible/aller à parent source*. Vous pouvez également accéder à cette commande en utilisant le bouton de la barre d'outils  $\stackrel{\circ}{\longrightarrow}$ . Pour plus d'information sur le mode de débogage, voir <u>Débogueur</u><sup>620</sup>.

Pas à pas sortant

Poursuit l'exécution jusqu'à la fin de l'étape actuelle (ou soit terminée à nouveau pour un autre item de la séquence), ou si une étape non liée s'achève. Cette commande passe par-dessus les calculs qui sont des entrées de l'étape actuelle. Vous pouvez également accéder à cette commande en utilisant le bouton de la barre d'outils  $\mathcal{F}$ . Pour plus d'information sur le mode de débogage, voir <u>Débogueur</u><sup>626</sup>.

Pas à pas principal

Poursuit l'exécution jusqu'à ce que le résultat de l'étape actuelle soit consommée ou qu'une étape soit exécutée qui n'est pas une entrée ou un enfant de la consommation. Cette commande sort du calcul actuel. Dépendant du contexte de mappage, cette commande vous dit environ ceci : *aller à droite/aller à parent cible/aller à enfant source*. Vous pouvez également accéder à cette commande en utilisant le bouton de la barre d'outils  $\Im$ . Pour plus d'information sur le mode de débogage, voir <u>Débogueur</u><sup>620</sup>.

#### Étape minimale

## 14.10 Affichage

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Affichage.

Afficher Annotations

Afficher les annotations dans le composant. Vous pouvez aussi activer cette option en cliquant sur la

touche de la barre d'outils Si l'icône **Afficher Types** est aussi active, les deux ensembles d'information sont affichés sous forme de grille (*voir la capture d'écran ci-dessous*). Vous pouvez aussi utiliser des annotations pour libeller les connexions. Pour les détails, voir <u>Paramètres de connexion</u><sup>(93)</sup>.

= F1060	
type	string
ann.	Revision identifier

Afficher Types

Afficher les types de données dans le composant. Vous pouvez aussi activer cette option en cliquant sur

la touche de la barre d'outils E. Si l'icône **Afficher Annotations** est aussi active, alors deux ensembles d'information sont enregistrements affiché sous forme de grille (*voir Afficher Annotations*).

Afficher la Bibliothèque dans l'en-tête Fonction

Affiche le nom de bibliothèque dans l'en-tête de la fonction. Vous pouvez aussi activer cette option en

cliquant sur la touche de la barre d'outils

Afficher astuces

Lorsque vous placez le curseur au-dessus d'un en-tête de fonction, vous verrez une info-bulle résumant ce que cette fonction fait. Avec l'option **Afficher astuces** activée, vous pouvez aussi voir l'information sur les types de données dans un composant.

Generation Strate Contemporation Strategy Strate

MapForce vous permet de configurer les paramètres XBRL suivants :

- Le langage de libellé des items XBRL et leurs annotations
- Les rôles de libellé préférés pour les noms d'item XBRL
- Le type spécifique de rôles de libellé des annotations pour les items XBRL
- Packages de Taxonomie XBRL personnalisés
- Afficher les connecteurs de composant sélectionnés/ connexions depuis la source vers la cible

Ces options vous permettent de surligner les connexions de manière sélective. Pour savoir comment ces options fonctionnent, voir <u>Connexions</u><sup>(8)</sup>.

Zoom

Ouvre le dialogue **Zoom**. Vous pouvez saisir le facteur de manière numérique et faire glisser le curseur pour changer le facteur zoom de manière interactive.

Précédent/ Suivant

Les commandes **Retour** et **Suivant** vous permettent de basculer entre les mappages précédents ou suivants sur lesquels vous avez travaillé, relatifs au mappage actuellement ouvert.

Barre de statut

Active/désactive la barre de statut visible en-dessous de la fenêtre Messages.

Bibliothèques/Gérer les Bibliothèques

Cliquez sur **Bibliothèques** pour activer/désactiver la fenêtre **Bibliothèques**. Cliquez sur **Gérer les bibliothèques** pour activer/désactiver la fenêtre **Gérer les bibliothèques**.

Messages

Active/désactive <u>la fenêtre des Messages</u><sup>35</sup>. Lorsque le code est généré, la fenêtre **Messages** est automatiquement activée pour montrer le résultat de validation.

🖃 Aperçu

Active/désactive la fenêtre d'aperçu<sup>33</sup>. Faites glisser le rectangle pour naviguer à travers le mappage.

Fenêtre de projet (éditions Professional et Enterprise)

Active/désactive la fenêtre **Projet**. Pour en savoir plus sur les projets, voir <u>Projets</u><sup>115</sup>.

■ Fenêtres Débogage (éditions Professional et Enterprise)

Le mode de débogage vous permet d'analyser le contexte responsable dans lequel une valeur particulière est produite. Cette information est disponible directement sur le mappage et dans les fenêtres **Valeurs**, **Contexte** et **Points d'arrêt**. Pour plus d'information, voir <u>À propos Mode Déboguer</u><sup>(631)</sup>.

# 14.11 Outils

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Outils.

Ressources globales

Ouvre la boîte de dialogue **Gérer les Ressources globales** où vous pouvez ajouter, éditer et supprimer des paramètres applicables à travers les multiples applications d'Altova. Pour plus d'informations, voir <u>Ressources globales Altova</u><sup>971</sup>.

Configuration active

Vous permet de sélectionner une configuration de ressource active actuelle depuis une liste de configurations. Pour créer et configurer différents types de ressources globales, voir <u>Ressources globales</u> <u>Altova</u> <sup>(971)</sup>.

Créer mappage inversé

Crée un mappage contrepassé du mappage actuellement actif, ce qui signifie que le composant source devient un composant cible et le composant cible devient la source. Notez que seuls les connexions directes entre des composants sont retenues dans le mappage contrepassé. Il est probable que le nouveau mappage ne sera pas valide ou approprié pour un aperçu dans le volet **Sortie**. Pour cette raison, le nouveau mappage devrait requérir l'édition manuelle.

Les données suivantes sont retenues :

- Connexions directes entre des composants
- Connexions directes entre des composants dans un mappage enchaîné
- Le type of connexion <sup>(10)</sup> : Standard, Contenu mixte, Copier-tout
- Paramètres de composant de passage
- Composants de base de données (éditions Professional et Enterprise)

Les données suivantes ne sont pas retenues :

- Connexions via les fonctions, filtres, etc.
- Fonctions définies par l'utilisateur
- Composants de service Web (Enterprise Edition)
- Gestionnaire de taxonomie XBRL(Enterprise Edition)

Le gestionnaire de taxonomie XBRL est un outil qui permet d'installer et de gérer des taxonomies XBRL.

Gestionnaire de schéma XML

Le Gestionnaire de schéma XML 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 activées par XBRL d'Altova. Pour plus d'informations, voir le <u>Gestionnaire de schéma</u><sup>(13)</sup>.

Personnaliser

Cette option vous permet de personnaliser l'interface d'utilisateur graphique de MapForce. Ceci inclut de montrer/ cacher les barres d'outils de même que de personnaliser les <u>menus</u><sup>104</sup> et les <u>raccourcis de</u> <u>clavier</u><sup>104</sup>.

Restaurer les barres d'outils et les fenêtres

Restaure les barres d'outils, les fenêtres d'assistants à la saisie, les fenêtres ancrées, etc. à leur affichage par défaut. Vous devez redémarrer MapForce pour que les changements prennent effet.

Options

Ouvre la boîte de dialogue **Options** qui vous permet de changer les paramètres par défaut de MapForce. Pour plus d'informations, voir <u>Options</u> <sup>1048</sup>.

### 14.11.1 Personnaliser les menus

Vous pouvez personnaliser des menus standard de MapForce de même que des menus contextuels (par ex., pour ajouter, modifier ou supprimer les commandes). Vous pouvez rétablir vos changements à l'état par défaut (**Rétablir**). Pour personnaliser les menus, allez à **Outils | Personnaliser** et cliquez sur l'onglet **Menu** (*voir la capture d'écran ci-dessous*).



#### Menu par défaut vs. MapForce Design

La barre *Menu par Défaut* est affichée lorsqu'aucun document n'est ouvert dans la fenêtre principale. La barre de menu *MapForce Design* est la barre de menu qui est affichée lorsqu'un ou plusieurs mappage sont ouverts. Chaque barre de menu peut être personnalisée séparément. Les changements de personnalisation effectués à une barre de menu n'affectent pas l'autre.
Pour personnaliser une barre de menu, la choisir depuis la liste déroulante *Afficher Menus pour*. Ensuite, cliquez sur l'onglet **Commandes** et glissez les commandes depuis le champ de liste *Commandes* dans la barre de menu ou dans un des menus.

#### Supprimer les commandes depuis les menus

Pour supprimer un menu ou une commande entière depuis un menu, procéder comme suit :

- 1. Sélectionnez Default Menu ou MapForce Design depuis la liste déroulante Afficher menus pour.
- 2. Lorsque le dialogue **Personnaliser** est ouvert, choisissez une commande de la barre d'outils que vous souhaitez supprimer ou une commande que vous souhaitez supprimer depuis un des menus.
- 3. Glissez la commande de la barre d'outils ou la commande depuis le menu. En alternative, cliquez avec la touche de droite sur la commande de la barre d'outils ou la commande de menu et sélectionnez **Supprimer**.

Vous pouvez réinitialiser toute barre de menu à son état d'installation originale en le sélectionnant depuis la liste déroulante *Afficher menus pour* et cliquez sur le bouton **Réinitialiser**.

#### Personnaliser les menus contextuels

Les menus contextuels qui apparaissent lorsque vous cliquez avec la touche de droite sur certains objets dans l'interface de l'application. Chacun de ces menus contextuels peuvent être personnalisés en procédant comme suit :

- 1. Choisissez le menu contextuel depuis la liste déroulante Sélectionner un menu contextuel. Ceci ouvre un menu contextuel respectif.
- 2. Ouvrez l'onglet **Commandes** et glissez une commande depuis la zone de liste *Commandes* dans le menu contextuel.
- Pour supprimer une commande depuis le menu contextuel, cliquez avec la touche de droite sur cette commande et sélectionnez Supprimer. En alternative, glissez la commande hors du menu contextuel.

Vous pouvez réinitialiser un menu contextuel dans son état par défaut en le choisissant dans la liste déroulante Sélectionner le menu contextuel cliquer sur la touche **Réinitialiser**.

#### Ombres de menu

Sélectionnez la case à cocher Ombres de menu pour ajouter des ombres à tous les menus.

## 14.11.2 Personnaliser les raccourcis de clavier

Vous pouvez définir ou changer les raccourcis de clavier dans MapForce comme suit : Sélectionnez **Outils** | **Personnaliser** et cliquez sur l'onglet **Clavier**. Pour attribuer un nouveau raccourci à une commande, suivez les étapes suivantes :

- 1. Sélectionnez la commande **Outils | Personnaliser** et cliquez sur l'onglet **Clavier** (*voir la capture d'écran ci-dessous*).
- 2. Cliquez sur la zone de liste modifiable Catégorie pour sélectionner le nom du menu.
- 3. Dans la zone de liste *Commandes*, sélectionnez la commande à laquelle vous souhaitez attribuer dans un nouveau raccourci.
- 4. Saisissez de nouvelles clés de raccourcis dans la zone de texte Appuyer sur Nouvelle clé de raccourci et cliquez sur Attribuer.

Commandes	Barres d'outils	Clavier	Menu	Options	
Catégorie :		Définir r	accourci	pour :	
Fichier 🗸		Par défaut 🗸 🗸		$\bigotimes$	
Commandes	11	Touche	s actuell	es :	
Aperçu ava	ant impressie 🔺				Attribuer
C++ Compiler su Conf. impre	rr Fichier d'€ ss ❤	Nouv.t	ouche de	e raccourci :	Supprimer
Description Générer un	: code en C#				Réinitialiser tout
					Their inclusion tout

Pour effacer la saisie dans la zone de texte *Appuyer sur Nouvelle clé de raccourci*, appuyez sur les clés de commandes : **Ctrl**, **Alt** ou **Shift**. Pour supprimer un raccourci, cliquez sur le raccourci que vous souhaitez supprimer dans la zone de liste *Clés actuelles* et cliquez sur **Supprimer**.

**Note :** Le Set accelerator for n'a pas de fonction pour le moment.

### Raccourcis clavier

Par défaut, MapForce attribue les raccourcis de clavier suivants:

F1	Menu d'aide
F2	Signet suivant (dans la fenêtre de sortie)
F3	Trouver suivant
F10	Activer la barre de menu
Num +	Agrandir le nœud d'item actuel
Num -	Réduire le nœud d'item
Num *	Agrandir tout depuis le nœud d'item actuel
CTRL + TAB	Passe entre les mappages ouverts
CTRL + F6	Feuilleter dans les fenêtres ouvertes
CTRL + F4	Ferme le document de mappage actif
Alt + F4	Ferme MapForce
Alt + F, F, 1	Ouvre le dernier fichier
Alt + F, T, 1	Ouvre le dernier projet

Nouveau fichier
Ouvrir fichier
Enregistrer fichier
Imprimer fichier
Sélectionner tout
Couper
Copier
Coller
Annuler
Rétablir
Supprimer composant (avec invitation)
Supprimer composant (sans invitation)
Recherche
Trouver suivant
Trouver précédent
Choisir item de composant suivant
Abandonner éditions/fermer dialogue

Return	Confirme une sélectior		

### Raccourcis fenêtre sortie

CTRL + F2
F2
SHIFT + F2
CTRL + SHIFT + F2

Insérer Supprimer/Signet Signet suivant Signet précédent Supprimer tous les signets

### **Raccourcis Zoom**

CTRL + roue de la souris en	Zoom avant
avant	
CTRL + roue de la souris en	Zoom arrière
arrière	
CTRL + 0 (Zéro)	Réinitialiser Zoom

# 14.12 Options

Vous pouvez modifier préférences générales et autres dans MapForce sortie en sélectionnant la commande **Outils | Options**. Les options disponibles sont les suivantes :

#### Généralités

Dans la section Généralités, vous pouvez définir les options suivantes :

- Afficher logo | Afficher sur démarrage : Affiche ou masque une image (splash screen) lorsque MapForce démarre.
- Mode Mappage Vous pouvez activer/désactiver l'arrière-plan dégradé dans le volet Mappage pane (Afficher l'arrière-plan dégradé). Vous pouvez également choisir de limiter l'affichage d'annotations à N lignes. Si le texte d'annotation contient plusieurs lignes, l'activation de cette option montre uniquement les premières N lignes( est la valeur que vous spécifiez) du composant. Ce paramètre s'applique également aux instructions SELECT visibles dans un composant.
- Encodage par défaut pour les nouveaux composants.

*Nom d'encodage :* L'encodage par défaut pour les nouveaux fichiers XML peut être défini en sélectionnant une option depuis la liste déroulante. Si un encodage à deux-ou-quatre-byte est sélectionné comme encodage par défaut (par ex., UTF-16, UCS-2 ou UCS-4), vous pouvez aussi choisir entre un tri little-endian et big-endian byte. Ce paramètre peut également être modifié individuellement pour chaque composant( voir <u>Changer paramètres du composants</u><sup>(80)</sup>).

*Tri d'octets :* Quand un document à encodage de caractère two-byte ou four-byte est enregistré, le document peut être enregistré soit avec un tri d'octets little-endian ou big-endian. Vous pouvez également spécifier si une marque de tri d'octets devrait être incluse ou non.

- Paramètres d'aperçu : L'option Utiliser la marche d'arrêt d'exécution dßefinit une marche d'arrêt d'exécution lorsque vous afficher le résultat de mappage dans le volet **Output**.
- Sur activation de l'onglet de Sortie : Vous pouvez générer la sortie vers les fichiers temporaires ou écrire la sortie directement dans un fichier de sortie (voir ci-dessous).

*Générer la sortie vers les fichiers temporaires :* Il s'agit de l'option par défaut. Si le chemin de fichier de sortie contient des dossiers qui n'existent pas encore, MapForce créera ces dossiers. Pour les éditions MapForce Professional et Enterprise : Si vous avez l'intention de déployer le mappage vers un serveur pour l'exécution, tout répertoire dans le champ doit exister sur le serveur ; sinon, une erreur d'exécution se produira. Voir aussi <u>Préparer des mappages pour l'exécution de serveur</u>

*Écrire directement sur les fichiers de sortie finaux :* Si le chemin de fichier de sortie contient des dossiers qui n'existent pas encore, une erreur de mappage se produira. Cette option écrase tout fichier de sortie existant sans demander de confirmation supplémentaire.

• Afficher le texte par étapes de N millions de caractères : Spécifie la taille maximum du texte affiché dans le volet de **Sortie** lorsque vous consultez des mappages qui génèrent des grands fichiers XML et texte. Si le texte de sortie dépasse cette valeur, vous devrez cliquer sur une touche **Charger plus** pour charger le bloc suivant. Pour plus d'informations, voir <u>Consulter et valider le sortie</u>

#### Édition

Dans la section Édition, vous pouvez définir les options suivantes :

- Aligner les composants sur glissage de souris : Spécifiez si des composants ou des fonctions doivent être alignés avec d'autres composants, alors que vous les glissez dans la fenêtre Mappage. Pour plus d'informations, voir <u>Aligner des composants</u><sup>[8]</sup>.
- Suppression de composant intelligent : MapForce vous permet de garder les connexions même après avoir supprimé quleques <u>composants de transformation</u><sup>(76)</sup>. Garder les connexions peut être particulièrement utile avec de multiples connexions enfant car vous ne devrez pas restaurer chaque connexion simple enfant manuellement après avoir supprimé le composant de transformation. Pour les détails, voir <u>Garder des connexions après avoir supprimé des composants</u><sup>(10)</sup>.

#### Messages

La section *Messages* vous permet d'activer des notifications de message telles que la suggestion de connecter des items ancêtre, informant sur la création de multiples composants, etc.

#### Génération (éditions Professional et Enterprise)

La section *Génération* vous permet de définir des paramètres pour la génération de code de programme et les fichiers d'exécution de MapForce Server. Pour plus d'informations, voir <u>Options de générateur de</u> code<sup>(20)</sup> et <u>Compiler des mappages dans les fichiers d'exécution de MapForce Server</u>.

#### 🗉 Java

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 tout chemin Java VM détecté automatiquement par MapForce. Pour plus de détails, voir Java

#### STATE XBRL (Enterprise Edition)

MapForce vous permet de configurer les paramètres (dans toute l'application) XBRL :

- Le langage de libellé des items XBRL et leurs annotations
- Les rôles de libellé préférés pour les noms d'item XBRL
- Le type spécifique de rôles de libellé des annotations pour les items XBRL
- Packs de taxonomie XBRL

#### Débogueur (éditions Professional et Enterprise)

Dans la section Débogueur, vous pouvez définir les paramètres de débogage suivants :

• Longueur maximum des valeurs stockées : Définit la longueur du string des valeurs affichées dans la fenêtre **Valeurs** (au moins 15 caractères). Veuillez noter que le réglage de la longueur de stockage à une valeur élevée peut vider la mémoire système disponible.

Garder l'historique du traçage complet : Instruit MapForce de garder l'historique de toutes les valeurs traitées par toutes les connecteurs de tous les composants dans le mappage pour la durée du débogage. Si cette option est activée, toutes les valeurs traitées par MapForce depuis le début de l'exécution de débogage seront stockées dans la mémoire et seront disponibles pour une analyse dans la fenêtre Valeurs, jusqu'à ce que vous cessiez de le déboguer. Il n'est pas recommandé d'activer cette option si vous êtes en train de déboguer des mappages nécessitant beaucoup de données, étant donné qu'elles pourraient ralentir l'exécution de débogage et vider la mémoire système disponible. Si cette option est désactivée, MapForce ne gardera que l'historique de trace la plus récente pour les nœuds liés à la position d'exécution actuelle.

#### Base de données (éditions Professional et Enterprise)

Dans la section *Base de données*, vous pouvez définir les paramètres de requête de la base de données. Pour les détails, voir <u>Paramètres de requête de base de données</u><sup>(05)</sup>.

#### Proxy de réseau

La section *Proxy de réseau* vous permet de configurer des paramètres de proxy personnalisés. Ces paramètres affectent la manière dont l'application se connecte à Internet. 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. Pour plus de détails, voir <u>Proxy de réseau</u><sup>(105)</sup>.

## 14.12.1 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, MapForce 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 MapForce.

Emplacement de bibliothèque Java VM					
Chemin vers jvm.dll:					
E.g., C:\Program Files (x86)\Java\jdk-11.0.9\bin\server\jvm.dll	Parcourir				
Laisser le champ vide pour la détection automatique du JVM.					
Important : La version Java bit doit être la même que celle de l'application Altova (32-bit).					
Note : Si la machine virtuelle Java a déjà été lancée dans le cadre de l'instance d'application actuelle, les changements effectués au niveau de l'emplacement de VM ne prendront effet qu'une fois l'application redémarrée.					

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 MapForce (32-bit, 64-bit) doit être la même que celle du JDK.
- Une fois avoir modifié le chemin Java VM, vous devrez éventuellement redémarrer MapForce pour que les nouveaux paramètres prennent effet.

La modification du chemin Java VM touche les zones suivantes :

- Connectivité JDBC
- Fonctions d'extension Java pour XSLT/XPath

Ce paramètre ne touche pas la génération du code Java.

## 14.12.2 Base de données

Cette section explique comment configurer les différents paramètres d'édition SQL. Vous pouvez accéder aux paramètres d'une des deux manières suivantes :

- En ouvrant le volet DB Query et en cliquant sur 🔤 (Options).
- En sélectionnant Outils | Options | base de données, puis la section pertinente.

Dans la section *Base de données*, vous pouvez définir les paramètres généraux d'édition SQL, les options du mode et de l'encodage des résultats, les paramètres de génération SQL et les polices. Pour de détails, voir la sous-section ci-dessous.

### Éditeur SQL

La section Éditeur SQL vous permet de changer les paramètres généraux d'édition SQL (voir la capture d'écran ci-dessous). Les paramètres disponibles sont les suivants :

SQL Editor	
General	Connect data source on execute
Retrieval ☑ Show timeout dialog	Execution timeout (in seconds): 60 Buffered amount (rows): 200
Entry Helpers  Automatically open  Fill buffer on connect  Fill buffer the first time it is needed	The Entry Helper Buffer is used by auto-completion and auto-insertion and requires some time to populate itself.
Text View Settings	

- Général Pour voir les différents éléments de la syntaxe SQL dans des couleurs distinctes, activez la coloration de la syntaxe. <anchor id="conndatasource"></anchor>Sélectionnez la case à cocher Connecter la source de données sur exécution pour se connecter automatiquement à la source de données pertinente dès qu'une instruction SQL est exécutée.
- *Extraction* Activer la case à cocher *Afficher dialogue délai d'expiration* pour vous permettre de changer les paramètres du délai d'expiration lorsque la période d'exécution autorisée est dépassée. Vous pouvez spécifier le temps maximum autorisé pour l'exécution SQL (*délai d'expiration*) en secondes. Vous pouvez aussi définir le nombre de lignes qui seront mises dans un buffer.
- Assistants à la saisie Pour permettre des suggestions de saisie semi-automatique dès que vous commencez à taper les instructions SQL, sélectionnez la case à cocher *ouvrir automatiquement* (voir aussi <u>Saisie semi-automatique</u><sup>220</sup>). Vous pouvez choisir quand vous remplissez le buffer d'assistant à la saisie : quand vous vous connectez à la source de données ou quand le buffer est utilisé pour la première fois. Veuillez noter que le remplissage du tampon peut prendre un certain temps. Utiliser la touche Supprimer la mémoire tampon pour réinitialiser la mémoire tampon.
- Paramètres du Mode Texte : Vous permet de définir différents paramètres du mode Texte tels que les marges, onglets, aides visuelles, options de surbrillance automatique et raccourcis clavier de navigation en mode Texte. Pour plus d'informations, voir <u>Fonctions de Mode Texte</u><sup>106</sup>.

### Encodage

Dans la section *Encodage*, vous pouvez spécifier les options d'encodage pour les fichiers de l'Éditeur SQL créés ou ouverts dans l'Éditeur SQL (*voir la cpature d'écran ci-dessous*).

Beneral: Encoding	Reset to Page Defaults
Default encoding for new SQL files	BOM Always create BOM if not UTF-8
<ul> <li>Little-endian byte order</li> <li>Big-endian byte order</li> </ul>	O Preserve detected BOM on saving
Open SQL files with unknown encoding as	
Unicode UTF-8	×

- Encodage par défaut pour les nouveaux fichiers SQL : Définir l'encodage par défaut pour les nouveaux fichiers de manière à ce que chaque nouveau document inclut la spécification d'encodage. Si un encodage à deux-ou-quatre-byte est sélectionné comme encodage par défaut (par ex., UTF-16, UCS-2 ou UCS-4), vous pouvez aussi choisir entre un tri little-endian et big-endian byte pour les fichiers SQL. L'encodage de fichiers existants n'est pas affecté par ce paramètre.
- *Ouvrir les fichiers SQL avec les encodages inconnus comme :* Vous pouvez choisir l'encodage avec lequel ouvrir un fichier SQL sans spécification d'encodage.
- *BOM:* Choisir si vous voulez créer le caractère byte order mark (BOM) pour l'encodage autre que UTF-8 ou s'il faut garder le BOM détecté.

Note : Les fichiers SQL qui n'ont pas de spécifications d'encodage sont enregistrés avec un encodage UTF-8.

### Géneration SQL

La section *Génération* vous permet de spécifier la syntaxe de génération d'instruction SQL pour les différents genres de base de données (*voir la capture d'écran ci-dessous*).

Statement generation Select a database: MS SQL Server Oracle Sybase IBM DB2 IBM iSeries MySQL Postgre SQL Informix Apply to all databa	options	<ul> <li>✓ Append semi-colons to statement e</li> <li>✓ Surround identifiers with escape ch</li> <li>✓ Generate SELECT statements with</li> </ul>	end haracters h full column list
Confirmation Message	en editor	enforces semicolons and SQL generation	n does not

Afin de définir des préférences de syntaxe pour une base de données spécifique, en choisir une depuis la liste, puis activez ou désactivez les trois options à droite. Vous pouvez aussi choisir d'appliquer le même ensemble de paramètre à toutes les bases de données (la case à cocher *Appliquer à toutes les bases de données*). Veuillez noter que l'utilisation des paramètres communs pour toutes les bases de données peut empêcher l'édition des données dans les bases de données Oracle et IBM DB2 et iSeries par le biais d'une connexion JDBC.

### Mode des Résultats

Vous pouvez configurer l'apparence de l'onglet **Résultats** du volet **DB Query** dans la section *Mode des Résultats* (*voir la capture d'écran ci-dessous*).

SQL Editor: Result	View		Reset to Page Defaults
Colors Show grid with alternating	colors		
Display Options Show horizontal lines Show vertical lines	⊡ Show lir ⊡ Show re	ne numbers esult toolbar	Show sorting arrows
Data Editing Use transactions to save of Rollback transactions	changes on failure	☐ Initialize c ☑ Show hint	cells with default values t when data editing is limited.

Sélectionnez la case à cocher *Afficher la grille avec des couleurs alternées* dans Afficher les lignes dans les onglets **Résultat** comme grille simple ou une grille avec des lignes alternantes blanches et colorées. Les couleurs alternatives sont configurables. Le groupe *Options d'affichage* vous permet de définir comment seront affichés des lignes de grille horizontales à verticales, les numéros de ligne et la barre d'outils **Résultat**.

Le groupe Édition des données vous permet de définir paramètres de transaction si les cellules doivent être remplies avec des valeurs par défaut, et si une indication doit être affichée lorsque l'édition des données est limitée.

#### Polices de texte

Dans la section *Polices de texte*, vous pouvez configurer les couleurs et paramètres de polices d'instructions SQL (*voir la capture d'écran ci-dessous*). Vous pouvez choisir le type de police, le style et la coloration syntaxique de travail les textes types qui apparaissent dans l'Éditeur SQL. Veuillez noter que la même police et la même taille sont utilisées pour tous les types de texte. Seul le style peut être modifié pour des types de texte individuels. Si vous accédez à la boîte de dialogue **Polices de texte** depuis le volet **DB Query**, vous pouvez cliquer sur le bouton **Reset to Page Defaults** pour restaurer les paramètres d'origine.

Polices de texte	
Police de l'éditeur texte v	Police
Par défaut	Consolas ~
Commentaire	Utiliser la même pour tous
Nombre	
Chaîne	Taille
Opérateur	10 ~
Identificateur	Utiliser la même pour tous
Variable	
Instruction	Styles
Type de données	В <u>I Ц</u>
Variable globale	
Fonction	Couleur de texte
	$\sim$
	Couleur d'arrière-plan
	Transparent 🗸
	Arrière-plan du Mode texte

## 14.12.3 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.

Network Proxy	
Proxy configuration System V	
Current proxy settings Test URL http://www.example.com	9
Found IE auto-proxy configuration. Methods WPAD (using test URL http://www.example.com) PAC resovled DIRECT (NO PROXY). Using no Proxy.	

#### Utiliser les paramètres de proxy de système

Utilise les paramètres Internet Explorer (IE) configurables par le biais des paramètres de proxy du système. Effectue également une requête des paramètres configurés avec netsh.exe winhttp.

#### Configuration de proxy automatique

Les options suivantes sont possibles :

- Paramètres auto-détection : Consulte un script WPAD (http://wpad.LOCALDOMAIN/wpad.dat) par le biais de DHCP ou DNS, et utilise ce script pour une configuration proxy.
- URL de script : Spécifie une URL HTTP dans un script configuration-auto-proxy (.pac) qui doit être utilisé pour cette configuration de proxy.
- *Recharger :* Réinitialise et recharge la configuration automatique de proxy actuelle. Cette action requiert Windows 8 ou plus, et peut prendre jusqu'à 30 sec avant de prendre effet.

#### Configuration de proxy manuelle

Spécifier manuellement le nom d'hôte et le port entièrement qualifiés pour les proxies des protocoles respectifs. Un scheme pris en charge peut être inclus dans le nom d'hôte (par exemple : http://hostname). Il n'est pas exigé que le schéma soit le même que le protocole respectif si le proxy prend en charge le schéma.

Network I	Proxy	
Proxy configur	ation Manual 🗸 🗸	
HTTP Proxy		Port 0
	Use this proxy server for all protocols	
SSL Proxy		Port 0
No Proxy for	r	
	Do not use the proxy server for local addresses	
Current prox	y settings	
Test URL ht	tp://www.example.com	5
(using test Using no Pr	URL http://www.example.com) roxy.	
	5	

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 transformation sur <u>Altova FlowForce</u> <u>Server</u>, 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).

# 14.13 Fenêtre

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Fenêtre.

#### Cascade

Cette commande réarrange toutes les fenêtres de document ouvertes de manière à ce qu'elles se présentent en cascade (donc, étalées) l'une sur l'autre.

Mosaïque horizontale

Cette commande réarrange toutes les fenêtres de document ouvertes sous forme de mosaïque horizontale, les rendant toutes visibles en même temps.

Mosaïque verticale

Cette commande réarrange toutes les fenêtres de document ouvertes mosaïque verticale, les rendant toutes visibles en même temps.

Thème classique/clair/sombre

MapForce vous permet de choisir des thèmes différents : *Classique*, *Clair* et *Sombre*. Les exemples de ces thèmes sont illustrés dans la capture d'écran ci-dessous. L'option par défaut est le thème classique.

🛃 <u>F</u> ile	<u>E</u> dit	<u>I</u> nsert	<u>P</u> roject	<u>C</u> omponent	Connection	Function
i 🗂 🖻	8é		<b>N</b> CH	ሯ 🖻 🛱 🕽	K 🏥 🖉 🕯	a 🛛 😻 🗸 🗄
Thème c	lassiq	ue				
🛃 <u>F</u> ile	<u>E</u> dit	<u>I</u> nsert	<u>P</u> roject	<u>C</u> omponent	Connection	Function
i 🗋 걸			<b>n</b> 🖂	👗 🖻 💼 🕽	< 69 cm m	2 😻 💵
Thème c	lair					
ᡖ <u>F</u> ile	<u>E</u> dit	<u>I</u> nsert	<u>P</u> roject	<u>C</u> omponent	Connection	Fu <u>n</u> ction
i 🗋 📂		• 6	S CI	👗 🖻 🛍 🕽	<   🛱 🕼 🕅	2 🛛 👽 💵

Thème sombre

1 <MappingName>

Se réfère au premier design de mappage ouvert. S'il y a plus de mappages ouverts en même temps, ils seront également recensés dans le menu contextuel.

Windows

Cette liste affiche toutes les fenêtres actuellement ouvertes et vous permet de basculer rapidement entre elles. Vous pouvez utiliser les raccourcis de clavier **Ctrl-TAB** ou **CTRL F6** pour basculer entre les fenêtres ouvertes.

# 14.14 Aide

Cette rubrique recense toutes les commandes de menu disponibles dans le menu Aide.

#### Sommaire

Ouvre le manuel d'aide sur écran de MapForce avec le Sommaire affiché dans le panneau de gauche de la fenêtre Aide. Le Sommaire présente un aperçu de l'ensemble du document Aide. Cliquer sur une entrée dans le Sommaire pour vous rendre dans cette section.

#### Index

Ouvre le manuel d'aide sur écran de MapForce avec l'Index de clavier affiché dans le panneau de gauche de la fenêtre Aide. L'index regroupe les mots-clés et vous permet de naviguer vers un thème en doublecliquant le mot-clé. Si un mot-clé est lié à plus d'un thème, une liste de ces thèmes s'affichera.

#### Recherche

Ouvre le manuel d'aide sur écran de MapForce avec le dialogue Recherche affiché dans le panneau de gauche de la fenêtre Aide. Pour rechercher un terme, saisir le terme dans le champ de saisie et appuyer (i) sur Entrée ou (ii) sur List topics. Le système d'aide effectuera une recherche complète dans l'ensemble de la documentation d'aide et retourne une liste des rubriques trouvées. Double-cliquer sur un élément pour l'afficher.

#### Activation logiciel

#### Mettre sous licence votre produit

Après avoir téléchargé votre logiciel de produits Altova, vous pourrez acquérir une licence - ou l'activer - en utilisant soit une clé d'évaluation gratuite ou en achetant une clé de licence permanente.

 Licence d'évaluation gratuite. Lorsque vous lancez le logiciel pour la première fois après l'avoir téléchargé et installé, le dialogue Activation du logiciel s'ouvrira. Vous y trouverez un bouton pour demander une licence d'évaluation gratuite. Saisir votre nom, entreprise et votre adresse email dans le dialogue qui apparaît et cliquer sur Requête. Un fichier de licence est envoyé à l'adresse e-mail que vous avez saisi et devrait arriver dans quelques minutes. Enregistrer le fichier de licence dans un endroit approprié.

Lorsque vous cliquez sur **Requête**, un champ de saisie apparaît en bas du dialogue Requête. Ce champ prend le chemin vers le fichier de licence. Chercher ou saisir le champ vers le fichier de licence et cliquer sur **OK**. (Dans le dialogue d'**Activation du logiciel**, vous pouvez aussi cliquer sur **Charger une Nouvelle Licence** pour accéder à un dialogue dans lequel le chemin vers le fichier de licence est saisi.) Le logiciel sera débloqué pour une période de 30 jours.

• Clé de licence permanente. Le dialogue Activation du logiciel contient un bouton pour acheter une clé de licence permanente. Cliquer sur ce bouton pour vous rendre à la boutique en ligne d'Altova, où vous pourrez acheter une clé de licence permanente pour votre produit. Votre licence vous sera envoyée par e-mail sous forme d'un fichier de licence contenant vos données de licence.

Il existe trois types de licences permanentes : *installée, utilisateur simultané*, et *utilisateur nommé*. Une licence installée déverrouille le logiciel sur un seul ordinateur. Si vous achetez une licence installée pour*N* ordinateurs, la licence permettra une utilisation du logiciel sur jusqu'à *N* ordinateurs. Une licence utilisateur concomitant pour *N* utilisateurs concomitants permet à *N* 

utilisateurs d'exécuter le logiciel simultanément. (Le logiciel peut être installé sur 10N ordinateurs.) Une licence utilisateur nommé autorise un utilisateur spécifique d'utiliser le logiciel sur jusqu'à 5 ordinateurs différents. Pour activer votre logiciel, cliquer sur **Charger une Nouvelle licence**, et, dans le dialogue qui apparaît, chercher ou saisir le chemin vers le fichier de licence et cliquer sur **OK**.

**Note :** en ce qui concerne les licences utilisateurs multiples, chaque utilisateur sera invité à saisir son nom dans le champ Nom.

Votre e-mail de licence et les différents moyens de mise sous licence de votre produit Altova (activation) à votre disposition

L'e-mail de licence que vous avez reçu de la part d'Altova contient votre fichier de licence en pièce jointe. Le fichier package a une extension de fichier .altova\_licenses.

Pour activer votre produit Altova, vous pouvez choisir une des étapes suivantes :

- Enregistrez le fchier de licence (.altova\_licenses) vers un emplacement approprié, double-cliquez sur le fichier de licence, saisissez tout détail requis dans le dialogue qui apparaît, et terminez en cliquant sur Appliquer clés.
- Enregistrez le fichier de licence (.altova\_licenses) vers un emplacement approprié. Dans votre produit Altova, sélectionnez la commande de menu Aide | Activation Logiciel, puis Télécharger une nouvelle licence. Recherchez ou saisissez le chemin du fichier de licence, et cliquez sur OK.
- Enregistrez le fichier de licence (.altova\_licenses) vers un emplacement approprié et chargez-le depuis cet emplacement vers le pool de licences de votre <u>Altova</u> <u>LicenseServer</u> 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 <u>Serveur de licence Altova</u> (*voir les détails ci-dessous*).

- Enregistrant la licence dans le dialogue d'Activation du logiciel. Dans le dialogue, cliquez sur Charger une Nouvelle Licence, puis cherchez et sélectionnez le fichier de licence. Cliquez sur OK pour confirmer le chemin vers le fichier de licence et confirmez toutes les données que vous avez saisies (votre nom dans le cas de licences multi-utilisateur). Terminez en cliquant Enregistrer.
- Mise sous licence par le biais du Serveur de licence Altova sur votre réseau : Pour acquérir une licence par le biais d'un Altova LicenseServer sur votre réseau, cliquez sur Utiliser Altova LicenseServer, situé en bas du dialogue Activation du logiciel. Choisissez l'appareil sur lequel le LicenseServer que vous souhaitez utiliser a été installé. Veuillez noter que l'auto-découverte des License Servers fonctionne par le biais d'une diffusion envoyée sur le LAN. Puisque les diffusions sont limitées à un sous-réseau, License Server doit se trouver sur le même sous-réseau

que l'appareil client pour la découverte automatique afin de fonctionner. Si elle ne fonctionne pas, saisir le nom du serveur. L'Altova LicenseServer doit disposer d'une licence pour votre produit Altova dans son pool de licence. Si une licence est disponible dans le pool de LicenseServer, cela sera indiqué dans le dialogue d'Activation du logiciel (*la capture d'écran ci-dessous affiche le dialogue dans Altova XMLSpy*). Cliquez sur Enregistrer pour acquérir la licence.

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-dé	Saisir un code-clé				
Pour activer votre logiciel, veuillez saisir ou sélectionner le nom du Altova LicenseServer su	r votre réseau.				
Altova LicenseServer : DEV02	ک 🗸				
다. 이 아이는 Nucleice vous a déjà été attribuée sur LicenseServer sous VIEPDEV02.vie.alto	va.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 désattribuer à tout moment une licence acquise par le biais de la Web UI du LicenseServer). Veuillez noter qu'un renvoi de la licence n'est applicable qu'aux seules licences sur appareil installées, pas aux licences concurrentes.

#### Extraire la licence

Vous pouvez consulter une licence du pool de licence pour une période de jusqu'à 30 jours pour que la licence puisse être stockée sur l'appareil de produit. Cela vous permet de travailler hors ligne, ce qui peut être utile, par exemple, si vous souhaitez travailler dans un environnement où vous ne pourrez pas accéder à votre Altova LicenseServer (par exemple, si votre produit Altova est installé sur un ordinateur portable et que vous vous trouvez en déplacement). Tant que la licence est extraite, LicenseServer affiche la licence comme étant utilisée ; elle ne peut donc pas être utilisée par une autre machine. La licence passe automatiquement à l'état d'archivage lorsque la période d'extraction expire. En alternative, une licence extraite peut être archivée à tout moment par le biais du bouton **Archiver** du dialogue d'**Activation du logiciel**.

Pour extraire une licence, procédez comme suit : (i) dans le dialogue d'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.

Les enregistrements de licence doivent correspondre à la même version majeure du produit duquel la licence a été extraite. Donc veillez à enregistrer une licence avant de mettre à niveau votre produit Altova vers la prochaine version majeure.

**Note :** afin de pouvoir effectuer des extractions de licence, la fonction d'extraction doit être activée sur le LicenseServer. Si la fonction n'a pas été activée, vous recevrez un message d'erreur à cet effet lorsque vous essayez de faire le « check out ». Dans ce cas, veuillez contacter votre administrateur de LicenseServer.

#### Copier code de support

Cliquer sur **Copier code de support** pour copier des détails de licence dans le presse-papiers. Il s'agit des données que vous devrez fournir en cas de demande d'assistance avec le <u>formulaire</u> <u>d'assistance en ligne</u>.

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 <u>site web Altova</u>. Pour plus d'informations concernant Altova LicenseServer et la mise sous licence par le biais d'Altova LicenseServer, voir la <u>documentation Altova LicenseServer</u>.

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.

Centre de support

Un lien qui vous mènera vers le Centre de support Altova sur Internet. Le Centre de support contient des FAQ, des forums de discussion pour toute sorte de problèmes et l'accès à l'équipe de support technique

#### d'Altova.

#### FAQ sur le web

Un lien menant à la base de données FAQ d'Altova sur Internet. La base de données FAQ est constamment mise à jour à la suite des problèmes rapportés par les clients.

E 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.

MapForce sur Internet

Un lien menant au <u>site web Altova</u> sur Internet. Vous pouvez en apprendre plus sur MapForce, les technologies et produits liés le le <u>site web Altova</u>.

#### Formation MapForce

Un lien menant à la page de Formation en ligne sur le <u>site web Altova</u>. Ici, vous pouvez choisir parmi une série de cours en ligne tenus par des formateurs experts Altova.

À propos de MapForce

Affiche la fenêtre d'accueil et le numéro de version de votre produit. Si vous utilisez la version 64-bit de MapForce, 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.

# 15 Générateur de code

Le générateur de code est une fonction MapForce intégrée qui vous permet de générer du code Java, C++ ou C# depuis des fichiers de mappage conçus avec MapForce. Vous pouvez générer du code non seulement depuis des mappages simples avec une seule source et cible de données, mais aussi depuis des mappages avec des sources multiples et des cibles multiples. Le résultat est une application complète proposant toutes les fonctions qui effectue l'opération de mappage pour vous. Une fois que vous générez le code, vous pouvez exécuter le mappage en exécutant l'application directement lors de la génération. Vous pouvez aussi importer le code généré dans votre propre application ou y ajouter vos propres fonctions.

L'objectif primaire du code généré est d'exécuter un mappage MapForce. Si votre mappage utilise des schémas XML ou DTD, vous pouvez générer optionnellement des bibliothèques schema wrapper pour des schémas XML/DTD utilisés par le mappage, qui vous permettent d'écrire ou de lire des données vers ou depuis des instances XML. Les bibliothèques schema wrapper vous permettent de travailler avec des données XML de manière programmatique, utilisant des types générés depuis le schéma, sans trop se soucier de l'API XML sous-jacent. Pour tout détail, voir <u>Générer du Code depuis les Schémas XML ou des DTD</u>

Langage cible	C++	C#	Java
Environnements de développement	Microsoft Visual Studio 2013, 2015, 2017, 2019, 2022	Microsoft Visual Studio 2013, 2015, 2017, 2019, 2022 Frameworks cibles : • .NET Framework • .NET Core 3.1 • .NET 5.0 • .NET 6.0	Java SE JDK 8 ou 11 (y compris OpenJDK) Eclipse 4.4 ou ultérieur Apache Ant (build.xml file)
Mises en œuvre XML DOM	MSXML 6.0 Apache Xerces 3	System.Xml	JAXP
Database API	ADO	ADO.NET	JDBC

Le code généré est exprimé en langages de programmation C++, Java ou C#.

**Note :** le code généré de MapForce peut être considéré thread-safe uniquement si l'XML DOM tiers sousjacent et les bibliothèques API de bases de données le sont. Bien que la thread safety du code généré ne peut pas être prouvée ou garantie de manière réaliste, il est probable que de multiples instances concurrentes du code de mappage seront exécutées avec succès dans la plupart des cas.

### C++

Vous pouvez configurer si la sortie générée C++ doit utiliser MSXML 6.0 ou Apache Xerces 3. MapForce génère le projet complet (.vcproj) et les fichiers de solution (.sln) pour toutes les versions prises en charge de Visual Studio (voir la table ci-dessus)Le code généré prend en charge MFC optionnellement.

Prérequis :

- 1. Pour compiler le code C++ généré, Windows SDK doit être installé sur votre ordinateur.
- Pour utiliser Xerces 3 pour C++, vous allez devoir l'installer et le construire par les instructions suivantes depuis <u>http://xerces.apache.org/xerces-c/</u>. Assurez-vous d'ajouter la variable d'environnement XERCES3 qui pointe vers le répertoire dans lequel Xerces a été installé, par exemple : C:\xerces-c-3.2.2. Aussi, la variable d'environnement PATH doit inclure le chemin où se trouvent les binaires Xerces, par exemple : %xerces3%\bin.
- Lorsque vous construisez un code C++ pour Visual Studio et utilisez une bibliothèque Xerces précompilée pour Visual C++, un paramètre de compilateur doit être changé dans tous les projets de la solution :
  - a) Sélectionnez tous les projets dans le Solution Explorer.
  - b) Dans le menu Projet, cliquez sur Propriétés.
  - c) Cliquez sur **Propriétés de configuration | C/C++ | Langage**.
  - d) Dans la liste de configurations, sélectionnez Toutes les Configurations.
  - e) Changer Treat wchar\_t as Built-in Type à No (/Zc:wchar\_t-).

#### C#

Le code C# généré peut être utilisé depuis tout langage de programmation .NET capable, tel que VB.NET, Managed C++, ou J#. Les fichiers de projet peuvent être générés pour toutes les versions prises en charge de Visual Studio (voir la table ci-dessus).

#### Java

La sortie Java générée est écrite par rapport à l'API Java pour Traitement XML (JAXP) et inclut un fichier Ant build et des fichiers de projet pour des versions prises en charge de Java et Eclipse (voir la table ci-dessus).

### Sortie générée

Le dossier de destination désigné inclura toutes les bibliothèques et tous les fichiers requis pour exécuter le mappage, à savoir :

- Un nombre variable de bibliothèques Altova requises par le mappage (par exemple, les bibliothèques de fonction Altova, les bibliothèques de bases de données)
- Une application de mappage complète. Lorsqu'elle est compilée et exécutée, l'application exécute la transformation de mappage.

#### Modèles de générateur de code

Le code généré prend en charge la personnalisation par le biais d'un <u>langage de modèle</u> appelé SPL (Spy Programming Language). Ceci peut être utile, par exemple, quand vous souhaitez personnaliser le code selon les conventions d'écriture de votre société, ou remplacer les bibliothèques spécifiques dans le code généré.

# 15.1 Les nouveautés...

#### Version 2022 R2

- Prise en charge de Visual Studio 2022.
- Prise en charge de . NET 6.0.

#### Version 2021 R2

• Lorsque vous générez le code C#, vous pouvez choisir .NET Core 3,1, NET 5.0 ou .NET 6.0 en tant qu'infrastructures cibles depuis les <u>options de génération de code</u> (ceci s'ajoute à la prise en charge existante pour les projets .NET Framework).

#### Version 2020 R2

• Le code généré pour des bibliothèques de wrapper de schéma XML propose maintenant un plus grand contrôle sur les espaces de noms d'élément et des préfixes. De nouvelles méthodes sont disponibles pour déclarer ou contourner des espaces de noms pour un élément, ou pour apposer un élément avec un espace de noms préfixé. Voir <u>Exemple : Purchase Order</u><sup>1145</sup>.

#### Version 2020

- Prise en charge de Visual Studio 2019
- Fin de la prise en charge de Visual Studio 2008

#### Version 2018

- Prise en charge étendue à Microsoft Visual Studio 2013, 2015, 2017
- Fin de la prise en charge de Visual Studio 2005 et Xerces 2.x

#### Version 2014

• Retrait de l'option de mode de compatibilité pour la génération de code

#### Version 2011

• Contient des corrections de bogues et des améliorations

#### Version 2010 R3

- Prise en charge de Microsoft Visual Studio 2010
- Prise en charge de MSXML 6.0 dans du code C++ généré
- Prise en charge de cibles 64-bit pour les projets C++ et C#

#### Version 2010

• Des facettes d'énumération provenant de schémas XML sont maintenant disponibles en tant que constantes symboliques dans les classes générées (en utilisant les modèles 2007r3)

#### Version 2009 sp1

#### 1068 Générateur de code

• Prise en charge d'Apache Xerces version 3.x (les anciennes versions commençant à partir de Xerces 2.6.x sont toujours prises en charge)

#### Version 2009

• La mise en place de mappage généré a été revue et corrigée pour prendre en charge les séquences et les regroupements. L'API n'a pas changé.

#### Version 2008 R2

- La prise en charge de la génération de fichiers de projet Visual Studio 2008 pour C# et C++ a été ajoutée
- Un code de mappage MapForce généré en C# et Java peut utiliser des lectures/écritures, des streams, des strings ou des documents DOM en tant que sources et cibles

#### Version 2008

- Les nouveaux modèles 2007 R3-style SPL ont encore été améliorés :
  - o Il est maintenant possible de supprimer des éléments uniques
  - L'accès aux métadonnées de schéma (par ex. noms d'élément, facettes, énumérations, occurrence, etc.) est possible
  - Les types complexes dérivés par extension sont maintenant générés en tant que classes dérivées

#### Version 2007 R3

- Le Générateur de code a été reconçu pour la version 2007 release 3 dans le but de simplifier l'utilisation du code généré, réduire le volume de code et augmenter la performance.
- Gestion des documents XML et des nœuds avec une propriété explicite pour éviter les pertes de mémoire et permettre un multi-threading
- Nouvelle syntaxe pour éviter les collisions de nom
- Nouveaux types de données pour une utilisation plus simple et une meilleure performance (types natifs si possible, nouveau null handling, ...)
- Les attributs ne sont plus générés en tant que collections
- Le contenu d'élément simple est maintenant aussi traité comme un attribut spécial, pour une plus grande homogénéité
- Nouveau modèle d'objet interne (important pour les modèles SPL personnalisés)
- Mode de compatibilité pour générer du code dans le style des releases plus anciens
- Les classes de type wrapper sont maintenant uniquement générées sur demande pour un code plus petit

# 15.2 Génération de code C++

Vous pouvez générer du code C++ pour Visual Studio 2013, 2015, 2017, 2019, 2022. Le code généré inclut les fichiers **.sln** et **.vcproj** pour Visual Studio. Veuillez noter la chose suivante lorsque vous générez du code :

- Certaines fonctions MapForce ne sont pas prises en charge dans du code de programme généré. Pour plus de détails, voir <u>Fonctions prises en charge dans le code généré</u><sup>1476</sup>.
- Vous pouvez générer du code soit depuis un seul design de mappage (.mfd), ou depuis un projet de mappage (.mfp). Si vous générez du code depuis un seul mappage, l'application résultante exécute la transformation de mappage respective. Si vous générez du code depuis un projet MapForce (.mfp) qui contient plusieurs mappages, l'application résultante exécute en vrac tous les mappages contenus dans le projet.
- Vous pouvez modifier les options de génération de code générales depuis le menu Outils | Options, dans l'onglet Génération. Voir <u>Options du Générateur de Code</u><sup>(20)</sup>.
- Vous pouvez changer le nom de l'application de mappage générée et d'autres paramètres provenant du menu Fichier | Paramètres de mappage . Le nom d'application par défaut est Mappage. Voir aussi <u>Changer les paramètres de réseau</u><sup>(112)</sup>.
- Pour plus d'information concernant les chemins de handling dans le code généré, voir <u>Chemins dans</u> les différents environnements d'exécution<sup>85</sup>.
- Si votre mappage contient des composants de base de données, vous pouvez consulter des paramètres spécifiques à la base de données en cliquant sur un composant de base de données puis en choisissant l'option de menu Composant | Propriétés. Voir <u>Paramètres de composant de base de</u> <u>données</u><sup>231</sup>.
- La prise en charge pour les connexions de base de données varie selon les plateformes et il existe des types de connexion qui ne sont pas pris en charge sur toutes les plateformes. Si votre mappage se connexte à une base de données, choisir une connexion de base de données qui est compatible avec l'environnement cible pour lequel vous générez du code. Pour plus de détails, voir <u>Mappages de base de données dans des environnements d'exécution variés</u>

Une solution C++ typique générée par MapForce comprend les éléments suivants :

- Plusieurs bibliothèques signées Altova requises par le mappage (toutes préfixées avec Altova).
- Le projet de mappage principal (dans cet exemple, **Mapping**), qui comprend l'application de mappage et des fichiers dépendants.



Exemple de solution C++ générée avec MapForce

Cette section inclut les rubriques suivantes :

- <u>Générer du code depuis un mappage</u><sup>1070</sup>
- <u>Générer du code depuis un projet de mappage</u><sup>1071</sup>
- <u>Construire le projet</u><sup>1072</sup>
- Exécuter l'application<sup>1072</sup>

## 15.2.1 Générer du code depuis un mappage

#### Pour générer du code C++ depuis un fichier de design de mappage (.mfd) :

- 1. Consulter et sélectionner les options générateur de code<sup>[20]</sup>.
- 2. Dans le menu Fichier, cliquer sur Générer code dans | C++ .
- Choisir un répertoire de destination pour les fichiers générés, puis cliquer sur OK pour confirmer. Le résultat de la génération de code (message d'erreur ou de succès) est affiché dans la fenêtre Messages.

Le nom par défaut de l'application générée est **Mapping**. Le cas échéant, vous pouvez le modifier, ainsi que d'autres paramètres depuis le dialogue <u>Paramètres de mappage</u><sup>112</sup>.

## 15.2.2 Générer du code depuis un projet de mappage

#### Pour générer du code depuis un projet de mappage (.mfp) :

- 1. Si vous ne l'avez pas encore fait, ouvrez le projet de mappage dans MapForce.
- 2. Cliquez avec la touche de droite dans la fenêtre de projet, puis cliquez sur Propriétés.

Project Settings		<b>—</b>
Project Name:	Project01	
Project <u>D</u> irectory:	C:\Project01\	
Output Settings		
Output Name:	Project01	
Output Directory:	C:\Project01\output\	Browse
Language:	Java 🔹	
Java Settings		
Base package name:	com.mapforce	
	ОК	Cancel

- 3. Consultez et modifiez les paramètres de projet le cas échéant (en particulier, assurez-vous que le langue cible et le répertoire de sortie sont définis correctement), puis cliquez sur **OK**.
- 4. Dans le menu **Projet**, cliquez sur Générer du code pour tout le projet.

Le progrès et le résultat du processus de génération de code (message de succès ou d'erreur) est affiché dans la fenêtre Messages.

Par défaut, le nom de l'application générée est le même que le nom du projet. Si le nom du projet contient des espaces, elles sont converties en traits de soulignement dans le code généré. Par défaut, le code est généré dans le même répertoire que le projet MapForce, dans le sous-répertoire **sortie**.

Pour changer le répertoire de sortie et le nom du projet, cliquer sur Projet dans la fenêtre Projet, puis sélectionner **Projet | Propriétés** depuis le menu. **Si votre projet de mappage contient des dossiers, vous pouvez modifier les paramètres de génération de code pour chaque dossier individuel (cliquer sur la touche de droite sur le dossier, puis sélectionner** Propriétés). Sinon, tous les dossiers de projet héritent des paramètres de projet MapForce.

## 15.2.3 Construire le projet

Une fois que vous avez généré le code C++, vous pourrez le construire dans Visual Studio. Pour construire le code généré :

1. Ouvrir le fichier de solution généré (.sln) dans Visual Studio.

Par défaut, le nom du fichier de solution est **Mapping.sln**, et il se trouve dans la sous-répertoire **Mapping** relatif au répertoire dans lequel vous avez enregistré le code généré. Si vous avez changé le nom d'application provenant des paramètres de mappage, puis le nom du fichier .sln est modifié conformément. Par exemple, si vous avez changé le nom d'application dans **MyApplication**, puis le fichier de solution est appelé **MyApplication**.

2. Dans le menu Build, cliquer sur Configuration Manager.

Configuration Manager				? <mark>×</mark>
Active solution configuration:		Active solut	tion platform:	
Debug		Win32		•
Debug Release		ploy):		
Unicode Debug			Platform	Build
<pre>Vnicode Release <new></new></pre>			Win32	<b>V</b>
<edit></edit>			Win32	
AltovaFunctions	Debug		Win32	
AltovaXML	Debug		Win32	
Mapping	Debug	-	Win32	. ▼
				Close

- Choisir la configuration de build requise (Debug, Release, Unicode Debug, Unicode Release). Veuillez noter que seuls les builds Unicode soutiennent l'ensemble des caractères Unicode dans XML et d'autres fichiers. Les builds non-Unicode fonctionnent avec la page de code locale de votre installation Windows.
- 4. Dans le menu **Build**, cliquer sur **Build Solution**.

## 15.2.4 Exécuter l'application

Une fois que vous compilez le projet Visual Studio, une application de ligne de commande est produite, appelée **Mapping.exe**. (Veuillez noter que si vous avez changé le nom de l'application à partir des paramètres de mappage, le nom du programme d'exécution sera modifié en conséquence.)

Vous pouvez situer l'application de mappage dans un des sous-répertoires relatifs au fichier .sln, selon l'option de build que vous avez choisi :

- Déboguer
- Release
- Débogage Unicode
- Unicode Release

Pour exécuter l'application, ouvrir une invite de commande, modifier le répertoire actuel vers le chemin du programme d'exécution, et l'exécuter, par exemple :

C:\codegen\DB\_CompletePOcpp\Mapping\Debug>Mapping.exe Mapping Application Finished

C:\codegen\DB\_CompletePOcpp\Mapping\Debug>\_

# 15.3 Génération de code C#

Vous pouvez générer le code C# code .NET pour tout mappage ou projet de mappage où tous les composants de mappage sont pris en charge dans C#. Pour un aperçu des composants pris en charge dans chaue langage, y compris C#, voir les <u>Fonctions prises en charge dans le code généré</u> . Veuillez noter que MapForce publie généralement des messages d'avertissement et de notification à chaque fois que vous tentez d'ajouter des composants qui ne sont pas pris en charge dans le langage spécifique.

Lorsque vous générez du code C#, vous pouvez cibler non seulement la plateforme .NET Framework mais aussi les plateformes .NET Core 3.1, .NET 5.0 et .NET6.0. Si vous choisissez la dernière, le résultat est une application desktop multiplateforme .NET (Core) qui peut être exécutée sur Windows, macOS et Linux.

Veuillez noter la chose suivante lorsque vous générez du code :

- Certaines fonctions MapForce ne sont pas prises en charge dans du code de programme généré. Pour plus de détails, voir <u>Fonctions prises en charge dans le code généré</u><sup>1476</sup>.
- Vous pouvez générer du code soit depuis un seul design de mappage (.mfd), ou depuis un projet de mappage (.mfp). Si vous générez du code depuis un seul mappage, l'application résultante exécute la transformation de mappage respective. Si vous générez du code depuis un projet MapForce (.mfp) qui contient plusieurs mappages, l'application résultante exécute en vrac tous les mappages contenus dans le projet.
- Vous pouvez modifier les options de génération de code générales depuis le menu Outils | Options, dans l'onglet Génération. Voir <u>Options du Générateur de Code</u><sup>(20)</sup>.
- Vous pouvez changer le nom de l'application de mappage générée et d'autres paramètres provenant du menu Fichier | Paramètres de mappage. Le nom d'application par défaut est Mappage. Voir aussi Changer les paramètres de réseau<sup>(112)</sup>.
- Pour plus d'information concernant les chemins de handling dans le code généré, voir <u>Chemins dans</u> les différents environnements d'exécution<sup>85</sup>.
- Si votre mappage contient des composants de base de données, vous pouvez consulter des paramètres spécifiques à la base de données en cliquant sur un composant de base de données puis en choisissant l'option de menu Composant | Propriétés. Voir <u>Paramètres de composant de base de</u> <u>données</u><sup>231</sup>.
- La prise en charge pour les connexions de base de données varie selon les plateformes et il existe des types de connexion qui ne sont pas pris en charge sur toutes les plateformes. Si votre mappage se connexte à une base de données, choisir une connexion de base de données qui est compatible avec l'environnement cible pour lequel vous générez du code. Pour plus de détails, voir <u>Mappages de base de données dans des environnements d'exécution variés</u>

Si votre mappage utilise des composants de base de données, assurez-vous de créer une connexion de base de données utilisant des pilotes de base de données compatibles avec la plateforme sur laquelle vous voulez exécuter le programme généré. Par exemple, les connexions de base de données ADO et les pilotes ODBC ne sont pas pris en charge sur Linux.

Une solution C# typique générée par MapForce comprend les éléments suivants :

- Des fichiers Solution (.sln) et project (.csproj) peuvent être ouverts dans Visual Studio.
- Plusieurs bibliothèques signées Altova requises par le mappage (toutes préfixées avec Altova).
- Le projet de mappage principal (dans cet exemple, Mapping), qui comprend l'application de mappage et des fichiers dépendants.



Exemple de solution C# générée avec MapForce

# 15.3.1 Sélectionner la plateforme cible

Avant de générer le code C#, vous pouvez définir la plate-forme cible au niveau du programme, comme suit :

- 1. Dans la menu Outils, cliquez sur les Options.
- 2. Cliquez sur l'onglet Generation.
- 3. Dans le groupe « paramètres C# », sélectionnez une valeur depuis le menu déroulant, si applicable.

Generation	
C±+ Settings Project File:	C <u>#</u> Settings Project File:
Microsoft® Visual Studio® 2015 $$	Microsoft® Visual Studio® 2015 ~
XML Library ● MSXML 6 ○ Xerces 3.x	Microsoft® Visual Studio® 2010 Microsoft® Visual Studio® 2013 Microsoft® Visual Studio® 2015 Microsoft® Visual Studio® 2017 Microsoft® Visual Studio® 2019
Library Type (i) <u>S</u> tatic Library (.LIB)	Microsoft® .NET Core 3.1 Microsoft® .NET 5.0
O Dynamic-Link Library (.DLL)	Wrapper Classes
MFC Support	Generate <u>W</u> rapper Classes

Sélectionnez l'option **Microsoft .NET Core 3.1, Microsoft .NET 5.0 ou Microsoft .NET 6.0** pour générer une solution Visual Studio ciblant les plate-formes respectives. Si vous devez cibler la plate-forme **.NET Framework** pour une version spécifique de Visual Studio, sélectionnez toute option de **Microsoft Visual Studio 2010-2019** - dans ce cas, la solution générée ciblera la version .NET Framework correspondant à la version respective de Visual Studio.

La case à cocher **Generate Wrapper Classes**, si elle est sélectionnée, crée des projets supplémentaires dans le code généré, un pour chaque schéma XML ou composant DTD présent dans le mappage. Les projets de wrapper générés vous permettent de manipuler de manière programmatique les instances XML respectives à l'extérieur du mappage. Les bibliothèques wrapper sont optionnelles et indépendantes du programme de mappage, bien que vous puissiez les utiliser depuis le programme de mappage, si nécessaire. Pour plus d'information sur les classes de wrapper, voir <u>Générer du Code depuis des Schémas XML ou DTD</u>

## 15.3.2 Générer du code depuis un mappage

Pour générer du code C# depuis un fichier de design de mappage (.mfd) :

- 1. Si vous ne l'avez pas déjà fait, créez un nouveau mappage ou ouvrez un mappage existant dans MapForce.
- 2. Sélectionnez C# en tant que langage de transformation du mappage. Pour ce faire, cliquez sur is bouton de la barre d'outils ou sélectionnez la commande de menu Sortie | C# (Sharp).
- 3. Définissez ou modifiez les options de génération du code C# telles que décrites dans <u>Sélectionner la</u> <u>plate-forme cible</u><sup>1075</sup>.
- 4. Sélectionnez la commande de menu Fichier | Générer code dans | C# (Sharp). Vous êtes désormais invité à sélectionner un répertoire cible.
- 5. Parcourez un répertoire de vote choix (par exemple, "C:\codegen\ConvertProducts"), puis cliquez sur **OK**.

La sortie de génération de code est affichée dans la fenêtre des Messages, par exemple :



En option, cliquez sur le lien dans la fenêtre Messages pour ouvrir la solution générée dans Visual Studio.

Veuillez noter que si vous choisissez de générer le code ciblant les plate-formes .NET 5, .NET 6.0 ou .NET, vous aurez besoin des SDK respectifs et une version compatible de Visual Studio. Pour votre package de téléchargement qui s'applique à votre système d'exploitation et à votre plate-forme, veuillez vous référer au site web de Microsoft (<u>https://dotnet.microsoft.com/download</u>).

Afin de revérifier le framework cible du projet de Visual Studio, cliquez avec le bouton droit sur le projet « Mapping » dans le Solution Explorer, sélectionnez **Propriétés** et observez la valeur affichée dans la liste déroulante **Target Framework**, par exemple :

Mapping 🕫 🗙			÷ \$
Application	Configuration: N/A	N/A	
Build		· Plation <u>ini</u> , INTA	
Build Events	Assembly <u>n</u> ame:	Defau <u>l</u> t namespace:	
Package	Mapping	Mapping	
Debug	Target framework:	O <u>u</u> tput type:	
Signing	.NET Core 3.1	<ul> <li>Console Application</li> </ul>	
Code Analysis	Startup <u>o</u> bject:		
Resources	(Not set)	~	

Le nom par défaut de l'application généré est **Mapping**. En option, vous pouvez le modifier ainsi que d'autres paramètres avant de générer le code depuis la boîte de dialogue <u>Paramètres de mappage</u><sup>112</sup>.

## 15.3.3 Générer du code depuis un projet de mappage

#### Pour générer du code depuis un projet de mappage (.mfp) :

- 1. Si vous ne l'avez pas encore fait, ouvrez le projet de mappage dans MapForce.
- 2. Cliquez avec la touche de droite dans la fenêtre de projet, puis cliquez sur Propriétés.

Project Settings		<b>—</b>
Project Name:	Project01	
Project Directory:	C:\Project01\	
Output Settings		
Output Name:	Project01	
Output Directory:	C:\Project01\output\	Browse
Language:	Java 🔻	
Java Settings		
B <u>a</u> se package name:	com.mapforce	
	OK	Cancel

3. Consultez et modifiez les paramètres de projet le cas échéant (en particulier, assurez-vous que le

langue cible et le répertoire de sortie sont définis correctement), puis cliquez sur OK.

4. Dans le menu **Projet**, cliquez sur Générer du code pour tout le projet.

Le progrès et le résultat du processus de génération de code (message de succès ou d'erreur) est affiché dans la fenêtre Messages.

Par défaut, le nom de l'application générée est le même que le nom du projet. Si le nom du projet contient des espaces, elles sont converties en traits de soulignement dans le code généré. Par défaut, le code est généré dans le même répertoire que le projet MapForce, dans le sous-répertoire **sortie**.

Pour changer le répertoire de sortie et le nom du projet, cliquer sur Projet dans la fenêtre Projet, puis sélectionner **Projet | Propriétés** depuis le menu. **Si votre projet de mappage contient des dossiers, vous pouvez modifier les paramètres de génération de code pour chaque dossier individuel (cliquer sur la touche de droite sur le dossier, puis sélectionner** Propriétés). Sinon, tous les dossiers de projet héritent des paramètres de projet MapForce.

## 15.3.4 Construire les projets .NET Framework

Si vous avez généré les cibles C# de la plate-forme .NET Framework, la manière la plus facile de construire le projet est le suivant :

1. Ouvrir le fichier de solution généré (.sln) dans Visual Studio.

Par défaut, le nom du fichier de solution est **Mapping.sln**, et il se trouve dans la sous-répertoire **Mapping** relatif au répertoire dans lequel vous avez enregistré le code généré. Si vous avez changé le nom d'application provenant des paramètres de mappage, puis le nom du fichier .sln est modifié conformément. Par exemple, si vous avez changé le nom d'application dans **MyApplication**, puis le fichier de solution est appelé **MyApplication**.

- 2. Dans le menu **Build**, cliquez sur **Manager de configuration**.
- 3. Choisir la configuration de build requise (déboguer, release).
- 4. Dans le menu **Build**, cliquez sur **Solution de build**.

### Exécuter l'application

Une fois que vous compilez le projet Visual Studio, une application de ligne de commande est produite, appelée **Mapping.exe**. Veuillez noter que si vous modifiez le nom de l'application de la boîte de dialogue <u>Paramètres de mappage</u><sup>(112)</sup>, le nom de l'exécutable est changé en conséquence.

Vous pouvez trouver l'application de mappage dans les sous-répertoires suivants relatifs au fichier .sln, dépendant l'option de build que vous avez choisi plus tôt :

- bin\Debug
- bin\Release

Pour exécuter l'application, ouvrez une invite de commande, modifiez le répertoire actuel vers le chemin du programme exécutable et exécutez-la, par exemple :

#### C:\codegen\DB\_CompletePOcs\Mapping\bin\Release>Mapping.exe Mapping Application Connecting to CustomersAndArticles database...

Finished

C:\codegen\DB\_CompletePOcs\Mapping\bin\Release>

## 15.3.5 Construire les projets .NET 5 et .NET Core

Pour construire et exécuter la solution générée .NET Core 3.1,.NET 5.0 ou .NET 6.0, vous devez avoir installé le SDK respectif. Pour le package à télécharger applicable à votre système d'exploitation et plate-forme, veuillez consulter le site Internet (<u>https://dotnet.microsoft.com/download</u>). Après avoir téléchargé et installé le package requis, vous pouvez facilement visualiser les SDK installés en saisissant la commande suivante à l'invite de commande :

dotnet --list-sdks

Si vous avez le SDK .NET Core 3.1, .NET 5.0 ou .NET 6.0, vous pouvez procéder à la construction du code généré, soit dans la ligne de commande ou dans Visual Studio. Les SDK .NET Core 3.1, .NET 5.0 et .NET 6.0 sont comptabiles avec Visual Studio 2019 (v16.7).

#### Construire à la ligne de saisie

Pour construire le code généré à la ligne de commande :

 Ouvrez une fenêtre d'invite de commande et modifiez le répertoire vers celui dans lequel le code de mappage a été généré. Par exemple, si vous sélectionnez C:\codegen\ConvertProducts en tant que répertoire cible et les paramètres de mappage par défaut, exécutez :

cd C:\codegen\ConvertProducts

2. Exécutez la commande suivante :

```
dotnet build Mapping\Mapping.sln --no-incremental --configuration Release --
verbosity normal --output Mapping\bin\release
```

**Note :** le chemin vers la solution de mappage et son nom peuvent être différents si vous avez modifié la propriété du **Nom de l'application** depuis la boîte de dialogue des <u>Paramètres de mappage</u><sup>112</sup>.

Le résultat d'un build réussi pourrait être semblable à celle ci-dessous :

c	Command Prompt	-		×
in	Copying file from "c:\codegen\ConvertProducts\Mapping\obj\Release\netco ng.pdb" to "c:\codegen\ConvertProducts\Mapping\bin\release\Mapping.pdb". 2>Done Building Project "c:\codegen\ConvertProducts\Mapping\Mapping.csproj"	reapp3 (Rebu	.1∖Map ild ta	р ^ г
(s	1>Done Building Project "c:\codegen\ConvertProducts\Mapping\Mapping.sln" (R ;)).	ebuild	ltarge	t
Bu	ild succeeded. θ Warning(s) θ Error(s)			
Ti	me Elapsed 00:00:12.09			
c:	\codegen\ConvertProducts>			v

### Construire avec Visual Studio

Pour construire le code de génération dans Visual Studio :

- 1. Double-cliquez sur **Mapping.sln** (ou le nom de solution personnalisé respectif) pour l'ouvrir dans Visual Studio.
- 2. Dans le menu Build, cliquez sur Gestionnaire de configuration.
- 3. Sélectionnez la configuration de build requise (déboguer, release) et la plate-forme de solution (x64, x86, tout CPU).
- 4. Dans le menu Build, sélectionnez la Solution de build (Ctrl+Shift+B).

### Exécuter le programme généré

Après avoir construit le code généré, un **Mapping.exe** (ou un programme avec un nom personnalisé, si configuré) est produit dans le répertoire de sortie du projet. Le chemin du répertoire de sortie dépend de la configuration de build que vous avez choisi avant (déboguer, release).

Si vous construisez le code de génération sur Linux, le programme exécutable généré sera simplement appelé **Mappage**, sans aucune extension.

Sur Windows, vous pouvez exécuter le programme en double-cliquant sur le programme exécutable dans Windows Explorer ou en appelant le programme exécutable depuis une fenêtre d'invite de la commande. Sur les autres plate-formes, vous allez devoir appeler le programme exécutable depuis le terminal avec une commande telle que :

#### ./Mapping

Veuillez noter que si le mappage a des références vers des fichiers d'entrée et de sortie sur le disque, le programme s'exécutera avec succès uniquement si tous les chemins peuvent être résolus. Par exemple, si vous avez configuré le mappage pour qu'un composant doit rechercher un fichier d'entrée dans le même répertoire utilisant un chemin relatif, alors le fichier respectif doit être présent dans le répertoire d'exécution du programme. Pour toute information sur les chemins de gestion à l'heure « mapping design time », voir <u>Utiliser</u> les chemins relatifs et absolus<sup>82</sup>.
# 15.4 Générer du code Java

Vous pouvez générer du code de programme pour Java SE JDK 8 ou 11 (y compris OpenJDK). Veuillez noter la chose suivante lorsque vous générez du code :

- Certaines fonctions MapForce ne sont pas prises en charge dans du code de programme généré. Pour plus de détails, voir <u>Fonctions prises en charge dans le code généré</u><sup>1476</sup>.
- Vous pouvez générer du code soit depuis un seul design de mappage (.mfd), ou depuis un projet de mappage (.mfp). Si vous générez du code depuis un seul mappage, l'application résultante exécute la transformation de mappage respective. Si vous générez du code depuis un projet MapForce (.mfp) qui contient plusieurs mappages, l'application résultante exécute en vrac tous les mappages contenus dans le projet.
- Vous pouvez modifier les options de génération de code générales depuis le menu Outils | Options, dans l'onglet Génération. Voir <u>Options du Générateur de Code</u><sup>(20)</sup>.
- Vous pouvez changer le nom de l'application de mappage générée et d'autres paramètres provenant du menu Fichier | Paramètres de mappage. Le nom d'application par défaut est Mappage. Voir aussi <u>Changer les paramètres de réseau</u><sup>(112)</sup>.
- Pour plus d'information concernant les chemins de handling dans le code généré, voir <u>Chemins dans</u> les différents environnements d'exécution<sup>(85)</sup>.
- Si votre mappage contient des composants de base de données, vous pouvez consulter des paramètres spécifiques à la base de données en cliquant sur un composant de base de données puis en choisissant l'option de menu Composant | Propriétés. Voir <u>Paramètres de composant de base de</u> <u>données</u><sup>281</sup>.
- La prise en charge pour les connexions de base de données varie selon les plateformes et il existe des types de connexion qui ne sont pas pris en charge sur toutes les plateformes. Si votre mappage se connexte à une base de données, choisir une connexion de base de données qui est compatible avec l'environnement cible pour lequel vous générez du code. Pour plus de détails, voir <u>Mappages de base de données dans des environnements d'exécution variés</u>

Un projet Java classique généré par MapForce contient les éléments suivants :

- Plusieurs packages signés Altova requis par le mappage (tout est préfixé avec com.altova).
- Le package **com.mapforce**, qui comprend l'application de mappage et les fichiers dépendants (comme indiqué ci-dessous, il est possible de changer le nom de ce package). Les deux fichiers les plus importants dans ce package sont les suivants :
  - L'application de mappage Java en tant qu'application de dialogue (MappingApplication.java).
  - o L'application de Java en tant qu'application de console (MappingConsole.java).
- Un fichier build.xml que vous pouvez exécuter avec Apache Ant pour compiler le projet et générer les fichiers JAR.



Exemple d'application Java générée par MapForce (Eclipse IDE)

Cette section inclut les rubriques suivantes :

- <u>Générer du code depuis un mappage</u><sup>1082</sup>
- Générer du code depuis un projet de mappage<sup>1083</sup>
- <u>Gérer des références JDBC</u>
- <u>Construire le projet avec Ant</u><sup>1084</sup>
- Exemple : Exécuter et compiler le code Java avec Eclipse et Ant

# 15.4.1 Générer du code depuis un mappage

Pour générer du code Java depuis le fichier design de mappage (.mfd) :

- 1. Revoir et sélectionner les options de générateur de code<sup>[201]</sup>.
- 2. Dans le menu Fichier, cliquer sur Générer code dans | Java.
- 3. Choisir un répertoire de destination pour les fichiers générés, puis cliquer sur **OK** pour confirmer. Le résultat de la génération de code (message d'erreur ou de succès) est affiché dans la fenêtre Messages.

Le nom par défaut de l'application générée est **Mapping**, et le nom par défaut du package de base est **com.mapforce**. Le cas échéant, vous pouvez les modifier à partir du dialogue <u>Paramètres de mappage</u><sup>(112)</sup>.

# 15.4.2 Générer du code depuis un projet de mappage

# Pour générer du code depuis un projet de mappage (.mfp) :

- 1. Si vous ne l'avez pas encore fait, ouvrez le projet de mappage dans MapForce.
- 2. Cliquez avec la touche de droite dans la fenêtre de projet, puis cliquez sur Propriétés.

Project Settings		×
Project Name:	Project01	
Project Directory:	C:\Project01\	
Output Settings		
Output Name:	Project01	
Output Directory:	C:\Project01\output\	Browse
<u>L</u> anguage:	Java 🔻	
Java Settings		
B <u>a</u> se package name:	com.mapforce	
	ОК	Cancel

- 3. Consultez et modifiez les paramètres de projet le cas échéant (en particulier, assurez-vous que le langue cible et le répertoire de sortie sont définis correctement), puis cliquez sur **OK**.
- 4. Dans le menu **Projet**, cliquez sur Générer du code pour tout le projet.

Le progrès et le résultat du processus de génération de code (message de succès ou d'erreur) est affiché dans la fenêtre Messages.

Par défaut, le nom de l'application générée est le même que le nom du projet. Si le nom du projet contient des espaces, elles sont converties en traits de soulignement dans le code généré. Par défaut, le code est généré dans le même répertoire que le projet MapForce, dans le sous-répertoire **sortie**.

Pour changer le répertoire de sortie et le nom du projet, cliquer sur Projet dans la fenêtre Projet, puis sélectionner **Projet | Propriétés** depuis le menu. **Si votre projet de mappage contient des dossiers, vous pouvez modifier les paramètres de génération de code pour chaque dossier individuel (cliquer sur la touche de droite sur le dossier, puis sélectionner** Propriétés). Sinon, tous les dossiers de projet héritent des paramètres de projet MapForce.

# 15.4.3 Construire le projet avec Ant

Apache Ant est une bibliothèque (et outil de ligne de commande) Java utilisée communément qui automatise la construction et la compilation de projets Java (voir <u>http://ant.apache.org/manual/</u>). Ant fonctionne avec des fichiers build (ces fichiers définissent les sources et les cibles à partir desquels le code doit être compilé, ainsi que toutes options de build spécifiques). Étant donné que tout projet généré par MapForce inclut un fichier **build.xml** reconnu par Ant, vous pouvez facilement construire des projets générés par MapForce avec Ant.

Ant peut être disponible sur votre système soit en tant qu'installation autonome, ou réuni avec Eclipse (ou d'autres IDE Java). Pour consulter des instructions pour apprendre comment installer Ant sur votre système, voir <u>http://ant.apache.org/manual/</u>. Pour consulter des instructions pour apprendre comment utiliser Ant dans Eclipse, voir la documentation Eclipse.

Vous pouvez vérifier rapidement si la version autonome d'Ant (pas celui lié avec Eclipse) est disponible sur votre système en ouvrant une invite de commande et en saisissant ant dans la ligne de commande. Lorsqu'Ant est déjà disponible, le message résultat sera semblable à : Buildfile: buildxml does not exist! Ce message indique qu'Ant est installé et il tente de construire un fichier **build.xml** mais ce dernier n'existe pas dans le répertoire actuel. Si vous exécutez Ant depuis un répertoire qui inclut un fichier **build.xml**, Ant exécute le fichier **build.xml** à la place, avec n'importe quelle options de build y est définie.

#### Pour construire un projet Java généré par MapForce avec Ant :

- 1. Ouvrir une invite de commande et naviguer vers le répertoire dans lequel le projet Java a été généré (veuillez noter que le répertoire doit contenir le fichier **build.xml**).
- 2. Dans l'invite de commande, saisir ant. Cela compilera et exécutera le code Java conformément aux options définies dans le fichier **build.xml**, par exemple :



#### Pour générer un fichier JAR avec Ant :

• Dans l'invite de commande, saisir ant jar.

Si vous avez besoin d'aide avec la syntaxe et les options de commande Ant, saisir ant -help dans la ligne de commande. Pour consulter un exemple, voir <u>Exemple : Générer et exécuter du Code Java</u>

# **15.4.4 Gérer les références JDBC**

Si le mappage se connecte à une base de données par le biais de JDBC, s'assurer que les pilotes JDBC utilisés par le mappage sont installés sur votre système. Pour consulter les paramètres JDBC actuels de tout composant de base de données, cliquer dessus, puis sélectionner **Component | Properties** depuis le menu. Pour plus d'informations, voir <u>Créer une connexion JDBC</u><sup>(12)</sup>.

Si le mappage utilise une connexion de base de données non-JDBC, la connexion peut être convertie en JDBC pendant la génération de code Java, pour proposer une compatibilité dans un environnement Java. Par exemple, des connexions ADO, ADO.NET et ODBC sont converties dans des connections JDBC lorsque vous générez du code Java depuis le mappage. Pour plus de détails, voir <u>Mappages de base de données dans des environnements d'exécution variés</u><sup>157</sup>.

Si vous effectuez un test de l'application générée dans Eclipse, le pilote JDBC peut devoir être ajouté en tant qu'entrée "classpath" dans l'exécution actuelle de la configuration. Sinon, la marche de test de l'application dans Eclipse peut échouer avec un message d'erreur comme : java.lang.ClassNotFoundException: com.mysql.jdbc.Driver. En prenant la base de données MySQL en tant qu'un exemple, vous pouvez ajouter le pilote JDBC en tant que dépendance dans Eclipse comme suit :

- 1. Générer le code Java depuis MapForce et importer le projet dans Eclipse en utilisant la commande de menu File | Import | Existing Projects into Workspace.
- 2. Dans le menu **Run** cliquer sur **Run Configurations**. Dans la partie gauche de la fenêtre, sour "Java Application", vous trouverez deux configurations appelées **MappingApplication** et **MappingConsole**.
- 3. Cliquer sur la configuration désirée (par exemple, **MappingApplication**).
- 4. Dans l'onglet Dependencies, cliquer sur Classpath entries, plus sur Add External JARs.
- 5. Chercher le fichier .jar du pilote JDBC, par exemple :



6. Cliquer sur **Run** pour exécuter le programme avec le pilote JDBC de base de données en tant que dépendance.

Si vous obtenez l'erreur de pilote JDBC pendant la marche de test du fichier Ant **build.xml**, ajouter le pilote JDBC dans le classpath de la tâche "test". L'exemple suivant montre une tâche "test" d'Ant qui inclut une référence au fichier .jar du pilote JDBC.

```
<target name="test" depends="compile">
<java classpath="C:\codegen\java\mysql_mapping"
classname="com.mapforce.MappingConsole" fork="true" failonerror="true">
<classpath>
<pathelement path="${classpath}"/>
<pathelement path="${classpath}"/>
<pathelement location="C:\jdbc\mysql\mysql-connector-java-5.1.16-bin.jar"/>
</classpath>
<arg line="${cmdline}"/>
</java>
</target>
```

Note: L'exemple ci-dessus part du principe que le fichier .jar existe sous le chemin suivant : C: \jdbc\mysql\mysql-connector-java-5.1.16-bin.jar. Assurez-vous de changer le chemin du fichier .jar selon votre environnement.

#### Inclure des pilotes JDBC dans le manifest de l'application

Si vous construisez des fichiers JAR depuis l'application Java générée, ajouter lune référence au pilote de base de données dans la section "manifest" du fichier **build.xml**. Cela garantit que la référence au pilote de base de données est disponible dans le fichier manifest (**MANIFEST.MF**) une fois que vous avez construit le projet.

#### Pour ajouter la référence au fichier manifest :

- 1. Situer l'élément manifest dans le fichier build.xml,
- Ajouter un nouvel élément appelés attribute, sachant que l'attribut "name" est "Class-Path" et que l'attribut "value" est le nom du fichier .jar. Par exemple, pour MySQL 5.1.16, le nouvel élément pourrait ressembler à :

<attribute name="Class-Path" value="mysql-connector-java-5.1.16-bin.jar"/>

L'élément **manifest** du fichier **build.xml** ressemble maintenant à la capture d'écran ci-dessous.

```
<manifest file="C:\codegen\java\mysql_mapping/META-INF/MANIFEST.MF"
mode="replace">
    <attribute name="Created-By" value="MapForce 2023"/>
    <attribute name="Main-Class" value="com.mapforce.MappingConsole"/>
    <attribute name="Class-Path" value="mysql-connector-java-5.1.16-bin.jar"/>
</manifest>
```

3. Copier le fichier JAR du pilote JDBC dans le dossier qui contient le fichier JAR de l'application générée.

# 15.4.5 Exemple : Générer et exécuter du code Java

Cet exemple vous guide à travers les étapes nécessaires pour générer une application Java avec MapForce, et la compiler en dehors de MapForce en utilisant Eclipse Integrated Development Environment (IDE) et Apache Ant. Après avoir achevé cet exemple, vous aurez créé et compilé une application Java complète qui exécute un des échantillons de mappage disponibles par défaut dans MapForce.

Si vous êtes déjà en mesure de compiler avec succès d'autres applications Java avec Eclipse et Ant, vous n'aurez à respecter aucune exigence particulière pour exécuter cet exemple. Sinon, veuillez noter les exigences préalables suivantes :

- Installer le Java Development Kit (JDK), Eclipse et Ant sur votre système. Eclipse contient généralement une version bundled d'Ant, mais vous pouvez aussi installer Ant séparément, voir aussi <u>Construire le projet avec Ant</u>
- Pour exécuter Eclipse avec OpenJDK, la variable d'environnement PATH exige généralement l'inclusion du chemin dans le répertoire bin JDK (par exemple, C:\Java\jdk-11.0.1\bin).
- Si vous exécutez Apache Ant standalone comme dans cet exemple, la variable d'environnement PATH doit aussi inclure le chemin vers le répertoire bin (par exemple, C:\apache-ant-1.10.5\bin), afin de pouvoir exécuter Ant sans de manière confortable sans devoir saisir le chemin complet vers le programme d'exécution depuis la ligne de commande.

Cet exemple utilise la configuration suivante :

- MapForce 64-bit
- OpenJDK 11.0.1 64-bit
- Eclipse IDE for Java Developers, version 2019-03 (4.11.0), 64-bit
- Apache Ant 1.10.6 (installé en standalone) et Apache Ant 1.10.5 (réuni avec Eclipse)

# Étape 1 : Générer du code Java

Pour générer du code Java dans MapForce :

- 1. Dans le menu Fichier, cliquer sur Ouvrir, et naviguer dans le mappage CompletePO.mfd disponible dans le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\.
- 2. Dans le menu **Sortie**cliquer sur **Java**. Cela modifie le langage de transformation sur Java.
- Dans le menu Fichier, cliquer sur Générer code dans | Java. Sur invitation, chercher le répertoire sur lequel le projet Java doit être enregistré. En ce qui concerne cet exemple, vous pouvez générer du code Java dans C:\codegen\java\CompletePO. Vous devrez vous référer à ce chemin dans des étapes suivantes.

# Étape 2 : Importer le projet dans Eclipse

Pour importer le projet dans Eclipse :

- 1. Si vous ne l'avez pas encore fait, exécutez Eclipse et passez à la perspective Java par défaut en utilisant la commande de menu **Window | Open Perspective**.
- 2. Dans le menu File, cliquer sur Import, puis sélectionner General | Existing Projects into Workspace.

lmport			×
Select Create new projects from an archive file or directory.		Ľ	5
Select an import wizard:			
<ul> <li>Constraints</li> <li></li></ul>			<b>^</b>
	1	Cance	1

3. Cliquer sur Next.

🖨 Import						×
Import Projects Select a directory to search	ch for existin	g Eclipse proje	cts.			
<ul> <li>Select root directory:</li> <li>Select archive file:</li> <li>Projects:</li> </ul>	C:\codegen	\java\Comple	tePO .		B <u>r</u> owse	
✓ Mapping (C:\code	egen\java\Co	ompletePO)		[	<u>S</u> elect A <u>D</u> eselect R <u>e</u> fresl	All All
Options Searc <u>h</u> for nested pro <u>C</u> opy projects into we Cl <u>o</u> se newly imported H <u>i</u> de projects that alr	jects orkspace d projects up eady exist in	on completior the workspace	1			
Working sets	ing sets		~		Ne <u>w</u> S <u>e</u> lect	
? < E	ack	<u>N</u> ext >	<u>F</u> inish		Cance	I

4. Chercher le dossier dans lequel vous avez précédemment enregistré le code généré, puis cliquer sur Finish. Le projet Java créé par MapForce est maintenant disponible dans le mode Package Explorer. Si vous ne pouvez pas consulter le mode Package Explorer, l'afficher en utilisant la commande de menu Window | Show View | Package Explorer.



# Étape 3 : Exécuter le projet en tant qu'application de dialogue

Pour exécuter le projet Java en tant qu'application GUI :

- 1. Dans le mode Package Explorer d'Eclipse, cliquer sur le fichier **MappingApplication.java** disponible dans le package **com.mapforce**.
- 2. Dans le menu Run, cliquer sur Run As | Java application.
- 3. Dans la fenêtre d'application MapForce, cliquer sur Start pour exécuter le mappage.

Mapforce Application			_	×
THIS APPLICATION I http://www.altova.c Please check the inp	NAS GENERATED BY MapForce 2020r2 om/mapforce ut and output files, and press the Star	t button		
Source instance of Customers.xsd:				^
C:/Users/altova/Documents/Altova	MapForce2020/MapForceExamples/Cu	ustomers.xml		
Source instance of Articles.xsd:				
C:/Users/altova/Documents/Altova	MapForce2020/MapForceExamples/Ar	ticles.xml		
Source instance of ShortPO.xsd:				
C:/Users/altova/Documents/Altova	MapForce2020/MapForceExamples/Sh	nortPO.xml		
Instance of CompletePO.xsd:				
CompletePO.xml				
				~
Start				
Started				>
Loading com.altova.io.F	ileInput@2d6a94e7			
Loading com.altova.io.F	ileInput@76a8c72d			
Loading com.altova.io.F	ileInput@1be209af			
Finished				
L				*

Si Eclipse rencontre la configuration de système ou des erreurs de run-time, vous serez notifié, voir aussi <u>Gérer</u> <u>des références JDBC</u>. Sinon, l'application Java exécute la transformation de mappage et génère **CompletePO.xml** dans le chemin de sortie (dans cet exemple le chemin de sortie est : **C: \codegen\Java\CompletePO**).

# Étape 4 : Exécuter le projet en tant qu'application de console

Pour exécuter le projet Java en tant qu'application de console :

- 1. Dans le mode Package Explorer d'Eclipse, cliquer sur le fichier **MappingConsole.java** disponible dans le package **com.mapforce**.
- 2. Dans le menu Run, cliquer sur Run As | Java application.



Si Eclipse détecte des erreurs de configuration de système ou de durée d'exécution, vous en serez informé voir aussi <u>Gérer des références JDBC</u><sup>(103)</sup>. Sinon, l'application Java exécute la transformation de mappage et génère **CompletePO.xml** dans le chemin de sortie (dans cet exemple le chemin de sortie est : **C:** \codegen\Java\CompletePO).

# Étape 5 : Construire le fichier JAR

Si Apache Ant est réuni avec Eclipse, vous pouvez construire le fichier JAR directement depuis l'IDE Eclipse. Sinon, il est nécessaires d'installer Apache Ant séparément pour pouvoir construire le fichier JAR dans la ligne de commande en dehors d'Eclipse.

#### Pour construire le fichier .jar avec Ant dans Eclipse :

 Dans le mode Package Explorer d'Eclipse, cliquer avec la touche de droite sur le fichier build.xml et pointer vers le menu Run As menu. Dans le menu Run As deux options possibles pour exécuter le fichier de build Ant sont affichées. Si vous choisissez la première option, Eclipse lance le build Ant avec les paramètres par défaut. Si vous choisissez la seconde option, vous pouvez changer les paramètres du build Ant avant de le lancer. Choisissez la seconde option.

0	Run As	>	湪	1 Ant Build	Alt+Shift+X, Q
*	Debug As	>	湪	2 Ant Build	
	Team	>		External Tools Configurations	
	Compare With	>	-	-	

2. Cliquer pour activer les cibles que vous souhaitez inclure dans le build Ant. Dans cet exemple, les cibles **test** et **jar** sont sélectionnées.

Edit Configuration	_	D X	
Edit configuration and launch.		0_	
Run an Ant build file.			
			_
Name: Mapping build.xml			]
📄 Main 🤣 Refresh 🔂 Build 🦧 Tar	gets 🛛 🍫 Classpath 💷 Properties 🛋 JRE 📧 Environment 🔲	Common	1
Check targets to execute:			
Name	Description		
🗌 回 🎯 compile			
🔽 👰 test [default]			
🗌 🖲 clean			
Manifest			
🗹 🛞 jar			
2 out of 5 selected			
Hide internal targets not selected for	execution		
larget execution order:			
test, jar		<u>O</u> rder	
	✓		
			1
	Re <u>v</u> ert	Apply	
?	Run	Close	

3. Cliquer sur **Run**. Eclipse exécute le fichier build Ant et affiche le résultat dans le mode Console.

#### Pour construire le fichier JAR dans la ligne de commande :

- 1. Assurez-vous que le répertoire dans lequel Apache Ant est installé (par exemple, **C:\apache-ant-1.10.5\bin**) est ajouté à la variable d'environnement PATH.
- Ouvrir une fenêtre d'invite de commande et changer le répertoire actuel vers celui dans lequel vous avez généré du code Java (dans cet exemple, C:\codegen\java\CompletePO). Il s'agit aussi du répertoire dans lequel le fichier build.xml a été généré.
- 3. Dans la ligne de commande, saisir :

ant clean j	ar
-------------	----

Ant exécute les cibles "clean" et "jar" depuis le fichier **build.xml** (y compris toutes cibles dépendantes), et affiche la sortie dans la ligne de commande :

Command Prompt	—	×
c:\codegen\java\CompletePO≻ant clean jar Buildfile: c:\codegen\java\CompletePO\build.xml		Â
clean:		
compile: [javac] Compiling 94 source files		
manifest:		
jar: [jar] Building jar: c:\codegen\java\CompletePO\Mapp	ing.jar	
BUILD SUCCESSFUL Total time: 3 seconds		
c:\codegen\java\CompletePO>		~

# 15.5 Intégrer du code généré par MapForce

Un code généré par MapForce peut être intégré, ou adapté dans votre application spécifique, bien que le résultat de la génération de code est une application complète et entièrement fonctionnelle. Voici quelques scénarios typiques dans lequel vous pourriez changer le code généré :

- Définir les fichiers de source ou cible personnalisés pour l'application de mappage
- Ajouter un code de gestion d'erreur personnalisé
- Dans du code généré C# ou Java, vous pouvez aussi modifier le type de données de l'entrée de mappage par le biais d'un programme (par exemple, de string à stream).

Cette section fournit des instructions comment atteindre ces objectifs, sur la base du mappage d'échantillon DB\_CompletePO.mfd disponible dans le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\.



DB\_CompletePO.mfd échantillon de mappage dans MapForce

Comme illustré ci-dessus, le mappage d'échantillon consiste en deux sources et un cible :

- ShortPO.xml est un fichier XML de source
- CustomersAndArticles.mdb est une base de données de source
- CompletePO.xml est la cible du fichier XML.

Dans le code généré, ces sources et ces cibles se traduiront en deux entrées et un paramètre de sortie fournis dans la méthode run qui exécute le mappage (comme décrit dans les rubriques suivantes). Pour l'instant, veuillez noter les points de base suivants concernant la génération de code :

- Le nombre de sources et de cibles dans le design de mappage corresponde au nombre des paramètres de mappage dans la méthode **run** dans le code généré.
- Si vous changez le nombre de sources ou de cibles du mappage, vous devrez régénérer le code en conséquence.
- Si vous modifiez le code généré, puis régénérez le code dans le même emplacement, tous les changements seront écrasés.

Si un mappage contient des composants de base de données, la méthode générée <u>run</u> inclut l'objet de connexion de base de données dans l'endroit approprié. Par exemple, si le mappage utilise trois sources (contenu de texte, contenu XML et une base de données) pour mapper vers un seul fichier de sortie, MapForce génère la méthode suivante <u>run</u>:

#### Java

```
void run(Input in1, Input in2, java.sql.Connection dbConn, Output out1);
```

L'ordre de l'argument est important. Comme vous le verrez dans les exemples suivants, vous pouvez modifier des paramètres dbConn, ou utiliser les paramètres par défaut générés par MapForce lors de l'intégration de votre code.

# 15.5.1 Exemple Java

Cet exemple utilise Eclipse en tant que IDE Java. Pour commencer, générer du code Java depuis le mappage d'échantillon **DB\_CompletePO** dans le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\, puis importer le projet dans Eclipse.



Exemple d'application Java générée par MapForce (Eclipse IDE)

Pour éditer l'application de console Java générée, situez la méthode main(String[] args) de votre application générée (voir la capture d'écran ci-dessus). Si vous n'avez pas modifié le nom de package de base par défaut avant de générer le code, cette méthode se trouve dans la classe MappingConsole du package com.mapforce. Sinon, il se trouve dans la classe MappingConsole de votre package défini personnellement.

Pour éditer l'application de dialogue Java généré, situez l'endroit dans le code dans lequel la méthode run est invoquée depuis votre application générée. Si vous n'avez pas modifié le nom de package de base par défaut avant de générer du code, la méthode run est invoquée depuis la classe appelée MappingFrame.java du package com.mapforce.

L'échantillon de code suivant illustre un extrait depuis la méthode principale dans l'application de console Java générée. Les sources et les cibles de mappage sont marqués en jaune et son définis en tant que paramètres dans la méthode run. Puisque ce mappage utilise une connexion de base de données, le paramètre correspondant dispose d'une structure spéciale. Concrètement, la connexion consiste dans le string de connexion (dans ce cas, jdbc:odbc:;DRIVER=Microsoft Access Driver

(\*.mdb);DBQ=CustomersAndArticles.mdb), ainsi que deux arguments vides conçus pour le **Nom** d'utilisateur et **Mot de passe** (en texte clair) pour les base de données où ces données sont nécessaires.

Veuillez noter que les chemins de fichier dans le code ci-dessous ont été modifiés d'absolu à relatif.

```
com.altova.io.Input ShortPO2Source =
com.altova.io.StreamInput.createInput("ShortPO.xml");
```

#### Pour définir des fichiers source ou cible de mappage personnalisé :

• Situer les paramètres passés dans la méthode run et les éditer comme exigé. Dans l'échantillon cidessus, com.altova.db.Dbs.newConnection et ShortPO2Source est l'entrée de mappage et CompletePO2Target est la sortie de mappage.

#### Pour ajouter un code de gestion d'erreur supplémentaire :

- Éditer le code sous le code catch (Exception e) (dans le cas d'une application de console Java)
- Éditer le code sous le code catch (Exception ex) (dans le cas d'une application de dialogue Java)

Pour consulter des instructions sur la manière de changer le type de données des paramètres fournis en tant qu'entrée/sortie de mappage, voir <u>Changer l'entrée et la sortie de manière programmatique</u><sup>1101</sup>.

# 15.5.2 Exemple C#

Cet exemple utilise l'IDE Visual Studio 2010. Pour commencer, générez du code C# depuis le mappage d'échantillon **DB\_CompletePO** dans le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\, puis ouvrez la solution dans Visual Studio.

Par défaut, le nom du fichier de solution est **Mapping.sln**, et il se trouve dans la sous-répertoire **Mapping** relatif au répertoire dans lequel vous avez enregistré le code généré. Si vous avez changé le nom d'application provenant des paramètres de mappage, puis le nom du fichier .sln est modifié conformément. Par exemple, si vous avez changé le nom d'application dans **MyApplication**, puis le fichier de solution est appelé **MyApplication**.



Exemple de solution C# générée avec MapForce

Ouvrez le fichier MappingConsole.cs, et situez la méthode main(String[] args. L'échantillon de code suivant illustre un extrait depuis la méthode principale. Les sources et les cibles de mappage sont définies en tant que paramètres dans la méthode Run. Puisque ce mappage lit des données depuis une base de données, il y a aussi un paramètre d'entrée qui est un string de connexion de base de données. Si nécessaire, vous pouvez modifier le string de connexion de la base de données.

Veuillez noter que les chemins de fichier dans le code ci-dessous ont été modifiés d'absolu à relatif.

#### Pour définir des fichiers source ou cible de mappage personnalisé :

• Situez les paramètres passés dans la méthode Run et les éditer comme exigé. Dans l'échantillon cidessus, l'entrée de mappage est un string de connexion dans **CustomersAndArticles.mdb** et ShortPO2Source. La sortie de mappage est CompletePO2Target.

#### Pour ajouter un code de gestion d'erreur supplémentaire :

• Éditer le code situé sous le code catch (Exception e)

Pour plus d'informations concernant la modification du type de données des paramètres fournis en tant qu'entrée/sortie de mappage, voir <u>Modifier l'entrée et la sortie par programmation</u><sup>[10]</sup>.

# 15.5.3 Exemple C++

Cet exemple utilise l'IDE Visual Studio 2010. Pour commencer, générer du code C++ depuis le mappage d'échantillon **DB\_CompletePO** dans le répertoire <Documents>\Altova\MapForce2023\MapForceExamples\, puis ouvrir la solution dans Visual Studio.

Par défaut, le nom du fichier de solution est **Mapping.sln**, et il se trouve dans la sous-répertoire **Mapping** relatif au répertoire dans lequel vous avez enregistré le code généré. Si vous avez changé le nom d'application provenant des paramètres de mappage, puis le nom du fichier .sln est modifié conformément. Par exemple, si vous avez changé le nom d'application dans **MyApplication**, puis le fichier de solution est appelé **MyApplication**.



Exemple de solution C++ générée avec MapForce

Ouvrir le fichier **Mapping.cpp**, et situer la méthode \_tmain. L'échantillon de code suivant illustre un extrait depuis cette méthode. Les sources et les cibles de mappage sont définis en tant que les paramètres dans la méthode Run. Puisque ce mappage lit des données depuis une base de données, il y a aussi un paramètre d'entrée qui est un string de connexion de base de données. Si nécessaire, vous pouvez modifier le string de connexion de la base de données.

Veuillez noter que les chemins de fichier dans le code ci-dessous ont été modifiés d'absolu à relatif.

```
MappingMapToCompletePO MappingMapToCompletePOObject;
MappingMapToCompletePOObject.Run(
    _T("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=CustomersAndArticles.mdb;
"),
    _T("ShortPO.xml"),
    _T("CompletePO.xml"));
```

### Pour définir des fichiers source ou cible de mappage personnalisé :

Situer les paramètres passés dans la méthode Run et les éditer comme exigé. Dans l'échantillon de code ci-dessus, l'entrée de mappage est un string de connexion dans la base de données
 CustomersAndArticles.mdb et \_T("shortP0.xml") . La sortie de mappage est \_T("CompleteP0.xml") .

#### Pour ajouter un code de gestion d'erreur supplémentaire :

• Éditer le code sous le code catch (CAltovaException& e).

# **15.5.4 Modifier l'entrée et la sortie par programmation**

Après avoir généré Java ou le code C# avec MapForce, vous pouvez optionnellement modifier le type de données de l'entrée ou la sortie de mappage en éditant le code généré. Plus spécifiquement, vous pouvez utiliser des objets de paramètres de mappage de types différents par rapport à ceux générés par défaut. Par exemple, au lieu d'avoir le mappage qui lit l'entrée d'un fichier sur le disque, vous pouvez fournir un string ou un objet flux en tant qu'entrée. Veuillez noter que cette fonction est spécifique au code généré dans C# ou uniquement Java.

Les types d'objet pris en charge en tant qu'entrée ou sortie sont recensés dans la première colonne de la table ci-dessous. Chaque colonne suivante spécifie des formats de données dans lesquels ce type spécifique est pris en charge. Pour une définition plus précise de chaque type, voir la section « Définitions de type » ci-dessous.

	XML	JSON*	Microsoft Excel*	EDI (inclut X12, HL7)*	FlexText*	CSV/Text
Fichiers	Oui	Oui	Oui	Oui	Oui	Oui
Flux	Oui	Oui	Oui	Oui	Oui	Oui
Strings	Oui	Oui	_	Oui	Oui	Oui
Lecture/Écri ture	Oui	Oui	-	Oui	Oui	Oui
Documents DOM	Oui	-	-	-	-	-

\* Les formats ne pris en charge que dans MapForce Enterprise Edition

#### Pour modifier le type de données de l'entrée ou la sortie de mappage :

- 1. Générez le code C# ou Java depuis le mappage.
- Dans le code généré, trouvez l'appel de la méthode <u>run</u> (dans Java) ou la méthode <u>Run</u> (dans C#), comme suit :
  - a. Si vous utilisez C#, ouvrez le fichier MappingConsole.cs.
  - b. Si vous utilisez Java, ouvrez **MappingConsole.java** (le programme de console) ou le fichier **MappingFrame.java** (le programme GUI).
- **Note :** le nom du fichier peut être différent si vous avez modifié le nom de l'application dans les <u>paramètres de</u> <u>mappage</u><sup>112</sup>. Par exemple, si vous l'avez modifié en « MyApp », alors le nom du fichier généré devient respectivement **MyAppConsole.js** et **MyAppConsole.java** et **MyAppFrame.java**.
  - 3. Créez une instance du type requis (voir la section « Définitions de type »).
  - 4. Fournissez les objets déclarés en tant que paramètres à la méthode **run** (dans Java) ou la méthode **Run** (dans C#), tel qu'affiché dans les exemples ci-dessous.

La méthode **run** est la méthode la plus importante des classes de mappage générées. Elle a un paramètre pour chaque source *statique* ou composant d'entrée dans le mappage, et un paramètre final pour le composant de sortie. Si votre mappage contient des composants qui traitent de multiples fichiers de manière <u>dynamique</u><sup>633</sup>, les paramètres respectifs n'apparaissent pas dans le code généré parce que, dans ce cas, les noms de fichier sont traités de manière dynamique à l'intérieur du mappage.

## Définitions de types

Dans C#, les types que vous pouvez fournir en tant que paramètres à la méthode Run sont des classes définies dans l'espace de nom Altova.IO. Les classes de base sont respectivement Altova.IO.Input et Altova.IO.Output.

Types (	C#
---------	----

Fichiers	Altova.IO.FileInput(string filename) Altova.IO.FileOutput(string filename)
Flux	Altova.IO.StreamInput(System.IO.Stream stream) Altova.IO.StreamOutput(System.IO.Stream stream)
Strings	Altova.IO.StringInput(string content) Altova.IO.StringOutput(System.Text.StringBuilder sbuilder)
Lecture/Écriture	Altova.IO.ReaderInput(System.IO.TextReader reader) Altova.IO.WriterOutput(System.IO.TextWriter writer)
Documents DOM	Altova.IO.DocumentInput(System.Xml.XmlDocument document) Altova.IO.DocumentOutput(System.Xml.XmlDocument document)

Dans Java, les types que pouvez fournir en tant que paramètres à la méthode **run** sont des classes définies dans le package com.altova.io. Les classes de base sont respectivement com.altova.io.Input et com.altova.io.Output.

<i>I ypes Java</i>
--------------------

Fichiers	<pre>com.altova.io.FileInput(String filename) com.altova.io.FileOutput(String filename)</pre>
Flux	com.altova.io.StreamInput(java.io.InputStream stream) com.altova.io.StreamOutput(String filename)
Strings	<pre>com.altova.io.StringInput(String content) com.altova.io.StringOutput()</pre>
Lecture/Écriture	com.altova.io.ReaderInput(java.io.Reader reader) com.altova.io.WriterOutput(java.io.Writer writer)
Documents DOM	<pre>com.altova.io.DocumentInput(org.w3c.dom.Document document) com.altova.io.DocumentOutput(org.w3c.dom.Document document)</pre>

# Exemple

Pour illustrer la modification de l'entrée et de la sortie par programmation, nous allons utiliser le mappage **ConvertProducts.mfd** en tant que modèle. Après avoir installé et exécuté MapForce au moins une fois, vous trouverez ce mappage dans le répertoire suivant : **C**:

\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\Tutorials.



#### ConvertProducts.mfd

Tel qu'illustré ci-dessus, le mappage convertit les données depuis un document XML source en un autre document XML. Nos objectifs sont les suivants :

- 1. Générer Java et le code de programme C# depuis ce mappage.
- 2. Changer le type de données du composant source en type de string.
- 3. Changer le type de données du compostant cible en type de string writer.

Pour générer le code de programme, ouvrez le mappage **ConvertProducts.mfd** et exécutez la commande **Fichier | Générer code dans | C#** (ou **Java**). Pour la portée de l'exemple, nous parterons du fait que les paramètres de mappage de **ConvertProducts.mfd** sont ceux par défaut.

😢 Paramètres de mappag	e	×
Génération de code Nom d'application :	Mapping	
Nom pack de base Java :	com.mapforce	
Rendre les chemins absolus dans le code généré		
Bibliothèques de référence avec des chemins relatifs aux fichiers XSLT / XQuery générés		
Assurer la convention de système de fichier local	e chemin Windows pour la sortie de chemin de fichier pour les fichiers d'un	

Cet exemple utilise les répertoires cibles suivants pour le code généré (modifiez le chemin, si nécessaire) :

- C:\codegen\cs\ConvertProducts, pour C#
- C:\codegen\java\ConvertProducts, pour Java

Après avoir généré le code de programme, ouvrez **MappingConsole.cs** (dans C#) ou **MappingConsole.java** (dans Java) et trouvez les lignes suivantes :

#### C#

```
Altova.IO.Input Products2Source = Altova.IO.StreamInput.createInput("Products.xml");
Altova.IO.Output ProductValuePairs2Target = new
Altova.IO.FileOutput("ProductValuePairs.xml");
```

#### Java

```
com.altova.io.Input Products2Source =
com.altova.io.StreamInput.createInput("Products.xml");
com.altova.io.Output ProductValuePairs2Target = new
com.altova.io.FileOutput("ProductValuePairs.xml");
```

Commentez les lignes ci-dessus et modifiez le code comme suit :

#### C#

```
.....
                                               <id>100</id>\r\n" +
                                        н.
                                               <color>blue</color>\r\n" +
                                               <size>XXL</size>\r\n" +
                                        н.
                                            </product>r\n" +
                                        "</products>\r\n");
System.IO.StringWriter writer = new System.IO.StringWriter(new
System.Text.StringBuilder());
Altova.IO.Output ProductValuePairs2Target = new Altova.IO.WriterOutput(writer);
try
{
    MappingMapToProductValuePairsObject.Run(Products2Source, ProductValuePairs2Target);
    // Print out the writer object
    Console.Write(writer.ToString());
finally
{
    Products2Source.Close();
    ProductValuePairs2Target.Close();
}
```

Java

```
//com.altova.io.Input Products2Source =
com.altova.io.StreamInput.createInput("Products.xml");
//com.altova.io.Output ProductValuePairs2Target = new
com.altova.io.FileOutput("ProductValuePairs.xml");
com.altova.io.Input Products2Source = new com.altova.io.StringInput("<?xml
version=\"1.0\" encoding=\"UTF-8\"?>\r\n" +
               "products xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\"products.xsd\">\r\n" +
               н. –
                  <product>\r\n" +
               н.
                     <id>100</id>\r\n" +
               н.
                     <color>blue</color>\r\n" +
               н.
                      <size>XXL</size>\r\n" +
               n.
                  </product>\r\n" +
               "</products>\r\n");
java.io.StringWriter writer = new java.io.StringWriter();
com.altova.io.Output ProductValuePairs2Target = new com.altova.io.WriterOutput(writer);
try {
   MappingMapToProductValuePairsObject.run(Products2Source, ProductValuePairs2Target);
   // Print out the writer object
   System.out.print(writer.toString());
   } finally {
      (Products2Source).close();
      ProductValuePairs2Target.close();
}
```

Dans les listes de code C# et Java ci-dessus, la chose suivante se passe :

- Les deux lignes originales qui fournissent l'entrée et la sortie à la méthode <u>run</u> ont été commentées.
   En conséquence, l'application de mappage ne lit plus les données depuis **Products.xml**. En fait, nous n'avions même pas eu besoin de copier ce fichier dans le répertoire de travail du programme.
- Le type Products2Source a été déclaré en tant que StringInput qui fournit le contenu du fichier XML d'entrée à traiter.
- Le type ProductValuePairs2Target a été déclaré en tant que type WriterOutput qui prend un string writer comme argument.
- Une fois que le mappage a été exécuté, les contenus du string writer sont imprimés sur la console.

#### Orientations d'utilisation pour les flux et objets Lecture/Writer

Lorsque vous utilisez des flux binaires ou objets Lecture/Writer en tant qu'entrée ou sortie du mappage, veuillez noter les points suivants :

- Les objets flux binaires et objets Lecture/Writer sont prévus être ouverts et prêts à l'utilisation avant d'appeler la méthode <u>run</u>.
- Par défaut, la méthode **run** ferme le flux quand elle a terminé. Pour prévenir ce comportement, insérez (ou décommentez) la ligne suivante avant d'appeler la méthode **run** :

```
C#
```

```
MappingMapToSomething.CloseObjectsAfterRun = false;
```

Java

MappingMapToSomething.setCloseObjectsAfterRun(false);

Note : veillez à modifier MappingMapToSomething vers le nom de l'objet de mappage, si applicable, à votre code généré.

#### Orientations d'utilisation pour les strings

Dans Java, le constructeur de *StringOutput* ne prend pas d'argument. Le contenu de string produit par le mappage peut être accédé en appelant la méthode getString(), par exemple :

Java

```
com.altova.io.Input Products2Source =
com.altova.io.StreamInput.createInput("Products.xml");
com.altova.io.StringOutput ProductValuePairs2Target = new com.altova.io.StringOutput();
try {
    // Run the mapping
    MappingMapToProductValuePairsObject.run(Products2Source, ProductValuePairs2Target);
    // Get the string object
    String str = ProductValuePairs2Target.getString().toString();
}
```

Dans C#, le constructeur de StringOutput prend le paramètre de type StringBuilder que vous devez déclarer à l'avance. Si l'objet StringBuilder contient déjà des données, la sortie de mappage y sera ajoutée.

C#

```
Altova.IO.Input Products2Source = Altova.IO.StreamInput.createInput("Products.xml");
System.Text.StringBuilder sb = new System.Text.StringBuilder();
Altova.IO.Output ProductValuePairs2Target = new Altova.IO.StringOutput(sb);

try
{
    // Run the mapping
    MappingMapToProductValuePairsObject.Run(Products2Source, ProductValuePairs2Target);
    // Get the string output
    String str = sb.ToString();
}
```

Pour exécuter ces listings de code, vous pouvez utiliser le même projet généré que dans l'exemple précédent. Veillez, toutefois, à copier le fichier **Products.xml** de **C**:

\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\Tutorials\ vers votre répertoire de travail du programme puisque le code de mappage lit les données depuis ce fichier.

# Orientations d'utilisation pour les documents DOM

Lorsque vous utilisez les documents DOM en tant qu'entrée ou sortie de mappage, veuillez noter les points suivants :

- L'instance de document fournie en tant que paramètre au constructeur DocumentOutput doit être vide.
- Après avoir appelé <u>run</u>, le document DOM généré par le constructeur de <u>DocumentOutput</u> contient déjà une sortie de mappage, et vous pouvez manipuler le document tel que requis.

# 15.6 Générer du code depuis des Schémas XML ou des DTDs

Lorsque vous générez du code depuis un mappage, MapForce génère une application complète qui exécute automatiquement toutes les étapes du mappage. En option, vous pouvez générer des bibliothèques pour tous les schémas XML utilisés dans le mappage. Celles-ci permettent à votre code de créer ou de lire facilement des instances XML qui sont utilisées ou créées par le code mappage.

Pour générer des bibliothèques pour tous les schémas XML utilisés dans le mappage, cochez la case à cocher **Generate Wrapper Classes** depuis les <u>options de générateur de code</u><sup>(20)</sup>. La prochaine fois que vous générez du code, MapForce créera non seulement une application de mappage, mais aussi des classes de wrapper pour tous les schémas utilisés dans le mappage, comme suit :

C++ ou C#	Java	Objectif
Altova	com.altova	Bibliothèque de base contenant un support de runtime commun, à l'identique pour chaque schéma.
AltovaXML	com.altova.xml	Bibliothèque de base contenant un support de runtime pour XML, à l'identique pour chaque schéma.
YourSchema	com. YourSchema	Une bibliothèque contenant des déclarations générées depuis le schéma d'entrée, nommé comme le fichier de schéma ou le DTD. Cette Bibliothèque est un wrapper DOM (W3C Document Object Model) qui vous permet de lire, modifier et créer des documents XML en toute sécurité. Toutes les données contenues dans le DOM, et il existe des méthodes pour extraire des données depuis le DOM, et pour mettre à jour et créer des données dans le DOM. Le code C++ généré prend en charge soit Microsoft MSXML soit Apache Xerces 3. La syntaxe pour utiliser le code généré est généralement semblable pour les deux implémentations DOM, sauf pour quelques légères différences (par exemple, Xerces prend en charge plus de fonctions surchargées). Le code C# généré utilise la bibliothèque standard .NET .XML en guise de mise en place DOM sous-jacente.
		Le code Java généré utilise JAXP (Java API pour traitement XML) en tant que l'interface DOM sous-jacente.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir <u>Intégrer des Bibliothèques Schema</u> <u>Wrapper</u><sup>(1118)</sup>).

En plus des bibliothèques de base recensées ci-dessus, certaines bibliothèques prises en charge sont aussi générées. Les bibliothèques prises en charge sont utilisées par les bibliothèques de base Altova et ne sont pas destinées pour les intégrations personnalisées, puisqu'elles sont soumises à des modifications.

## Génération de nom et espaces de noms

MapForce génère des classes correspondant à tous les éléments déclarés ou les types complexes qui redéfinissent tout type complexe dans votre Schéma XML, préservant la dérivation de classe telle que définie par les extensions des types complexes dans votre Schéma XML. Dans le cas de schémas complexes qui importent des composants de schéma depuis plusieurs espaces de noms, MapForce préserve cette information en générant les espaces de noms C# ou C++ ou les packages Java appropriés.

Généralement, le générateur de code tente de préserver les noms pour les espaces de noms générés, les classes et les membres provenant du schéma XML original. Les caractères qui ne sont pas valides dans les identifieurs dans le langage cible sont remplacés par "\_". Les noms qui se heurtent avec d'autres noms ou des mots réservés sont rendus uniques en ajoutant un nombre. La génération de nom peut être influée en changeant les paramètres par défaut dans le modèle <u>SPL</u><sup>[200]</sup>.

Les espaces de noms provenant du Schéma XML sont convertis en packages dans Java ou des espaces de noms dans du code C# ou C++, en utilisant le préfixe d'espace de noms depuis le schéma en tant qu'espace de noms de code. La bibliothèque complète est contenue dans un package ou dans un espace de noms dérivé depuis le nom de fichier, pour que vous puissiez utiliser plusieurs bibliothèques générées dans un programme sans conflits de noms.

# Types de données

Schéma XML dispose d'un modèle de type de données plus élaboré que Java, C# ou C++. Le générateur de code convertit les types de Schéma XML dans des types primitifs spécifiques au langage, ou dans les classes fournies avec la bibliothèque Altova. Les types complexes et les types dérivés définis dans le schéma sont convertis en classes dans la bibliothèque générée. Les facettes d'énumération provenant de types simples sont converties dans des constantes symboliques.

Le mappage des types simples peut être configuré dans le modèle SPL, voir <u>Référence SPL</u><sup>1203</sup>.

Si vos fichiers d'instance XML utilisent des types de schéma liés à l'heure et la durée, ceux-ci sont convertis en classes natives Altova dans le code généré. Pour plus d'informations concernant les classes de bibliothèque, voir :

- <u>Référence aux classes générées (C++)</u><sup>1153</sup>
- <u>Référence aux classes générées (C#)</u>
- <u>Référence aux classes générées (Java)</u><sup>(118)</sup>

Pour plus d'informations concernant la conversion de type et d'autres détails applicables dans chaque langage, voir :

• À propos des Bibliothèques de Schema Wrapper (C++)

- À propos des Schema Wrapper Libraries (C#)
- <u>À propos des Schema Wrapper Libraries (Java)</u><sup>1116</sup>

## Gestion de mémoire

Une arborescence DOM est constituée de nœuds, qui sont toujours possédées par un document DOM spécifique - même si le nœud ne fait pas actuellement partie du document. Toutes les classes générées sont des références aux nœuds DOM qu'elles représentent, par des valeurs. Cela signifie que l'attribution d'une instance d'une classe générée ne copie pas la valeur, elle crée uniquement une référence supplémentaire dans la même donnée.

# Prise en charge de Schéma XML

Les structures de Schéma XML suivant sont traduites dans du code :

a) Espace des noms XML

b) Types simple :

- Types de schéma XML Built-in
- Types simples dérivés par extension
- Types simples dérivés par restriction
- Facettes
- Enumérations
- Motifs

c) Types complexes :

- Nœud anyType Built-in
- Types complexes définis par l'utilisateur
- Dérivé par extension : Mappé vers les classes dérivées
- Dérivé par restriction
- Contenu complexe
- Contenu simple
- Contenu mixte

Les fonctions de Schéma XML avancées suivantes ne sont pas prises en charge (ou pas entièrement prises en charge) dans les classes de wrapper générées :

- Caractères génériques : xs:any et xs:anyAttribute
- Modèles de contenu (séquence, choix, tout). Le compositeur de niveau supérieur est disponible dans <u>SPL</u><sup>(20)</sup>, mais n'est pas mis en place par des classes générées.
- Les valeurs par défaut et fixées pour les attributs. Celles-ci sont disponibles dans <u>SPL</u><sup>[203]</sup>, mais ne sont pas configurées ou mises en place par des classes générées.
- Les attributs xsi:type, les types abstraits. Lorsque vous souhaitez écrire l'attribut xsi:type, utiliser la méthode SetXsiType() des classes générées.
- Types d'union : toutes les combinaisons ne sont pas prises en charge.
- Les groupes de substitution sont partiellement pris en charge (résolus en tant que "choice").
- Attribut nillable="true" et xsi:nil
- Contraintes d'unicité
- Contraintes d'identité (key et keyref)

# 15.6.1 À propos des Schema Wrapper Libraries (C++)

# Types de caractère

Le code C++ généré peut être compilé avec ou sans prise en charge Unicode. Selon ce paramètre, les types  $string_type$  et tstring seront tous les deux définis en tant que std::string ou std::wstring, consistant en des caractères larges ou étroits. Pour utiliser les caractères Unicode dans votre fichier XML qui ne sont pas représentables avec l'ensemble de caractères 8-bit actuels, la prise en charge Unicode doit être activée. Prêtez une attention particulière aux macros  $_T()$ . Ce macro garantit que les constantes de string sont stockées correctement, que vous compiliez pour les programmes Unicode ou non-Unicode.

# Types de données

Le mappage par défaut des types Schéma XML dans les types de données C++ est :

Schéma XML	C++	Remarques
xs:string	string_type	string_type est défini en tant que std::string ou std:wstring
xs:boolean	bool	
xs:decimal	double	C++ n'a pas de type décimal, donc double est utilisé.
xs:float, xs:double	double	
xs:integer	int64	xs:integer a une plage illimitée, mappée surint64 pour des raisons d'efficacité.
xs:nonNegativeInteger	unsignedint64	voir ci-dessus
xs:int	int	
xs:unsignedInt	unsigned int	
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	altova::DateTime <sup>1153</sup>	
xs:duration	altova::Duration <sup>1156</sup>	
xs:hexBinary and xs:base64Binary	std::vector <unsigned char&gt;</unsigned 	L'encodage et le décodage des données binaires est effectué automatiquement.
xs:anySimpleType	string_type	

Tous les types de Schéma XML qui ne sont pas contenus dans la liste sont des types dérivés, et mappés sur le même C++ que leur type de base respectif.

# Classes générées

Pour chaque type dans le schéma, une classe est générée qui contient un membre pour chaque attribut et élément du type. Les membres sont nommés comme les attributs ou éléments dans le schéma original (en cas d'une collision éventuelle, un nombre est ajouté). Pour les types simples, des opérateurs d'attribution et de conversion sont générés. Pour des types simples avec des facettes d'énumération, les méthodes GetEnumerationValue() et SetEnumerationValue(int) peuvent être utilisés ensemble avec des constantes générées pour chaque valeur d'énumération. De plus, la méthode <code>StaticInfo()</code> permet l'accès à l'information de schéma en tant qu'un des types suivants :

altova::meta::SimpleType<sup>(16)</sup> altova::meta::ComplexType<sup>(16)</sup>

Les classes générées depuis des types complexes comprennent la méthode <code>SetXsiType()</code>, qui vous permet de définir l'attribut <code>xsi:type</code> du type. Cette méthode est utile si vous souhaitez créer des éléments d'instance XML d'un type dérivé.

Outre les classes pour les types déclarés dans le schéma XML, une classe de document (identifiée par "CDoc", voir ci-dessous) est générée. Elle contient tous les éléments root possibles, en tant que membres, ainsi que plusieurs autres méthodes. Pour plus d'informations concernant la classe, voir <u>[YourSchema]::</u> [CDoc]

Note: Le véritable nom de la classe dépend du nom du schéma .xsd.

Pour chaque attribut ou élément membre d'un type de schéma, une nouvelle classe est générée. Pour plus d'informations concernant ces classes, voir :

[YourSchema]::MemberAttribute [YourSchema]::MemberElement

Note: Les véritables noms de la classe dépendent du nom de l'attribut ou de l'élément membre du schéma.

Voir aussi Exemple : Utiliser les bibliothèques de Schema Wrapper<sup>(112)</sup>.

## Gestion d'erreur

Les erreurs sont rapportées par exceptions. Les classes d'exception suivantes sont définies dans l'espace de noms altova:

Classe	Classe de Base	Description
Erreur	std::logic_error	Erreur de logique de programme interne (indépendant des données d'entrée).
Exception	std::runtime_error	Classe de Base pour les erreurs de runtime.
InvalidArgumentsException	Exception	Une méthode a été appelée avec des valeurs d'argument invalide.

ConversionException	Exception	Exception lancée lorsqu'une conversion de type échoue.
StringParseException	ConversionException	Une valeur dans l'espace lexical ne peut pas être converti dans un espace de valeur.
ValueNotRepresentableExcept ion	ConversionException	Une valeur dans l'espace de valeur ne peut pas être converti dans un espace lexical.
OutOfRangeException	ConversionException	Une valeur de source ne peut pas être représentée dans un domaine cible.
InvalidOperationException	Exception	Une opération a été tentée qui n'est pas valide dans le contexte donné.
DataSourceUnavailableExcept ion	Exception	Un problème est survenu pendant le chargement d'une instance XML.
DataTargetUnavailableExcept ion	Exception	Un problème est survenu pendant l'enregistrement d'une instance XML.

Toutes les classes d'exception contiennent un texte de message et un pointeur dans une exception intérieure possible.

Méthode	Objectif	
string_type message()	Retourne une description textuelle de l'exception.	
std::exception inner()	Retourne l'exception qui a causé cette exception, si disponible, ou NULL.	

## Accéder à l'information de schéma

La bibliothèque générée permet d'accéder à des informations de schéma statique par le biais des classes suivantes. Toutes les méthodes sont déclarées en tant que const. Les méthodes qui retournent une de ces classes des métadonnées retournent un objet NULL si la propriété respective n'existe pas.

altova::meta::Attribute<sup>(115)</sup> altova::meta::ComplexType<sup>(116)</sup> altova::meta::Element<sup>(116)</sup> altova::meta::SimpleType<sup>(116)</sup>

# 15.6.2 À propos des Schema Wrapper Libraries (C#)

Le mappage par défaut des types Schéma XML dans les types de données C# est le suivant.

Schéma XML	C#	Remarques
xs:string	string	
xs:boolean	bool	
xs:decimal	decimal	xs:decimal dispose une plage et une précision illimitée, mappé vers les décimales pour des raisons d'efficacité.
xs:float, xs:double	double	
xs:long	long	
xs:unsignedLong	ulong	
xs:int	int	
xs:unsignedInt	uint	
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	Altova.Types.DateTime <sup>(168)</sup>	
xs:duration	Altova.Types.Duration <sup>1172</sup>	
xs:hexBinary and xs:base64Binary	byte[]	L'encodage et le décodage des données binaires est effectué automatiquement.
xs:anySimpleType	string	

Tous les types de Schéma XML qui ne sont pas contenus dans la liste sont des types dérivés, et mappés sur le même C# que leur type de base respectif.

# Classes générées

Pour chaque type dans le schéma, une classe est générée qui contient un membre pour chaque attribut et élément du type. Les membres sont nommés comme les attributs ou éléments dans le schéma original (en cas d'une collision éventuelle, un nombre est ajouté). Pour les types simples, des opérateurs d'attribution et de conversion sont générés. Pour des types simples avec des facettes d'énumération, les méthodes GetEnumerationValue() et SetEnumerationValue(int) peuvent être utilisés ensemble avec des constantes générées pour chaque valeur d'énumération. De plus, la méthode StaticInfo() permet l'accès à l'information de schéma en tant qu'un des types suivants :

Altova.Xml.Meta.SimpleType<sup>4177</sup> Altova.Xml.Meta.ComplexType<sup>4176</sup>

Les classes générées depuis des types complexes comprennent la méthode <code>setXsiType()</code>, qui vous permet de définir l'attribut <code>xsi:type</code> du type. Cette méthode est utile si vous souhaitez créer des éléments d'instance XML d'un type dérivé.

Outre les classes pour les types déclarés dans le schéma XML, une classe de document (identifiée par "Doc", voir ci-dessous) est générée. Elle contient tous les éléments root possibles, en tant que membres, ainsi que plusieurs autres méthodes. Pour plus d'informations concernant la classe, voir <u>[YourSchema].[Doc]</u><sup>(178)</sup>.

Note: Le véritable nom de la classe dépend du nom du schéma .xsd.

Pour chaque attribut ou élément membre d'un type de schéma, une nouvelle classe est générée. Pour plus d'informations concernant ces classes, voir :

[YourSchemaType].MemberAttribute<sup>(181)</sup> [YourSchemaType].MemberElement<sup>(182)</sup>

Note: Les véritables noms de la classe dépendent du nom de l'attribut ou de l'élément membre du schéma.

## Gestion d'erreur

Les erreurs sont rapportées par exceptions. Les classes d'exception suivantes sont définies dans l'espace de noms Altova :

Classe	Classe de Base	Description
ConversionException	Exception	Exception lancée lorsqu'une conversion de type échoue
StringParseException	ConversionException	Une valeur dans l'espace lexical ne peut pas être converti dans un espace de valeur.
DataSourceUnavailableException	System.Exception	Un problème est survenu pendant le chargement d'une instance XML.
DataTargetUnavailableException	System.Exception	Un problème est survenu pendant l'enregistrement d'une instance XML.

#### NET suivantes sont utilisées communément :

Classe	Description
System.Exception	Classe de Base pour les erreurs de runtime
System.ArgumentException	Une méthode a été appelée avec des valeurs d'argument invalide, ou bien une conversion de type a échoué.
System.FormatException	Une valeur dans l'espace lexical ne peut pas être converti dans un espace de valeur.
System.InvalidCastException	Une valeur ne peut pas être converie dans un autre type.
System.OverflowException	Une valeur de source ne peut pas être représentée dans un domaine cible.

## Accéder à l'information de schéma

La bibliothèque générée permet d'accéder à des informations de schéma statique par le biais des classes suivantes :

Altova.Xml.Meta.Attribute<sup>(1175</sup> Altova.Xml.Meta.ComplexType<sup>(1176</sup> Altova.Xml.Meta.Element<sup>(1176</sup> Altova.Xml.Meta.SimpleType<sup>(1177</sup>)</sup>

Les propriétés qui retournent une de ces classes des métadonnées retournent null si la propriété respective n'existe pas.

# 15.6.3 À propos des Schema Wrapper Libraries (Java)

Le mappage par défaut des types Schéma XML dans les types de données Java est le suivant :

Schéma XML	Java	Remarques
xs:string	String	
xs:boolean	booléenne	
xs:decimal	java.math.BigDecimal	
xs:float, xs:double	double	
xs:integer	java.math.BigInteger	
xs:long	long	
xs:unsignedLong	java.math.BigInteger	Java ne dispose pas de types non signés.
xs:int	int	
xs:unsignedInt	long	Java ne dispose pas de types non signés.
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	com.altova.types.DateTim e <sup>1184</sup>	
xs:duration	com.altova.types.Duratio n <sup>1189</sup>	
xs:hexBinary and xs:base64Binary	byte[]	L'encodage et le décodage des données binaires est effectué automatiquement.
xs:anySimpleType	string	
Tous les types de Schéma XML qui ne sont pas contenus dans la liste sont des types dérivés, et mappés sur le même type Java que leur type de base respectif.

### Classes générées

Pour chaque type dans le schéma, une classe est générée qui contient un membre pour chaque attribut et élément du type. Les membres sont nommés comme les attributs ou éléments dans le schéma original (en cas d'une collision éventuelle, un nombre est ajouté). Pour les types simples, des opérateurs d'attribution et de conversion sont générés. Pour des types simples avec des facettes d'énumération, les méthodes GetEnumerationValue() et SetEnumerationValue(int) peuvent être utilisés ensemble avec des constantes générées pour chaque valeur d'énumération. De plus, la méthode StaticInfo() permet l'accès à l'information de schéma en tant qu'un des types suivants :

com.altova.xml.meta.SimpleType<sup>1194</sup> com.altova.xml.meta.ComplexType<sup>1133</sup>

Les classes générées depuis des types complexes comprennent la méthode <code>SetXsiType()</code>, qui vous permet de définir l'attribut <code>xsi:type</code> du type. Cette méthode est utile si vous souhaitez créer des éléments d'instance XML d'un type dérivé.

Outre les classes pour les types déclarés dans le schéma XML, une classe de document (identifiée par "Doc", voir ci-dessous) est générée. Elle contient tous les éléments root possibles, en tant que membres, ainsi que plusieurs autres méthodes. Pour plus d'informations concernant la classe, voir <u>com. [YourSchema]. [Doc]</u><sup>(195)</sup>.

Note: Le véritable nom de la classe dépend du nom du schéma .xsd.

Pour chaque attribut ou élément membre d'un type de schéma, une nouvelle classe est générée. Pour plus d'informations concernant ces classes, voir :

com.[YourSchema].[YourSchemaType].MemberAttribute<sup>1198</sup> com.[YourSchema].[YourSchemaType].MemberElement<sup>1193</sup>

Note: Les véritables noms de la classe dépendent du nom de l'attribut ou de l'élément membre du schéma.

#### Gestion d'erreur

Les erreurs sont rapportées par exceptions. Les classes d'exception suivantes sont définies dans l'espace de noms com.altova:

Classe	Classe de Base	Description
SourceInstanceUnvailableException	Exception	Un problème est survenu pendant le chargement d'une instance XML.
TargetInstanceUnavailableException	Exception	Un problème est survenu pendant l'enregistrement d'une instance XML.

De plus, les exceptions Java suivantes sont utilisées communément :

Classe	Description
java.lang.Error	Erreur de logique de programme interne (indépendant des données d'entrée)
java.lang.Exception	Classe de Base pour les erreurs de runtime
java.lang.IllegalArgumentsException	Une méthode a été appelée avec des valeurs d'argument invalide, ou bien une conversion de type a échoué.
java.lang.ArithmeticException	Exception lancée lorsqu'une conversion de type numérique échoue.

### Accéder à l'information de schéma

La bibliothèque générée permet d'accéder à des informations de schéma statique par le biais des classes suivantes :

```
<u>com.altova.xml.meta.Attribute</u><sup>(1192</sup>
<u>com.altova.xml.meta.ComplexType</u>
<u>com.altova.xml.meta.Element</u><sup>(1193</sup>
<u>com.altova.xml.meta.SimpleType</u><sup>(1194)</sup>
```

Les propriétés qui retournent une de ces classes des métadonnées retournent null si la propriété respective n'existe pas.

# 15.6.4 Intégrer des Bibliothèques Schema Wrapper

Pour utiliser les bibliothèques Altova dans votre projet personnalisé, veuillez vous référer aux bibliothèques depuis votre projet ou les inclure dans votre projet, comme indiqué ci-dessous pour chaque langage.

#### C#

Pour intégrer les bibliothèques Altova dans un projet C# existant :

- Une fois que MapForce génère du code depuis un schéma (par exemple, YourSchema.xsd), construire la solution générée YourSchema.sln dans Visual Studio. Cette solution se trouve dans un dossier de projet avec le même nom en tant que le schéma.
- 2. Cliquer avec la touche de droite dans le projet existant dans Visual Studio, et sélectionner **Ajouter Référence**.
- 3. Dans l'onglet Chercher, naviguer dans les bibliothèques suivantes : Altova.dll, AltovaXML.dll et YourSchema.dll situés dans le répertoire de sortie des projets générés (par exemple, bin\Debug).

CO Add Reference
.NET COM Projects Browse Recent
Look in: 🌗 Debug 🗸 🎯 🎓 🛄 🕶 Name
S Altova.dll AltovaXML.dll Library.dll
۰
File name:
Files of type:       Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
OK Cancel

#### C++

Le moyen le plus simple d'intégrer les bibliothèques dans un projet C++ existant est d'ajouter les fichiers de projet généré dans votre solution. Par exemple, assumons que vous avez généré du code depuis une schéma appelé Library.xsd et sélectionné c:\codegen\cpp\library en tant que répertoire cible. Les bibliothèques générés dans ce cas sont disponibles sous :

- c:\codegen\cpp\library\Altova.vcxproj
- c:\codegen\cpp\library\AltovaXML\AltovaXML.vcxproj
- c:\codegen\cpp\library\Library.vcxproj

Tout d'abord, ouvrir la solution c:\codegen\cpp\library\Library.sln générée et l'intégrer dans Visual Studio.

Ensuite ouvrir votre solution Visual Studio existante (dans Visual Studio 2010, dans cet exemple), cliquer avec la touche de droite, choisir **Add | Existing Project**, et ajouter les fichiers de projet recensés ci-dessus, un par un. Soyez patient tant que Visual Studio parse les fichiers. Ensuite cliquer avec la touche de droite sur votre projet et choisir **Propriétés**. Dans le dialogue Pages de propriété, sélectionner **Common Properties | Framework and References**, puis cliquer sur **Add New Reference**. Ensuite, sélectionner et ajouter chacun des projets suivants : *Altova*, *AltovaXML* et *Library*.

🕫 Add Reference	? 🗙
Projects	
Project Name	Project Directory
Altova	C:\codegen\cpp\library\Altova\
AltovaXML	C:\codegen\cpp\library\AltovaXML\
Library	C:\codegen\cpp\library\Library\
<	
	OK Cancel

Voir aussi la documentation MSDN pour utiliser les fonctions depuis une bibliothèque personnalisée, comme applicables à votre version de Visual Studio, par exemple :

- Si vous avez choisi de générer des bibliothèques statiques, voir <u>https://msdn.microsoft.com/en-us/library/ms235627(v=vs.100).aspx</u>
- Si vous avez choisi de générer des bibliothèques dynamique, voir <u>https://msdn.microsoft.com/en-us/library/ms235636(v=vs.100).aspx</u>

L'option de générer des bibliothèques statiques ou dynamiques est disponible dans les options de la génération de code (voir <u>Options de Générateur de Code</u><sup>1201</sup>).

#### Java

Un des moyens d'intégrer les packages Altova dans vos projets Java est de copier le répertoire **com** du code généré vers le répertoire qui stocke les packages de source de votre projet Java (par exemple, **C: \Workspace\MyJavaProject\src**). Par exemple, nous partons du principe de vous avez généré du code dans **c:\codegen\java\library**. Les classes Altova générées dans ce cas sont disponibles sous **c: \codegen\java\library\com**.

Une fois avoir copié les bibliothèques, actualiser le projet. Pour actualiser le projet dans Eclipse, le sélectionner dans le Package Explorer, et appuyer sur F5. Pour actualiser le projet dans NetBeans IDE 8.0, sélectionner la commande de menu **Source | Scan for External Changes**.

Une fois que vous avez effectué l'opération copy, les packages Altova sont disponibles dans le Package Explorer (dans le cas d'Eclipse), ou sous "Source Packages" dans le volet Projets (dans le cas de NetBeans IDE).



Packages Altova dans Eclipse 4.4



Packages Altova dans NetBeans IDE 8.0.2

# **15.6.5 Exemple: Bibliothèques de livres**

Cet exemple illustre comment utiliser les bibliothèques de schema wrapper générées afin d'écrire ou de lire des documents XML conforme en terme de programmation au schéma. Avant d'utiliser le code d'échantillon, veuillez prendre le temps de comprendre la structure du schéma d'exemple inclus.

Le schéma utilisé dans cet exemple décrit une bibliothèque de livres. La définition complète du schéma est affichée ci-dessous. Enregistrer cet extrait de code en tant que **Library.xsd** si vous souhaitez obtenir les mêmes résultats que dans cet exemple. Vous aurez besoin de ce schéma pour générer les bibliothèques de code utilisées dans cet exemple.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.nanonull.com/LibrarySample"</pre>
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.nanonull.com/LibrarySample" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="Library">
     <xs:complexType>
         <xs:sequence>
            <xs:element name="Book" type="BookType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="LastUpdated" type="xs:dateTime"/>
      </xs:complexType>
   </r></r></r>
   <xs:complexType name="BookType">
      <xs:sequence>
         <xs:element name="Title" type="xs:string"/>
         <xs:element name="Author" type="xs:string" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
      <xs:attribute name="Format" type="BookFormatType" use="required"/>
   </xs:complexType>
   <xs:complexType name="DictionaryType">
      <xs:complexContent>
         <xs:extension base="BookType">
            <xs:sequence>
               <xs:element name="FromLang" type="xs:string"/>
               <xs:element name="ToLang" type="xs:string"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:simpleType name="BookFormatType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Hardcover"/>
         <xs:enumeration value="Paperback"/>
         <xs:enumeration value="Audiobook"/>
         <xs:enumeration value="E-book"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

Library est un élément racide d'un complexType qui peut être représenté graphiquement comme suit dans le mode Schéma de XMLSpy:



Comme indiqué ci-dessus, la bibliothèque contient un attribut **LastUpdated** (défini en tant que xs:dateTime), et stocke une séquence de livres. Chaque livre est un xs:complexType et possède deux attributs : un **ID** (défini en tant que xs:integer), et un **Format**. Le format de chaque livre peut être hardcover, paperback, audiobook, ou e-book. Dans le schéma, **Format** est défini en tant que xs:simpleType qui utilise une énumération des valeurs mentionnées ci-dessus.

Chaque livre présente aussi un élément **Title** (défini en tant que xs:string), et un ou plusieurs **Author** éléments (définis en tant que xs:string).

La bibliothèque peut aussi contenir des livres qui sont des dictionnaires. Dictionaries ont le type DictionaryType, qui est dérivé par extension depuis BookType. En d'autres termes, un dictionnaire hérite de tous les attributs et éléments d'un Book (livre), plus deux éléments supplémentaires : **FromLang** et **ToLang**, comme illustré ci-dessous.



Les éléments FromLang et ToLang stockent le langage de source et de destination du dictionnaire.

Un fichier d'instance XML valide conformément au schéma ci-dessus peut donc ressembler à l'exemple affiché dans l'exemple ci-dessous (s'il se trouve dans le même répertoire que le fichier de schéma) :

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="nl:DictionaryType">
 <Title>English-German Dictionary</Title>
 <Author>John Doe</Author>
 <FromLang>English</FromLang>
 <ToLang>German</ToLang>
 </Book>
</Library>
```

La rubrique suivante illustre comment lire ou écrire dans un tel fichier par le biais d'un programme. Pour commencer, générer le code de schema wrapper depuis le schéma ci-dessus, en utilisant les étapes décrites dans <u>Générer du code depuis des Schémas XML ou des DTD</u><sup>1100</sup>.

# 15.6.5.1 Lire et écrire des documents XML (C++)

Après avoir généré du code depuis le <u>schéma d'exemple<sup>(112)</sup></u>, une application C++ test est créée, avec plusieurs bibliothèques Altova de support.

## À propos des bibliothèques C++ générées

La classe centrale du code généré est la classe CDoc qui représente le document XML. Une telle classe est générée pour tous les schémas et son nom dépend du nom de fichier de schéma. Comme indiqué dans le diagramme, cette classe fournit des méthodes pour charger les documents depuis des fichiers, des streams binaires ou des strings (ou pour enregistrer des documents dans des fichiers, des streams, des strings). Pour consulter une liste de tous les membres exposés par cette classe, voir la référence de classe ([YourSchema]::[CDoc]<sup>1100</sup>).

	CDoc
	Library:MemberElement <membertype->CLibraryType,MemberIndex-&gt;_altova_mi_altova_CDoc_altova_Library&gt;</membertype->
0	«constructor» CDoc(in unnamed1:int)
	«constructor» CDoc(in init:«const» CDoc&)
	operator=(in other:«const» CDoc&);void
	<u>StaticInfo():ComplexType</u>
	SetXsiType():void
	LoadFromFile(in fileName:«const» string type&);CDoc
	LoadFromString(in xml:«const» string type&):CDoc
	<u>LoadFromBinary(in data:«const» vector&lt; Ty-&gt; unsigned_char&gt;&amp;):CDoc</u>
	SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool):void
	SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in omitXmlDecl:bool):void
	SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in encoding:«const» string_type&):void
	SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:«const» string_type&):void
	SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in encoding:«const» string_type&, in bBigEndian:bool, in bBOM:bool):void
	SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:«const» string_type&, in bBigEndian:bool, in bBOM:bool):void
	SaveToString(in prettyPrint:bool):string_type
	SaveToString(in prettyPrint:bool, in omitXmlDecl:bool):string_type
	SaveToBinary(in prettyPrint:bool):vector<_Ty-> unsigned_char>
	SaveToBinary(in prettyPrint:bool, in encoding:« <mark>const</mark> » string_type&):vector<_Ty-> unsigned_char>
	SaveToBinary(in prettyPrint:bool, in encoding:« <mark>const</mark> » string_type&, in bBigEndian:bool, in bBOM:bool):vector<_Ty-> unsigned_char>
	<u>CreateDocument():CDoc</u>
	DestroyDocument():void
	SetDTDLocation(in dtdLocation:«const» string_type&):void
	SetSchemaLocation(in schemaLocation:«const» string_type&):void
	DeclareAllNamespacesFromSchema(in node:TypeBase&):void

Le champ Library de la classe CDoc représente la racine réelle du document. Library est un élément dans le fichier XML, donc dans le code C++ il a une classe de modèle en tant que type (MemberElement). Le modèle de classe expose des méthodes et des propriétés pour interagir avec l'élément Library. En général, chaque attribut et chaque élément d'un type dans le schéma est saisi dans le code généré avec les classes de modèle MemberAttribute et MemberElement, respectivement. Pour plus d'informations, voir la référence de classe [YourSchema]::MemberAttribute

La classe CLibraryType est générée depuis le type complexe **LibraryType** dans le schéma. Veuillez noter que la classe CLibraryType contient deux champs : Book et LastUpdated. Conformément à la logique déjà mentionnée ci-dessus, ces champs correspondent à l'élément **Book** et à l'attribut **LastUpdated** dans le schéma, et vous permettent de manipuler par le biais d'un programme (apposer, supprimer, etc.) des éléments et des attributs dans le document d'instance XML.

СLibraryТуре
LastUpdated:MemberAttribute <membertype->DateTime,MemberIndex-&gt;_altova_mi_altova_CLibraryType_altova_LastUpdated,EnumOffset-&gt;0,EnumCount-&gt;0&gt; Book:MemberElement<membertype->CBookType,MemberIndex-&gt;_altova_mi_altova_CLibraryType_altova_Book&gt;</membertype-></membertype->
<constructor» clibrarytype(in="" unnamed1:int)<br="">«constructor» CLibraryType(in init:«const» CLibraryType&amp;) operator=(in other:«const» CLibraryType&amp;):void <u>StaticInfo():ComplexType</u></constructor»>
«struct» Book

Puisque le DictionaryType est un type complexe dérivé depuis **BookType** dans le schéma, cette relation se retrouve aussi dans les classes générées. Comme illustré dans le diagramme, la classe CDictionaryType hérite la classe CBookType.

CBookType	
ID:MemberAttribute <membertype->long_long,MemberIndex-&gt;_altova_mi_altova_CBookType_altova_ID,EnumOffset-&gt;0,EnumCount-&gt;0&gt;         Format:MemberAttribute<membertype->string_type,MemberIndex-&gt;_altova_mi_altova_CBookType_altova_Format,EnumOffset-&gt;1,EnumCount-&gt;4&gt;         Title:MemberElement<membertype->CstringType,MemberIndex-&gt;_altova_mi_altova_CBookType_altova_Title&gt;         Author:MemberElement<membertype->CstringType,MemberIndex-&gt;_altova_mi_altova_CBookType_altova_Author&gt;</membertype-></membertype-></membertype-></membertype->	
<ul> <li>«constructor» CBookType(in unnamed1:int)</li> <li>«constructor» CBookType(in init:«const» CBookType&amp;)</li> <li>operator=(in other:«const» CBookType&amp;):void</li> <li><u>StaticInfo():ComplexType</u></li> <li>SetXsiType():void</li> </ul>	
∠ «visibility»	«visibility»
CDictionaryType	
FromLang:MemberElement < MemberType-> CstringType,MemberIndex-> _altova_mi_altova_CDictionaryType_altova_Fro     ToLang:MemberElement < MemberType-> CstringType,MemberIndex-> _altova_mi_altova_CDictionaryType_altova_ToLan	mLang> g>
<ul> <li>«constructor» CDictionaryType(in unnamed1:int)</li> <li>«constructor» CDictionaryType(in init:«const» CDictionaryType&amp;)</li> <li>operator=(in other:«const» CDictionaryType&amp;):void</li> <li><u>StaticInfo():ComplexType</u></li> <li>SetXsiType():void</li> </ul>	

Si votre schéma XML définit des types simples en tant qu'énumérations, les valeurs énumérées deviennent disponibles en tant que valeurs enum dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un enum membre de la classe CBookFormatType.

## Écrire un document XML

1. Ouvrir la solution **LibraryTest.sln** dans Visual Studio générée depuis le schéma Library mentionné plus tôt dans cet exemple.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir <u>Intégrer des Bibliothèques Schema</u> <u>Wrapper</u>).

2. Dans Solution Explorer, ouvrir le fichier LibraryTest.cpp, et éditer la méthode Example() comme indiqué ci-dessous.

#include <ctime> // required to get current time
using namespace Doc; // required to work with Altova libraries

```
void Example()
ł
   // Create a new, empty XML document
  CDoc libDoc = CDoc::CreateDocument();
   // Create the root element <Library> and add it to the document
  CLibraryType lib = libDoc.Library.append();
   // Get current time and set the "LastUpdated" attribute using Altova classes
   time_t t = time(NULL);
   struct tm * now = localtime( & t );
   altova::DateTime dt = altova::DateTime(now->tm_year + 1900, now->tm_mon + 1, now-
>tm_mday, now->tm_hour, now->tm_min, now->tm_sec);
  lib.LastUpdated = dt;
   // Create a new <Book> and add it to the library
  CBookType book = lib.Book.append();
   // Set the "ID" attribute of the book
  book.ID = 1;
   // Set the "Format" attribute of the <Book> using an enumeration constant
   book.Format.SetEnumerationValue( CBookFormatType::k_Paperback );
   // Add the <Title> and <Author> elements, and set values
  book.Title.append() = _T("The XML Spy Handbook");
   book.Author.append() = _T("Altova");
   // Append a dictionary (book of derived type) and populate its attributes and elements
   CDictionaryType dictionary = CDictionaryType(lib.Book.append().GetNode());
   dictionary.ID = 2;
  dictionary.Format.SetEnumerationValue( CBookFormatType::k_E_book);
  dictionary.Title.append() = _T("English-German Dictionary");
  dictionary.Author.append() = _T("John Doe");
   dictionary.FromLang.append() = _T("English");
  dictionary.ToLang.append() = _T("German");
   // Since dictionary a derived type, set the xsi:type attribute of the book element
  dictionary.SetXsiType();
   // Optionally, set the schema location
  libDoc.SetSchemaLocation(_T("Library.xsd"));
   // Save the XML document to a file with default encoding (UTF-8),
  // "true" causes the file to be pretty-printed.
  libDoc.SaveToFile(_T("GeneratedLibrary.xml"), true);
  // Destroy the document
  libDoc.DestroyDocument();
}
```

3. Appuyer sur **F5** pour lancer le débogage. Si le code a été exécuté avec succès, un fichier **GeneratedLibrary.xml** est créé dans le répertoire de sortie de solution.

#### Lire un document XML

- 1. Ouvrir la solution LibraryTest.sln dans Visual Studio.
- 2. Enregistrer le code ci-dessous en tant que **Library1.xml** dans un répertoire qui peut être lu par le code de programme (par exemple, le même répertoire que **LibraryTest.sln**).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="nl:DictionaryType">
<Title>English-German Dictionary</Title>
<Author>John Doe</Author>
 <FromLang>English</FromLang>
<ToLang>German</ToLang>
 </Book>
</Library>
```

3. Dans Solution Explorer, ouvrir le fichier LibraryTest.cpp, et éditer la méthode Example() comme indiqué ci-dessous.

```
using namespace Doc;
void Example()
{
   // Load XML document
   CDoc libDoc = CDoc::LoadFromFile(_T("Library1.xml"));
   // Get the first (and only) root element <Library>
  CLibraryType lib = libDoc.Library.first();
   // Check whether an element exists:
   if (!lib.Book.exists())
   {
      tcout << "This library is empty." << std::endl;</pre>
      return;
   }
   // iteration: for each <Book>...
   for (Iterator<CBookType> itBook = lib.Book.all(); itBook; ++itBook)
   {
      // output values of ISBN attribute and (first and only) title element
      tcout << "ID: " << itBook->ID << std::endl;</pre>
      tcout << "Title: " << tstring(itBook->Title.first()) << std::endl;</pre>
      // read and compare an enumeration value
```

```
if (itBook->Format.GetEnumerationValue() == CBookFormatType::k_Paperback)
        tcout << "This is a paperback book." << std::endl;

// for each <Author>...
for (CBookType::Author::iterator itAuthor = itBook->Author.all(); itAuthor; +
+itAuthor)
        tcout << "Author: " << tstring(itAuthor) << std::endl;

// alternative: use count and index
for (unsigned int j = 0; j < itBook->Author.count(); ++j)
        tcout << "Author: " << tstring(itBook->Author.gil) << std::endl;

// Destroy the document
libDoc.DestroyDocument();
}</pre>
```

4. Press F5 to start debugging.

## 15.6.5.2 Lire et écrire des documents XML (C#)

Après avoir généré du code depuis le schéma de bibliothèque (voir <u>Schéma d'exemple<sup>(1121)</sup>)</u>, une application C# test est créée, avec plusieurs bibliothèques Altova.

### À propos des bibliothèques C# générées

La classe centrale du code généré est la classe Doc2qui représente le document XML. Une telle classe est générée pour tous les schémas et son nom dépend du nom de fichier de schéma. Veuillez noter que cette classe est appelée Doc2 pour éviter un conflit possible avec le nom d'espace de noms. Comme indiqué dans le diagramme, cette classe fournit des méthodes pour charger les documents depuis des fichiers, des streams binaires ou des strings (ou pour enregistrer des documents dans des fichiers, des streams, des strings). Pour consulter une description de cette classe, voir la référence de classe (<u>YourSchema].[Doc]</u> ).

	Doc2
	Library:MemberElement_Library
	C# Properties
٥	<u>«GetAccessor, property» StaticInfo():ComplexType</u>
	Methods
	LoadFromFile(in filename:string):Doc2
	LoadFromString(in xmlstring:string):Doc2
	LoadFromBinary(in binary:byte[*]):Doc2
	SaveToFile(in filename:string, in prettyPrint:bool):void
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool):void
	SaveToFileWithLineEnd(in filename:string, in prettyPrint:bool, in omitXmIDecl:bool, in lineend:string):void
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:string):void
	SaveToFile(in filename:string, in prettyPrint:bool, in encoding:string, in lineend:string):void
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:string, in lineend:string):void
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:string, in bBigEndian:bool, in bBOM:bool, in lineend:string):void
	SaveToString(in prettyPrint:bool):string
	SaveToString(in prettyPrint:bool, in omitXmlDecl:bool):string
	SaveToBinary(in prettyPrint:bool):byte[*]
	SaveToBinary(in prettyPrint:bool, in encoding:string):byte[*]
۲	SaveToBinary(in prettyPrint:bool, in encoding:string, in bBigEndian:bool, in bBOM:bool):byte[*]
	CreateDocument():Doc2
۲	CreateDocument(in encoding:string):Doc2
	SetDTDLocation(in dtdLocation:string):void
۲	SetSchemaLocation(in schemaLocation:string):void
	DeclareAllNamespacesFromSchema(in node:TypeBase):void
۲	«constructor» Doc2(in init:System.Xml.XmlNode)
٥	SetXsiType():void

Le membre Library de la classe Doc2 représente la véritable racine du document.

Conformément aux règles de génération de code mentionnées dans À propos des Schema Wrapper Libraries (C#)<sup>[113]</sup>, les membres de classes sont générés pour chaque attribut et pour chaque élément d'un type. Dans le code généré, le nom de ce type de classes de membre est préfixé avec MemberAttribute\_ et MemberElement\_, respectivement. Des exemples de telles classes sont MemberAttribute\_ID et MemberElement\_Author, générés depuis l'élément **Author** et l'attribut **ID** d'un livre, respectivement (dans le diagramme ci-dessous, ils s'agit de classes imbriquées sous BookType). Ces classes vous permettent de manipuler à l'aide d'un programme les éléments et les attributs correspondants dans l'instance du document XML (par exemple, apposer, supprimer, définir une valeur, etc). Pour plus d'informations, voir la référence de classe <u>[YourSchemaType].MemberAttribute</u><sup>[18]</sup> et <u>[YourSchemaType].MemberElement</u><sup>[18]</sup>.

Puisque le **DictionaryType** est un type complexe dérivé depuis **BookType** dans le schéma, cette relation se retrouve aussi dans les classes générées. Comme illustré dans le diagramme, la classe DictionaryType hérite la classe BookType.



Si votre schéma XML définit des types simples en tant qu'énumérations, les valeurs énumérées deviennent disponibles en tant que valeurs Enum dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un Enum membre de la classe BookFormatType.

### Écrire un document XML

1. Ouvrir la solution **LibraryTest.sln** dans Visual Studio généré depuis le schéma Library mentionné plus tôt dans cet exemple.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir <u>Intégrer des Bibliothèques Schema</u> <u>Wrapper</u>). 2. Dans Solution Explorer, ouvrir le fichier LibraryTest.cs, et éditer la méthode Example() comme indiqué ci-dessous.

```
protected static void Example()
{
    // Create a new XML document
    Doc2 doc = Doc2.CreateDocument();
    // Append the root element
   LibraryType root = doc.Library.Append();
    // Create the generation date using Altova DateTime class
    Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
    // Append the date to the root
    root.LastUpdated.Value = dt;
    // Add a new book
    BookType book = root.Book.Append();
    // Set the value of the ID attribute
   book.ID.Value = 1;
    // Set the format of the book (enumeration)
    book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
    // Set the Title and Author elements
    book.Title.Append().Value = "The XMLSpy Handbook";
    book.Author.Append().Value = "Altova";
   // Append a dictionary (book of derived type) and populate its attributes and
elements
   DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
   dictionary.ID.Value = 2;
   dictionary.Title.Append().Value = "English-German Dictionary";
    dictionary.Format.EnumerationValue = BookFormatType.EnumValues.eE_book;
    dictionary.Author.Append().Value = "John Doe";
    dictionary.FromLang.Append().Value = "English";
    dictionary.ToLang.Append().Value = "German";
    // Since it's a derived type, make sure to set the xsi:type attribute of the
book element
   dictionary.SetXsiType();
    // Optionally, set the schema location (adjust the path if
    // your schema is not in the same folder as the generated instance file)
   doc.SetSchemaLocation("Library.xsd");
    // Save the XML document with the "pretty print" option enabled
    doc.SaveToFile("GeneratedLibrary.xml", true);
}
```

 Appuyer sur F5 pour lancer le débogage. Si le code a été exécuté avec succès, un fichier GeneratedLibrary.xml est créé dans le répertoire de sortie de solution (généralement, bin/Debug).

#### Lire un document XML

1. Ouvrir la solution LibraryTest.sln dans Visual Studio.

2. Enregistrer le code-ci-dessous en tant que **Library.xml** dans le répertoire de sortie du projet (par défaut, **bin/Debug**). C'est le fichier qui sera lu par le code du programme.

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd" LastUpdated="2016-
02-03T17:10:08.4977404">
 <Book ID="1" Format="E-book">
 <Title>The XMLSpy Handbook</Title>
 <Author>Altova</Author>
 </Book>
 <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="nl:DictionaryType">
 <Title>English-German Dictionary</Title>
 <Author>John Doe</Author>
 <FromLang>English</FromLang>
 <ToLang>German</ToLang>
 </Book>
</Library>
```

3. Dans Solution Explorer, ouvrir le fichier LibraryTest.cs, et éditer la méthode Example() comme indiqué ci-dessous.

```
protected static void Example()
{
   // Load the XML file
   Doc2 doc = Doc2.LoadFromFile("Library.xml");
   // Get the root element
   LibraryType root = doc.Library.First;
   // Read the library generation date
   Altova.Types.DateTime dt = root.LastUpdated.Value;
   string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
   Console.WriteLine("The library generation date is: " + dt_as_string);
   // Iteration: for each <Book>...
   foreach (BookType book in root.Book)
    {
     // Output values of ID attribute and (first and only) title element
     Console.WriteLine("Title: " + book.Title.First.Value);
     // Read and compare an enumeration value
     if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
        Console.WriteLine("This is a paperback book.");
     // Iteration: for each <Author>
     foreach (xs.stringType author in book.Author)
        Console.WriteLine("Author: " + author.Value);
     // Determine if this book is of derived type
```

```
if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
         // Find the value of the xsi:type attribute
         string xsiTypeValue =
book.Node.Attributes.GetNamedItem("xsi:type").Value;
         // Get the namespace URI and the lookup prefix of this namespace
         string namespaceUri = book.Node.NamespaceURI;
         string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);
         // if this book has DictionaryType
         if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
         {
            // output additional fields
            DictionaryType dictionary = new DictionaryType(book.Node);
           Console.WriteLine("Language from: " +
dictionary.FromLang.First.Value);
           Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
         }
         else
         {
            throw new Exception("Unexpected book type");
         }
      }
    }
    Console.ReadLine();
}
```

4. Appuyer sur **F5** pour lancer le débogage. Si le code a été exécuté avec succès, **Library.xml** sera lu par le code de programme, et ses contenus seront affichés en tant que sortie de console.

#### Lire et écrire des éléments et des attributs

Les valeurs des éléments et des attributs peut être accédé en utilisant la propriété value de la classe d'élément ou d'attribut du membre généré, par exemple :

```
// Output values of ID attribute and (first and only) title element
Console.WriteLine("ID: " + book.ID.Value);
Console.WriteLine("Title: " + book.Title.First.Value);
```

Pour obtenir la valeur de l'élément **Title** dans cet exemple particulier, nous avons aussi utilisé la méthode First(), étant donné que c'est le premier (et seul) élément **Title** d'un livre. Dans les cas où vous devez choisir un élément spécifique depuis une liste par index, utiliser la méthode At().

La classe générée pour chaque élément de membre d'un type met en place l'interface standard System.Collections.IEnumerable. Cela permet de boucler dans plusieurs éléments du même type. Dans cet exemple particulier, vous pouvez boucler à travers tous les livres d'un objet Bibliothèque comme suit :

```
// Iteration: for each <Book>...
foreach (BookType book in root.Book)
```

{

}

// your code here...

Pour ajouter un nouvel élément, utiliser la méthode Append(). Par exemple, le code suivant appose l'élément racine au document :

```
// Append the root element to the library
LibraryType root = doc.Library.Append();
```

Vous pouvez définir la valeur d'un attribut (comme ID dans cet exemple) comme suit :

```
// Set the value of the ID attribute
book.ID.Value = 1;
```

#### Lire et écrire des valeurs d'énumération

Si votre schéma XML définit des types simples en tant qu'énumérations, les valeurs énumérées deviennent disponibles en tant que valeurs Enum dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un Enum.



Pour attribuer des valeurs d'énumération à un objet, utiliser du code comme celui ci-dessous :

```
// Set the format of the book (enumeration)
book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
```

Vous pouvez lire ces valeurs d'énumération provenent de documents d'instance XML comme suit :

```
// Read and compare an enumeration value
if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
Console.WriteLine("This is a paperback book.");
```

Si une condition "if" n'est pas suffisante, créer un interrupteur pour déterminer chaque valeur d'énumération et traiter comme requis.

#### Travailler avec des types xs:dateTime et xs:duration

Si le schéma à partir duquel vous avez généré du code utilise les types heure et durée comme xs:dateTime, ou xs:duration, ils sont convertis en classes natives Altova dans un code généré. C'est pourquoi, pour écrire une valeur date ou durée vers le document XML, procéder comme suit :

- 1. Construit un objet <u>Altova.Types.DateTime</u><sup>1128</sup> OU <u>Altova.Types.Duration</u><sup>1172</sup> (soit depuis System.DateTime, ou en utilisant des parties comme des heures et des minutes, voir <u>Altova.Types.DateTime</u><sup>1128</sup> et <u>Altova.Types.Duration</u><sup>1172</sup> pour plus d'informations).
- 2. Définir l'objet en tant que la valeur de l'élément ou de l'attribut requis, par exemple :

```
// Create the library generation date using Altova DateTime class
Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
// Append the date to the root
root.LastUpdated.Value = dt;
```

Pour lire une date ou une durée depuis un document XML, procéder comme suit :

- 1. Déclarer la valeur d'élément (ou d'attribut) comme objet <u>Altova.Types.DateTime</u><sup>[163]</sup> OU <u>Altova.Types.Duration</u><sup>[1172]</sup>.
- 2. Formate l'élément ou l'attribut requis, par exemple :

```
// Read the library generation date
Altova.Types.DateTime dt = root.LastUpdated.Value;
string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
Console.WriteLine("The library generation date is: " + dt_as_string);
```

Pour plus d'informations, voir la référence de classe <u>Altova.Types.DateTime</u><sup>1168</sup> et Altova.Types.Duration<sup>1172</sup>.

#### Travailler avec des types dérivés

Si votre schéma XML définit des types dérivés, vous pouvez préserver la dérivation de type dans des documents XML que vous créez ou chargez par le biais d'un programme. Prendre le schéma utilisé dans cet exemple, l'extrait de code suivant illustre comment créer un nouveau livre de type dérivéDictionaryType:

```
// Append a dictionary (book of derived type) and populate its attributes and elements
DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
dictionary.ID.Value = 2;
dictionary.Title.Append().Value = "English-German Dictionary";
dictionary.Author.Append().Value = "John Doe";
dictionary.FromLanguage.Append().Value = "English";
dictionary.ToLanguage.Append().Value = "German";
// Since it's a derived type, make sure to set the xsi:type attribute of the book element
dictionary.SetXsiType();
```

Veuillez noter qu'il est important de définir l'attribut xsi:type du livre récemment créé. Cela garantit que le type de livre sera interprété correctement par le schéma losrque le document XML est validé.

Lorsque vous chargez des données depuis un document XML, l'extrait de code suivant montre comment identifier un livre de type dérivé DictionaryType dans l'instance XML chargée. Premièrement, le code trouve la valeur de l'attribut xsi:type du nœud de livre. Si l'espace de noms URI de ce nœud est http://www.nanonull.com/LibrarySample et si le préfixe et type de lookup URI correspond à la valeur de l'attribut xsi:type, alors il s'agit d'un dictionnaire :

```
// Determine if this book is of derived type
      if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
          // Find the value of the xsi:type attribute
          string xsiTypeValue = book.Node.Attributes.GetNamedItem("xsi:type").Value;
          // Get the namespace URI and the lookup prefix of this namespace
          string namespaceUri = book.Node.NamespaceURI;
          string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);
          // if this book has DictionaryType
          if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
          {
             // output additional fields
             DictionaryType dictionary = new DictionaryType(book.Node);
            Console.WriteLine("Language from: " + dictionary.FromLang.First.Value);
            Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
         else
          {
              throw new Exception("Unexpected book type");
```

# 15.6.5.3 Lire et écrire des documents XML (Java)

Après avoir généré du code depuis le schéma de bibliothèque (voir <u>Schéma d'exemple<sup>(1121</sup>)</u>, un projet de text Java est créé, avec plusieurs bibliothèques Altova.

## À propos des bibliothèques Java générées

La classe centrale du code généré est la classe Doc2 qui représente le document XML. Une telle classe est générée pour tous les schémas et son nom dépend du nom de fichier de schéma. Veuillez noter que cette classe est appelée Doc2 pour éviter un conflit possible avec le nom d'espace de noms. Comme indiqué dans le diagramme, cette classe fournit des méthodes pour charger les documents depuis des fichiers, des streams binaires ou des strings (ou pour enregistrer des documents dans des fichiers, des streams, des strings). Pour consulter une description de cette classe, voir la référence de classe <u>com. [YourSchema].[Doc]</u>

	Doc2
	Library: MemberElement_Library
•	getStaticInfo():ComplexType
	loadFromFile(in filename:String):Doc2
	loadFromString(in xmlstring:String):Doc2
	loadFromBinary(in binary:byte[*]):Doc2
	saveToFile(in filename:String, in prettyPrint:boolean):void
	saveToFile(in filename:String, in prettyPrint:boolean, in omitXmIDecl:boolean):void
	saveToFile(in filename:String, in prettyPrint:boolean, in encoding:String):void
	saveToFile(in filename:String, in prettyPrint:boolean, in omitXmIDecl:boolean, in encoding:String):void
	saveToFile(in filename:String, in prettyPrint:boolean, in encoding:String, in bBigEndian:boolean, in bBOM:boolean):void
	saveToFile(in filename:String, in prettyPrint:boolean, in omitXmIDecl:boolean, in encoding:String, in bBigEndian:boolean, in bBOM:boolean]:void
	saveToString(in prettyPrint:boolean):String
	saveToString(in prettyPrint:boolean, in omitXmIDecl:boolean):String
	saveToBinary(in prettyPrint:boolean):byte[*]
	saveToBinary(in prettyPrint:boolean, in encoding:String):byte[*]
	saveToBinary(in prettyPrint:boolean, in encoding:String, in bBigEndian:boolean, in bBOM:boolean):byte[*]
	createDocument():Doc2
	setSchemaLocation(in schemaLocation:String):void
	<u>declareAlINamespacesFromSchema(in node:TypeBase):void</u>
	«constructor» Doc2(in init:org.w3c.dom.Node)
	setXsiType():void
	«static» MemberElement_Library

Le membre Library de la classe Doc2 représente la véritable racine du document.

Conformément aux règles de génération de code mentionnées dans À propos du code Java généré<sup>[110]</sup>, les membres de classes sont générés pour chaque attribut et pour chaque élément d'un type. Dans le code généré, le nom de ce type de classes de membre est préfixé avec MemberAttribute\_ et MemberElement\_, respectivement. Dans le diagramme ci-dessus, des exemples de ce type de classes sont MemberAttribute\_ID et MemberElement\_Author, générés depuis l'élément **Author** et l'attribut **ID** d'un livre, respectivement. Ces classes vous permettent de manipuler à l'aide d'un programme les éléments et les attributs correspondants dans l'instance du document XML (par exemple, apposer, supprimer, définir une valeur, etc). Pour plus d'informations, voir la référence de classe <u>com.[YourSchema].</u>[YourSchemaType].MemberElement

Puisque le **DictionaryType** est un type complexe dérivé depuis **BookType** dans le schéma, cette relation se retrouve aussi dans les classes générées. Comme illustré dans le diagramme, la classe DictionaryType hérite la classe BookType.



Si votre schéma XML définit des types simples en tant qu'énumerations, les valeurs énumérées deviennent disponibles en tant que valeurs Enum dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un Enum membre de la classe BookFormatType.

## Écrire un document XML

- 1. Dans le menu Fichier d'Eclipse, cliquer sur Import, choisir Existing Projects into Workspace, et cliquer sur Next.
- 2. À côté de **Select root directory**, cliquer sur **Browse**, choisir le répertoire dans lequel vous souhaitez générer le code Java, et cliquer sur **Finish**.
- 3. Dans Eclipse Package Explorer, agrandir le package **com.LibraryTest** et ouvrir le fichier **LibraryTest.java**.

Tout en prototypant une application depuis un schéma XML changeant fréquemment, vous devrez éventuellement générer fréquemment du code dans le même répertoire, de manière à ce que les changements de schéma sont réfléchis immédiatement dans le code. Veuillez noter que l'application de test généré et les bibliothèques Altova sont écrasées à chaque fois que vous générez du code dans le même répertoire cible. C'est pourquoi il ne faut pas ajouter du code à l'application de test généré. Au lieu de cela, veuillez intégrer les bibliothèques dans votre projet (voir Intégrer des Bibliothèques Schema Wrapper<sup>(118)</sup>).

4. Éditer la méthode Example() comme indiqué ci-dessous.

```
protected static void example() throws Exception {
      // create a new, empty XML document
     Doc2 libDoc = Doc2.createDocument();
      // create the root element <Library> and add it to the document
     LibraryType lib = libDoc.Library.append();
      // set the "LastUpdated" attribute
      com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
      lib.LastUpdated.setValue(dt);
      // create a new <Book> and populate its elements and attributes
      BookType book = lib.Book.append();
      book.ID.setValue(java.math.BigInteger.valueOf(1));
     book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
     book.Title.append().setValue("The XML Spy Handbook");
      book.Author.append().setValue("Altova");
      // create a dictionary (book of derived type) and populate its elements and
attributes
     DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
     dict.ID.setValue(java.math.BigInteger.valueOf(2));
     dict.Title.append().setValue("English-German Dictionary");
     dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
      dict.Author.append().setValue("John Doe");
      dict.FromLang.append().setValue("English");
      dict.ToLang.append().setValue("German");
     dict.setXsiType();
      // set the schema location (this is optional)
     libDoc.setSchemaLocation("Library.xsd");
     // save the XML document to a file with default encoding (UTF-8). "true" causes the
file to be pretty-printed.
      libDoc.saveToFile("Library1.xml", true);
 }
```

 Construire le projet Java et l'exécuter. Si le code a été exécuté avec succès, un fichier Library1.xml est créé dans le répertoire de projet.

#### Lire un document XML

- 1. Dans le menu Fichier d'Eclipse, cliquer sur Import, choisir Existing Projects into Workspace, et cliquer sur Next.
- À côté de Select root directory, cliquer sur Browse, choisir le répertoire dans lequel vous souhaitez générer le code Java, et cliquer sur Finish.
- 3. Enregistrer le code ci-dessous en tant que Library1.xml dans un répertoire local (vous devrez vous référer au chemin du fichier Library1.xml depuis l'échantillon de code ci-dessous).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
```

- 4. Dans Eclipse Package Explorer, agrandir le package **com.LibraryTest** et ouvrir le fichier **LibraryTest.java**.
- 5. Éditer la méthode Example() comme indiqué ci-dessous.

```
protected static void example() throws Exception {
     // load XML document from a path, make sure to adjust the path as necessary
    Doc2 libDoc = Doc2.loadFromFile("Library1.xml");
     // get the first (and only) root element <Library>
    LibraryType lib = libDoc.Library.first();
     // check whether an element exists:
     if (!lib.Book.exists()) {
       System.out.println("This library is empty.");
       return;
     }
     // read a DateTime schema type
     com.altova.types.DateTime dt = lib.LastUpdated.getValue();
     System.out.println("The library was last updated on: " + dt.toDateString());
     // iteration: for each <Book>...
     for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
       BookType book = (BookType) itBook.next();
        // output values of ID attribute and (first and only) title element
        System.out.println("ID: " + book.ID.getValue());
        System.out.println("Title: " + book.Title.first().getValue());
        // read and compare an enumeration value
        if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
           System.out.println("This is a paperback book.");
        // for each <Author>...
        for (java.util.Iterator itAuthor = book.Author.iterator(); itAuthor
              .hasNext();)
           System.out.println("Author: " + ((com.Doc.xs.stringType)
itAuthor.next()).getValue());
```

```
// find the derived type of this book
        // by looking at the value of the xsi:type attribute, using DOM
        org.w3c.dom.Node bookNode = book.getNode();
        if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
           // Get the value of the xsi:type attribute
           String xsiTypeValue =
bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();
           // Get the namespace URI and lookup prefix of this namespace
           String namespaceUri = bookNode.getNamespaceURI();
          String lookupPrefix = bookNode.lookupPrefix(namespaceUri);
           // If xsi:type matches the namespace URI and type of the book node
           if (namespaceUri == "http://www.nanonull.com/LibrarySample"
                 && ( xsiTypeValue.equals(lookupPrefix + ":DictionaryType" )))
                                                                                   {
              // ...then this is a book of derived type (dictionary)
              DictionaryType dictionary = new DictionaryType( book.getNode());
              // output the value of the "FromLang" and "ToLang" elements
              System.out.println("From language: " +
dictionary.FromLang.first().getValue());
              System.out.println("To language: " + dictionary.ToLang.first().getValue());
           else
              // throw an error
              throw new java.lang.Error("This book has an unknown type.");
           }
        }
     }
   }
```

6. Construire le projet Java et l'exécuter. Si le code est exécuté avec succès, Library1.xml sera lu par le code de programme, et ses contenus seront affichés dans le mode Console.

#### Lire et écrire des éléments et des attributs

Les valeurs des éléments et des attributs peut être accédé en utilisant la méthode getValue() de la classe d'élément ou d'attribut du membre généré, par exemple :



Pour obtenir la valeur de l'élément **Title** dans cet exemple particulier, nous avons aussi utilisé la méthode first(), étant donné que c'est le premier (et seul) élément **Title** d'un livre. Dans les cas où vous devez choisir un élément spécifique depuis une liste par index, utiliser la méthode at().

Pour itérer dans plusieurs éléments, utiliser soit une itération basée sur l'index ou java.util.Iterator. Par exemple, vous pouvez itérer dans les livres d'une bibliothèque comme suit :

```
// index-based iteration
for (int j = 0; j < lib.Book.count(); ++j ) {
    // your code here
}
// alternative iteration using java.util.Iterator
for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
    // your code here
   }
</pre>
```

Pour ajouter un nouvel élément, utiliser la méthode Append(). Par exemple, le code suivant appose un élément de racine videLibrary dans le document :

```
// create the root element <Library> and add it to the document
LibraryType lib = libDoc.Library3.append();
```

Une fois qu'un élément est apposé, vous pouvez définir la valeur d'un de ses éléments ou un attribut en utilisant la méthode setValue().

```
// set the value of the Title element
book.Title.append().setValue("The XML Spy Handbook");
// set the value of the ID attribute
book.ID.setValue(java.math.BigInteger.valueOf(1));
```

#### Lire et écrire des valeurs d'énumération

Si votre schéma XML définit des types simples en tant qu'énumerations, les valeurs énumérées deviennent disponibles en tant que valeurs Enum dans le code généré. Dans le schéma utilisé dans ces exemple, un format de livre peut être hardcover, paperback, e-book, etc. Dans le code généré, ces valeurs peuvent donc être disponibles via un Enum (voir le diagramme de classe BookFormatType ci-dessus). Pour attribuer des valeurs d'énumération à un objet, utiliser du code comme celui ci-dessous :

```
// set an enumeration value
book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
```

Vous pouvez lire ces valeurs d'énumération provenent de documents d'instance XML comme suit :

```
// read an enumeration value
if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
System.out.println("This is a paperback book."
```

Si une condition "if" n'est pas suffisante, créer un interrupteur pour déterminer chaque valeur d'énumération et traiter comme requis.

#### Travailler avec des types xs:dateTime et xs:duration

Si le schéma à partir duquel vous avez généré du code utilise les types heure et durée comme xs:dateTime, ou xs:duration, ils sont convertis en classes natives Altova dans un code généré. C'est pourquoi, pour écrire une valeur date ou durée vers le document XML, procéder comme suit :

- 1. Construit un objet <u>com.altova.types.DateTime</u><sup>(1184)</sup> OU <u>com.altova.types.Duration</u><sup>(1189)</sup>.
- 2. Définir l'objet en tant que la valeur de l'élément ou de l'attribut requis, par exemple :

```
// set the value of an attribute of DateTime type
com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
lib.LastUpdated.setValue(dt);
```

Pour lire une date ou une durée depuis un document XML :

- 1. Déclarer la valeur d'élément (ou d'attribut) comme objet <u>com.altova.types.DateTime</u><sup>1189</sup> ou com.altova.types.Duration<sup>1189</sup>.
- 2. Formate l'élément ou l'attribut requis, par exemple :

```
// read a DateTime type
com.altova.types.DateTime dt = lib.LastUpdated.getValue();
System.out.println("The library was last updated on: " + dt.toDateString());
```

Pour plus d'informations, voir la référence de classe <u>com.altova.types.DateTime</u><sup>1184</sup> et <u>com.altova.types.Duration</u><sup>1188</sup>.

#### Travailler avec des types dérivés

Si votre schéma XML définit des types dérivés, vous pouvez préserver la dérivation de type dans des documents XML que vous créez ou chargez par le biais d'un programme. Prendre le schéma utilisé dans cet exemple, l'extrait de code suivant illustre comment créer un nouveau livre de type dérivéDictionaryType:

```
// create a dictionary (book of derived type) and populate its elements and attributes
DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
dict.ID.setValue(java.math.BigInteger.valueOf(2));
dict.Title.append().setValue("English-German Dictionary");
dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
dict.Author.append().setValue("John Doe");
dict.FromLang.append().setValue("English");
dict.ToLang.append().setValue("German");
dict.setXsiType();
```

Veuillez noter qu'il est important de définir l'attribut xsi:type du livre récemment créé. Cela garantit que le type de livre sera interprété correctement par le schéma losrque le document XML est validé.

Lorsque vous chargez des données depuis un document XML, l'extrait de code suivant montre comment identifier un livre de type dérivé DictionaryType dans l'instance XML chargée. Premièrement, le code trouve la valeur de l'attribut xsi:type du nœud de livre. Si l'espace de noms URI de ce nœud est http://www.nanonull.com/LibrarySample et si le préfixe et type de lookup URI correspond à la valeur de l'attribut xsi:type, alors il s'agit d'un dictionnaire :

```
// find the derived type of this book
// by looking at the value of the xsi:type attribute, using DOM
org.w3c.dom.Node bookNode = book.getNode();
if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
```

```
// Get the value of the xsi:type attribute
String xsiTypeValue = bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();
// Get the namespace URI and lookup prefix of the book node
String namespaceUri = bookNode.getNamespaceURI();
String lookupPrefix = bookNode.lookupPrefix(namespaceUri);
// If xsi:type matches the namespace URI and type of the book node
if (namespaceUri == "http://www.nanonull.com/LibrarySample"
&& ( xsiTypeValue.equals(lookupPrefix + ":DictionaryType" ))) {
// ...then this is a book of derived type (dictionary)
DictionaryType dictionary = new DictionaryType( book.getNode());
// output the value of the "FromLang" and "ToLang" elements
System.out.println("From language: " + dictionary.FromLang.first().getValue());
System.out.println("To language: " + dictionary.ToLang.first().getValue());
}
else
// throw an error
throw new java.lang.Error("This book has an unknown type.");
```

# 15.6.6 Exemple: Purchase Order

Cet exemple illustre comment travailler avec le code de programme généré depuis un schéma XML "main" qui importe d'autres schémas. Chacun des schémas importés a un espace de noms cible différent. L'objectif est de créer par programme un document XML dans lequel tous les éléments sont préfixés conformément à leur espace de noms. Plus précisément, le document XML créé depuis votre code C++, C# ou Java doit ressembler à l'exemple ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
representation="http://NamespaceTest.com/Purchase Main.xsd"
           xmlns:p="http://NamespaceTest.com/Purchase"
           xmlns:o="http://NamespaceTest.com/OrderTypes"
           xmlns:c="http://NamespaceTest.com/CustomerTypes"
           xmlns:cmn="http://NamespaceTest.com/CommonTypes"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <p:OrderDetail>
       <o:Item>
           <o:ProductName>Lawnmower</o:ProductName>
           <o:Quantity>1</o:Quantity>
           <o:UnitPrice>148.42</o:UnitPrice>
       </o:Item>
   </p:OrderDetail>
   <p:PaymentMethod>VISA</p:PaymentMethod>
   <p:CustomerDetails>
       <c:Name>Alice Smith</c:Name>
       <c:DelivervAddress>
           <cmn:Line1>123 Maple Street</cmn:Line1>
```

```
<cmn:Line2>Mill Valley</cmn:Line2>
</c:DeliveryAddress>
<c:BillingAddress>
<cmn:Line1>8 Oak Avenue</cmn:Line1>
<cmn:Line2>Old Town</cmn:Line2>
</c:BillingAddress>
</p:CustomerDetails>
</p:Purchase>
```

Le schéma principal utilisé dans ces exemple est appelé **Main.xsd**. Comme indiqué dans l'extrait de code cidessous, il importe trois autres schémas : **CommonTypes.xsd**, **CustomerTypes.xsd** et **OrderTypes.xsd**. Pour obtenir les mêmes résultats que dans cet exemple, enregistrer tous les extraits de code ci-dessous dans des fichiers et utiliser les mêmes noms de fichier que ci-dessus. Veuillez noter que le schéma mappe chacun des préfixes **ord**, **pur**, **cmn**, et **cust** vers des espaces de noms (Order types, Purchase types, Common types, et Customer types, respectivement). Cela signifie que, dans le code généré, les classes correspondant à Orders, Purchases, Customers, etc, seront disponibles sous leur espace de noms respectif.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
      targetNamespace="http://NamespaceTest.com/Purchase"
      xmlns:ord="http://NamespaceTest.com/OrderTypes"
      xmlns:pur="http://NamespaceTest.com/Purchase"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      xmlns:cust="http://NamespaceTest.com/CustomerTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"</pre>
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:import schemaLocation="CustomerTypes.xsd"</pre>
namespace="http://NamespaceTest.com/CustomerTypes" />
    <xs:import schemaLocation="OrderTypes.xsd"</pre>
namespace="http://NamespaceTest.com/OrderTypes" />
    <xs:element name="Purchase">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="OrderDetail" type="ord:OrderType" />
                <xs:element name="PaymentMethod" type="cmn:PaymentMethodType" />
                <xs:element ref="pur:CustomerDetails" />
            </xs:sequence>
        </xs:complexType>
    </r></r></r>
    <xs:element name="CustomerDetails" type="cust:CustomerType" />
</xs:schema>
```

Main.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://NamespaceTest.com/CommonTypes"
    elementFormDefault="qualified">
    <xs:complexType name="AddressType">
        <xs:complexType name="AddressType">
        <xs:sequence>
```

```
<xs:element name="Line1" type="xs:string"/>
         <xs:element name="Line2" type="xs:string"/>
      </xs:sequence>
  </xs:complexType>
   <xs:simpleType name="PriceType">
      <xs:restriction base="xs:decimal">
         <xs:fractionDigits value="2"/>
      </ms:restriction>
   </xs:simpleType>
   <xs:simpleType name="PaymentMethodType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="VISA"/>
         <xs:enumeration value="MasterCard"/>
         <xs:enumeration value="Cash"/>
         <xs:enumeration value="AMEX"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

CommonTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://NamespaceTest.com/CustomerTypes"
xmlns:cmn="http://NamespaceTest.com/CommonTypes"
elementFormDefault="qualified">
<xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
<xs:complexType name="CustomerType">
<xs:complexType name="CustomerType">
<xs:sequence>
<xs:element name="Name" type="xs:string" />
<xs:element name="DeliveryAddress" type="cmn:AddressType" />
<xs:sequence>
</xs:sequence>
</xs:sequence>
</xs:scomplexType>
</xs:schema>
```

CustomerTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://NamespaceTest.com/OrderTypes"
wmlns:cmn="http://NamespaceTest.com/CommonTypes"
elementFormDefault="qualified">
<xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
<xs:complexType name="OrderType">
<xs:complexType name="OrderType">
<xs:sequence>
<xs:element maxOccurs="unbounded" name="Item">
<xs:complexType>
<xs:sequence>
```

#### OrderTypes.xsd

Pour terminer cet exemple, suivre les étapes suivantes :

- 1. Enregistrer tous les schémas depuis les extraits de code ci-dessus dans les fichiers sur le disque, en vous assurant que vous préservez les noms de fichier indiqués.
- Générer le code de schema wrapper depuis le schéma Main.xsd ci-dessus, en utilisant les étapes décrites dans <u>Générer du code depuis des Schémas XML ou des DTD</u><sup>(108)</sup>. Une fois cette étape achevée, vous aurez généré un programme compilable dans le langage de votre choix (C++, C#, ou Java).
- Ajouter du code dans votre programme C++, C# ou Java depuis un des extraits de code d'exemple, le cas échéant :
  - Espaces de noms XML et Préfixes (C++)<sup>1148</sup>
  - Espaces de noms XML et Préfixes (C#)<sup>[1150]</sup>
  - Espaces de noms XML et Préfixes (Java)<sup>(1151)</sup>

# 15.6.6.1 Espaces de noms XML et Préfixes (C++)

Après avoir généré du code depuis le <u>schéma d'exemple</u><sup>[1145</sup>], une application C++ test est créée, avec plusieurs bibliothèques Altova de support. Rappelez-vous que le schéma d'exemple (**Main.xsd**) a plusieurs déclarations d'espace de noms. Par conséquent, le code généré comprend des espaces de noms qui correspondent aux alias d'espace de noms (préfixes) provenant du schéma : <u>Main::ord</u>, <u>Main::pur</u>, <u>Main::cmn</u> et <u>Main::cust</u>.

En général, afin de contrôler les espaces de noms XML et les préfixes avec l'aide des bibliothèques de schéma, vous disposez des méthodes suivantes :

- <u>DeclareAllNamespacesFromSchema()</u><sup>(113)</sup>. Appeler cette méthode si vous souhaitez déclarer les mêmes espaces de noms dans votre instance XML que dans le schéma. Sinon, si vous souhaitez utiliser d'autres espaces de noms que ceux utilisés dans cet exemple, vous devriez utiliser <u>DeclareNamespace()</u>. La méthode <u>DeclareAllNamespacesFromSchema()</u> n'est pas utilisée dans cet exemple parce que nous souhaitons utiliser spécifiquement des éléments XML avec des préfixes qui sont légèrement différents de ceux déclarés dans le schéma.
- <u>DeclareNamespace()</u><sup>[115]</sup>. Appeler cette méthode pour créer ou contourner l'attribut de préfixe d'espace de noms existant sur un élément. L'élément doit déjà être créé en utilisant soit la méthode append() soit appendWithPrefix(), comme illustré ci-dessous.
- <u>appendWithPrefix().</u><sup>[1167]</sup> Utiliser cette méthode pour apposer un élément d'instance avec un préfixe spécifique. Pour créer l'instance XML illustrée dans cet exemple, il a suffit d'appeler cette méthode pour l'élément root uniquement. Tous les autres éléments ont été apposés en utilisant uniquement

<u>append()</u><sup>(167)</sup>, et leurs préfixes ont été ajoutés automatiquement sur la base de leurs espaces de noms, conformément aux règles ci-dessus.

L'extrait de code ci-dessous vous montre comment créer un document XML avec plusieurs déclarations d'espace de noms et des noms d'élément préfixés. En particulier, il génère une instance Purchase Order comme illustré dans l'<u>Example : Purchase Order</u> <sup>113</sup>. Plus important encore, à des fins d'illustration, certains préfixes sont contournés dans l'instance XML (c'est à dire qu'ils ne sont tout à fait identiques que ceux déclarés dans le schéma).

```
void Example()
{
  // Create the XML document and append the root element
  Main::pur::CMain doc = Main::pur::CMain::CreateDocument();
  Main::pur::CPurchaseType purchase = doc.Purchase.appendWithPrefix(_T("p"));
  // Set schema location
  doc.SetSchemaLocation(_T("Main.xsd"));
  // Declare namespaces on root element
  purchase.DeclareNamespace(_T("o"), _T("http://NamespaceTest.com/OrderTypes"));
  purchase.DeclareNamespace(_T("c"), _T("http://NamespaceTest.com/CustomerTypes"));
  purchase.DeclareNamespace(_T("cmn"), _T("http://NamespaceTest.com/CommonTypes"));
  // Append the OrderDetail element
  Main::ord::COrderType order = purchase.OrderDetail.append();
  Main::ord::CItemType item = order.Item.append();
  item.ProductName.append() = _T("Lawnmower");
  item.Quantity.append() = 1;
  item.UnitPrice.append() = 148.42;
  // Append the PaymentMethod element
  Main::cmn::CPaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
  paymentMethod.SetEnumerationValue(Main::cmn::CPaymentMethodTypeType::k_VISA);
  // Append the CustomerDetails element
  Main::cust::CCustomerType customer = purchase.CustomerDetails.append();
  customer.Name.append() = _T("Alice Smith");
  Main::cmn::CAddressType deliveryAddress = customer.DeliveryAddress.append();
  deliveryAddress.Line1.append() = _T("123 Maple Street");
  deliveryAddress.Line2.append() = _T("Mill Valley");
  Main::cmn::CAddressType billingAddress = customer.BillingAddress.append();
  billingAddress.Line1.append() = _T("8 Oak Avenue");
  billingAddress.Line2.append() = _T("Old Town");
  // Save to file and release object from memory
  doc.SaveToFile(_T("Main1.xml"), true);
  doc.DestroyDocument();
}
```

## 15.6.6.2 Espaces de noms XML et Préfixes (C#)

Après avoir généré du code depuis le schéma de bibliothèque (voir <u>Schéma d'exemple</u><sup>1145</sup>), une application C# test est créée, avec plusieurs bibliothèques Altova. Rappelez-vous que le schéma d'exemple (**Main.xsd**) a plusieurs déclarations d'espace de noms. Par conséquent, le code généré comprend des espaces de noms qui correspondent aux alias d'espace de noms (préfixes) provenant du schéma : <u>Main.ord</u>, <u>Main.pur</u>, <u>Main.cmn</u>, et <u>Main.cust</u>.

En général, afin de contrôler les espaces de noms XML et les préfixes avec l'aide des bibliothèques de schéma, vous disposez des méthodes suivantes :

- <u>DeclareAllNamespacesFromSchema()</u><sup>IIII</sup>. Appeler cette méthode si vous souhaitez déclarer les mêmes espaces de noms dans votre instance XML que dans le schéma. Sinon, si vous souhaitez utiliser d'autres espaces de noms que ceux utilisés dans cet exemple, vous devriez utiliser <u>DeclareNamespace()</u>. La méthode <u>DeclareAllNamespacesFromSchema()</u> n'est pas utilisée dans cet exemple parce que nous souhaitons utiliser spécifiquement des éléments XML avec des préfixes qui sont légèrement différ<u>en</u>ts de ceux déclarés dans le schéma.
- <u>DeclareNamespace()</u><sup>(112)</sup>. Appeler cette méthode pour créer ou contourner l'attribut de préfixe d'espace de noms existant sur un élément. L'élément doit déjà être créé en utilisant soit la méthode <u>Append()</u> soit <u>AppendWithPrefix()</u>, comme illustré ci-dessous.
- <u>AppendWithPrefix().</u><sup>(IIII)</sup> Utiliser cette méthode pour apposer un élément d'instance avec un préfixe spécifique. Pour créer l'instance XML illustrée dans cet exemple, il a suffit d'appeler cette méthode pour l'élément root uniquement. Tous les autres éléments ont été apposés en utilisant uniquement <u>Append()</u><sup>(IIIII)</sup>, et leurs préfixes ont été ajoutés automatiquement sur la base de leurs espaces de noms, conformément aux règles ci-dessus.

L'extrait de code ci-dessous vous montre comment créer un document XML avec plusieurs déclarations d'espace de noms et des noms d'élément préfixés. En particulier, il génère une instance Purchase Order comme illustré dans l'<u>Example : Purchase Order</u> <sup>1145</sup>. Plus important encore, à des fins d'illustration, certains préfixes sont contournés dans l'instance XML (c'est à dire qu'ils ne sont tout à fait identiques que ceux déclarés dans le schéma).

```
protected static void Example()
{
  // Create the XML document and append the root element
  pur.Main2 doc = pur.Main2.CreateDocument();
  pur.PurchaseType purchase = doc.Purchase.AppendWithPrefix("p");
   // Set schema location
  doc.SetSchemaLocation(@"Main.xsd");
  // Declare namespaces on root element
  purchase.DeclareNamespace("o", "http://NamespaceTest.com/OrderTypes");
  purchase.DeclareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
  purchase.DeclareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");
  // Append the OrderDetail element
  ord.OrderType order = purchase.OrderDetail.Append();
  ord.ItemType item = order.Item.Append();
  item.ProductName.Append().Value = "Lawnmower";
  item.Quantity.Append().Value = 1;
```

}

```
item.UnitPrice.Append().Value = 148.42M;
// Append the PaymentMethod element
cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.Append();
paymentMethod.EnumerationValue = cmn.PaymentMethodTypeType.EnumValues.eVISA;
// Append the CustomerDetails element
cust.CustomerType customer = purchase.CustomerDetails.Append();
customer.Name.Append().Value = "Alice Smith";
cmn.AddressType deliveryAddress = customer.DeliveryAddress.Append();
deliveryAddress.Line1.Append().Value = "123 Maple Street";
deliveryAddress.Line2.Append().Value = "Mill Valley";
cmn.AddressType billingAddress = customer.BillingAddress.Append();
billingAddress.Line1.Append().Value = "8 Oak Avenue";
billingAddress.Line2.Append().Value = "Old Town";
// Save to file
doc.SaveToFile("PurchaseOrder.xml", true);
```

# 15.6.6.3 Espaces de noms XML et Préfixes (Java)

Après avoir généré du code depuis le schéma de bibliothèque (voir <u>Schéma d'exemple</u><sup>1145</sup>), une application Java de test est créée, avec plusieurs bibliothèques Altova. Rappelez-vous que le schéma d'exemple (**Main.xsd**) a plusieurs déclarations d'espace de noms. Par conséquent, le code généré comprend des espaces de noms qui correspondent aux alias d'espace de noms (préfixes) provenant du schéma : com.Main.ord, com.Main.pur, com.Main.cmn, et com.Main.cust.

En général, afin de contrôler les espaces de noms XML et les préfixes avec l'aide des bibliothèques de schéma, vous disposez des méthodes suivantes :

- <u>declareAllNamespacesFromSchema()</u><sup>[115]</sup>. Appeler cette méthode si vous souhaitez déclarer les mêmes espaces de noms dans votre instance XML que dans le schéma. Sinon, si vous souhaitez utiliser d'autres espaces de noms que ceux utilisés dans cet exemple, vous devriez utiliser <u>declareNamespace()</u>. La méthode <u>declareAllNamespacesFromSchema()</u> n'est pas utilisée dans cet exemple parce que nous souhaitons utiliser spécifiquement des éléments XML avec des préfixes qui sont légèrement différ<u>en</u>ts de ceux déclarés dans le schéma.
- <u>declareNamespace()</u><sup>[113]</sup>. Appeler cette méthode pour créer ou contourner l'attribut de préfixe d'espace de noms existant sur un élément. L'élément doit déjà être créé en utilisant soit la méthode append() soit appendWithPrefix(), comme illustré ci-dessous.
- <u>appendWithPrefix().</u><sup>[1139]</sup> Utiliser cette méthode pour apposer un élément d'instance avec un préfixe spécifique. Pour créer l'instance XML illustrée dans cet exemple, il a suffit d'appeler cette méthode pour l'élément root uniquement. Tous les autres éléments ont été apposés en utilisant uniquement <u>append()</u><sup>[113]</sup>, et leurs préfixes ont été ajoutés automatiquement sur la base de leurs espaces de noms, conformément aux règles ci-dessus.

L'extrait de code ci-dessous vous montre comment créer un document XML avec plusieurs déclarations d'espace de noms et des noms d'élément préfixés. En particulier, il génère une instance Purchase Order comme illustré dans l'<u>Example : Purchase Order</u> <sup>116</sup>. Plus important encore, à des fins d'illustration, certains préfixes sont contournés dans l'instance XML (c'est à dire qu'ils ne sont tout à fait identiques que ceux déclarés dans le schéma).

```
protected static void example() throws Exception {
   // Create the XML document and append the root element
   com.Main.pur.Main2 doc = com.Main.pur.Main2.createDocument();
   com.Main.pur.PurchaseType purchase = doc.Purchase.appendWithPrefix("p");
   // Set schema location
   doc.setSchemaLocation("Main.xsd");
  // Declare namespaces on root element
   purchase.declareNamespace("o", "http://NamespaceTest.com/OrderTypes");
   purchase.declareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
   purchase.declareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");
   // Append the OrderDetail element
   com.Main.ord.OrderType order = purchase.OrderDetail.append();
   com.Main.ord.ItemType item = order.Item.append();
   item.ProductName.append().setValue("Lawnmower");
   item.Quantity.append().setValue(1);
   java.math.BigDecimal price = new java.math.BigDecimal("148.42");
  item.UnitPrice.append().setValue(price);
  // Append the PaymentMethod element
   com.Main.cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
   paymentMethod.setEnumerationValue(com.Main.cmn.PaymentMethodTypeType.EVISA);
   // Append the CustomerDetails element
   com.Main.cust.CustomerType customer = purchase.CustomerDetails.append();
  customer.Name.append().setValue("Alice Smith");
   com.Main.cmn.AddressType deliveryAddress = customer.DeliveryAddress.append();
   deliveryAddress.Line1.append().setValue("123 Maple Street");
   deliveryAddress.Line2.append().setValue("Mill Valley");
   com.Main.cmn.AddressType billingAddress = customer.BillingAddress.append();
   billingAddress.Line1.append().setValue("8 Oak Avenue");
  billingAddress.Line2.append().setValue("Old Town");
   // Save to file
   doc.saveToFile("PurchaseOrder.xml", true);
}
```
# 15.7 Référence aux classes générées (C++)

Ce chapitre contient une description des classes C++ avec MapForce depuis un Schéma XML ou DTD (voir <u>Générer du code depuis des Schémas XML ou DTDs</u> (100)). Vous pouvez intégrer ces classes dans votre code pour lire, modifier et écrire des documentation XML.

**Note:** Le code généré peut inclure d'autres classes de support, qui ne sont pas recensées ici et qui sont sujettes à des modifications.

## 15.7.1 altova::DateTime

Cette classe vous permet de traiter des attributs XML ou des éléments qui ont des types de date et d'heure, comme xs:dateTime.

#### Constructeurs

Nom	Description
DateTime()	Initialise une nouvelle instance de la classe DateTime à 12:00:00 minuit, le 1er janvier, 0001.
DateTime(int64 value, short timezone)	Initialise une nouvelle instance de la classe DateTime. Le paramètre value représente le nombre de tics (intervales de 100 nanosecondes) qui se sont écoulés depuis 12:00:00 minuit, le 1er janvier, 0001.
DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second)	Initialise une nouvelle instance de la classe DateTime à l'année, mois, jour, heure, minute et seconde fournis en tant qu'argument.
DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second, short timezone)	Initialise une nouvelle instance de la classe DateTime à l'année, mois, jour, heure, minute, seconde et fuseau horaire fournis en tant qu'argument. Le fuseau horaire est exprimé en minutes et peut être positif ou négatif. Par exemple, le fuseau horaire "UTC-01:00" est exprimé en tant que "-60".

Nom	Description
unsigned char Day() const	Retourne le jour du mois de l'objet DateTime actuel. Les valeurs de retour s'étendent de 1 à 31.
int DayOfYear() const	Retourne le jour de l'année de l'objet DateTime actuel. Les valeurs de retour s'étendent de 1 à 366.
bool HasTimezone() const	Retourne Booléenne <b>true</b> si l'objet DateTime actuel présente un fuseau horaire défini ; <b>false</b> sinon.

Nom	Description
unsigned char Hour() const	Retourne l'heure de l'objet DateTime actuel. Les valeurs de retour s'étendent de 0 à 23.
static bool IsLeapYear(int year)	Retourne Booléenne true si l'année de la classe DateTime est une année bissextile ; false sinon.
unsigned char Minute() const	Retourne la minute de l'objet DateTime actuel. Les valeurs de retour s'étendent de 0 à 59.
unsigned char Month() const	Retourne le mois de l'objet DateTime actuel. Les valeurs de retour s'étendent de 1 à 12.
int64 NormalizedValue() const	Retourne la valeur de l'objet DateTime exprimé en tant que Coordinated Universal Time (UTC).
double Second() const	Retourne la seconde de l'objet DateTime actuel. Les valeurs de retour s'étendent de 0 à 59.
void SetTimezone(short tz)	Définit le fuseau horaire de l'objet DateTime actuel à la valeur du fuseau horaire fourni en tant qu'argument. L'argument <b>tz</b> est exprimé en minutes et peut êtr positif ou négatif.
short Timezone() const	Retourne le fuseau horaire, en minutes de l'objet DateTime actuel. Avant d'utiliser cette méthode, veuillez vous assurer que l'objet comporte réellement un fuseau horaire, en appelan la méthode HasTimezone().
int64 Value() const	Retourne la valeur de l'objet DateTime object, exprimé dans le nombre de tics (intervalles de 100 nanosecondes) qui se sont écoulés depuis 12:00:00 minuit, le 1er janvier, 0001.
int Weekday() const	Retourne le jour de la semaine de l'objet DateTime actuel, en tant qu'un entier. Les valeurs s'étendent de 0 à 6, 0 étant lundi (ISO- 8601).
int Weeknumber() const	Retourne le nombre de semaines dans l'année de l'objet DateTime actuel. Les valeurs de retour sont conformes à ISO-8601.
int WeekOfMonth() const	Retourne le nombre de semaines dans le mois de l'objet DateTime actuel. Les valeurs de retour sont conformes à ISO-8601.
int Year() const	Retoune l'année de l'objet DateTime actuel.

### Exemple

void Example()
{

// initialize a new DateTime instance to 12:00:00 midnight, January 1st, 0001
altova::DateTime dt1 = altova::DateTime();

// initialize a new DateTime instance using the year, month, day, hour, minute, and

```
second
   altova::DateTime dt2 = altova::DateTime(2015, 11, 10, 9, 8, 7);
   // initialize a new DateTime instance using the year, month, day, hour, minute,
second, and UTC +01:00 timezone
   altova::DateTime dt = altova::DateTime(2015, 11, 22, 13, 53, 7, 60);
   // Get the value of this DateTime object
   std::cout << "The number of ticks of the DateTime object is: " << dt.Value() <<</pre>
std::endl;
  // Get the year
   cout << "The year is: " << dt.Year() << endl;</pre>
   // Get the month
  cout << "The month is: " << (int)dt.Month() << endl;</pre>
  // Get the day of the month
   cout << "The day of the month is: " << (int) dt.Day() << endl;</pre>
   // Get the day of the year
  cout << "The day of the year is: " << dt.DayOfYear() << endl;</pre>
   // Get the hour
   cout << "The hour is: " << (int) dt.Hour() << endl;</pre>
   // Get the minute
  cout << "The minute is: " << (int) dt.Minute() << endl;</pre>
  // Get the second
   cout << "The second is: " << dt.Second() << endl;</pre>
   // Get the weekday
  cout << "The weekday is: " << dt.Weekday() << endl;</pre>
   // Get the week number
   cout << "The week of year is: " << dt.Weeknumber() << endl;</pre>
   // Get the week in month
   cout << "The week of month is: " << dt.WeekOfMonth() << endl;</pre>
   // Check whether a DateTime instance has a timezone
   if (dt.HasTimezone() == TRUE)
      // output the value of the Timezone
      cout << "The timezone is: " << dt.Timezone() << endl;</pre>
   }
   else
   {
      cout << "No timezone has been defined." << endl;</pre>
   // Construct a DateTime object with a timezone UTC+01:00 (Vienna)
  altova::DateTime vienna_dt = DateTime(2015, 11, 23, 14, 30, 59, +60);
   // Output the result in readable format
   cout << "The Vienna time: "
       << (int) vienna_dt.Month()
       << "-" << (int) vienna_dt.Day()
       << " " << (int) vienna_dt.Hour()
       << ":" << (int) vienna_dt.Minute()
       << ":" << (int) vienna_dt.Second()
       << endl;
```

```
// Convert the value to UTC time
DateTime utc_dt = DateTime(vienna_dt.NormalizedValue());
// Output the result in readable format
cout << "The UTC time:
                          п
   << (int) utc_dt.Month()
   << "-" << (int) utc_dt.Day()
   << " " << (int) utc_dt.Hour()
   << ":" << (int) utc_dt.Minute()
   << ":" << (int) utc_dt.Second()
   << endl;
// Check if a year is a leap year
int year = 2016;
if( altova::DateTime::IsLeapYear(year) )
{ cout << year << " is a leap year" << endl; }
else
{ cout << year << " is not a leap year" << endl; }
```

## 15.7.2 altova::Duration

Cette classe vous permet de traiter des attributs XML ou des éléments de type xs: duration .

#### Constructeurs

}

Nom	Description
Duration()	Initialise une nouvelle instance de la classe Duration dans une valeur vide.
Duration(const DayTimeDuration& dt)	Initialise une nouvelle instance de la classe Duration dans une durée définie par l'argument dt (voir <u>altova::DayTimeDuration</u> <sup>(1133)</sup> ).
Duration(const YearMonthDuration& ym)	Initialise une nouvelle instance de la classe Duration dans la durée définie par l'argument ym (voir <u>altova::YearMonthDuration</u> <sup>1159</sup> ).
Duration(const YearMonthDuration& ym, const DayTimeDuration& dt)	Initialise une nouvelle instance de la classe Duration dans les durées définies par les arguments dt et ym (voir <u>altova::YearMonthDuration</u> <sup>1159</sup> et <u>altova::DayTimeDuration</u> <sup>1159</sup> ).

Nom	Description
int Days() const	Retourne le nombre de jours dans l'instance Duration actuelle.
DayTimeDuration DayTime() const	Retourne le jour et la durée de temps dans l'instance Duration actuelle, exprimé en tant qu'un objet DayTimeDuration (voir <u>altova::DayTimeDuration</u> <sup>1155</sup> ).

Nom	Description
int Hours() const	Retourne le nombre d'heures dans l'instance Duration actuelle.
bool IsNegative() const	Retourne Boléenne <b>true</b> si l'instance Duration actuelle est négative.
bool IsPositive() const	Retourne Boléenne true si l'instance Duration actuelle est positive.
int Minutes() const	Retourne le nombre de minutes dans l'instance Duration actuelle.
int Months() const	Retourne le nombre de mois dans l'instance Duration actuelle.
double Seconds() const	Retourne le nombre de secondes dans l'instance Duration actuelle.
YearMonthDuration YearMonth() const	Retourne la durée d'année et de mois dans l'instance Duration actuelle, exprimée en tant qu'un objet YearMonthDuration (Voir altova::YearMonthDuration <sup>(153)</sup> ).
int Years() const	Retourne le nombre des années dans l'instance Duration actuelle.

## Exemple

L'extrait de code suivant illustre la création d'un nouvel objet Duration, ainsi que la lecture de valeurs en provenant.

```
void ExampleDuration()
{
   // Create an empty Duration object
   altova::Duration empty_duration = altova::Duration();
   // Create a Duration object using an existing duration value
   altova::Duration duration1 = altova::Duration(empty_duration);
   // Create a YearMonth duration of six years and five months
   altova::YearMonthDuration yrduration = altova::YearMonthDuration(6, 5);
   // Create a DayTime duration of four days, three hours, two minutes, and one second
   altova::DayTimeDuration dtduration = altova::DayTimeDuration(4, 3, 2, 1);
   // Create a Duration object by combining the two previously created durations
   altova::Duration duration = altova::Duration(yrduration, dtduration);
   // Get the number of years in this Duration instance
   cout << "Years: " << duration.Years() << endl;</pre>
   // Get the number of months in this Duration instance
   cout << "Months: " << duration.Months() << endl;</pre>
   // Get the number of days in this Duration instance
   cout << "Days: " << duration.Days() << endl;</pre>
   // Get the number of hours in this Duration instance
```

cout << "Hours: " << duration.Hours() << endl; // Get the number of hours in this Duration instance cout << "Minutes: " << duration.Minutes() << endl; // Get the number of seconds in this Duration instance cout << "Seconds: " << duration.Seconds() << endl;</pre>

# 15.7.3 altova::DayTimeDuration

Cette classe vous permet de traiter des types de durée de schéma XML qui consistent en une partie jour et une partie heure.

#### Constructeurs

}

Nom	Description
DayTimeDuration()	Initialise une nouvelle instance de la classe DayTimeDuration dans une valeur vide.
DayTimeDuration(int days, int hours, int minutes, double seconds)	Initialise une nouvelle instance de la classe DayTimeDuration dans le nombre de jours, heures, minutes, secondes fournis en tant qu'argument.
explicit DayTimeDuration(int64 value)	Initialise une nouvelle instance de la classe DayTimeDuration dans une durée qui consiste en autant de tics (intervalles de 100 nanosecondes) que fournis dans l'argument value.

Nom	Description
int Days() const	Retourne le nombre de jours dans l'instance DayTimeDuration actuelle.
int Hours() const	Retourne le nombre d'heures dans l'instance DayTimeDuration actuelle.
bool IsNegative() const	Retourne Booléenne true si l'instance DayTimeDuration actuelle est négative.
bool IsPositive() const	Retourne Booléenne true si l'instance DayTimeDuration actuelle est positive.
int Minutes() const	Retourne le nombre de minutes dans l'instance DayTimeDuration actuelle.
double Seconds() const	Retourne le nombre de secondes dans l'instance DayTimeDuration actuelle.

int64	Value(	) const	R
	varue(	/ CONSC	- 12

etourne la valeur (en tics) de l'instanceDayTimeDuration actuelle.

## 15.7.4 altova::YearMonthDuration

Cette classe vous permet de traiter des types de durée de schéma XML qui consistent en une partie année et mois.

#### Constructeurs

Nom	Description
YearMonthDuration()	Initialise une nouvelle instance de la classe YearMonthDuration dans une valeur vide.
YearMonthDuration(int years, int months)	Initialise une nouvelle instance de la classe YearMonthDuration dans le nombre des années et des mois fournis dans les arguments <b>années</b> et <b>mois</b> .
explicit YearMonthDuration(int value)	Initialise une nouvelle instance de la classe YearMonthDuration dans une durée qui consiste en autant de tics (intervalles de 100 nanosecondes) comme fournir dans l'argument <b>value</b> .

#### Méthodes

Nom	Description
bool IsNegative() const	Retourne Booléennen true si l'instance YearMonthDuration actuelle est négative.
bool IsPositive() const	Retourne Booléennen true si l'instance YearMonthDuration actuelle est positive.
int Months() const	Retourne le nombre de mois dans l'instance YearMonthDuration actuelle.
int Value() const	Retourne la valeur (en tics) de l'instance YearMonthDuration actuelle.
int Years()	Retourne le nombre d'années dans l'instance YearMonthDuration actuelle.

## 15.7.5 altova::meta::Attribute

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des attributs. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un attribut dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un attribut particulier défini dans le schéma XML.

### Méthodes

Nom	Description
SimpleType GetDataType()	Retourne le type du contenu d'attribut.
<pre>string_type GetLocalName()</pre>	Retourne le nom local de l'attribut.
<pre>string_type GetNamespaceURI()</pre>	Retourne l'URI d'espace de noms de l'attribut.
<pre>bool IsRequired()</pre>	Retourne true si l'attribut est requis.

## Opérateurs

Nom	Description
bool operator()	Retourne true si ce n'est pas l'attribut NULL.
<pre>bool operator!()</pre>	Retourne true s'il s'agit de l'attribut NULL.

## 15.7.6 altova::meta::ComplexType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types complexes. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types complexes dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type complexes particulier défini dans le schéma XML.

Nom	Description
Attribute FindAttribute(const char_type* localName, const char_type* namespaceURI)	Trouve l'attribut avec le nom local spécifié et l'URI d'espace de noms
Element FindElement(const char_type* localName, const char_type* namespaceURI)	Trouve l'élément avec le nom local spécifié et l'URI d'espace de noms.
<pre>std::vector<attribute> GetAttributes()</attribute></pre>	Retourne une liste de tous les attributs.
ComplexType GetBaseType()	Retourne le type de base de ce type.
SimpleType GetContentType()	Retourne le type simple du contenu.
<pre>std::vector<element> GetElements()</element></pre>	Retourne une liste de tous les éléments.
<pre>string_type GetLocalName()</pre>	Retourne le nom local du type.

Nom	Description
<pre>string_type GetNamespaceURI()</pre>	Retourne l'URI d'espace de noms du type.

## Opérateurs

Nom	Description
bool operator()	Retourne true si ce n'est pas le ComplexType NULL.
bool operator!()	Retourne true s'il s'agit du ComplexType NULL.

## 15.7.7 altova::meta::Element

Cette classe vous permet d'accéder à des informations concernant des classes générées depuis des éléments de schéma. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un élément dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un élément particulier défini dans le schéma XML.

## Méthodes

Nom	Description
ComplexType GetDataType()	Retourne le type de l'élément. Noter qu'il s'agit toujours d'un type de complexe même s'il est déclaré en tant que simple dans le schéma original. Utiliser GetContentType() de l'objet retourné pour obtenir le type de contenu simple.
<pre>string_type GetLocalName()</pre>	Retourne le nom local de l'élément.
unsigned int GetMaxOccurs()	Retourne la valeur maxOccurs définie dans le schéma.
unsigned int GetMinOccurs()	Retoune la valeur minOccurs définie dans le schéma.
<pre>string_type GetNamespaceURI()</pre>	Retourne l'URI d'espace de noms de l'élément .

## Opérateurs

Nom	Description
bool operator()	Retourne true si ce n'est pas l'élément NULL.
bool operator!()	Retourne true s'il s'agit de l'élément NULL.

## 15.7.8 altova::meta::SimpleType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types simples. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à

#### 1162 Générateur de code

propos des instances particulières des types simples dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type simple particulier défini dans le schéma XML.

### Méthodes

Nom	Description
SimpleType GetBaseType()	Retourne le type de base de ce type.
<pre>std::vector<string_type> GetEnumerations()</string_type></pre>	Retourne une liste de toutes les facettes d'énumération.
unsigned int GetFractionDigits()	Retourne la valeur de cette facette.
unsigned int GetLength()	Retourne la valeur de cette facette.
string_type GetLocalName()	Retourne le nom local du type.
<pre>string_type GetMaxExclusive()</pre>	Retourne la valeur de cette facette.
string_type GetMaxInclusive()	Retourne la valeur de cette facette.
unsigned int GetMaxLength()	Retourne la valeur de cette facette.
<pre>string_type GetMinExclusive()</pre>	Retourne la valeur de cette facette.
<pre>string_type GetMinInclusive()</pre>	Retourne la valeur de cette facette.
unsigned int GetMinLength()	Retourne la valeur de cette facette.
string_type GetNamespaceURI()	Retourne l'URI d'espace de noms du type.
<pre>std::vector<string_type> GetPatterns()</string_type></pre>	Retourne une liste de toutes les facettes de motif.
unsigned int GetTotalDigits()	Retourne la valeur de cette facette.
WhitespaceType GetWhitespace()	Retourne la valeur de la facette d'espace blanc, qui est un de : • Whitespace_Unknown • Whitespace_Preserve • Whitespace_Replace • Whitespace_Collapse

## Opérateurs

Nom	Description
bool operator()	Retourne true si ce n'est pas de SimpleType NULL.
<pre>bool operator!()</pre>	Retourne true s'il s'agit du SimpleType NULL.

# 15.7.9 [YourSchema]::[CDoc]

Lorsque le code est généré depuis un schéma XML, le code généré fournit une classe de document avec le même nom que le schéma. Cette classe contient tous les éléments root possibles en tant que membres, ainsi que les méthodes suivantes. Veuillez noter que, dans les noms de méthode ci-dessous, "CDoc" représente le nom de la classe de document générée elle-même.

Nom	Description
static CDoc CreateDocument()	Crée un nouveau document XML vide. Doit être publié en utilisant DestroyDocument().
static void DeclareAllNamespacesFromSchema(Ele mentType& node)	Déclare tous les espaces de noms depuis le Schéma XML dans l'élément fournit en dans qu'argument (généralement, l'élément racine XML). L'appel de cette méthode est utile si votre schéma a plusieurs déclarations d'espace de noms, chacune mappée dans un préfixe, et que vous souhaitez les déclarer tous dans l'élément fourni en tant qu'argument.
void DestroyDocument()	Détruit un document. Toutes les références au document et à ses nœuds sont invalidées. Cela doit être appelé lorsque vous terminez de travailler avec un document.
<pre>static CDoc LoadFromBinary(const std:vector<unsigned char="">&amp; xml)</unsigned></pre>	Charge un document XML depuis un byte array.
<pre>static CDoc LoadFromFile(const string_type&amp; fileName)</pre>	Charge un document XML depuis un fichier.
<pre>static CDoc LoadFromString(const string_type&amp; xml)</pre>	Charge un document XML depuis un string.
std::vector <unsigned char=""> SaveToBinary(bool prettyPrint)</unsigned>	Enregistre un document XML sous un byte array. Lorsque défini sur true, l'argument prettyPrint reformate le document XML pour une meilleure lecture.
<pre>std::vector<unsigned char=""> SaveToBinary(bool prettyPrint, const string_type &amp; encoding)</unsigned></pre>	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
<pre>std::vector<unsigned char=""> SaveToBinary(bool prettyPrint, const string_type &amp; encoding, bool bBigEndian, bool bBOM)</unsigned></pre>	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
void SaveToFile(const string_type & fileName, bool prettyPrint)	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
void SaveToFile(const string_type & fileName, bool omitXmlDecl)	Enregistre un document XML sous un fichier. Si l'argument omitXmlDecl est défini pour être true, la déclaration XML ne sera pas rédigée.

Nom	Description
<pre>void SaveToFile(const string_type &amp; fileName, bool omitXmlDecl, const string_type &amp; encoding)</pre>	Enregistre un document XML sous un fichier avec l'encodage spécifié. Si l'argument omitXmlDecl est défini pour être true, la déclaration XML ne sera pas rédigée.
<pre>void SaveToFile(const string_type &amp; fileName, bool prettyPrint, bool omitXmlDecl, const string_type &amp; encoding, bool bBigEndian, bool bBOM)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
<pre>void SaveToFile(const string_type &amp; fileName, bool prettyPrint, bool omitXmlDecl, const string_type &amp; encoding, bool bBigEndian, bool bBOM, const string_type &amp; lineend)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et la fin de ligne spécifiée. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
	Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir <u>Options de Générateur de Code</u> ).
<pre>void SaveToFile(const string_type&amp; fileName, bool prettyPrint, bool omitXmlDecl, const string_type &amp; encoding, const string_type &amp; lineend)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et la fin de ligne spécifiée.
	Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir <u>Options de Générateur de Code</u> <sup>(20)</sup> ).
<pre>void SaveToFile(const string_type &amp; fileName, bool prettyPrint, const string_type &amp; encoding)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
<pre>void SaveToFile(const string_type&amp; fileName, bool prettyPrint, const string_type &amp; encoding, bool bBigEndian, bool bBOM)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
<pre>void SaveToFile(const string_type&amp; fileName, bool prettyPrint, const string_type &amp; encoding, bool bBigEndian, bool bBOM, const string_type &amp; lineend)</pre>	Enregistre un document XML sous un fichier, avec l'encodage spécifié et la fin de ligne spécifiée. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
	Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir <u>Options de Générateur de Code</u> <sup>1201</sup> ).
void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, const string_type & lineend)	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et la fin de ligne spécifiée.
	Cette méthode est uniquement disponible si vous générez le code pour la bibliothèque XML Xerces3 (voir <u>Options de Générateur de Code</u> <sup>(20)</sup> ).

Nom	Description
<pre>string_type SaveToString(bool prettyPrint)</pre>	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel.
<pre>string_type SaveToString(bool prettyPrint, bool omitXmlDecl)</pre>	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel. Si l'argument omitXmlDecl est défini pour être true, la déclaration XML ne sera pas rédigée.
<pre>void SetDTDLocation(const string_type &amp; dtdLocation)</pre>	Ajoute une déclaration DOCTYPE avec l'ID de système spécifiée. Un élément root doit déjà exister. Cette méthode n'est pas prise en charge pour MSXML, puisqu'il n'est pas possible d'ajouter une déclaration DOCTYPE dans un document dans la mémoire.
<pre>void SetSchemaLocation(const string_type &amp; schemaLocation)</pre>	Ajoute un attribut xsi:schemaLocation ou xsi:noNamespaceSchemaLocation à l'élément root. Un élément root doit déjà exister.

# 15.7.10 [YourSchema]::[ElementType]

Cette classe propose des méthodes pour manipuler des éléments XML depuis votre schéma. Des méthodes de cette classe peuvent être appelées sur des éléments, pas sur le document XML lui-même. Noter qu'il n'est pas nécessaire d'instancier la classe directement pour appeler des méthodes de cette classe. Tout élément créé en utilisant les méthodes append() OU appendWithPrefix() est de type [ElementType].

Nom	Description
<pre>void DeclareNamespace(const string_type prefix, const string_type nsURI)</pre>	Cette méthode prend deux arguments qui sont tous les deux de type string : le préfixe et l'URI d'espace de noms que vous souhaitez utiliser. Le préfixe fournit en tant qu'argument sera mappé dans la valeur URI d'espace de noms fournie en tant qu'argument. Si le préfixe fourni en tant qu'argument est vide, la méthode crée ou contourne la déclaration d'espace de noms par défaut dans l'élément. Par exemple, assumons que le document XML a un élément XML appelé "purchase". Si vous appelez
	<pre>purchase.DeclareNamespace(_T("ord"), _T("http://OrderTypes"));</pre>
	Alors le document XML devient
	<purchase xmlns:ord="http://OrderTypes"></purchase>
	Un autre exemple ; si vous appelez :

Nom	Description	
	<pre>purchase.DeclareNamespace(_T(""),</pre>	
	_T("http://OrderTypes"));	
	Alors le document XML devient	
	<purchase xmlns="http://OrderTypes"></purchase>	
	Note : l'espace de noms déclaré est utilisé lorsque les éléments ou les attributs enfants ultérieurs,	
	1. si l'espace de noms enfant est le défaut, utiliser le	
	préfixe vide. 2. si l'espace de noms enfant est égal à l'espace de	
	<ol> <li>Sinon, chercher le préfixe le plus proche en commençant du parent vers le haut, en utilisant l'algorithme lookup décrit dans la section "B.2: Namespace Prefix Lookup" at</li> </ol>	
	<ul> <li><u>https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html</u>.</li> <li>4. Si aucun préfixe n'a été trouvé pour l'espace de noms élément, utiliser un préfixe vide.</li> </ul>	

# 15.7.11 [YourSchema]::MemberAttribute

Lorsque le code est généré depuis un schéma XML, une classe telle que celle-ci est créée pour chaque attribut de membre d'un type.

Nom	Description	
bool exists()	Retourne true si l'attribut existe.	
int GetEnumerationValue()	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou "Invalid" si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.	
altova::meta::Attribute info()	Retourne un objet pour l'information de schéma de requête (voir <u>altova::meta::Attribute</u> <sup>(159</sup> ).	
void remove()	Supprime l'attribut depuis son élément parent.	

Nom	Description
void SetEnumerationValue(int)	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.

# 15.7.12 [YourSchema]::MemberElement

Lorsque le code est généré depuis un schéma XML, une classe telle que celle-ci est créée pour chaque élément de membre d'un type. Dans les descriptions ci-dessous, "MemberType" représente le nom de l'élément de membre lui-même.

Nom	Description	
<pre>Iterator<membertype> all()</membertype></pre>	Retourne un objet pour itérer des instances de l'élément de membre.	
MemberType append()	Crée un nouvel élément et l'ajoute à son parent.	
MemberType appendWithPrefix(string_type prefix)	Crée un nouvel élément ayant le préfixe fournit en tant qu'argument, et l'appose à son parent. Pour consulter un exemple, voir <u>Exemple : Purchase Order</u> <sup>1145</sup> .	
unsigned int count()	Retourne le décompte des éléments.	
<pre>int GetEnumerationValue() Généré pour des types d'énumération uniquement. des constantes générée pour les valeurs possibles, la valeur ne correspond pas à une des valeurs énum schéma.</pre>		
<pre>bool exists()</pre>	Retourne true si au moins un élément existe.	
MemberType first()	Retourne la première instance de l'élément de membre.	
MemberType operator[](unsigned int index)	Retourne l'élément de membre spécifié par l'index.	
altova::meta::Element info()	Retourne un objet pour l'information de schéma de requête (voir <u>altova::meta::Element</u> <sup>116</sup> ).	
MemberType last()	Retourne la dernière instance de l'élément de membre.	
void remove()	Supprime toutes les occurrences de l'élément depuis son parent.	
void removeAT(unsigned int index)	Supprime l'occurrence de l'élément spécifié par l'index.	
void SetEnumerationValue(int)	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.	

# 15.8 Référence aux classes générées (C#)

Ce chapitre contient une description des classes C# générées avec MapForce epuis un schéma XML ou DTD (voir <u>Générer du code depuis des Schémas XML ou DTDs</u> 1009). Vous pouvez intégrer ces classes dans votre code pour lire, modifier et écrire des documentation XML.

**Note:** Le code généré peut inclure d'autres classes de support, qui ne sont pas recensées ici et qui sont sujettes à des modifications.

## 15.8.1 Altova.Types.DateTime

Cette classe vous permet de traiter des attributs XML ou des éléments qui ont des types de date et d'heure, comme xs:dateTime.

#### Constructeurs

	Nom	Description	
<b>=</b> Ø	DateTime(DateTime obj)	Initialise une nouvelle instance de la classe DateTime dans l'objet DateTime fourni en tant qu'argument.	
=0	DateTime(System.DateTime newvalue)	Initialise une nouvelle instance de la classe DateTime dans l'objet System.DateTime fourni en tant qu'argument.	
<b>≓</b> ∳	DateTime(int year, int month, int day, int hour, int minute, double second, int offsetTZ)	Initialise une nouvelle instance de la classe DateTime à l'offset d'année, mois, jour, heure, minute, seconde et fuseau horaire fournis en tant qu'arguments.	
=\$	DateTime(int year, int month, int day, int hour, int minute, double second)	Initialise une nouvelle instance de la classe DateTime à l'année, mois, jour, heure, minute et seconde fournis en tant qu'arguments.	
<b>=</b>	DateTime(int year, int month, int day)	Initialise une nouvelle instance de la classe DateTime à l'année, mois et jour fournis en tant qu'argument.	

### Propriétés

	Nom	Description	
<b>*</b>	bool HasTimezone	Obtient une valeur Booléenne qui indique si DateTime a un fuseau horaire.	
	static DateTime Now	Obtient un objet DateTime qui est défini sur la date et l'heure actuelles sur cet ordinateur.	
<b>*</b>	short TimezoneOffset	Obtient ou définit l'offset de fuseau horaire, en minutes de l'objet DateTime.	

	Nom	Description	
8	System.DateTime Value	Obtient ou définit l'offset de l'objet DateTime en tant qu'une valeur System.DateTime.	

	Nom	Description	
=	int CompareTo(object obj)	La classe DateTime met en place l'interface IComparable. Cette méthode compare l'instance actuelle de DateTime avec un autre objet et retourne un entier qui indique si l'instance actuelle précèdes, suit, ou se produit dans la même position dans l'ordre de tri que l'autre objet. Voir aussi <u>https://msdn.microsoft.com/en-</u> us/library/system.icomparable.compareto(v=vs.110).aspx	
€Ŵ	override bool Equals(object obj)	Retourne <b>true</b> si l'objet spécifié est égal à l'objet actuel ; <b>false</b> sinon.	
-≡∳	System.DateTime GetDateTime(bool correctTZ)	Retourne un objet System.DateTime depuis l'instance Altova.Types.DateTime. L'argument booléen correctTZ spécifie si le temps de l'objet retourné doit être ajusté conformément au fuseau horaire de l'instance actuel Altova.Types.DateTime.	
=0	override int GetHashCode()	Retourne le code hash de l'instance actuelle.	
≡∳	int GetWeekOfMonth()	Retourne le nombre de semaines dans le mois sous forme d'un entier.	
⊴\$	<pre>static DateTime Parse( string s )</pre>	Crée un objet DateTime depuis le string fournit en tant qu'argument. Par exemple, les valeurs de string d'échantillon suivantes sont converties avec succès dans un objet DateTime : 2015-01-01T23:23:23 01/01/2015 2015-11 23:23:23 Une exception est soulevée si le string ne peut pas être converti en un objet DateTime. Veuillez noter que cette méthode est statique et qu'elle peut uniquement être appelée dans la classe Altova.Types.DateTime elle-même, pas dans une instance de la classe	
	static DateTime Parse(string s, DateTimeFormat format)	Crée un objet DateTime depuis un string, en utilisant le format fourni en tant qu'argument. Pour la liste des formats possibles, voir <u>Altova.Types.DateTimeFormat</u> <sup>(117)</sup> . Une exception est soulevée si le string ne peut pas être converti en un objet DateTime.	

	Nom	Description	
		Veuillez noter que cette méthode est statique et qu'elle peut uniquement être appelée dans la classe Altova.Types.DateTime elle-même, pas dans une instance de la classe.	
=0	override string ToString()	Convertit l'objet DateTime en un string.	
<b>≡</b> ∳	string ToString(DateTimeFormat format)	Convertit l'objet DateTime en un string, utiliser le format fournit en tant qu'argument. Pour la liste des formats possibles, voir <u>Altova.Types.DateTimeFormat</u> <sup>1170</sup> .	

## Opérateurs

Nom	Description	
!=	Détermine si DateTime a n'est pas égal à DateTime b.	
<	Détermine si DateTime a est inférieur à DateTime b.	
<=	Détermine si DateTime a est inférieur ou égal à DateTime b.	
==	Détermine si DateTime a est égal à DateTime b.	
>	Détermine si DateTime a est supérieur à DateTime b.	
>=	Détermine si DateTime a est supérieur ou égal à DateTime b.	

### Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

utiliser Altova.Types;

L'extrait de code suivant illustre plusieurs moyens de créer des objets DateTime :

```
protected static void DateTimeExample1()
{
    // Create a DateTime object from the current system time
    Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
    Console.WriteLine("The current time is: " + dt.ToString());
    // Create an Altova DateTime object from parts (no timezone)
    Altova.Types.DateTime dt1 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33);
    Console.WriteLine("My custom time is : " + dt1.ToString());
    // Create an Altova DateTime object from parts (with UTC+60 minutes timezone)
    Altova.Types.DateTime dt2 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33, 60);
```

```
Console.WriteLine("My custom time with timezone is : " + dt2.ToString());
// Create an Altova DateTime object by parsing a string
Altova.Types.DateTime dt3 = Altova.Types.DateTime.Parse("2015-01-01T23:23:23");
Console.WriteLine("Time created from string: " + dt3.ToString());
// Create an Altova DateTime object by parsing a string formatted as schema date
Altova.Types.DateTime dt4 = Altova.Types.DateTime.Parse("2015-01-01",
DateTimeFormat.W3_date);
Console.WriteLine("Time created from string formatted as schema date: " +
dt4.ToString());
}
```

The following code listing illustrates various ways to format DateTime objects:

```
protected static void DateTimeExample2()
{
  // Create a DateTime object from the current system time
  Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
  // Output the unformatted DateTime
  Console.WriteLine("Unformatted time: " + dt.ToString());
  // Output this DateTime formatted using various formats
  Console.WriteLine("S_Days:
                                 " + dt.ToString(DateTimeFormat.S_Days));
                              " + dt.ToString(DateTimeFormat.S_Seconds));
  Console.WriteLine("S_Seconds:
  Console.WriteLine("W3_date:
                                " + dt.ToString(DateTimeFormat.W3_date));
  Console.WriteLine("W3_dateTime:
                               " + dt.ToString(DateTimeFormat.W3_dateTime));
  Console.WriteLine("W3_gDay:
  Console.WriteLine("W3_gMonth:
                                " + dt.ToString(DateTimeFormat.W3_gDay));
                                " + dt.ToString(DateTimeFormat.W3_gMonth));
  Console.WriteLine("W3_gMonthDay:
                                " + dt.ToString(DateTimeFormat.W3_gMonthDay));
  Console.WriteLine("W3_gYear:
                                " + dt.ToString(DateTimeFormat.W3_gYear));
  Console.WriteLine("W3_time:
                                " + dt.ToString(DateTimeFormat.W3_time));
}
```

## 15.8.2 Altova.Types.DateTimeFormat

Le type DateTimeFormat enum comporte les valeurs constantes suivantes :

Valeur	Description	Exemple
S_DateTime	Formate la valeur en tant que dateTime standard, avec une précision d'un dix millionième de seconde, y compris le fuseau horaire.	2015-11-12 12:19:03.9019132+01:00
<b>S_Days</b> Formate la valeur en tant que nombre de jours passés depuis l'époque UNIX.		735913,6318973451087962962963

Valeur Description		Exemple
S_Seconds Formate la valeur en tant que le nombre de seconde écoulées depuis l'époque UNIX, avec une précision d'un dix millionième d'une seconde.		63582937678,0769062
W3_date	Formate la valeur en tant que date de schéma.	12/11/2015
W3_dateTimeFormate la valeur en tant que dateTime de schéma.		2015-11-12T15:12:14.5194251
W3_gDay Formate la valeur en tant que gDay de schéma.		12 (en partant du principe que la date est le 12 du mois)
W3_gMonth	Formate la valeur en tant que gMonth de schéma.	11 (en partant du principe que le mois est novembre)
W3_gMonthDay	Formate la valeur en tant que gMonthDay de schéma.	11-12 (en partant du principe que la date est le 12 novembre)
W3_gYear	Formate la valeur en tant que gYear de schéma.	2015 (en partant du principe que l'année est 2015)
W3_gYearMonth	Formate la valeur en tant que gYearMonth de schéma.	2015-11 (en partant du principe que l'année est 2015 et que le mois est novembre)
W3_time	Formate la valeur en tant qu'heure de schéma, avec une précision d'un dix millionième d'une seconde.	15:19:07.5582719

# 15.8.3 Altova.Types.Duration

Cette classe vous permet de traiter des attributs XML ou des éléments de type xs: duration .

### Constructeurs

	Nom	Description
=0	Duration(Duration obj)	Initialise une nouvelle instance de la classe Duration dans l'objet Duration fournie en tant qu'argument.
.≡∳	Duration(Duration newvalue)	Initialise une nouvelle instance de la classe Duration dans l'objet System.TimeSpan fournie en tant qu'argument.

	Nom	Description
=0	Duration(long ticks)	Initialise une nouvelle instance de la classe Duration dans la le nombre de tics fournie en tant qu'argument.
- <b>:</b>	Duration(int newyears, int newmonths, int days, int hours, int minutes, int seconds, double partseconds, bool bnegative)	Initialise une nouvelle instance de la classe Duration dans une durée construite depuis des parties fournies en tant qu'argument.

## Propriétés

	Nom	Description
<b>*</b>	int Months	Obtient ou définit le nombre des mois dans l'instance actuelle de Duration.
<b>*</b>	System.TimeSpan Value	Obtient ou définit la valeur (en tant que System.TimeSpan) de l'instance actuelle de Duration.
<b>*</b>	int Years	Retourne le nombre des années dans l'instance actuelle de Duration.

	Nom	Description	
<b>=</b>	override bool Equals(object other)	Retourne true	si l'objet spécifié est égal à l'objet actuel ; <b>false</b> sinon.
=\$	override int GetHashCode()	Retourne le co	de hash de l'instance actuelle.
=0	<pre>bool IsNegative()</pre>	Retourne <b>true</b> négative.	si l'instance actuelle de Duration représente une durée
= <b>\</b>	static Duration Parse( string s, ParseType pt )	Retourne un objet Altova. Types. Duration parsé depuis le string fourni en tant qu'argument, en utilisant le type de parsage fourni en tant qu'argument. Valeurs de type de parsage valides :	
		DURATION	Parse la durée en partant du principe que la partie de l'année, du mois, du jour et de la durée de temps existe.
		YEARMONT H	Parse la durée en partant du principe que seules les parties d'année et de mois existent.
		DAYTIME	Parse la durée en partant du principe que la partie du jour et d'heure existent.
		Veuillez noter appelée dans l	que cette méthode est statique et peut uniquement être la classe elle-même, pas dans une instance de la classe.

	Nom	Description
=♦	override string ToString()	Convertit l'instance actuelle Duration en un string. Par exemple, une durée de 3 heures, 4 minutes et 5 secondes peut être convertie en "PT3H4M5S".
<b>=</b>	string ToYearMonthString()	Convertit l'instance actuelle Duration en un string, en utilisant le type de parsage "Year and Month".

### Opérateurs

Nom	Description
! =	Détermine si Duration a n'est pas égal à Duration b.
==	Détermine si Duration a est égal à Duration b.

#### Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

#### using Altova.Types;

L'extrait de code suivant illustre plusieurs moyens de créer des objets Duration :

```
protected static void DurationExample1()
{
   // Create a new time span of 3 hours, 4 minutes, and 5 seconds
  System.TimeSpan ts = new TimeSpan(3, 4, 5);
   // Create a Duration from the time span
  Duration dr = new Duration(ts);
  // The output is: PT3H4M5S
  Console.WriteLine("Duration created from TimeSpan: " + dr.ToString());
   // Create a negative Altova.Types.Duration from 6 years, 5 months, 4 days, 3 hours,
      // 2 minutes, 1 second, and .33 of a second
   Duration dr1 = new Duration(6, 5, 4, 3, 2, 1, .33, true);
   // The output is: -P6Y5M4DT3H2M1.33S
   Console.WriteLine("Duration created from parts: " + drl.ToString());
  // Create a Duration from a string using the DAYTIME parse type
  Duration dr2 = Altova.Types.Duration.Parse("-P4DT3H2M1S", Duration.ParseType.DAYTIME);
   // The output is -P4DT3H2M1S
   Console.WriteLine("Duration created from string: " + dr2.ToString());
   // Create a duration from ticks
   Duration dr3 = new Duration(System.DateTime.UtcNow.Ticks);
   // Output the result
```

}

Console.WriteLine("Duration created from ticks: " + dr3.ToString());

L'extrait de code suivant illustre comment obtenir des valeurs depuis des objets Duration :

```
protected static void DurationExample2()
{
    // Create a negative Altova.Types.Duration from 6 years, 5 months, 4 days, 3 hours,
    // 2 minutes, 1 second, and .33 of a second
    Duration dr = new Duration(6, 5, 4, 3, 2, 1, .33, true);
    // The output is: -P6Y5M4DT3H2M1.33S
    Console.WriteLine("The complete duration is: " + dr.ToString());

    // Get only the year and month part as string
    string dr1 = dr.ToYearMonthString();
    Console.WriteLine("The YEARMONTH part is: " + dr1);

    // Get the number of years in duration
    Console.WriteLine("Years: " + dr.Years);

    // Get the number of months in duration
    Console.WriteLine("Months: " + dr.Months);
}
```

## 15.8.4 Altova.Xml.Meta.Attribute

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des attributs. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un attribut dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un attribut particulier défini dans le schéma XML.

#### Propriétés

	Nom	Description
<b>M</b>	SimpleType DataType	Retourne le type du contenu d'attribut.
<b>M</b>	string LocalName	Retourne le nom local de l'attribut.
<b>M</b>	string NamespaceURI	Retourne l'URI d'espace de noms de l'attribut.
8	XmlQualifiedName QualifiedName	Retourne le nom qualifié de l'attribut.
2	bool Required()	Retourne true si l'attribut est requis.

# 15.8.5 Altova.Xml.Meta.ComplexType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types complexes. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types complexes dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type complexes particulier défini dans le schéma XML.

### Propriétés

	Nom	Description
2	Attribute[] Attributes	Retourne une liste de tous les attributs.
<b>*</b>	ComplexType BaseType	Retourne le type de base de ce type ou null si aucun type de base type n'existe.
2	SimpleType ContentType	Retourne le type simple du contenu.
2	Element[] Elements	Retourne une liste de tous les éléments.
2	string LocalName	Retourne le nom local du type.
2	string NamespaceURI	Retourne l'URI d'espace de noms du type.
2	XmlQualifiedName QualifiedName	Retourne le nom qualifié de ce type.

## Méthodes

	Nom	Description
=¢	ComplexType BaseType	Retourne le type de base de ce type.
.≡∳	bool Equals(obj)	Contrôle si deux objets d'info se réfèrent au même type, basé dans la comparaison du nom qualifié. Retourne true si le type a le même nom qualifié.
<b></b>	Attribute FindAttribute(string localName, string namespaceURI)	Trouve l'attribut avec le nom local spécifié et l'URI d'espace de noms
=∳	Element FindElement(string localName, string namespaceURI)	Trouve l'élément avec le nom local spécifié et l'URI d'espace de noms.

## 15.8.6 Altova.Xml.Meta.Element

Cette classe vous permet d'accéder à des informations concernant des classes générées depuis des éléments de schéma. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos

des instances particulières d'un élément dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un élément particulier défini dans le schéma XML.

## Propriétés

	Nom	Description
<b>M</b>	ComplexType DataType	Retourne le type de l'élément. Noter qu'il s'agit toujours d'un type de complexe même s'il est déclaré en tant que simple dans le schéma original. Utiliser la propriété ContentType de l'objet retourné pour obtenir le type de contenu simple.
2	string LocalName	Retourne le nom local de l'élément.
2	int MaxOccurs	Retourne la valeur maxOccurs définie dans le schéma.
2	int MinOccurs	Retourne la valeur minOccurs définie dans le schéma.
2	string NamespaceURI	Retourne l'URI d'espace de noms de l'élément .
<b>*</b>	XmlQualifiedName QualifiedName	Retourne le nom qualifié de l'élément.

# 15.8.7 Altova.Xml.Meta.SimpleType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types simples. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types simples dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type simple particulier défini dans le schéma XML.

## Propriétés

	Nom	Description
2	SimpleType BaseType	Retourne le type de base de ce type.
2	<pre>string[] Enumerations</pre>	Retourne une liste de toutes les facettes d'énumération.
2	int FractionDigits	Retourne la valeur de cette facette.
2	int Length	Retourne la valeur de cette facette.
2	string LocalName	Retourne le nom local du type.
2	string MaxExclusive	Retourne la valeur de cette facette.
2	string MaxInclusive	Retourne la valeur de cette facette.
2	int MaxLength	Retourne la valeur de cette facette.
	string MinExclusive	Retourne la valeur de cette facette.

	Nom	Description
2	string MinInclusive	Retourne la valeur de cette facette.
2	int MinLength	Retourne la valeur de cette facette.
2	string NamespaceURI	Retourne l'URI d'espace de noms du type.
2	<pre>string[] Patterns</pre>	Retourne les facettes de motif, ou null si aucun motif n'est spécifié.
	XmlQualifiedName QualifiedName	Retourne le nom qualifié de ce type.
2	int TotalDigits	Retourne la valeur de cette facette.
2	WhitespaceType Whitespace	Retourne la facette de normalisation d'espace blanc.

# 15.8.8 [YourSchema].[Doc]

Lorsque le code est généré depuis un schéma XML, le code généré fournit une classe de document avec le même nom que le schéma. Cette classe contient tous les éléments root possibles en tant que membres, ainsi que les membres recensés ci-dessous. Veuillez noter que, dans les noms de méthode ci-dessous, "Doc" représente le nom de la classe de document générée elle-même.

	Nom	Description
=0	<b>static</b> Doc CreateDocument()	Crée un nouveau document XML vide.
ų	<b>static</b> Doc CreateDocument( <b>string</b> encoding)	Crée un nouveau document XML vide, avec l'encodage de type "encoding".
.≡♦	<b>static</b> void DeclareAllNamespacesFromSchem a(nœud <b>Altova.Xml.TypeBase</b> node)	Déclare tous les espaces de noms depuis le Schéma XML dans l'élément fournit en dans qu'argument (généralement, l'élément racine XML). L'appel de cette méthode est utile si votre schéma a plusieurs déclarations d'espace de noms, chacune mappée dans un préfixe, et que vous souhaitez les déclarer tous dans l'élément fourni en tant qu'argument.
<b>=</b>	<b>static</b> Doc LoadFromBinary( <b>byte</b> [] binary)	Charge un document XML depuis un byte array.
<b>=</b>	<pre>static Doc LoadFromFile(string filename)</pre>	Charge un document XML depuis un fichier.
≡∳	<b>static</b> Doc LoadFromString( <b>string</b> xmlstring)	Charge un document XML depuis un string.

	Nom	Description
<b>=</b>	<b>byte</b> [] SaveToBinary( <b>bool</b> prettyPrint)	Enregistre un document XML sous un de byte, avec un formatage "pretty-print" optionnel.
<b>#</b>	<pre>byte[] SaveToBinary(bool prettyPrint, string encoding)</pre>	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
ų	<pre>byte[] SaveToBinary(bool prettyPrint, string encoding, bool bBigEndian, bool bBOM)</pre>	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié, l'ordre de byte et BOM (Byte Order Mark).
=∳	<pre>void SaveToFile(string fileName, bool prettyPrint)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
≡∳	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
<b></b>	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
<b>.</b>	<pre>void SaveToFile(string fileName, bool prettyPrint, string encoding, string lineend)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et le(s) caractère(s) de fin de ligne.
÷	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding, string lineend)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié et le(s) caractère(s) de fin de ligne. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
- <b>:</b>	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding, bool bBigEndian, bool bBOM, string lineend)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié, l'ordre de byte, BOM (Byte Order Mark) et le(s) caractère(s) de fin de ligne. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
-=•	<pre>void SaveToFileWithLineEnd(string fileName, bool prettyPrint, bool omitXmlDecl, string lineend)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel et le(s) caractère(s) de fin de ligne. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
≡Ŵ	<b>string</b> SaveToString( <b>bool</b> prettyPrint)	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
.≡∳	<pre>string SaveToString(bool prettyPrint, bool omitXmlDecl)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.

	Nom	Description
₩	<b>void</b> SetDTDLocation( <b>string</b> dtdLocation)	Ajoute une déclaration DOCTYPE avec l'ID de système spécifiée. Un élément root doit déjà exister.
.≡∳	<b>void</b> SetSchemaLocation( <b>string</b> schemaLocation)	Ajoute un attribut xsi:schemaLocation ou xsi:noNamespaceSchemaLocation à l'élément root. Un élément root doit déjà exister.

# 15.8.9 [YourSchema].[ElementType]

Cette classe propose des méthodes pour manipuler des éléments XML depuis votre schéma. Des méthodes de cette classe peuvent être appelées sur des éléments, pas sur le document XML lui-même. Noter qu'il n'est pas nécessaire d'instancier la classe directement pour appeler des méthodes de cette classe. Tout élément créé en utilisant les méthodes Append() ou AppendWithPrefix() est de type [ElementType].

	Nom	Description
- <b>=</b> \$	void DeclareNamespace(string prefix, string nsURI)	Cette méthode prend deux arguments qui sont tous les deux de type string : le préfixe et l'URI d'espace de noms que vous souhaitez utiliser. Le préfixe fournit en tant qu'argument sera mappé dans la valeur URI d'espace de noms fournie en tant qu'argument. Si le préfixe fourni en tant qu'argument est vide, la méthode crée ou contourne la déclaration d'espace de noms par défaut dans l'élément. Par exemple, assumons que le document XML a un élément XML appelé "purchase". Si vous appelez
		<pre>purchase.DeclareNamespace("ord",     "http://OrderTypes");</pre>
		Alors le document XML devient
		<purchase xmlns:ord="http://OrderTypes"></purchase>
		Un autre exemple ; si vous appelez :
		<pre>purchase.DeclareNamespace("", "http://OrderTypes");</pre>
		Alors le document XML devient
		<pre><purchase xmlns="http://OrderTypes"></purchase></pre>
		Note : l'espace de noms déclaré est utilisé lorsque les

Nom	Description
	éléments ou les attributs enfants ultérieurs, conformément aux règles suivantes :
	<ol> <li>si l'espace de noms enfant est le défaut, utiliser le préfixe vide.</li> <li>si l'espace de noms enfant est égal à l'espace de noms parent, utiliser le préfixe parent.</li> <li>Sinon, chercher le préfixe le plus proche en commençant du parent vers le haut, en utilisant l'algorithme lookup décrit dans la section "B.2: Namespace Prefix Lookup" at https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-</li> </ol>
	<ul> <li><u>20021022/namespaces-algorithms.html</u>.</li> <li>4. Si aucun préfixe n'a été trouvé pour l'espace de noms élément, utiliser un préfixe vide.</li> </ul>

# 15.8.10 [YourSchemaType].MemberAttribute

Lorsque le code est généré depuis un schéma XML, une classe est créée pour chaque attribut membre d'un type. Dans les descriptions ci-dessous, "AttributeType" représente le type de l'attribut de membre lui-même.

### Méthodes

	Nom	Description
=0	bool Exists()	Retourne true si l'attribut existe.
=0	void Remove()	Supprime l'attribut depuis son élément parent.

## Propriétés

	Nom	Description
2	<b>int</b> EnumerationValue	Généré pour des types d'énumération uniquement. Définit ou obtient la valeur d'attribut en utilisant une des constantes générée pour les valeurs possibles. Retourne <b>Invalid</b> si les valeurs ne correspondent pas à une des valeurs énumérées dans le schéma.
2	Altova.Xml.Meta.Attribute Info	Retourne un objet pour l'information de schéma de requête (voir <u>Altova.Xml.Meta.Attribute</u> <sup>1175</sup> ).
<b>*</b>	AttributeType Value	Définit ou obtient la valeur d'attribut.

# 15.8.11 [YourSchemaType].MemberElement

Lorsque le code est généré depuis un schéma XML, une classe avec les membres suivants est créée pour chaque élément de membre d'un type. La classe met en place l'interface standard System.Collections.IEnumerable, donc il peut être utilisé avec l'instruction foreach.

Dans les descriptions ci-dessous, "MemberType" représente le type de l'élément de membre lui-même.

### Méthodes

	Nom	Description
=0	MemberType Append()	Crée un nouvel élément et l'ajoute à son parent.
≓∳	MemberType AppendWithPrefix(string prefix)	Crée un nouvel élément ayant le préfixe fournit en tant qu'argument, et l'appose à son parent. Pour consulter un exemple, voir <u>Exemple : Purchase Order</u> <sup>(1145)</sup> .
=0	MemberType At( <b>int</b> index)	Retourne l'élément de membre spécifié par l'index.
<b>=</b>	System.Collections.IEnumerator GetEnumerator()	Retourne un objet pour itérer des instances de l'élément de membre.
<b>=</b>	<pre>void Remove()</pre>	Supprime toutes les occurrences de l'élément depuis son parent.
=0	<b>void</b> RemoveAt( <b>int</b> index)	Supprime l'occurrence de l'élément spécifié par l'index.

## Propriétés

	Nom	Description
	int Count	Retourne le décompte des éléments.
2	<b>int</b> EnumerationValue	Généré pour des types d'énumération uniquement. Définit ou obtient la valeur d'élément en utilisant une des constantes générée pour les valeurs possibles. Retourne <b>Invalid</b> si les valeurs ne correspondent pas à une des valeurs énumérées dans le schéma.
2	bool Exists	Retourne true si au moins un élément existe.
2	MemberType First	Retourne la première instance de l'élément de membre.
2	Altova.Xml.Meta.Element Info	Retourne un objet pour l'information de schéma de requête (voir <u>Altova.Xml.Meta.Element</u> <sup>1176</sup> ).
2	MemberType Last	Retourne la dernière instance de l'élément de membre.
<b>P</b>	MemberType <b>this</b> [ <b>int</b> index]	Retourne l'élément de membre spécifié par l'index.

	Nom	Description
Ш	MemberType Value	Définit ou obtient le contenu d'élément (uniquement généré si l'élément peut avoir un contenu simple ou mixte).

# **15.9** Référence aux classes générées (Java)

Ce chapitre contient une description des classes Java générées avec MapForce depuis un schéma XML ou DTD (voir <u>Générer du code depuis des Schémas XML ou DTDs</u> (1000). Vous pouvez intégrer ces classes dans votre code pour lire, modifier et écrire des documentation XML.

**Note:** Le code généré peut inclure d'autres classes de support, qui ne sont pas recensées ici et qui sont sujettes à des modifications.

## 15.9.1 com.altova.types.DateTime

Cette classe vous permet de traiter des attributs XML ou des éléments qui ont des types de date et d'heure, comme xs:dateTime.

#### Constructeurs

	Nom	Description
• <sup>c</sup>	<pre>public DateTime()</pre>	Initialise une nouvelle instance de la classe DateTime dans une valeur vide.
• <sup>c</sup>	<pre>public DateTime(DateTime newvalue)</pre>	Initialise une nouvelle instance de la classe DateTime dans l'objet DateTime fourni en tant qu'argument.
● <sup>c</sup>	<pre>public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, int newoffsetTZ)</pre>	Initialise une nouvelle instance de la classe DateTime à l'année, mois, jour, heure, minute, seconde, la partie fractionnaire de la seconde et fuseau horaire fournis en tant qu'argument. La partie fractionnaire de la seconde newpartsecond doit se situer entre 0 et 1. L'offset de fuseau horaire newoffsetTZ peut être soit positif soit négatif et s'exprime en minutes.
• <sup>c</sup>	<pre>public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond)</pre>	Initialise une nouvelle instance de la classe DateTime dans l'année, le mois, le jour, l'heure, la minute, la seconde et la partie fractionnelle d'une seconde fournie en tant qu'arguments.
• •	<pre>public DateTime(int newyear, int newmonth, int newday)</pre>	Initialise une nouvelle instance de la classe DateTime pour l'année, le mois et le jour fournis en tant qu'arguments.
• <sup>c</sup>	<pre>public DateTime(Calendar newvalue)</pre>	Initialise une nouvelle instance de la classe DateTime dans la valeur java.util.Calendar fournie en tant qu'argument.

	Nom	Description
• <sup>S</sup>	<pre>static DateTime now()</pre>	Retourne l'heure actuelle en tant qu'un objet DateTime.

	Nom	Description	
• <sup>\$</sup>	<b>static</b> DateTime parse(String s)	Retourne un objet DateTime parsé depuis la valeur de string fournie en tant qu'argument. Par exemple, les valeurs de string d'échantillon suivantes sont converties avec succès dans un objet DateTime :	
		2015-11-24T12:54:47.969+01:00 2015-11-24T12:54:47 24/11/2015	
۲	<pre>int getDay()</pre>	Retourne le jour de l'instance actuelle de DateTime.	
۲	<b>int</b> getHour()	Retourne l'heure de l'instance actuelle de DateTime.	
۲	<pre>int getMillisecond()</pre>	Retourne la milliseconde de l'instance actuelle de DateTime, en tant qu'une valeur d'entier.	
۲	<pre>int getMinute()</pre>	Retourne la minute de l'instance actuelle DateTime.	
۲	<pre>int getMonth()</pre>	Retourne le mois de l'instance actuelle DateTime.	
۰	<b>double</b> getPartSecond()	Retourne la partie fractionnaire de la seconde de l'instance actuelle de DateTime, en tant que valeur <b>double</b> . La valeur de retour est supérieure à zéro et inférieure à un, par exemple :	
		0.313	
۲	<b>int</b> getSecond()	Retourne la seconde de l'instance actuelle DateTime.	
۰	<pre>int getTimezoneOffset()</pre>	Retourne l'offset de fuseau horaire, en minutes de l'instance actuelleDateTime. Par exemple, le fuseau horaire "UTC-01:00" serait retourné en tant que :	
		-60	
۲	Calendar getValue()	Retourne l'instance actuelle de DateTime en tant qu'une valeur java.util.Calendar.	
۰	<b>int</b> getWeekday()	Retourne le jour dans la semaine de l'instance actuelle DateTime. Les valeurs s'étendent de 0 à 6, 0 étant lundi (ISO- 8601).	
۲	<pre>int getYear()</pre>	Retoune l'année de l'instance actuelle DateTime.	
۲	<pre>int hasTimezone()</pre>	Retourne l'information concernant le fuseau horaire de l'instance actuelle DateTime. Les valeurs de retour possibles sont :	
		CalendarBase.TZ_MISSING <b>Un offset</b> ( pas défini.	de fuseau horaire n'est
		CalendarBase.TZ_UTC Le fuseau	horaire est UTC.
		CalendarBase.TZ_OFFSET Un offset of été défini.	de fuseau d'horaire a

	Nom	Description		
۲	<b>void</b> setDay( <b>int</b> nDay )	Définit le jour de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
•	<b>void</b> setHasTimezone( <b>int</b> nHasTZ )	Définit l'information de fuseau horaire de l'instance actuelle DateTime à la valeur fournie en tant qu'argument. Cette méthode peut être utilisée pour supprimer l'information de fuseau horaire ou définir le fuseau horaire sur UTC (Coordinated Universal Time). Les valeurs valides pour l'argument nHasTZ :		
		CalendarBase.TZ_MISSIN G	Configurer l'offset de fuseau horaire sur non-défini.	
		CalendarBase.TZ_UTC	Configurer le fuseau horaire sur UTC.	
		CalendarBase.TZ_OFFSET	Si l'objet actuel a un offset de fuseau horaire, ne pas le modifier.	
۲	<b>void</b> setHour( <b>int</b> nHour )	Définit l'heure de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
۲	<pre>void setMinute( int nMinute )</pre>	Définit la minute de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
۲	<pre>void setMonth( int nMonth )</pre>	Définit le mois de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
۲	<b>void</b> setPartSecond( <b>double</b> nPartSecond )	Définit la partie fractionnaire de la seconde de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
۲	<b>void</b> setSecond( <b>int</b> nSecond )	Définit la seconde de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
۰	<pre>void setTimezoneOffset( int nOffsetTZ )</pre>	Définit l'offset de fuseau horaire de l'instance actuelle DateTime à la valeur fournie en tant qu'argument. La valeur nOffsetTZ doit être un entier (positif ou négatif) et doit être exprimée en minutes.		
۲	<b>void</b> setYear( <b>int</b> nYear )	Définit l'année de l'instance actuelle DateTime à la valeur fournie en tant qu'argument.		
۲	String toString()	Retourne la représentation de string de l'instance actuelle DateTime, par exemple :		
		2015-11-24T15:50:56.968+01:00		

## Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

import com.altova.types.\*;

L'extrait de code suivant illustre plusieurs moyens de créer des objets DateTime :

```
protected static void DateTimeExample1()
{
    // Initialize a new instance of the DateTime class to the current time
    DateTime dt = new DateTime(DateTime.now());
    System.out.println("DateTime created from current date and time: " + dt.toString());
    // Initialize a new instance of the DateTime class by supplying the parts
    DateTime dt1 = new DateTime(2015, 11, 23, 14, 30, 24, .459);
    System.out.println("DateTime from parts (no timezone): " + dt1.toString());
    // Initialize a new instance of the DateTime class by supplying the parts
    DateTime dt2 = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
    System.out.println("DateTime from parts (with negative timezone): " + dt2.toString());
    // Initialize a new instance of the DateTime class by parsing a string value
    DateTime dt3 = DateTime.parse("2015-11-24T12:54:47.969+01:00");
    System.out.println("DateTime parsed from string: " + dt3.toString());
}
```

The following code listing illustrates getting values from DateTime objects:

```
protected static void DateTimeExample2()
   {
      // Initialize a new instance of the DateTime class to the current time
     DateTime dt = new DateTime(DateTime.now());
      // Output the formatted year, month, and day of this DateTime instance
      String str1 = String.format("Year: %d; Month: %d; Day: %d;", dt.getYear(),
dt.getMonth(), dt.getDay());
      System.out.println(str1);
      // Output the formatted hour, minute, and second of this DateTime instance
      String str2 = String.format("Hour: %d; Minute: %d; Second: %d;", dt.getHour(),
dt.getMinute(), dt.getSecond());
      System.out.println(str2);
      // Return the timezone (in minutes) of this DateTime instance
      System.out.println("Timezone: " + dt.getTimezoneOffset());
      // Get the DateTime as a java.util.Calendar value
      java.util.Calendar dt_java = dt.getValue();
      System.out.println("" + dt_java.toString());
      // Return the day of week of this DateTime instance
      System.out.println("Weekday: " + dt.getWeekday());
      // Check whether the DateTime instance has a timezone defined
      switch(dt.hasTimezone())
      {
         case CalendarBase.TZ_MISSING:
```

```
System.out.println("No timezone.");
    break;
    case CalendarBase.TZ_UTC:
        System.out.println("The timezone is UTC.");
        break;
    case CalendarBase.TZ_OFFSET:
        System.out.println("This object has a timezone.");
        break;
    default:
        System.out.println("Unable to determine whether a timezone is defined.");
        break;
    }
}
```

The following code listing illustrates changing the timezone offset of a DateTime object:

```
protected static void DateTimeExample3()
{
    // Create a new DateTime object with timezone -0100 UTC
    DateTime dt = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
    // Output the value before the change
    System.out.println("Before: " + dt.toString());
    // Change the offset to +0100 UTC
    dt.setTimezoneOffset(60);
    // Output the value after the change
    System.out.println("After: " + dt.toString());
}
```
# 15.9.2 com.altova.types.Duration

Cette classe vous permet de traiter des attributs XML ou des éléments de type xs: duration .

### Constructeurs

	Nom	Description
• •	Duration(Duration newvalue)	Initialise une nouvelle instance de la classe Duration dans l'objet Duration fournie en tant qu'argument.
e <sup>c</sup>	Duration( <b>int</b> newyear, <b>int</b> newmonth, <b>int</b> newday, <b>int</b> newhour, <b>int</b> newminute, <b>int</b> newsecond, <b>double</b> newpartsecond, <b>boolean</b> newisnegative)	Initialise une nouvelle instance de la classe Duration dans une durée construite depuis des parties fournies en tant qu'argument.

## Méthodes

	Nom	Description	
• <sup>S</sup>	<pre>static Duration getFromDayTime( int newday, int newhour, int newminute, int newsecond, double newpartsecond )</pre>	Retourne un objet Durati heures, minutes, seconde fourni en tant qu'argumen	on créé depuis le nombre de jours, es et partie de seconde fractionnelle t.
• <sup>\$</sup>	<pre>static Duration getFromYearMonth( int newyear, int newmonth )</pre>	Retourne un objet Durati de mois fourni en tant qu'	on créé depuis le nombre d'années et argument.
• <sup>\$</sup>	<pre>static Duration parse( String s )</pre>	Retourne un objet Durati qu'argument. Par exemple peut être utilisé pour crée mois, d'un jour, d'une heu parties fractionnelles 0.33 négative, apposer le signe	on créé depuis le string fourni en tant e, le string -P1Y1M1DT1H1M1.333S r une durée négative d'une année, d'un ure, d'une minute, d'une seconde et des 3 d'une seconde. Pour créer une durée e moins ( - ) au string.
• <sup>\$</sup>	<b>static</b> Duration parse( String s, ParseType pt )	Retourne un objet Duration créé depuis le string fourni en tant qu'argument, en utilisant un format de parsage spécifique. Le format de parsage peut être un des suivants :	
		ParseType.DAYTIME	Peut être utilisé lorsque le string <b>s</b> consiste en un des éléments suivants : jours, heures, minutes, secondes, parties fractionnelles de la seconde, par exemple - P4DT4H4M4.774S.

	Nom	Description	
		ParseType.DURATION	Peut être utilisé lorsque le string <b>s</b> consiste en un des éléments suivants : années, mois, jours, heures, minutes, secondes, parties fractionnelles de la seconde, par exemple P1Y1M1DT1H1M1.333S.
		ParseType.YEARMON TH	Peut être utilisé lorsque le string <b>s</b> consiste en un des éléments suivants : années, mois. Par exemple : P3Y2M.
۲	<pre>int getDay()</pre>	Retourne le nombre de jou	Irs dans l'instance Duration actuelle.
۲	<b>long</b> getDayTimeValue()	Retourne la valeur de jour l'instance Duration actue ignorés.	et d'heure (en millisecondes) de Ile. Les années et le mois sont
۲	<b>int</b> getHour()	Retourne le nombre d'heu	res dans l'instance Duration actuelle.
۲	<pre>int getMillisecond()</pre>	Retourne le nombre de mi actuelle.	llisecondes dans l'instance Duration
۲	<pre>int getMinute()</pre>	Retourne le nombre de minutes dans l'instance Duration actuelle.	
۲	<pre>int getMonth()</pre>	Retourne le nombre de mois dans l'instance Duration actuelle.	
۲	<b>double</b> getPartSecond()	Retourne le nombre de parties de secondes fractionnelles dans l'instance Duration actuelle.	
۲	<pre>int getSecond()</pre>	Retourne le nombre de secondes dans l'instance Duration actuelle.	
۲	<b>int</b> getYear()	Retourne le nombre des a actuelle.	nnées dans l'instance Duration
۰	<b>int</b> getYearMonthValue()	Retourne la valeur (en moi Duration actuelle. Les jou sont ignorés.	s) de l'année et du mois de l'instance urs, heures, secondes et millisecondes
۲	<b>boolean</b> isNegative()	Retourne Boléenne <b>true</b> s négative.	i l'instance Duration actuelle est
۲	<pre>void setDayTimeValue(long 1)</pre>	Définit la durée dans le no qu'argument, ne touchant	mbre de millisecondes fournies en tant que la partie jour et heure de la durée.
۲	<b>void</b> setNegative( <b>boolean</b> isnegative )	Convertit l'instance actuel	e Duration en une durée négative.
۲	<pre>void setYearMonthValue(int 1)</pre>	Définit la durée dans le no qu'argument. Seule la part	mbre de mois fournis en tant tie années et mois de la durée est

	Nom	Description
		touchée.
•	String toString()	Retourne la représentation de string de l'instance actuelle Duration, par exemple : -P4DT4H4M4.774S
•	String toYearMonthString()	Retourne la représentation de string de la partie YearMonth de l'instance actuelle Duration, par exemple : P1Y2M

## Exemples

Avant d'utiliser les extraits de code suivants dans votre programme, s'assurer que les types Altova sont importés :

```
import com.altova.types.*;
import com.altova.types.Duration.ParseType;
```

L'extrait de code suivant illustre plusieurs moyens de créer des objets Duration :

```
protected static void ExampleDuration()
{
  // Create a negative duration of 1 year, 1 month, 1 day, 1 hour, 1 minute, 1 second,
   // and 0.333 fractional second parts
  Duration dr = new Duration(1, 1, 1, 1, 1, 1, .333, true);
   // Create a duration from an existing Duration object
   Duration dr1 = new Duration(dr);
  // Create a duration of 4 days, 4 hours, 4 minutes, 4 seconds, .774 fractional second
parts
  Duration dr2 = Duration.getFromDayTime(4, 4, 4, 4, .774);
   // Create a duration of 3 years and 2 months
  Duration dr3 = Duration.getFromYearMonth(3, 2);
   // Create a duration from a string
  Duration dr4 = Duration.parse("-P4DT4H4M4.774S");
   // Create a duration from a string, using specific parse formats
   Duration dr5 = Duration.parse("-PlYlMlDTlHlMl.333S", ParseType.DURATION);
   Duration dr6 = Duration.parse("P3Y2M", ParseType.YEARMONTH);
   Duration dr7 = Duration.parse("-P4DT4H4M4.774S", ParseType.DAYTIME);
}
```

L'extrait de code suivant illustre l'obtention et la configuration d'objets Duration :

```
protected static void DurationExample2()
ł
   // Create a duration of 1 year, 2 month, 3 days, 4 hours, 5 minutes, 6 seconds,
       // and 333 milliseconds
   Duration dr = new Duration(1, 2, 3, 4, 5, 6, .333, false);
   // Output the number of days in this duration
   System.out.println(dr.getDay());
   // Create a positive duration of one year and 333 milliseconds
   Duration dr1 = new Duration(1, 0, 0, 0, 0, 0, 0, .333, false);
   // Output the day and time value in milliseconds
   System.out.println(dr1.getDayTimeValue());
   // Create a positive duration of 1 year, 1 month, 1 day, 1 hour, 1 minute, 1 second,
       // and 333 milliseconds
   Duration dr2 = new Duration(1, 1, 1, 1, 1, 1, 1, .333, false);
   // Output the year and month value in months
   System.out.println(dr2.getYearMonthValue());
   // Create a positive duration of 1 year and 1 month
   Duration dr3 = new Duration(1, 1, 0, 0, 0, 0, 0, false);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Set the DayTime part of duration to 1000 milliseconds
   dr3.setDayTimeValue(1000);
   // Output the value
  System.out.println("The duration is now: " + dr3.toString());
   // Set the YearMonth part of duration to 1 month
   dr3.setYearMonthValue(1);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Output the year and month part of the duration
   System.out.println("The YearMonth part of the duration is: " +
dr3.toYearMonthString());
}
```

## 15.9.3 com.altova.xml.meta.Attribute

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des attributs. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un attribut dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un attribut particulier défini dans le schéma XML.

### Méthodes

	Nom	Description
۲	SimpleType getDataType()	Retourne le type du contenu d'attribut.
۲	String getLocalName()	Retourne le nom local de l'attribut.

	Nom	Description
۲	String getNamespaceURI()	Retourne l'URI d'espace de noms de l'attribut.
۲	<b>boolean</b> isRequired()	Retourne true si l'attribut est requis.

# 15.9.4 com.altova.xml.meta.ComplexType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types complexes. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types complexes dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type complexes particulier défini dans le schéma XML.

### Méthodes

	Nom	Description
•	Attribute findAttribute(String localName, String namespaceURI)	Trouve l'attribut avec le nom local spécifié et l'URI d'espace de noms
•	Element findElement(String localName, String namespaceURI)	Trouve l'élément avec le nom local spécifié et l'URI d'espace de noms.
۲	Attribute[] GetAttributes()	Retourne une liste de tous les attributs.
۲	ComplexType getBaseType()	Retourne le type de base de ce type.
۲	SimpleType getContentType()	Retourne le type simple du contenu.
۲	<pre>Element[] GetElements()</pre>	Retourne une liste de tous les éléments.
۲	String getLocalName()	Retourne le nom local du type.
۲	String getNamespaceURI()	Retourne l'URI d'espace de noms du type.

# 15.9.5 com.altova.xml.meta.Element

Cette classe vous permet d'accéder à des informations concernant des classes générées depuis des éléments de schéma. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières d'un élément dans un document XML. Au lieu, il vous permet d'obtenir des informations par le biais d'un programme concernant un élément particulier défini dans le schéma XML.

Mé	thod	es
----	------	----

	Nom	Description
•	ComplexType getDataType()	Retourne le type de l'élément. Noter qu'il s'agit toujours d'un type de complexe même s'il est déclaré en tant que simple dans le schéma original. Utiliser getContentType() de l'objet retourné pour obtenir le type de contenu simple.
۲	String getLocalName()	Retourne le nom local de l'élément.
۲	<pre>int getMaxOccurs()</pre>	Retourne la valeur maxOccurs définie dans le schéma.
۲	<pre>int getMinOccurs()</pre>	Retoune la valeur minOccurs définie dans le schéma.
۲	String getNamespaceURI()	Retoune l'URI d'espace de noms de l'élément .

# 15.9.6 com.altova.xml.meta.SimpleType

Cette classe vous permet d'accéder à des informations de schéma concernant des classes générées depuis des types simples. Veuillez noter que cette classe n'est pas destinée à fournir des informations dynamiques à propos des instances particulières des types simples dans un document XML. Au lieu, elle vous permet d'obtenir des informations par le biais d'un programme concernant un type simple particulier défini dans le schéma XML.

### Méthodes

	Nom	Description
۲	SimpleType getBaseType()	Retourne le type de base de ce type.
۲	<pre>String[] getEnumerations()</pre>	Retourne un array de toutes les facettes d'énumération.
۲	<pre>int getFractionDigits()</pre>	Retourne la valeur de cette facette.
۲	<pre>int getLength()</pre>	Retourne la valeur de cette facette.
۲	String getLocalName()	Retourne le nom local du type.
۲	String getMaxExclusive()	Retourne la valeur de cette facette.
۲	String getMaxInclusive()	Retourne la valeur de cette facette.
۲	<pre>int getMaxLength()</pre>	Retourne la valeur de cette facette.
۲	String getMinExclusive()	Retourne la valeur de cette facette.
۲	String getMinInclusive()	Retourne la valeur de cette facette.
۲	<pre>int getMinLength()</pre>	Retourne la valeur de cette facette.
۲	String getNamespaceURI()	Retourne l'URI d'espace de noms du type.

	Nom	Description
۲	<pre>String[] getPatterns()</pre>	Retoune un array de toutes les facettes de motif.
۲	<pre>int getTotalDigits()</pre>	Retourne la valeur de cette facette.
•	<b>int</b> getWhitespace()	Retourne la valeur de la facette d'espace blanc, qui est un de : com.altova.typeinfo.WhitespaceType.Whitespace_Unknown com.altova.typeinfo.WhitespaceType.Whitespace_Preserv e com.altova.typeinfo.WhitespaceType.Whitespace_Replace com.altova.typeinfo.WhitespaceType.Whitespace_Collaps e

# 15.9.7 com.[YourSchema].[Doc]

Lorsque le code est généré depuis un schéma XML, le code généré fournit une classe de document avec le même nom que le schéma. Cette classe contient tous les éléments root possibles en tant que membres, ainsi que les membres recensés ci-dessous. Veuillez noter que, dans les noms de méthode ci-dessous, "Doc" représente le nom de la classe de document générée elle-même.

## Méthodes

	Nom	Description
● <sup>S</sup>	<pre>static Doc createDocument()</pre>	Crée un nouveau document XML vide.
• <sup>\$</sup>	<pre>static void declareAllNamespacesFromSchem a(com.altova.xml.ElementType node)</pre>	Déclare tous les espaces de noms depuis le Schéma XML dans l'élément fournit en dans qu'argument (généralement, l'élément racine XML). L'appel de cette méthode est utile si votre schéma a plusieurs déclarations d'espace de noms, chacune mappée dans un préfixe, et que vous souhaitez les déclarer tous dans l'élément fourni en tant qu'argument.
• <sup>s</sup>	static Doc loadFromBinary(byte[] xml)	Charge un document XML depuis un byte array.
● <sup>S</sup>	static Doc loadFromFile(String fileName)	Charge un document XML depuis un fichier.
● <sup>S</sup>	static Doc loadFromString(String xml)	Charge un document XML depuis un string.
۲	<pre>byte[] saveToBinary(boolean prettyPrint)</pre>	Enregistre un document XML sous un de byte, avec un formatage "pretty-print" optionnel.
۲	<pre>byte[] saveToBinary(boolean prettyPrint, String encoding)</pre>	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
•	byte[] saveToBinary(boolean prettyPrint, String encoding, boolean bigEndian, boolean writeBOM)	Enregistre un document XML sous un byte array, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La

	Nom	Description
		marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
•	void saveToFile(String fileName, boolean prettyPrint)	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel.
•	<pre>void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage UTF-8. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
•	<pre>void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl, String encoding)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
•	<pre>void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl, String encoding, boolean bBigEndian, boolean bBOM)</pre>	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
۲	void saveToFile(String fileName, boolean prettyPrint, String encoding)	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié.
•	void saveToFile(String fileName, boolean prettyPrint, String encoding, boolean bBigEndian, boolean bBOM)	Enregistre un document XML sous un fichier, avec un formatage "pretty-print" optionnel, avec l'encodage spécifié. La marque Byte order et Unicode byte-order peuvent être spécifiées pour les encodages Unicode.
۲	String saveToString(boolean prettyPrint)	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel.
•	String saveToString(boolean prettyPrint, boolean omitXmlDecl)	Enregistre un document XML sous un string, avec un formatage "pretty-print" optionnel. Si omitXmlDecl est true, la déclaration XML ne sera pas rédigée.
•	void setSchemaLocation(String schemaLocation)	Ajoute un attribut xsi:schemaLocation ou xsi:noNamespaceSchemaLocation à l'élément root. Un élément root doit déjà exister.

# 15.9.8 com.[YourSchema].[ElementType]

Cette classe propose des méthodes pour manipuler des éléments XML depuis votre schéma. Des méthodes de cette classe peuvent être appelées sur des éléments, pas sur le document XML lui-même. Noter qu'il n'est pas nécessaire d'instancier la classe directement pour appeler des méthodes de cette classe. Tout élément créé en utilisant les méthodes <u>append()</u> ou <u>appendWithPrefix()</u> est de type ElementType.

## Méthodes

	Nom	Description
•	void declareNamespace(String prefix, String nsURI)	Cette méthode prend deux arguments qui sont tous les deux de type string : le préfixe et l'URI d'espace de noms que vous souhaitez utiliser. Le préfixe fournit en tant qu'argument sera mappé dans la valeur URI d'espace de noms fournie en tant qu'argument. Si le préfixe fourni en tant qu'argument est vide, la méthode crée ou contourne la déclaration d'espace de noms par défaut dans l'élément. Par exemple, assumons que le document XML a un élément XML appelé "purchase". Si vous appelez
		<pre>purchase.declareNamespace("ord",     "http://OrderTypes");</pre>
		Alors le document XML devient
		<pre><purchase xmlns:ord="http://OrderTypes"></purchase></pre>
		Un autre exemple ; si vous appelez :
		<pre>purchase.declareNamespace("",     "http://OrderTypes");</pre>
		Alors le document XML devient
		<pre><purchase xmlns="http://OrderTypes"></purchase></pre>
		<ul> <li>Note : l'espace de noms déclaré est utilisé lorsque les éléments ou les attributs enfants ultérieurs, conformément aux règles suivantes :</li> <li>1. si l'espace de noms enfant est le défaut, utiliser le préfixe vide.</li> <li>2. si l'espace de noms enfant est égal à l'espace de noms parent, utiliser le préfixe parent.</li> <li>3. Sinon, chercher le préfixe le plus proche en</li> </ul>
		<ul> <li>commençant du parent vers le haut, en utilisant l'algorithme lookup décrit dans la section "B.2: Namespace Prefix Lookup" at <u>https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html</u>.</li> <li>Si aucun préfixe n'a été trouvé pour l'espace de noms élément, utiliser un préfixe vide.</li> </ul>

# 15.9.9 com.[YourSchema].[YourSchemaType].MemberAttribute

Lorsque le code est généré depuis un schéma XML, une classe est créée pour chaque attribut membre d'un type. Dans les descriptions ci-dessous, "AttributeType" représente le type de l'attribut de membre lui-même.

### Méthodes

	Nom	Description
۲	<b>boolean</b> exists()	Retourne true si l'attribut existe.
•	<b>int</b> getEnumerationValue()	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou <b>Invalid</b> si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.
۲	<pre>com.altova.xml.meta.Attribute getInfo()</pre>	Retourne un objet pour l'information de schéma de requête (voir <u>com.altova.xml.meta.Attribute</u> <sup>1192</sup> ).
۲	AttributeType getValue()	Définit ou obtient la valeur d'attribut.
۲	<pre>void remove()</pre>	Supprime l'attribut depuis son élément parent.
•	<b>void</b> setEnumerationValue( <b>int</b> )	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.
۲	<b>void</b> setValue(AttributeType value)	Définit la valeur d'attribut.

# 15.9.10 com.[YourSchema].[YourSchemaType].MemberElement

Lorsque le code est généré depuis un schéma XML, une classe avec les membres suivants est créée pour chaque élément de membre d'un type. Dans les descriptions ci-dessous, "MemberType" représente le type de l'élément de membre lui-même.

### Méthodes

	Nom	Description
۲	MemberType append()	Crée un nouvel élément et l'ajoute à son parent.
•	MemberType appendWithPrefix(String prefix)	Crée un nouvel élément ayant le préfixe fournit en tant qu'argument, et l'appose à son parent. Pour consulter un exemple, voir <u>Exemple : Purchase Order</u> <sup>(1145)</sup> .
۲	MemberType at( <b>int</b> index)	Retourne l'instance de l'élément membre en tant que l'index spécifié.
۲	int count()	Retourne le décompte des éléments.

	Nom	Description
۲	<b>boolean</b> exists()	Retourne true si au moins un élément existe.
۲	MemberType first()	Retourne la première instance de l'élément de membre.
۲	<b>int</b> getEnumerationValue()	Généré pour des types d'énumération uniquement. Retourne une des constantes générée pour les valeurs possibles, ou <b>Invalid</b> si la valeur ne correspond pas à une des valeurs énumérées dans le schéma.
۲	<pre>com.altova.xml.meta.Element getInfo()</pre>	Retourne un objet pour l'information de schéma de requête (voir <u>com.altova.xml.meta.Element</u> <sup>1193</sup> ).
۲	MemberType getValue()	Obtient le contenu de l'élément (uniquement généré si l'élément peut avoir un contenu simple ou mixte).
۲	java.util.Iterator iterator()	Retourne un objet pour itérer des instances de l'élément de membre.
۲	MemberType last()	Retourne la dernière instance de l'élément de membre.
۲	<pre>void remove()</pre>	Supprime toutes les occurrences de l'élément depuis son parent.
۲	<pre>void removeAt(int index)</pre>	Supprime l'occurrence de l'élément spécifié par l'index.
•	<b>void</b> setEnumerationValue( <b>int</b> index)	Généré pour des types d'énumération uniquement. Passer une des constantes générées pour les valeurs possibles dans cette méthode pour définir la valeur.
۲	<b>void</b> setValue(MemberType value)	Définit le contenu d'élément (uniquement généré si l'élément peut avoir un contenu simple ou mixte).

# **15.10** Astuces de Génération de Code

### Résoudre les exceptions "Mémoire insuffisante" pendant la compilation Java

Mappages complexes avec des schémas larges Peuvent produire une grande quantité de code, qui peut entraîner une exception java.lang.OutofMemory pendant la compilation en utilisant Ant. Pour corriger cela :

• Ajouter la variable d'environnement ANT\_OPTS, qui définit des options Ant spécifiques comme la mémoire à attribuer à l'ordinateur, et configurer sa valeur telle qu'indiqué ci-dessous.

Edit User Variable	
Variable <u>n</u> ame:	ANT_OPTS
Variable <u>v</u> alue:	-server -Xmx512m -Xms512m
	OK Cancel

• Pour vous assurer que le compileur et le code généré soient exécutés dans le même processus qu'Ant, changer l'attribut fork, dans **build.xml**, sur false.

Vous devrez éventuellement personnaliser les valeurs selon la quantité de mémoire disponible dans votre appareil et la taille du projet sur lequel vous travaillez. Pour plus de détails, voir votre documentation Java VM.

Lorsque vous exécutez la commande ant jar, vous recevrez éventuellement un message d'erreur de type "[...] archive contains more than 65535 entities". Afin d'éviter cela, il est recommandé d'utiliser Ant 1.9 ou plus, et dans le fichier **build.xml**, d'ajouter zip64mode="as-needed" dans l'élément <jar>.

### Réserver des noms de méthode

Lors de la personnalisation de la génération de code en utilisant les fichiers SPL fournis, il peut être nécessaire de réserver des noms pour éviter toute collision avec d'autres symboles. Pour ce faire :

- 1. Naviguer dans le répertoire d'installation du programme, par exemple, C:\Program Files\Altova\MapForce2023.
- 2. Dans le sous-répertoire **spl**, trouver le répertoire correspondant au langage de programme, par exemple, **..\spl\java**.
- 3. Ouvrir le fichier **settings.spl** et insérer une nouvelle ligne dans la section reserve par ex. **reserve** "myReservedWord".
- 4. Régénérer le code de programme.

# 15.11 Options du Générateur de Code

Pour consulter ou modifier les paramètres MapForce applicables pour la génération de code :

• Dans le menu Outils, cliquer sur Options, puis cliquer sur Génération.

C±+ Settings Project File:	C <u>#</u> Settings Project File:
Microsoft® Visual Studio® 2019 ∨	Microsoft® Visual Studio® 2019 ~
XML Library	
© <u>M</u> S∧ML 6 ○ <u>X</u> erces 3.x	
Library Type	
<u> <u> S</u>tatic Library (.LIB) </u>	
O Dynamic-Link Library (.DLL)	Wrapper Classes
MFC Support	Generate <u>W</u> rapper Classes
Server Execution File	
Convert all ADO and ODBC Database Conn	ections to <u>J</u> DBC
✓ Ignore Digital Signatures (unsupported by Ma	apForce Server)
Generate for ManForce Server version:	most current ~
denorate for maproree berver version.	

#### Les paramètres disponibles sont les suivants

Paramètres C++	Définit les paramètre de compileur spécifique pour l'environnement C++, concrètement :	
	<ul> <li>La version Visual Studio (2013, 2015, 2017, 2019, 2022)</li> <li>La bibliothèque XML (MSXML, Xerces 3.x)</li> <li>Si les bibliothèques statiques ou dynamiques doivent être générées</li> <li>Si le code doit être généré avec ou sans la prise en charge MFC</li> </ul>	

Paramètres C#	<ul> <li>Sélectionnez l'option Microsoft .NET Core 3.1 ou Microsoft .NET 5.0 ou Microsoft .NET 6.0 pour générer une solution Visual Studio ciblant les plateformes respectives.</li> <li>Si vous devez cibler la plate-forme .NET Framework pour une version spécifique de Visual Studio, sélectionnez toute option de Microsoft Visual Studio 2010-2019 - dans ce cas, la solution générée ciblera la version .NET Framework correspondant à la version respective de Visual Studio.</li> </ul>
Classes Wrapper	Vous permet de générer des classes wrapper pour les schémas, voir <u>Générer</u> <u>du code depuis des Schémas XML ou des DTDs</u> <sup>1103</sup> . Ces classes de wrapper peuvent être utilisées par un code personnalisé qui contient le code généré par MapForce.
Fichier d'exécution serveur	Ces options sont applicables lorsque vous compilez des mappages dans les fichiers d'exécution MapForce Server. Ils ne touchent pas la génération de code C#, C++, ou Java. Pour plus d'informations, voir <u>Compiler des Mappages</u> sur les fichiers d'exécution MapForce Server <sup>957</sup> .

# 15.12 Référence SPL

Cette section donne un aperçu de SPL (Spy Programming Language), le langage de modèle du générateur de code.

Il set recommandé d'avoir une certaine expérience de la programmation, et de maîtriser les opérateurs, les fonctions, les variables et les classes, ainsi que les bases de la programmation orientée vers l'objet - très largement utilisé dans SPL.

Les modèles utilisés par MapForce sont fournis dans le dossier ...\MapForce\spl. Vous pouvez utiliser ces fichiers en tant qu'aide pour vous aider à développer vos propres modèles.

### Comment fonctionne le générateur de code

Les entrées dans le générateur de code sont les fichiers de modèle (.spl) et le modèle d'objet fourni par MapForce. Les fichiers de modèle contiennent des instructions SPL pour créer des fichiers, lire des informations depuis le modèle d'objet et effectuer des calculs, ponctué de fragments de code littéral dans le langage de programmation cible.

Le fichier de modèle est interprété par le générateur de code et les produit des fichiers de code source **.cpp, .java, .cs**, des fichiers de projet, ou tout autre type de fichier dépendant du modèle. Le code de source peut ensuite être compilé dans un fichier exécutable qui accède aux données XML décrites par le fichier de schéma.

Les fichiers SPL ont accès à une grande variété d'informations recueillie depuis les schémas de source. Veuillez noter qu'un fichier SPL n'est pas lié à un schéma spécifique, mais permet l'accès à tous les schémas. Veuillez vous assurer d'écrire vos fichiers SPL génériquement et d'éviter les structures qui s'appliquent aux schémas spécifiques.

### Exemple : Créer un nouveau fichier dans SPL

Voici un fichier SPL de base. Il crée un fichier nommé **test.cpp**, et place l'instruction include à l'intérieur. La commande close achève le modèle.

```
[create "test.cpp"]
#include "stdafx.h"
[close]
```

## 15.12.1 Structure SPL de base

Un fichier SPL contient du texte littéral de sortie, mêlé d'instructions de générateur de code.

Les instructions du générateur de code sont contenues dans des crochets '[' et ']'. Plusieurs instructions peuvent être contenues dans une paire de crochets. Des instructions supplémentaires doivent être séparées par une nouvelle ligne ou un double point ':'.

Des exemples valides sont :

```
[\$x = 42
\$x = \$x + 1]
```

ou

```
[\$x = 42: \$x = \$x + 1]
```

### Ajouter du texte à des fichiers

Le texte non contenu par [ et ], est écrit directement dans le fichier de sortie actuel. S'il n'y a pas de fichier de sortie actuel, le texte est ignoré (voir <u>Utiliser les fichiers</u> comment créer un fichier de sortie).

Pour produire des crochets littéraux, les échapper avec une barre oblique inversée : \[ et \]; pour produire une barre oblique inversée utiliser \\.

### Commentaires

Les commentaires se trouvant dans un bloc d'instructions commencent toujours avec un caractère ', et se terminent sur la ligne suivante, ou à un caractère de bloc de fermeture ].

## 15.12.2 Déclarations

Les instructions suivantes sont évaluées tout en parsant le fichier de modèle SPL. Elles **ne sont pas** touchées par les instructions de contrôle du flux comme des conditions, des boucles ou des sous-routines, et sont toujours évaluées exactement une seule fois.

Ces mots-clé, comme tous les mots-clé dans SPL, ne sont pas sensibles à la casse.

Veuillez noter que toutes ces déclarations doivent se trouver dans un bloc délimité par des crochets.

map ... to ...

map mapname key to value [, key to value ]...

Cette instruction ajoute des informations dans un mappage. Voir ci-dessous pour des utilisations spécifiques.

map schemanativetype schematype to typespec

Le type de Schéma XML built-in spécifié sera mappé dans le type natif spécifié ou la classe, en utilisant le formateur spécifié. Ce paramètre s'applique uniquement à la génération de code pour la version 2007r3 et plus. Typespec est un type natif ou un nom de classe, suivi par une virgule, suivi par l'instance de classe de formateur.

Exemple :

map schemanativetype "double" en "double,Altova::DoubleFormatter"

map type ... to ...

map type schematype to classname

Le type de Schéma XML built-in spécifié sera mappé dans la classe spécifiée. Ce paramètre s'applique uniquement à la génération de code pour la version 2007 ou en-dessous.

Exemple :

map type "float" to "CSchemaFloat"

### default ... is ...

default setting is value

Cette instruction vous permet de toucher la manière dont les noms de classe et de membre sont dérivés depuis le Schéma XML. Veuillez noter que les noms de paramètre sont sensibles à la casse.

Exemple :

default "InvalidCharReplacement" is "\_"

Nom du paramètre	Explication	
ValidFirstCharSet	Les caractères autorisés pour lancer un identifiant	
ValidCharSet	Les caractères autorisés pour d'autres caractères dans un identifiant	
InvalidCharReplacement Le caractère qui remplacera tous les caractères dans les noms qui trouvent pas dans ValidCharSet		
AnonTypePrefix	Préfixe pour les noms des types anonymes*	
AnonTypeSuffix	Suffixe pour les noms des types anonymes*	
ClassNamePrefix	Préfixe pour des noms de classe générés	
ClassNameSuffix	Suffixe pour les noms de classe générés	
EnumerationPrefix	Préfixe pour les constantes symboliques déclarées pour les valeurs d'énumération	
EnumerationUpperCase	"on" pour convertir les noms de constante d'énumération pour la casse majuscule	

Nom du paramètre	Explication
FallbackName	Si un nom consiste uniquement en caractères qui ne sont pas dans ValidCharSet, utiliser celui-là

\* Les noms des types anonymes sont construits à partir de AnonTypePrefix + nom d'élément + AnonTypeSuffix

#### reserve

reserve word

Ajoute le mot spécifié pour la liste des mots réservés. Cela garantit qu'il ne sera jamais généré en tant que classe ou de nom de membre.

Exemple :

reserve "while"

### include

Inclut le fichier spécifié en tant que source SPL. Cela vous permet de partager votre modèle dans plusieurs fichiers pour une édition et une gestion plus simple.

include filename

Exemple :

include "Module.cpp"

## 15.12.3 Variables

Tout fichier SPL non-trivial nécessitera des variables. Certaines variables sont <u>prédéfinies</u><sup>1207</sup> par le générateur de code, et des nouvelles variables peuvent être créées simplement en y attribuant des valeurs.

Le caractère **\$** est utilisé lors d'une **déclaration** ou **utilisation** d'une variable, un nom de variable est toujours préfixé par **\$**. Les noms de variable sont **sensibles à la casse**.

Types de variables :

- entier aussi utilisé en tant que booléenne, où 0 est faux et tout le reste est vrai
- string
- objet fourni par MapForce
- iterator voir instruction foreach<sup>(212</sup>)

Les types de variables sont déclarés par la première affectation :

[\$x = 0]

x est maintenant un entier.

[\$x = "teststring"]

x est maintenant traité en tant qu'un string.

### Strings

Les constantes de string sont toujours entourées dans des guillemets doubles, comme dans l'exemple cidessus. In et It dans des guillemets doubles sont interprétés en tant que newline et tab, I" est un guillemet double littéral, et II est une barre oblique inversée. Les constantes de string peuvent aussi s'étendre sur plusieurs lignes.

La concaténation de string utilise le caractère &

```
[$BasePath = $outputpath & "/" & $JavaPackageDir]
```

### Objets

Les objets représentent l'information contenue dans le schéma XML, les structures de base de données, les fichiers de texte et les mappages. Les objets ont des **propriétés**, qui peuvent être accédées en utilisant l'opérateur .. Il n'est pas possible de créer de nouveaux objets dans SPL (ils sont prédéfinis par le générateur de code, dérivés depuis le mapping) d'entrée, mais il est possible d'attribuer des objets aux variables.

Exemple:

```
class [=$class.Name]
```

Cet exemple produit le mot "class", suivi par une espace et la valeur de la propriété **Name** property de l'objet **\$class**.

# 15.12.4 Variables prédéfinies

Une fois que le fichier de schéma a été analysé par le générateur de code, les objets dans la table ci-dessous existent dans le Template Engine.

Nom	Туре	Description
\$schematype	entier	1 pour DTD, 2 pour Schema XML
\$TheLibrary	Bibliothèque <sup>1216</sup>	La bibliothèque dérivée depuis le Schéma XML ou le DTD
\$module	string	Nom du Schéma de source sans extension

Nom	Туре	Description
\$outputpath	string	Le chemin de sortie spécifié par l'utilisateur ou le chemin de sortie par défaut

Uniquement pour la génération C++ :

Nom	Туре	Descriptio	on
\$domtype	entier	1 pour MSXML, 2 pour Xerces	
\$libtype	entier	1 pour static LIB, 2 pour DLL	
\$mfc	booléenne	True si la prise en charge MFC est activée	
\$VSVersion	entier	Spécifie la version Visual Studio. Valeurs valides :	
		0	Aucun projet Visual Studio
		2010	Visual Studio 2010
		2013	Visual Studio 2013
		2015	Visual Studio 2015
		2017	Visual Studio 2017
		2019	Visual Studio 2019

Uniquement pour la génération C# :

Nom	Туре	Description	
\$VSVersion	entier	Spécifie la version Visual Studio. Valeurs valides :	
		0	Aucun proiet Visual Studio
		2010	Visual Studio 2010
		2013	Visual Studio 2013
		2015	Visual Studio 2015
		2017	Visual Studio 2017
		2010	Visual Studio 2010
		2019	

## 15.12.5 Créer des fichiers de sortie

Ces instructions sont utilisées pour créer des fichiers de sortie depuis une génération de code. Veuillez noter que toutes ces instructions doivent se trouver dans un bloc délimité par des crochets.

#### create

#### create filename

crée un nouveau fichier. Le fichier doit être fermé avec l'instruction **close**. Toutes les sorties suivantes sont écrites dans le fichier spécifié.

Exemple :

```
[create $outputpath & "/" & $JavaPackageDir & "/" & $application.Name & ".java"]
package [=$JavaPackageName];
public class [=$application.Name]Application {
    ...
}
[close]
```

#### close

ferme le fichier de sortie actuel.

#### =\$variable

Writes écrit la valeur de la variable spécifiée dans le fichier de sortie actuel.

Exemple :

```
[\$x = 20+3]
The result of your calculation is [=\$x] - so have a nice day!
```

#### La fichier de sortie sera :

The result of your calculation is 23 - so have a nice day!

#### write

write string

écrit le string dans le fichier de sortie actuel.

Exemple :

[write "C" & \$name]

Il peut aussi s'écrire :

C[=\$name]

filecopy ... to ...

filecopy source to target

cope le fichier de source dans le fichier cible, sans interprétation.

Exemple :

```
filecopy "java/mapforce/mapforce.png" to $outputpath & "/" & $JavaPackageDir &
    "/mapforce.png"
```

## 15.12.6 Opérateurs

Les opérateurs dans SPL fonctionnent comme dans la plupart des langages de programmation.

Liste des opérateurs SPL dans un ordre de précédence décroissant :

<ol> <li>Accéder à la propriété d'obje</li> </ol>
---------------------------------------------------

- () Regroupement d'expression
- true constante booléenne "true"
- false constante booléenne "false"
- & Concaténation de string
- Signe pour le nombre négatif
- not Négation logique
- \* Multiplier
- / Diviser
- % Modulo
- + Ajouter
- Soustraire
- <= Inférieur à ou égal
- < Inférieur à
- >= Supérieur à ou égal
- > Supérieur à
- = Égal

<>	Non égal
and	Conjonction logique (avec une évaluation de court circuit)
or	Disjonction logique (avec une évaluation de court circuit)

```
= Attribution
```

## 15.12.7 Conditions

SPL vous permet d'utiliser les instructions "if" standard. La syntaxe est comme suit :

if condition statements else statements endif

#### ou, sans else :

if condition statements endif

Note: La condition n'est pas contenue dans des parenthèses.

Comme dans tout autre langage de programmation, les conditions sont construites avec des opérateurs <sup>(210</sup> de logique et comparaison.

Exemple :

```
[if $namespace.ContainsPublicClasses and $namespace.Prefix <> ""]
    whatever you want ['inserts whatever you want, in the resulting file]
[endif]
```

### Switch

SPL contient aussi une instruction à choix multiple.

Syntaxe :

```
switch $variable
    case X:
        statements
    case Y:
    case Z:
        statements
    default:
        statements
```

#### endswitch

Les libellés de cas doivent être des constantes ou des variables.

L'instruction switch dans SPL ne tombe pas à travers les cas (comme dans C), il n'y a donc pas de nécessité pour une instruction de "break".

## 15.12.8 Collections et foreach

### Collections et itérateurs

Une collection contient plusieurs objets - comme un array ordinaire. Les itérateurs résolvent le problème du stockage et l'incrémentation des indexes array lorsque vous accédez aux objets.

Syntaxe :

foreach iterator in collection statements next

Exemple :

Exemple 2 :

```
[foreach $i in 1 To 3
Write "// Step " & $i & "\n"
` Do some work
next]
```

Dans la première ligne :

**\$classes** est l' <u>objet global</u> <sup>(20)</sup> de tous les types générés. Il s'agit d'une collection d'objets de classe uniques.

**Foreach** passe à travers tous les items dans \$classes, et exécute le code suivant l'instruction, jusqu'à l'instruction **next**, pour chacun d'entre eux.

Dans chaque itération, **\$class** est attribué à l'objet de classe suivant. Il vous suffit de travailler avec l'objet de classe au lieu d'utiliser classes[i]->Name() comme pour un travail en C++.

Tous les itérateurs de collection possèdent les propriétés supplémentaires suivantes:

Index	L'index suivant, en commençant avec 0
lsFirst	true si l'objet actuel est le premier de la collection (index est 0)
lsLast	true si l'objet actuel est le derner de la collection
Current	L'objet actuel (cela est implicite s'il n'est pas spécifié et peut être exclu)

Exemple :

```
[foreach $enum in $facet.Enumeration
    if not $enum.IsFirst
        ], [
    endif
    ]"[=$enum.Value]"[
next]
```

## 15.12.9 Sous-routines

Le générateur de code prend en charge des sous-routines sous la forme de procédures ou de fonctions.

Fonctions :

- Passage de valeurs par valeur et par référence
- Paramètres locaux/globaux (local dans le cadre des sous-routines)
- Variables locales
- Invocation récursive (les sous-routines peuvent s'appeler elles-même)

## 15.12.9.1 Déclaration de sous-routine

### Sous-routine

Exemple de syntaxe :

```
Sub SimpleSub()
    ... lines of code
EndSub
```

- Sub est le mot-clé qui dénote la procédure.
- **SimpleSub** est le nom attribué à la sous-routine.
- Λα parenthèse Ronde peut contenir une liste de paramètre.
- Le bloc de code d'une sous-routine est lancé immédiatement après la parenthèse de fermeture de paramètre.
- EndSub dénote la fin du code de bloc.
- **Note:** La déclaration récursive ou en cascade de la sous-routine n'est pas permise, donc une sous-routine peut ne pas contenir une autre sous-routine.

### Paramètres

Les paramètres peuvent aussi être passés par des procédures utilisant la syntaxe suivante :

- Tous les paramètres doivent être des variables
- Les variables doivent être préfixées par le caractère \$
- Les variables locales sont définies dans une sous-routine
- Les variables globales sont déclarées explicitement, en-dehors des sous-routines
- Plusieurs paramètres sont séparés par le caractère virgule "," dans des parenthèses arrondies
- Des paramètres peuvent passer des valeurs

### Paramètres - des valeurs passées

Des paramètres peuvent être passées de deux manières, par des valeurs et par des références, en utilisant les mots-clés **ByVal** et **ByRef** respectivement.

Syntaxe :

```
' define sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
] ...
```

- **ByVal** spécifie que le paramètre est passé par valeur. Veuillez noter que la plupart des objets peut uniquement être passé par la référence.
- ByRef spécifie que le paramètre est passé par la référence. Il s'agit du défaut si ni ByVal ou ByRef est spécifié.

### La fonction retourne des valeurs

Pour retourner une valeur depuis une sous-routine, utiliser l'instruction **return**. Une telle fonction peut être appelée depuis l'intérieur d'une expression.

Exemple :

```
' define a function
[Sub MakeQualifiedName( ByVal $namespacePrefix, ByVal $localName )
if $namespacePrefix = ""
  return $localName
else
  return $namespacePrefix & ":" & $localName
endif
EndSub
]
```

## 15.12.9.2 Invocation de sous-routine

Utilisercall pour invoquer une sous-routine, suivi par le nom de procédure et les paramètres, le cas échéant.

Call SimpleSub()

ou

```
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
```

### Invocation de fonction

Pour invoquer une fonction (toute sous-routine qui contient une instruction **return**), il suffit d'utiliser son nom dans une expression. Ne pas utiliser l'instruction **call** pour appeler les fonctions. Exemple :

```
$QName = MakeQualifiedName($namespace, "entry")
```

## 15.12.9.3 Exemple de sous-routine

L'exemple suivant montre la déclaration de sous-routine et l'invocation.

```
[create $outputpath & $module & "output.txt"
' define sub SimpleSub()
Sub SimpleSub()
]SimpleSub() called
[endsub
' execute sub SimpleSub()
Call SimpleSub()
$ParamByValue = "Original Value"
]ParamByValue = [=$ParamByValue]
[$ParamByRef = "Original Value"
]ParamByRef = [=$ParamByRef]
' define sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
]CompleteSub called.
       param = [=$param]
       paramByValue = [=$paramByValue]
       paramByRef = [=$paramByRef]
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]
      Set values inside Sub
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]CompleteSub finished.
[endsub
' run sub CompleteSub()
```

```
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
]
ParamByValue=[=$ParamByValue]
ParamByRef=[=$ParamByRef]
[
Fermer
]
```

# 15.12.10 Types intégrés

La section décrit les propriétés des types intégrés utilisés dans les <u>variables prédéfinies</u><sup>1207</sup> qui décrivent le schéma parsé.

## 15.12.10.1 Bibliothèque

Cet objet représente la bibliothèque toute entière générée depuis le Schéma XML ou le DTD.

Propriétés	Туре	Description
SchemaNamespaces	Espace de noms <sup>(216</sup> collection	Espaces de noms dans cette bibliothèque
SchemaFilename	string	Nom du fichier XSD ou DTD à partir duquel cette bibliothèque est dérivée
SchemaType	entier	1 pour DTD, 2 pour Schéma XML
Guid	string	Une ID unique globale
CodeName	string	Nom de bibliothèque généré (dérivé du nom de fichier de schéma)

## 15.12.10.2 Espace de noms

Un objet d'espace de noms par espace de noms de Schéma XML est généré. Les composants de schéma qui ne se trouvent pas dans un espace de noms sont contenus dans un objet d'espace de noms spécial avec un NamespaceURI vide. Veuillez noter que pour DTD, les espaces de noms sont aussi dérivés des attributs dont les noms commencent avec "xmlns".

Propriétés	Туре	Description
CodeName	string	Nom pour le code généré (dérivé du préfixe)
LocalName	string	Préfixe d'espace de noms
NamespaceURI	string	URI d'espace de noms

Propriétés	Туре	Description
Types	Type <sup>(1217)</sup> collection	Tous les types contenus dans cet espace de noms
Library	Bibliothèque <sup>1216</sup>	Bibliothèque contenant cet espace de noms

# 15.12.10.3 Type

Cet objet représente un type simple ou complexe. Il est utilisé pour générer une classe dans le langage cible. Il existe un type supplémentaire par bibliothèque qui représente le document, qui a tous les éléments racine possibles en tant que membres.

Des types anonymes ont un LocalName vide.

Propriétés	Туре	Description
CodeName	string	Nom pour le code généré (dérivé depuis le nom local ou la déclaration de parent)
LocalName	string	Nom original dans le schéma
Namespace	Namespace <sup>1216</sup>	Espace de noms contenant ce type
Attributes	Membre <sup>(1218)</sup> collection	Attributs contenus dans ce type*
Elements	Membre <sup>(1213)</sup> collection	Élément enfant contenu dans ce type
IsSimpleType	booléen	True pour les types ssimples, false pour les types complexes
IsDerived	booléen	True si ce type est dérivé d'un autre type, qui est aussi représenté par un objet Type
IsDerivedByExtension	booléen	True si ce type est dérivé par extension
IsDerivedByRestriction	booléen	True si ce type est dérivé par restriction
IsDerivedByUnion	booléen	True si ce type est dérivé par union
lsDerivedByList	booléen	True si ce type est dérivé par list
BaseType	Туре	Le type de base de ce type (si IsDerived est true)
IsDocumentRootType	booléen	True si ce type représente le document lui-même
Library	Library <sup>1216</sup>	Bibliothèque contenant ce type
IsFinal	booléen	True si déclaré en tant que final dans le schéma
lsMixed	booléen	True si ce type peut avoir du contenu mixte

Propriétés	Туре	Description
lsAbstract	booléen	True si ce type est déclaré en tant qu'abstrait
IsGlobal	booléen	True si ce type est déclaré globalement dans le schéma
IsAnonymous	booléen	True si ce type est déclaré localement dans un élément

Pour les types simples uniquement :

Propriétés	Туре	Description
IsNativeBound	booléen	True si la liaison de type native existe
NativeBinding	NativeBinding <sup>(219</sup>	Liaison native pour ce type
Facettes	Facettes <sup>1219</sup>	Facettes de ce type
Whitespace	string	Raccourci vers la facette Whitespace

\* Les types complexes avec du contenu de texte (il s'agit des types avec du contenu mixte et des complexType avec simpleContent) ont un membre d'attribut non nommé supplémentaire qui représente le contenu de texte.

# 15.12.10.4 Membre

Cet objet représente un attribut ou un élément dans le Schéma XML. Il est utilisé pour créer des memnres de classe de types.

Propriétés	Туре	Description
CodeName	string	Nom pour le code généré (dérivé depuis le nom local ou la déclaration de parent)
LocalName	string	Nom original dans le schéma. Vide pour le membre spécial représentant le contenu de texte des types complexes.
NamespaceURI	string	L'URI d'espace de noms de cet Élément/Attribut dans le cadre du documents/streams de l'instance XML.
DeclaringType	<u>Type</u> <sup>[217</sup>	Type déclarant le membre à l'origine (égal au ContainingType pour les membres non hérités)
ContainingType	<u>Type</u> <sup>1217</sup>	Type où cela est un membre de
DataType	<u>Type</u> <sup>1217</sup>	Type de données du contenu de ce membre

Propriétés	Туре	Description
Bibliothèque	Bibliothèque <sup>(216</sup>	Bibliothèque contenant le DataType de ce membre
IsAttribute	booléenne	Vrai pour les attributs, faux pour les éléments
IsOptional	booléenne	Vrai si minOccurs = 0 ou attribut optionnel
IsRequired	booléenne	Vrai si minOccurs > 0 ou attribut requis
IsFixed	booléenne	Vrai pour des attributs fixes, la valeur se trouve dans la propriété Default
IsDefault	booléenne	Vrai pour des attributs par défaut, la valeur se trouve dans la propriété Default
IsNillable	booléenne	Vrai pour des éléments nillables
IsUseQualified	booléenne	Vrai si NamespaceURI n'est pas vide
MinOccurs	entier	minOccurs, comme dans le schéma. 1 pour des attributs requis
MaxOccurs	entier	maxOccurs, comme dans le schéma. 0 pour les attributs interdits, -1 pour non lié
Default	string	Valeur par défaut

# 15.12.10.5 NativeBinding

Cet objet représente la liaison d'un type simple vers un type natif dans le langage de programmation cible, comme spécifié par le mappage "schemanativetype".

Propriétés	Туре	Description
ValueType	string	Type natif
ValueHandler	string	Instance de classe de formateur

# 15.12.10.6 Facettes

Cet objet représente toutes les facettes d'un type simple. Les facettes héritées sont fusionnées avec les facettes déclarées explicitement. Si une facette Length est en effet, MinLength et MaxLength sont définies pour la même valeur.

Propriétés	Туре	Description
DeclaringType	Туре	Les facettes de type sont déclarées sur

Propriétés	Туре	Description
Whitespace	string	"preserve", "collapse" ou "replace"
MinLength	entier	Valeur de Facette
MaxLength	entier	Valeur de Facette
MinInclusive	entier	Valeur de Facette
MinExclusive	entier	Valeur de Facette
MaxInclusive	entier	Valeur de Facette
MaxExclusive	entier	Valeur de Facette
TotalDigits	entier	Valeur de Facette
FractionDigits	entier	Valeur de Facette
List	Collection de facette	Toutes les facettes en tant que liste

## Facette

Cet objet représente une seule facette avec sa valeur calculée effective pour un type spécifique.

Propriétés	Туре	Description
LocalName	string	Nom de facette
NamespaceURI	string	Espace de noms de facette
FacetType	string	Une de "normalization", "lexicalspace", "valuespace-length", "valuespace-enum" ou "valuespace-range"
DeclaringType	<u>Type</u> <sup>1217</sup>	Type sur lequel cette facette est déclarée
FacetCheckerName	string	Nom du vérificateur de facette (depuis le map schemafacet)
FacetValue	string ou entier	Valeur réelle de cette facette

# 16 L'API de MapForce

L'API basée sur COM permet aux clients d'accéder à la fonctionnalité de MapForce depuis un code personnalisé ou une application, et d'automatiser un large éventail de tâches.

L'API COM de MapForce suit les spécifications communes pour les serveurs Automation telles que définies par Microsoft. MapForce est automatiquement enregistré en tant qu'objet de serveur COM pendant l'installation. Une fois que l'objet de serveur COM est enregistré, vous pouvez l'invoquer depuis des applications et des langages de script qui prennent en charge des appels COM par programme. Ceci permet l'accès à l'API de MapForce non seulement depuis des environnements de développement utilisant .NET, C++ et Visual Basic, mais aussi depuis des langages de script comme JScript et VBScript.

Veuillez noter les points suivants :

- Si vous utilisez l'API de MapForce pour créer une application que vous voulez envoyer à d'autres clients, MapForce doit être installé sur chaque ordinateur client. Aussi, votre code d'intégration personnalisé doit être déployé vers (ou votre application installée sur) chaque ordinateur client.
- Certaines méthodes API, telles que Document.GenerateOutput, requièrent que la fenêtre principale de MapForce soit visible ou que MapForce (exécuté en tant que serveur COM) soit intégré à l'intérieur de l'interface utilisateur graphique. Si vous devez exécuter des mappages sans aucune assistance à travers divers plate-formes, envisagez l'utilisation de MapForce Server (https://www.altova.com/mapforce-server).

# 16.1 Accéder à l'API

Pour accéder à l'API COM de MapForce, une nouvelle instance de l'objet Application doit être créée dans votre application (ou script). Une fois que cet objet est créé, vous pouvez interagir avec MapForce en invoquant ses méthodes et propriétés telles que requises (par exemple, créer un nouveau document, ouvrir un document existant, générer un code de mappage, etc.).

## Exigences préalables

Pour rendre l'objet MapForce COM disponible dans votre projet de Visual Studio, ajoutez une référence au fichier MapForce type library (.tlb). Les instructions suivantes s'appliquent à Visual Studio 2013, mais sont semblables dans d'autres versions de Visual Studio :

- 1. Sur le menu **Projet**, cliquez sur **Ajouter référence**.
- 2. Cliquez sur **Parcourir** et sélectionnez le fichier **MapForce.tlb** situé dans le dossier d'installation MapForce.

### Un exemple de client API de MapForce dans C# est disponible à : C: \Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\API\C#.

Dans Java, l'API de MapForce est disponible par les bibliothèques pont Java-COM. Ces bibliothèques sont disponibles dans le dossier d'installation de MapForce : C:\Program Files (x86) \Altova\MapForce2023\JavaAPI (veuillez noter que ce chemin est valide quand MapForce 32-bit est exécuté sur Windows 64-bit, autrement, ajustez le chemin en conséquence).

- AltovaAutomation.dll: un wrapper JNI pour les serveurs d'automatisation d'Altova
- AltovaAutomation.jar: classes de Java pour accéder aux serveurs d'automatisation d'Altova
- MapForceAPI.jar: classes de Java qui wrappent l'interface d'automatisation de MapForce
- MapForceAPI\_JavaDoc.zip: un fichier Javadoc contenant une documentation d'aide pour l'API de Java

Pour permettre l'accès au serveur d'automatisation de MapForce directement depuis le code Java, les bibliothèques ci-dessus doivent être dans la classpath de Java.

Un exemple de client API de MapForce dans Java est disponible sous : C: \Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\API\Java.

Dans les langages de script tels que JScript ou VBScript, l'objet COM de MapForce est accessible par le Microsoft Windows Script Host (voir <u>https://msdn.microsoft.com/en-us/library/9bbdkx3k.aspx</u>). De tels scripts peuvent être écrits avec un éditeur de texte et n'ont pas besoin de compilation, puisqu'ils sont exécutés par Windows Script Host empaqueté avec Windows. (Pour vérifier si Windows Script Host est exécuté, saisissez wscript.exe /? à l'invite de commande). Plusieurs fichiers d'exemple JScript qui appellent l'API de MapForce sont disponibles sous : **C**:

### \Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\API\JScript.

**Note :** Pour MapForce 32-bit, le nom enregistré ou l'identificateur programmatique (Progld) de l'objet COM est MapForce.Application. Pour MapForce 64-bit, le nom est MapForce\_x64.Application. Néanmoins, n'oubliez pas que le programme d'appel accédera aux entrée de registre CLASSES dans sa propre ruche de registre, ou groupe (32-bit ou 64-bit). C'est pourquoi, si vous exécutez des scripts en utilisant l'invite de commande standard et Windows Explorer sur Windows 64-bit, les entrées de registre 64-bit seront accédées, et pointeront vers le MapForce 64-bit. Pour cette raison, si les deux MapForce 32-bit

et 64-bit sont installés, une gestion spéciale est nécessaire afin de pouvoir appeler le MapForce 32-bit. Par exemple, si on part du principe que Windows Scripting Host est le programme d'appel, procéder comme suit :

- 1. Changer le répertoire actuel à C:\Windows\SysWOW64.
- 2. Dans la ligne de commande, saisir **wscript.exe** suivi du chemin vers le script que vous souhaitez exécuter, par exemple :

```
wscript.exe "C:\Users\...
\Documents\Altova\MapForce2023\MapForceExamples\API\JScript\start.js"
```

### Lignes directrices

Les lignes directrices suivantes devraient être prises en compte dans votre code client :

- Ne gardez pas de références aux objets en mémoire plus longtemps que vous en avez besoin. Si un utilisateur interagit entre deux appels de votre client, il n'y a pas de garantie que ces références sont toujours valides.
- Ayez conscience que si votre code client a un incident, les instances de MapForce demeurent tout de même dans le système. Pour les détails comment éviter les messages d'erreur, voir <u>Gestion</u> d'erreur<sup>122</sup>.
- Libérez les références explicitement si vous utilisez les langages tels que C++.

### Créer l'objet de l'application

La syntaxe pour créer l'objet d'Application de démarrage dépend du langage de programmation, tel qu'expliqué dans l'exemple ci-dessous :

#### C#

```
// Créez une nouvelle instance de MapForce par l'interface d'automatisation.
MapForceLib.Application objMapForce = new MapForceLib.Application();
```

#### Java

```
// Démarrez MapForce en tant que serveur COM.
com.altova.automation.MapForce.Application objMapForce = new Application();
// les serveurs COM démarrent de manière invisible, donc nous les rendons visibles
objMapForce.setVisible(true);
```

#### JScript

```
// Accédez à une instance en cours d'exécution, ou créez une nouvelle instance de
MapForce.
try {
        objMapForce = WScript.GetObject ("", "MapForce.Application");
        // afficher l'application s'il s'agit d'une nouvelle instance
        objMapForce.Visible = true;
   }
catch(err) { WScript.Echo ("Ne peut pas accéder à ou créer MapForce.Application"); }
```

#### VBA

```
' Créez une nouvelle instance de MapForce.
Dim objMapForce As Application
Set objMapForce = CreateObject("MapForce.Application")
```

#### VBScript

```
' Accédez à une instance en cours d'exécution, ou créez une nouvelle instance de
MapForce.
Set objMapForce = GetObject("MapForce.Application");
```

#### Visual Basic

```
Dim objMapForce As MapForceLib.Application = New MapForceLib.Application
```
# 16.2 Le modèle d'objet

Le point de départ de chaque application qui utilise l'API de MapForce est l'objet d'<u>Application</u><sup>(23)</sup>. Toutes les autres interfaces sont accédées à travers l'objet d'application en tant que point de départ.

Le modèle d'objet de l'API de MapForce peut être représenté comme suit (chaque niveau d'indentation indique une relation enfant-parent avec le niveau juste au-dessus) :



Pour toute information sur la création d'une instance d'objet d'Application, voir <u>Accéder à l'API</u><sup>1222</sup>. Pour toute référence aux objets exposés par l'API, voir <u>Référence d'objet</u><sup>1253</sup>.

# 16.3 Gestion d'erreur

L'API de MapForce renvoie les erreurs de deux manières différentes. Chaque méthode API renvoie un HRESULT. Cette valeur de renvoi informe l'appelant sur les défaillances pendant l'exécution de la méthode. Si l'appel a réussi, la valeur de renvoi est égale à s\_ok. Les programmeurs C/C++ utilisent généralement HRESULT pour détecter des erreurs.

Visual Basic, les langages de scripts, et d'autres environnements de développement de haut niveau ne donnent pas accès au HRESULT de renvoi d'un appel COM. Ils utilisent le deuxième mécanisme de communication de l'erreur pris en charge par l'API de MapForce, l'interface IErrorInfo. Si une erreur se produit, l'API crée un nouvel objet qui met en œuvre l'interface IErrorInfo. L'environnement de développement prend cette interface et remplit son propre mécanisme de communication d'erreur à base des informations fournies.

Le texte suivant décrit comment gérer les erreurs survenues depuis l'API de MapForce dans les différents environnements de développement.

## Visual Basic

Une manière commune de gérer des erreurs dans Visual Basic est de définir un gestionnaire d'erreurs. Ce gestionnaire d'erreur peut être défini avec la déclaration Sur Erreur. Normalement, le gestionnaire affiche un message d'erreur et nettoie un peu pour éviter des références détachées et tout genre de fuites de ressources. Visual Basic remplit son propre objet Err avec des informations de l'interface IErrorInfo.

```
Sub Validate()
'placer les déclarations de variable ici
'définir le gestionnaire d'erreurs
On Error GoTo ErrorHandler
'si la génération échoue, l'exécution de programme continue à ErrorHandler :
objMapForce.ActiveDocument.GenerateXSLT()
'le code supplémentaire est mis ici
'quitter
Exit Sub
ErrorHandler :
MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
 "Description: " & Err.Description)
End Sub
```

## JavaScript

La mise en œuvre de Microsoft de JavaScript (JScript) fournit un mécanisme « try-catch » pour gérer les erreurs soulevées par les appels COM. Il est très semblable à l'approche de VisualBasic dans le sens que vous déclarez également un objet d'erreur contenant les informations nécessaires.

```
function Generate() {
  // veuillez insérer les déclarations de variable ici
  try {
    objMapForce.ActiveDocument.GenerateXSLT();
    }
  catch (Error) {
    sError = Error.description;
    nErrorCode = Error.number & 0xffff;
  return false;
    }
  return true;
}
```

# C/C++

C/C++ vous donne accès facilement à HRESULT de l'appel COM et à IErrorInterface.

```
HRESULT hr;
// Call GenerateXSLT() de MapForce API
if(FAILED(hr = ipDocument->GenerateXSLT()))
{
IErrorInfo *ipErrorInfo = Null;
if(SUCCEEDED(::GetErrorInfo(0, &ipErrorInfo)))
{
BSTR bstrDescr;
ipErrorInfo->GetDescription(&bstrDescr);
// gérer l'information d'erreur
wprintf(L"Error message:\t%s\n",bstrDescr);
::SysFreeString(bstrDescr);
// publier info d'erreur
ipErrorInfo->Release();
}
}
```

# 16.4 Exemple Projet C#

Après avoir installé MapForce, un exemple de projet client de MapForce API pour C# est disponible dans le répertoire C:\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\API.

Pour compiler et exécuter l'exemple, ouvrez la solution .sln file dans Visual Studio et exécutez **Déboguer** | **Démarrer débogage** ou **Déboguer** | **Démarrer sans débogage**.

Note : Si vous avez un système d'exploitation 64-bit et que vous utilisez une installation 32-bit installation de MapForce, ajoutez la plate-forme x86 dans le Manager de configuration de la solution et générez l'exemple en utilisant cette configuration. Une nouvelle plateforme x86 (pour la solution active dans Visual Studio) peut être créée dans le nouveau dialogue de la plateforme de solution (Build | Manager de configuration |plateforme de solution active | <New...>).

Lorsque vous exécutez un exemple, un formulaire Windows est affiché contenant des boutons invoquant des opérations de base de MapForce :

- Démarrer MapForce
- Créer un nouveau design de mappage
- Ouvrez le fichier CompletePO.mfd file du dossier ...\MapForceExamples (veuillez noter que vous allez éventuellement devoir ajuster le chemin pour indiquer le dossier \MapForceExamples sur votre appareil)
- Générez le code C# dans un répertoire temp
- Arrêter MapForce

💀 Form1	_	×
Start MapForce Create new mapping Open CompletePO.mfd Generate C# code		
OnDocumentOpened Event On/Off Shutdown MapForce		

## Liste de codes

Cette liste est dotée de commentaires pour faciliter la compréhension. Le code consiste essentiellement en une série de gestionnaires pour les boutons dans l'interface utilisateur affichée ci-dessus.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```
using System.Data;
using System.Drawing;
using System.Ling;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication2
public partial class Form1 : Form
ł
public Form1()
InitializeComponent();
}
// Une instance de MapForce accédée par le biais de son interface d'automatisation.
MapForceLib.Application MapForce;
// Emplacement d'exemples installés avec MapForce
String strExamplesFolder;
private void Form1_Load(object sender, EventArgs e)
}
// gestionnaire pour le bouton « Démarrer MapForce »
private void StartMapForce_Click(object sender, EventArgs e)
if (MapForce == null)
{
Cursor.Current = Cursors.WaitCursor;
// si nous n'avons pas d'instance de MapForce, nous en créons une et la rendons visible.
MapForce = new MapForceLib.Application();
MapForce.Visible = true;
// localiser des exemples installés avec MapForce.
int majorVersionYear = MapForce.MajorVersion + 1998;
strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My Documents\
\Altova\\MapForce" + Convert.ToString(majorVersionYear) + "\\MapForceExamples\\";
Cursor.Current = Cursors.Default;
}
else
// si nous avons déjà une instance de MapForce en cours d'exécution, nous changeons son
indicateur de visibilité.
MapForce.Visible = !MapForce.Visible;
}
}
// gestionnaire pour le bouton « Open CompletePO.mfd »
private void openCompletePO Click(object sender, EventArgs e)
```

```
if (MapForce == null)
StartMapForce_Click(null, null);
// Ouvrir un des exemples de fichiers installés avec le produit.
MapForce.OpenDocument(strExamplesFolder + "CompletePO.mfd");
}
// gestionnaire pour le bouton « Créer un nouveau mappage »
private void newMapping_Click(object sender, EventArgs e)
if (MapForce == null)
StartMapForce_Click(null, null);
// Créer un nouveau mappage
MapForce.NewMapping();
}
// gestionnaire pour le bouton « Arrêter MapForce »
// arrêter l'instance d'application en libérant l'objet COM.
private void shutdownMapForce_Click(object sender, EventArgs e)
if (MapForce != null)
// permettre l'arrêt de MapForce en libérant l'IU
MapForce.Visible = false;
// explicitement libérer l'objet COM
try
ł
while (System.Runtime.InteropServices.Marshal.ReleaseComObject(MapForce) > 0) ;
finally
ł
// éviter un accès plus tardif de cet objet.
MapForce = null;
}
// gestionnaire pour le bouton « Générer le code C# »
private void generateCppCode_Click(object sender, EventArgs e)
if (MapForce == null)
listBoxMessages.Items.Add("start MapForce first.");
// les erreurs COM sont renvoyées à C# comme exceptions. Nous utilisons un bloc try/catch
pour les gérer.
try
{
MapForceLib.Document doc = MapForce.ActiveDocument;
listBoxMessages.Items.Add("Active document " + doc.Name);
doc.GenerateCHashCode();
```

```
}
catch (Exception ex)
// L'appel COM n'a pas été couronné de succès.
// Probablement qu'aucune instance d'application n'a été lancée ou aucun document n'est
ouvert.
MessageBox.Show("COM error: " + ex.Message);
}
delegate void addListBoxItem_delegate(string sText);
// appeler du thread d'interface utilisateur
private void addListBoxItem(string sText)
listBoxMessages.Items.Add(sText);
}
// méthode wrapper pour permettre d'appeler les méthodes de contrôle IU depuis un thread
de travail
void syncWithUIthread(Control ctrl, addListBoxItem_delegate methodToInvoke, String sText)
ł
// Control.Invoke: Exécute sur le thread IU, mais le thread appelant attend la saisie
avant de poursuivre.
// Control.BeginInvoke: Exécute sur le thread IU et le thread appelant n'attend pas
l'achèvement.
if (ctrl.InvokeRequired)
ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
}
// gestionnaire d'événement pour l'événement OnDocumentOpened
private void handleOnDocumentOpened(MapForceLib.Document i_ipDocument)
String sText = "";
if (i_ipDocument.Name.Length > 0)
sText = "Document " + i_ipDocument.Name + " was opened!";
else
sText = "A new mapping was created.";
// nous devons synchroniser le thread appelant avec le thread IU car
// les événements COM sont déclenchés depuis un thread de travail
addListBoxItem_delegate methodToInvoke = new addListBoxItem_delegate(addListBoxItem);
// appeler syncWithUIthread avec les arguments suivants :
// 1 - listBoxMessages - zone de liste pour afficher les messages depuis les événements
COM
// 2 - methodToInvoke - un déléqué C# qui indique la méthode qui sera appelée depuis le
thread IU
// 3 - sText - le texte à afficher dans la zone de liste
syncWithUIthread(listBoxMessages, methodToInvoke, sText);
}
private void checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
if (MapForce != null)
```

} } }

{
if (checkBoxEventOnOff.Checked)
MapForce.OnDocumentOpened += new
MapForceLib.\_IApplicationEvents\_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
else
MapForce.OnDocumentOpened -= new
MapForceLib.\_IApplicationEvents\_OnDocumentOpenedEventHandler(handleOnDocumentOpened);

# 16.5 Exemple Projet Java

Une fois que vous avez installé MapForce, un exemple de projet client de MapForce API pour Java est disponible dans le répertoire **C**:

\Users\<username>\Documents\Altova\MapForce2023\MapForceExamples\API.

Vous pouvez tester l'exemple de Java directement depuis la ligne de commande, utilisant le fichier de commandes BuildAndRun.bat, ou vous pouvez compiler et exécuter l'exemple de projet depuis l'intérieur d'Eclipse. Voir plus bas pour des instructions sur l'utilisation de ces procédures.

## Liste de fichier

Le dossier d'exemples de Java contient toutes les fichiers requis pour exécuter l'exemple de projet. Ces fichiers sont recensés ci-dessous :

Java-COM bridge : DLL part
Java-COM bridge : Java library part
Java classes of the MapForce API
Code source de l'exemple Java
Fichier de commandes pour compiler et exécuter un exemple de code depuis l'invite de la ligne de commande. Attend le dossier où Java Virtual Machine réside en tant que paramètre.
Fichier d'assistance pour le projet Eclipse
Fichier de projet Eclipse
Le fichier Javadoc contenant une documentation d'aide pour Java API

# Ce que fait l'exemple

L'exemple démarre MapForce et effectue quelques opérations, y compris ouvrir et fermer les documents. Quand il a terminé, MapForce reste ouvert. Vous devez le fermer manuellement.

## Exécuter l'exemple depuis la ligne de commande

Pour exécuter l'exemple depuis la ligne de commande, ouvrez une fenêtre d'invite de commande, allez au dossier Java du dossier d'exemples API (*voir ci-dessus pour l'emplacement*), puis tapez :

buildAndRun.bat "<Path-to-the-Java-bin-folder>"

Le dossier binaire de Java doit être celui d'un kit de développement de Java (JDK) 7 ou d'une installation ultérieure sur votre ordinateur.

Appuyez sur la clé Retour. La source de Java dans RunMapForce. java sera compilée, puis exécutée.

## Charger l'exemple dans Eclipse

Ouvrez Eclipse et utilisez la commande Fichier | Importer... Commande | Général | Projets existants dans flux de travail pour ajouter un fichier de projet Eclipse (.project) situé dans le dossier Java du dossier d'exemples API (*voir au-dessus pour l'emplacement*). Le projet RunMapForce apparaîtra ensuite dans votre Package Explorer ou Navigator.

Sélectionnez le projet, puis la commande Run as | Java Application pour exécuter l'exemple.

**Note :** Vous pouvez sélectionner un nom de classe ou de méthode de l'API de Java et appuyez sur F1 pour recevoir de l'aide pour cette classe ou méthode.

#### Liste de code source de Java

Le code source de Java dans l'exemple de fichier RunMapForce.java iest recensé ci-dessous avec des commentaires.

```
// access general JAVA-COM bridge classes
import java.util.Iterator;
import com.altova.automation.libs.*;
// access XMLSpy Java-COM bridge
import com.altova.automation.MapForce.*;
import com.altova.automation.MapForce.Enums.ENUMProgrammingLanguage;
/**
 * Un exemple simple qui démarre le serveur COM et réalise quelques opérations sur celui-
ci.
 * N'hésitez pas à élargir.
 * /
public class RunMapForce
ł
public static void main(String[] args)
// une instance de l'application.
Application mapforce = null;
// au lieu d'utiliser la gestion d'erreur COM, utilisez le mécanisme d'exception de Java.
try
// Démarrez MapForce en tant que serveur COM.
mapforce = new Application();
// les serveurs COM démarrent de manière invisible, donc nous les rendons visibles
mapforce.setVisible(true);
// Les lignes suivantes se joignent aux événements de l'application en utilisant une mise
en œuvre par défaut
// pour les événements et écrase l'une de ses méthodes.
// Si vous souhaitez écraser tous les événements de document, il est plus judicieux de
dériver votre classe d'écoute
// de DocumentEvents et met en œuvre toutes les méthodes de cette interface.
mapforce.addListener(new ApplicationEventsDefaultHandler()
ł
@Override
public void onDocumentOpened(Document i ipDoc) throws AutomationException
String name = i_ipDoc.getName();
```

```
if (name.length() > 0)
System.out.println("Document " + name + " was opened.");
else
System.out.println("A new mapping was created.");
});
// Localisez des exemples installés avec le produit.
int majorVersionYear = mapforce.getMajorVersion() + 1998;
String strExamplesFolder = System.getenv("USERPROFILE") + "\\Documents\\Altova\\MapForce"
+ Integer.toString(majorVersionYear) + "\\MapForceExamples\\";
// créez un nouveau mappage MapForce et générez le code c++
Document newDoc = mapforce.newMapping();
ErrorMarkers err1 = newDoc.generateCodeEx(ENUMProgrammingLanguage.eCpp);
afficher(err1);
// ouvrez CompletePO.mfd et générez le code c++
Document doc = mapforce.openDocument(strExamplesFolder + "CompletePO.mfd");
ErrorMarkers err2 = doc.generateCodeEx(ENUMProgrammingLanguage.eCpp);
afficher(err2);
doc.close();
doc = null;
System.out.println("Watch MapForce!");
}
catch (AutomationException e)
// e.printStackTrace();
}
finally
{
// Assurez-vous que MapForce peut être arrêté correctement.
if (mapforce != null)
mapforce.dispose();
// Puisque le serveur COM a été rendu visible et l'est encore, il poursuivra son
exécution
// et devra être arrêté manuellement.
System.out.println("Now close MapForce!");
}
public static void display(ErrorMarkers err) throws AutomationException
Iterator<ErrorMarker> itr = err.iterator();
if (err.getCount() == 0)
System.out.print("Code generation completed successfully.\n");
while (itr.hasNext())
String sError = "";
Object element = itr.next();
if (element instanceof ErrorMarker)
```

```
sError = ((ErrorMarker)element).getText();
System.out.print("Error text: " + sError + "\n");
}
}
```

# 16.6 Exemples JScript

Une fois que vous avez installé MapForce, plusieurs fichiers d'exemple JScript sont disponibles dans le répertoire C:\Users\<username>\Documents\Altova\MapForce<%PRODYEAR%\MapForceExamples\API.

Les exemples de fichiers peuvent être exécutés d'une des deux manières :

- Depuis la ligne de commande : Ouvrez une fenêtre d'invite de commandes et saisissez le nom d'un des exemples de scripts (par exemple, start.js). L'Hôte de script Windows qui est empaqueté avec Windows exécutera le script.
- Depuis Windows Explorer : Dans Windows Explorer, parcourez pour le fichier JScript et double-cliquez sur celui-ci. L'Hôte de script Windows qui est empaqueté avec Windows exécutera le script. Après l'exécution du script, la console de commande se ferme automatiquement.

Les exemples de fichiers suivants sont inclus :

Start.js	Démarrez Mapforce enregistré comme serveur d'automatisation ou connectez-vous à une instance d'exécution (voir <u>Start Application</u> <sup>(227)</sup> ).
DocumentAccess.js	Montre comment ouvrir, itérer et fermer les documents (voir <u>Accès simple aux</u> <u>documents</u> ).
GenerateCode.js	Montre comment appeler la génération de code utilisant JScript (voir <u>Générer le</u> <u>code</u> <sup>1238</sup> ).
Readme.txt	Fournit l'aide basique pour exécuter les scripts.

Cette documentation comprend en plus quelques listes de code JScript supplémentaires :

- <u>Exemple : Génération de code 1241</u>
- <u>Exemple : Exécution de mappage</u><sup>1243</sup>
- Exemple : Project Support<sup>1247</sup>

# 16.6.1 Démarrer Application

La liste de code JScript ci-dessous démarre l'application et l'arrête. Si une instance de l'application est déjà en cours d'exécution, l'instance en cours sera renvoyée. Pour exécuter un script, démarrez-le depuis la fenêtre d'invite de commande ou depuis Windows Explorer, voir également <u>Accéder à l'API</u><sup>[22]</sup>.

```
// Initialisez l'objet COM de l'application. Ceci démarrera une nouvelle instance de
l'application et
// renvoie son objet COM principal. Dépendant des paramètres COM, l'objet COM principal
d'une
// application déjà en cours peut être renvoyé.
try { objMapForce = WScript.GetObject("", "MapForce.Application"); }
```

```
catch(err) {}
if( typeof( objMapForce ) == "undefined" )
{
try { objMapForce = WScript.GetObject("", "MapForce_x64.Application") }
catch(err)
 WScript.Echo( "Ne peut accéder à ou créer MapForce.Application" );
 WScript.Quit();
 }
}
// si nouvellement lancée, l'application démarrera sans son IU visible. Mettez-la à
visible.
objMapForce.Visible = true;
WScript.Echo(objMapForce.Edition + " a démarré avec succès. ");
objMapForce.Visible = false; // arrêtera l'application si elle n'a plus de connexions COM
//objMapForce.Visible = true; // maintiendra l'application en cours d'exécution avec l'IU
visible
```

# 16.6.2 Accès simple aux documents

La liste JScript ci-dessous montre comment ouvrir les documents, définir un document en tant que document actif, itérer à travers des documents ouverts et fermer des documents.

```
// Initialisez l'objet COM de l'application. Ceci démarrera une nouvelle instance de
l'application et
// renvoie son objet COM principal. Dépendant des paramètres COM, l'objet COM principal
d'une
// application déjà en cours peut être renvoyé.
try { objMapForce = WScript.GetObject("", "MapForce.Application"); }
catch(err) {}
if( typeof( objMapForce ) == "undefined" )
try { objMapForce = WScript.GetObject("", "MapForce_x64.Application") }
catch(err)
 {
WScript.Echo( "Ne peut accéder à ou créer MapForce.Application" );
WScript.Quit();
}
}
// si nouvellement lancée, l'application démarrera sans son IU visible. Mettez-la à
visible.
objMapForce.Visible = true;
// ******************************** code snippet pour "Simple Document Access"
*****
```

```
// Localisez des exemples via la variable d'environnement USERPROFILE. Le chemin doit
être adapté aux versions majeures de release.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objMapForce.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\
\Altova\\MapForce" + majorVersionYear + "\\MapForceExamples\\";
objMapForce.Documents.OpenDocument(strExampleFolder + "CompletePO.mfd");
objMapForce.Documents.OpenDocument(strExampleFolder + "Altova_Hierarchical_DB.mfd");
// ****************************** code snippet pour "Iteration"
*****
// parcourez tous les documents ouverts utilisant un énumérateur JScript
pour (var iterDocs = new Enumerator(objMapForce.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
ł
objName = iterDocs.item().Name;
WScript.Echo("Nom du document : " + objName);
}
// parcourez tous les documents ouverts utilisant l'accès basé sur l'index pour la
collection des documents
pour (i = objMapForce.Documents.Count; i > 0; i--)
objMapForce.Documents.Item(i).Close();
// ********************************* code snippet pour "Iteration"
//objMapForce.Visible = false; // arrêtera l'application si elle n'a plus de connexions
COM
objMapForce.Visible = true; // maintiendra l'application en cours avec l'IU visible
```

Le code recensé ci-dessus est disponible en tant qu'exemple de fichier (voir <u>Exemples JScript</u><sup>1223</sup>). Pour exécuter le script, démarrez-le d'une fenêtre d'invite de commandes ou depuis Windows Explorer.

# 16.6.3 Générer le code

La liste JScript ci-dessous montre comment ouvrir des documents, définir un document en tant que document actif, itérer à travers les documents ouverts, et générez le code C++.

```
// Initialisez l'objet COM de l'application. Ceci démarrera une nouvelle instance de
l'application et
// renvoie son objet COM principal. Dépendant des paramètres COM, l'objet COM principal
d'une
// application déjà en cours peut être renvoyé.
try { objMapForce = WScript.GetObject("", "MapForce.Application"); }
```

```
catch(err) {}
if( typeof( objMapForce ) == "undefined" )
{
try { objMapForce = WScript.GetObject("", "MapForce_x64.Application") }
catch(err)
WScript.Echo( "Ne peut accéder à ou créer MapForce.Application" );
WScript.Quit();
}
}
// si nouvellement lancée, l'application démarrera sans son IU visible. Mettez-la à
visible.
objMapForce.Visible = true;
// ********************************** code snippet pour "Simple Document Access"
*****
// Localisez des exemples via la variable d'environnement USERPROFILE. Le chemin doit
être adapté aux versions majeures de release.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objMapForce.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\
\Altova\\MapForce" + majorVersionYear + "\\MapForceExamples\\";
objMapForce.Documents.OpenDocument(strExampleFolder + "CompletePO.mfd");
//objMapForce.Documents.OpenDocument(strExampleFolder + "Altova_Hierarchical_DB.mfd");
objMapForce.Documents.NewDocument();
// ********************************** code snippet pour "Simple Document Access"
****
// ****************************** code snippet pour "Iteration"
objText = "";
// parcourez tous les documents ouverts utilisant un énumérateur JScript et générez le
code c++
pour (var iterDocs = new Enumerator(objMapForce.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
objText += "Generated c++ code result for document " + iterDocs.item().Name + " :\n";
objErrorMarkers = iterDocs.item().generateCodeEx(1); // ENUMProgrammingLanguage.eCpp = 1
bSuccess = true;
for (var iterErrorMarkers = new Enumerator(objErrorMarkers); !iterErrorMarkers.atEnd();
iterErrorMarkers.moveNext())
bSuccess = false;
objText += "\t" + iterErrorMarkers.item().Text + "\n";
}
if (bSuccess)
 objText += "\tCode generation completed successfully.\n";
```

Le code recensé ci-dessus est disponible en tant qu'exemple de fichier (voir <u>Exemples JScript</u><sup>1237</sup>). Pour exécuter le script, démarrez-le d'une fenêtre d'invite de commandes ou depuis Windows Explorer.

# 16.6.4 Générer le code (alternatif)

L'exemple JScript suivant montre comment charger un document existant et générer différents types de code de mappage pour celui-ci.

```
// ----- commencer l'exemple JScript ----
// Générer le code pour le mappage existant.
// fonctionne avec Windows scripting host.
// ----- fonction d'aide -----
function Exit(strErrorText)
{
WScript.Echo(strErrorText);
WScript.Quit(-1);
}
function ERROR(strText, objErr)
if (objErr != null)
Exit ("ERROR: (" + (objErr.number & 0xffff) + ")" + objErr.description + " - " +
strText);
else
Exit ("ERROR: " + strText);
}
// -----
                         _____
// ----- PRINCIPAL -----
// ----- crée le Shell et FileSystemObject du windows scripting
try
```

```
objWshShell = WScript.CreateObject("WScript.Shell");
 objFS0 = WScript.CreateObject("Scripting.FileSystemObject");
}
catch(err)
 { Exit("Can't create WScript.Shell object"); }
// ----- ouvrir MapForce ou accéder à une instance d'exécution et la rendre visible
try
{
 objMapForce = WScript.GetObject ("", "MapForce.Application");
objMapForce.Visible = true; // supprimer cette ligne pour effectuer le traitement en
arrière-plan
}
catch(err) { WScript.Echo ("Ne peut pas accéder à ou créer MapForce.Application"); }
// ---- ouvrez un mappage existant. adaptez ceci à vos besoins !
objMapForce.OpenDocument(objFSO.GetAbsolutePathName ("Test.mfd"));
// ----- accédez au mappage pour avoir accès aux méthodes de génération de code
var objDoc = objMapForce.ActiveDocument;
// ----- définir les propriétés de sortie de génération de code et appeler toutes les
méthodes de génération de code.
// ----- adaptez les répertoires de sortie à vos besoins
try
// ----- la génération de code utilise quelques-unes de ces options
var objOptions = objMapForce.Options;
// ----- générer XSLT -----
objOptions.XSLTDefaultOutputDirectory = "C:\\test\\TestCOMServer\\XSLT";
 objDoc.GenerateXSLT();
// ----- générer le code Java -----
 objOptions.CodeDefaultOutputDirectory = "C:\\test\\TestCOMServer\\Java";
 objDoc.GenerateJavaCode();
// ----- générer le code CPP, utiliser les mêmes options de code cpp comme la dernière
fois --
 objOptions.CodeDefaultOutputDirectory = "C:\\test\\TestCOMServer\\CPP";
 objDoc.GenerateCppCode();
// ----- générer le code C#, utiliser les options de code C# code comme la dernière fois
 objOptions.CodeDefaultOutputDirectory = "C:\\test\\TestCOMServer\\CHash";
objDoc.GenerateCHashCode();
}
catch (err)
 { ERROR ("en générant XSL ou le code de programme", err); }
// masquer MapForce pour permettre son arrêt
objMapForce.Visible = false;
```

// ----- fin de l'exemple -----

# 16.6.5 Exécuter un mappage

L'exemple Jscript suivant montre comment charger un document existant avec un mappage simple, accéder à ses composants, définir des noms de fichier d'instance d'entrée et de sortie et exécuter le mappage.

```
/*
Cet exemple de fichier réalise les opérations suivantes :
Charger le document de mappage de MapForce.
Trouver le composant source et cible.
Définir les noms de fichier d'instance d'entrée et de sortie.
Exécuter la transformation.
Fonctionne avec l'hôte de scripting Windows.
*/
// ---- assistance générale -----
function Exit( message )
{
WScript.Echo( message );
WScript.Quit(-1);
}
function ERROR( message, err )
if( err != null )
Exit( "ERROR: (" + (err.number & 0xffff) + ") " + err.description + " - " + message );
else
Exit( "ERROR: " + message );
}
// ---- constantes MapForce -----
var eComponentUsageKind_Unknown = 0;
var eComponentUsageKind_Instance = 1;
var eComponentUsageKind_Input = 2;
var eComponentUsageKind_Output = 3;
// ---- assistance MapForce -----
// Cherche dans le mappage spécifié pour un composant par le nom et le renvoie.
// Si pas trouvé, renvoie une erreur.
function FindComponent( mapping, component_name )
{
```

```
var components = mapping.Components;
for( var i = 0 ; i < components.Count ; ++i )</pre>
 {
var component = components.Item( i + 1 );
if( component.Name == component_name )
return component;
throw new Error( "Cannot find component with name " + component_name );
}
// Parcourt les composants dans un mappage et renvoie le premier qu'il trouve agissant en
tant que
// composant source (c'est-à-dire qui a des connexions à sa droite).
function GetFirstSourceComponent( mapping )
var components = mapping.Components;
for( var i = 0 ; i < components.Count ; ++i )</pre>
 {
var component = components.Item( i + 1 );
if( component.UsageKind == eComponentUsageKind_Instance &&
 component.HasOutgoingConnections )
 {
return component;
 }
throw new Error( "Cannot find a source component" );
}
// Parcourt les composants dans un mappage et renvoie le premier qu'il trouve agissant en
tant que
// composant cible (c'est-à-dire qui a des connexions à sa gauche).
function GetFirstTargetComponent( mapping )
ł
var components = mapping.Components;
for( var i = 0 ; i < components.Count ; ++i )</pre>
 {
var component = components.Item( i + 1 );
if( component.UsageKind == eComponentUsageKind_Instance &&
component.HasIncomingConnections )
 {
return component;
 }
throw new Error( "Cannot find a target component" );
}
function IndentTextLines( s )
ł
return "\t" + s.replace( /\n/g, "\n\t" );
}
```

```
function GetAppoutputLineFullText( oAppoutputLine )
{
var s = oAppoutputLine.GetLineText();
var oAppoutputChildLines = oAppoutputLine.ChildLines;
var i;
for( i = 0 ; i < oAppoint ChildLines.Count ; ++i )</pre>
 {
 oAppoutputChildLine = oAppoutputChildLines.Item( i + 1 );
 sChilds = GetAppoutputLineFullText( oAppoutputChildLine );
 s += "\n" + IndentTextLines( sChilds );
 }
return s;
}
// Créer un string bien formaté depuis AppOutputLines
function GetResultMessagesString( oAppoutputLines )
var s1 = "Transformation result messages:\n";
var oAppoutputLine;
var i;
for( i = 0 ; i < oAppoutputLines.Count ; ++i )</pre>
 {
 oAppoutputLine = oAppoutputLines.Item( i + 1 );
 s1 += GetAppoutputLineFullText( oAppoutputLine );
 s1 += "\n";
 }
return s1;
}
// ---- PRINCIPAL -----
var wshShell;
var fso;
var mapforce;
// créer le Shell et FileSystemObject du windows scripting system
try
{
wshShell = WScript.CreateObject( "WScript.Shell" );
 fso = WScript.CreateObject( "Scripting.FileSystemObject" );
}
catch( err )
{ ERROR( "Can't create windows scripting objects", err ); }
// ouvrir MapForce ou accéder à une instance en cours d'exécution
try
{
```

```
mapforce = WScript.GetObject( "", "MapForce.Application" );
}
catch( err )
 { ERROR( "Can't access or create MapForce.Application", err ); }
try
// Rendre l'IU de MapForce visible. Il s'agit d'une exigence API pour la génération de
sortie.
mapforce.Visible = true;
// ouvrir un mappage existant.
var sMapForceExamplesPath = fso.BuildPath(
 wshShell.SpecialFolders( "MyDocuments" ),
"Altova\\MapForce2023\\MapForceExamples" );
var sDocFilename = fso.BuildPath( sMapForceExamplesPath, "PersonList.mfd" );
var doc = mapforce.OpenDocument( sDocFilename );
// Trouver des composants existants par le nom dans le mappage principal.
// Notez que les noms des composants ne sont pas nécessairement uniques étant donné que
le nom du composant d'un schéma
// est dérivé de son nom de fichier du schéma.
var source_component = FindComponent( doc.MainMapping, "Employees" );
var target_component = FindComponent( doc.MainMapping, "PersonList" );
// Si vous ne connaissez pas les noms des composants pour une raison ou une autre, vous
pourrez
// utiliser les fonctions suivantes à la place de FindComponent.
//var source_component = GetFirstSourceComponent( doc.MainMapping );
//var target_component = GetFirstTargetComponent( doc.MainMapping );
// spécifier les fichiers d'entrée et de sortie désirés.
 source_component.InputInstanceFile = fso.BuildPath( sMapForceExamplesPath,
"Employees.xml" );
target_component.OutputInstanceFile = fso.BuildPath( sMapForceExamplesPath,
"test_transformation_results.xml" );
// Effectuer la transformation.
// Vous pouvez utiliser doc.GenerateOutput() si vous n'avez pas besoin de messages de
résultat.
// Si vous avez un mappage avec plus d'un compostant cible et vous voulez
// exécuter la transformation uniquement pour un seul composant cible,
// appelez target_component.GenerateOutput() à la place.
var result_messages = doc.GenerateOutputEx();
var summary_info =
"Transformation performed from " + source_component.InputInstanceFile + "\n" +
"to " + target_component.OutputInstanceFile + "\n\n" +
GetResultMessagesString( result_messages );
WScript.Echo( summary_info );
}
catch( err )
{
```

```
ERROR( "Failure", err );
}
```

# 16.6.6 **Project Tasks**

L'exemple JScript suivant montre comment utiliser l'API de MapForce pour automatiser les tâches pertinentes aux projets de MapForce. Avant d'exécuter l'exemple, assurez-vous d'éditer la variable strSamplePath pour qu'elle vous dirige vers le dossier de votre installation de MapForce : C: \Users\<username>\Documents\Altova\MapForce2023\MapForceExamples.

Afin d'exécuter avec succès toutes les opérations dans cet exemple ci-dessous, vous aurez besoin de l'édition Enterprise de MapForce. Si vous avez l'édition Professional, commentez les lignes qui insèrent le projet WebService.

```
var objMapForce = null;
var objWshShell = null;
var objFS0 = null;
// !!! adaptez le chemin suivant à vos besoins. !!!
var strSamplePath = "C:\\Users\\<username>\\Documents\\Altova\\MapForce2023\
\MapForceExamples\\";
function Exit(strErrorText)
{
WScript.Echo(strErrorText);
WScript.Quit(-1);
}
function ERROR(strText, objErr)
if (objErr != null)
Exit ("ERROR: (" + (objErr.number & 0xfff) + ")" + objErr.description + " - " +
strText);
else
Exit ("ERROR: " + strText);
}
function CreateGlobalObjects ()
// le Shell et FileSystemObject de l'hôte de scripting windows souvent utile
try
 {
objWshShell = WScript.CreateObject("WScript.Shell");
 objFS0 = WScript.CreateObject("Scripting.FileSystemObject");
}
catch(err)
 { Exit("Can't create WScript.Shell object"); }
```

```
// créer la connexion de MapForce
// s'il y a une instance d'exécution de MapForce (qui n'a jamais eu de connexion) -
utilisez-la,
// autrement, nous créons automatiquement une nouvelle instance
try
 {
objMapForce = WScript.GetObject ("", "MapForce.Application");
}
catch(err)
 { Exit("Can't access or create MapForce.Application"); }
}
}
// -----
// imprimer les items d'arborescence du projet et leurs propriétés de manière récursive.
// -----
function PrintProjectTree( objProjectItemIter, strTab )
while ( ! objProjectItemIter.atEnd() )
 {
// obtenir l'item de projet actuel
objItem = objProjectItemIter.item();
try
 {
// ----- imprimer les propriétés communes
 strGlobalText += strTab + "[" + objItem.Kind + "]" + objItem.Name + "\n";
// ----- imprimer les propriétés de génération de code, si disponible
try
{
if ( objItem.CodeGenSettings_UseDefault )
 strGlobalText += strTab + " Use default code generation settings\n";
else
 strGlobalText += strTab + " code generation language is " +
objItem.CodeGenSettings_Language +
" output folder is " + objItem.CodeGenSettings_OutputFolder + "\n";
 }
catch( err ) {}
// ----- imprimer les paramètres WSDL, si possible
try
 {
strGlobalText += strTab + " WSDL File is " + objItem.WSDLFile +
" Qualified Name is " + objItem.QualifiedName + "\n";
 }
catch( err ) {}
 ł
catch( ex )
 { strGlobalText += strTab + "[" + objItem.Kind + "]\n" }
// ---- recurse
 PrintProjectTree( new Enumerator( objItem ), strTab + ' ' );
```

```
objProjectItemIter.moveNext();
 }
}
// ----
// Charger l'exemple de projet installé avec MapForce.
// ------
function LoadSampleProject()
{
// fermer le projet ouvert
objProject = objMapForce.ActiveProject;
if ( objProject != null )
objProject.Close();
// ouvrir l'exemple de projet et itérer à travers celui-ci.
objProject = objMapForce.OpenProject(strSamplePath + "MapForceExamples.mfp");
// dump properties of all project items
strGlobalText = '';
PrintProjectTree( new Enumerator (objProject), ' ' )
WScript.Echo( strGlobalText );
objProject.Close();
}
// ---
// Créer un nouveau projet avec quelques dossiers, mappages et un
// projet de service Web.
// --
function CreateNewProject()
ł
try
{
// créer un nouveau projet et spécifier le fichier où le mettre.
objProject = objMapForce.NewProject(strSamplePath + "Sample.mfp");
// créer une structure de dossier simple
objProject.CreateFolder( "New Folder 1");
objFolder1 = objProject.Item(1);
 objFolder1.CreateFolder( "New Folder 2");
 objFolder2 = ( new Enumerator( objFolder1 ) ).item(); // an alternative to Item(0)
// ajouter deux différents mappages à la structure de dossier
objFolder1.AddFile( strSamplePath + "DB Altova SQLXML.mfd");
 objMapForce.Documents.OpenDocument(strSamplePath + "InspectionReport.mfd");
 objFolder2.AddActiveFile();
// écraser les paramètres de génération de code pour ce dossier
objFolder2.CodeGenSettings_UseDefault = false;
 objFolder2.CodeGenSettings_OutputFolder = strSamplePath + "SampleOutput"
 objFolder2.CodeGenSettings_Language = 1; //C++
// insérer le projet de service Web basé sur un fichier wsdl depuis les exemples
installés
objProject.InsertWebService( strSamplePath + "TimeService/TimeService.wsdl",
```

```
"{http://www.Nanonull.com/TimeService/}TimeService",
"TimeServiceSoap",
true );
objProject.Save();
if ( ! objProject.Saved )
WScript.Echo("problem occurred when saving project");
// vider l'arborescence de projet
 strGlobalText = '';
PrintProjectTree( new Enumerator (objProject), ' ' )
WScript.Echo( strGlobalText );
}
catch (err)
 { ERROR("while creating new project", err ); }
}
// -----
                                        _____
// Générer le code pour une sous-arborescence de projet. Combiner le code par défaut
// paramètres de génération et paramètres surchargés.
11 --
function GenerateCodeForNewProject()
{
// puisque le projet de service Web contient uniquement des mappages initiaux,
// nous ne générons du code que pour notre dossier client.
// les paramètres de génération de code du projet sont utilisés pour le Dossierl,
// tandis que le Dossier2 fournit les valeurs écrasées.
objFolder = objProject.Item(1);
objFolder1.GenerateCode();
}
CreateGlobalObjects();
objMapForce.Visible = true;
LoadSampleProject();
CreateNewProject();
GenerateCodeForNewProject();
// supprimer pour arrêter l'application quand le script se termine
// objMapForce.Visible = false;
```

# 16.7 Référence d'objet

Cette section fournit une référence aux objets de l'API COM de MapForce. Les objets sont décrits de manière générique puisque l'API peut être utilisée avec tout langage qui prend en charge l'appel d'un objet COM. Pour les exemples spécifiques au langage, voir :

- Exemple Projet C# <sup>1228</sup>
- Exemple Projet Java 1233
- Exemples JScript<sup>1237</sup>

La référence API contient deux principales sections, chacune décrivant les interfaces et les types d'énumération utilisés respectivement dans l'API. Les valeurs d'énumération contiennent à la fois le nom de string et la valeurs numériques. Si votre environnement d'écriture de scripts ne prend pas en charge les énumérations, utilisez les valeurs numériques à la place.

Dans .NET, pour chaque interface de l'interface d'automatisation COM de MapForce, une classe .NET existe dotée du même nom. Les types COM seront également convertis en type .NET approprié. Par exemple, un type tel que Long dans l'API COM apparaîtrait en tant que System.Int32 dans .NET.

Dans Java, veuillez noter les variations de la syntaxe suivantes :

- **Classes et noms de classe**. Pour chaque interface de l'interface d'automatisation de MapForce, une classe Java existe dotée du nom de l'interface.
- Noms de méthodes. Les noms de méthodes sur l'interface Java sont les mêmes que ceux utilisés sur les interfaces COM mais commencent par une minuscule pour être conforme aux conventions d'affectation de noms de Java. Pour avoir accès aux propriétés COM, les méthodes Java qui attribuent les préfixes « get » et « set » au nom de propriété peuvent être utilisées. Si une propriété ne prend pas en charge l'accès en écriture, aucune méthode setter n'est disponible. Par exemple, pour la propriété du Nom de l'interface Document, les méthodes de Java getName et setName sont disponibles.
- Énumérations. Pour chaque énumération définie dans l'interface d'automatisation, une énumération Java est définie avec les mêmes nom et valeurs.
- Événements et gestionnaires d'événement. Pour chaque interface dans l'interface d'automatisation qui prend en charge les événements, une interface Java avec le même nom plus 'Événement' est disponible. Pour simplifier la surcharge d'événements simples, une classe Java est fournie avec des implémentations par défaut pour tous les événements. Le nom de cette classe Java est le nom de l'interface d'événement plus « DefaultHandler ». Par exemple :

Application // classe Java pour accéder à l'application ApplicationEvents // interface d'événements pour l'application ApplicationEventsDefaultHandler // gestionnaire par défaut pour « ApplicationEvents »

# 16.7.1 Interfaces

# 16.7.1.1 Application

L'interface de l'application est l'interface d'un objet d'application de MapForce. Elle représente le point d'accès principal pour l'application MapForce elle-même. Cette interface est le point de départ pour toute opération supplémentaire avec MapForce ou pour extraire ou créer d'autres objets d'automatisation liés à

MapForce. Pour toute information concernant la création d'une instance de l'objet d'application, voir <u>Accéder à</u>

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent
- Options
- Projet
- Documents

Statut de l'application :

- visible
- Nom
- Quitter
- Statut
- WindowHandle

#### Designs de MapForce :

- NewDocument
- OpenDocument
- OpenURL
- ActiveDocument

#### Projets de MapForce :

- NewProject
- OpenProject
- ProjetActif

#### Génération de code de MapForce :

• HighlightSerializedMarker

#### Ressources globales :

- GlobalResourceConfig
- GlobalResourceFile

#### Information de version :

- Édition
- IsAPISupported
- MajorVersion
- MinorVersion

#### Propriétés

Nom	Description
ActiveDocument <sup>1256</sup>	Lecture seule.

Nom	Description
	Retourne l'objet d'automatisation du document actif actuel. Cette propriété retourne de la même manière que Documents.ActiveDocument.
ProjetActif <sup>1256</sup>	Lecture seule. Retourne l'objet d'automatisation du projet actif actuel.
Application <sup>1257</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Documents	Lecture seule. Retourne une collection de tous les documents actuellement ouverts.
Édition	Lecture seule. Renvoie l'édition de l'application, par ex., « Altova MapForce Enterprise Edition » pour l'édition d'enterprise.
<u>GlobalResourceConfig</u> <sup>1258</sup>	Obtient ou définit le nom du fichier de configuration de la ressource globale active. Par défaut, le fichier est appelé <b>GlobalResources.xml</b> .
	Le fichier de configuration peut être renommé et enregistré sous tout emplacement. Vous pouvez donc avoir de multiples fichiers XML de ressources globales. Toutefois, seulement un de ces fichiers XML de ressources globales peut être actif par application, à usage unique, et uniquement les définitions contenues dans ce fichier seront disponibles pour l'application.
GlobalResourceFile <sup>1258</sup>	Obtient ou définit le fichier de définition de la ressource globale. Par défaut, le fichier est appelé GlobalResources.xml.
ISAPISupported <sup>1259</sup>	Lecture seule. Renvoie true si l'API est prises en charge dans cette version de MapForce.
LibraryImports <sup>1259</sup>	Lecture seule. Obtient une collection de bibliothèques importées. Dans l'interface utilisateur graphique de MapForce, celles-ci correspondent aux entrées de la fenêtre <b>Gérer les</b> <b>bibliothèques</b> , ajoutées au niveau de l'application.
MajorVersion <sup>1259</sup>	Lecture seule. Obtient le numéro de version majeur de MapForce. La version est calculée commençant en 1998 et incrémentée de un chaque année. Par exemple, la version majeure est « 18 » pour la release 2016.
MinorVersion <sup>1280</sup>	Lecture seule. Le numéro de version mineur du produit, par exemple 2 pour 2006 R2 SP1.

Nom	Description
<u>Nom</u> (260)	Lecture seule. Le nom de l'application.
Options <sup>(26)</sup>	Lecture seule. Cette propriété donne accès aux options qui configurent le code de génération.
Parent <sup>(126)</sup>	Lecture seule. L'objet parent selon le modèle d'objet.
ServicePackVersion <sup>1261</sup>	Lecture seule. Le numéro de version du Service Pack, par exemple 1 pour 2016 R2 SP1.
Statut <sup>1262</sup>	Lecture seule. Le statut de l'application. Il s'agit d'une des valeurs de l'énumération ENUMApplicationStatus.
Visible <sup>1262</sup>	<b>True</b> si MapForce est affiché à l'écran (cependant, cela pourrait être couvert par d'autres applications ou être iconisé).
	<b>False</b> si MapForce est masqué. La valeur par défaut pour MapForce lorsqu'il est automatiquement démarré suite à une requête depuis le serveur Automation Application est false. Dans tous les autres cas, la propriété est initialisée par <b>true</b> .
	On peut dire qu'une instance d'application qui est visible est contrôlée par l'utilisateur (et éventuellement par des clients connectés par le biais de l'interface automation). Elle ne s'arrêtera que si elle reçoit une requête explicite de l'utilisateur. Pour arrêter une instance d'application, définissez sa visibilité à false et effacez toutes les références de cette instance à l'intérieur de votre programme. L'instance d'application s'arrêtera automatiquement quand aucun autre client COM n'a de références y afférentes.
WindowHandle <sup>1263</sup>	Lecture seule. Extrait Window Handle de l'application.

# Méthodes

Nom	Description
HighlightSerializedMarker <sup>1288</sup>	Utilisez cette méthode pour marquer un emplacement dans un fichier de mappage qui a été précédemment sérialisé. Si le document correspondant n'a pas encore été chargé, il sera chargé en premier. Voir Document.GenerateCodeEx pour une méthode d'extraction d'un marqueur sérialisé.

Nom	Description
NewDocument <sup>(264</sup>	Crée un nouveau document vide. Le nouveau document ouvert devient le ActiveDocument. Cette méthode est une forme raccourcie de Documents.NewDocument.
NewProject <sup>1264</sup>	Crée un nouveau document vide. Le projet actuel est fermé. Le nouveau projet est accessible sous ActiveProject.
NewWebServiceProject <sup>1265</sup>	Crée un nouveau projet Web Service vide. Le nouveau projet est accessible sous ActiveProject. Cette méthode est uniquement disponible dans MapForce Enterprise Edition.
OpenDocument <sup>1266</sup>	Charge un fichier de document précédemment enregistré et poursuit le travail dans le fichier. Le nouveau document ouvert devient le ActiveDocument. Cette méthode est une forme raccourcie de Documents.OpenDocument.
OpenProject <sup>1265</sup>	Ouvre un projet existant de Mapforce (*.mfp). Le projet actuel est fermé. Le projet récemment ouvert est accessible sous ActiveProject.
OpenURL (266	Charge un fichier de document précédemment enregistré depuis un emplacement URL. Permet la fourniture du nom d'utilisateur et du mot de passe.
Quitter 267	Se déconnecte de MapForce pour permettre à l'application de se fermer. Appeler cette méthode est optionnel puisque MapForce assure le suivi de connexions COM externes et reconnaît automatiquement une déconnexion. Pour plus d'information sur un arrêt automatique, voir la propriété visible.

# Événements

Nom	Description
OnDocumentOpened <sup>1267</sup>	Cet événement est déclenché quand un document existant ou un nouveau document est ouvert. L'événement de fermeture correspondant est Document.OnDocumentClosed.
OnProjectOpened <sup>1267</sup>	Cet événement est déclenché quand un document existant ou un nouveau document est chargé dans l'application. L'événement de fermeture correspondant est Document.OnDocumentClosed.
OnShutdown <sup>1268</sup>	Cet événement est déclenché quand l'application est en cours d'arrêt.

# 16.7.1.1.1 Propriétés

## 16.7.1.1.1.1 DocumentActif

Retourne l'objet d'automatisation du document actif actuel. Cette propriété retourne de la même manière que Documents.ActiveDocument.

## Signature

ActiveDocument
----------------

## Erreurs

Code d'erreur :	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de retour a été spécifiée.

## 16.7.1.1.1.2 ProjetActif

Retourne l'objet d'automatisation du projet actif actuel.

## Signature

ActiveProject : Projet

#### Erreurs

Code d'erreur :	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de retour a été spécifiée.

## 16.7.1.1.1.3 Application

Extrait l'objet de l'application de premier niveau.

## Signature

Application : Application

## Erreurs

Code d'erreur :	Description		
1000	L'objet d'application n'est plus valide.		
1001	L'adresse invalide pour le paramètre de retour a été spécifiée.		

## 16.7.1.1.1.4 Documents

Retourne une collection de tous les documents actuellement ouverts.

## Signature

Documents : Documents

### Erreurs

Code d'erreur	rreur Description	
1000	L'objet d'application n'est plus valide.	
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

# 16.7.1.1.1.5 Édition

Renvoie l'édition de l'application, par ex., « Altova MapForce Enterprise Edition » pour l'édition d'enterprise.

## Signature

Edition : String

#### Erreurs

Code d'erreur	Description		
1000	L'objet d'application n'est plus valide.		
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.		

## 16.7.1.1.1.6 GlobalResourceConfig

Obtient ou définit le nom du fichier de configuration de la ressource globale active. Par défaut, le fichier est appelé **GlobalResources.xml**.

Le fichier de configuration peut être renommé et enregistré à tout emplacement. Vous pouvez donc avoir de multiples fichiers XML des Ressources globales. Toutefois, uniquement un seul fichier XML des Ressources globales peut être actif, par application, et une seule fois, et seules les définitions contenues dans ce fichier seront disponibles pour l'application.

## Signature

GlobalResourceConfig : String

#### Erreurs

Error code	Description		
1000	L'objet d'application n'est plus valide.		
1001	Adresse invalide a été spécifiée pour le paramètre de renvoi.		

## 16.7.1.1.1.7 GlobalResourceFile

Obtient ou définit le fichier de définition de la ressource globale. Par défaut, le fichier est appelé GlobalResources.xml.

#### Signature

GlobalResourceFile : String

#### Erreurs

Error code	Description
1000	L'objet d'application n'est plus valide.
1001	Adresse invalide a été spécifiée pour le paramètre de renvoi.

## 16.7.1.1.1.8 IsAPISupported

Renvoie true si l'API est prise en charge dans cette version de MapForce.

### Signature

IsAPISupported : booléen

### Erreurs

Code d'erreur	Description
1001	Adresse invalide a été spécifiée pour le paramètre de renvoi.

## 16.7.1.1.1.9 LibraryImports

Obtient une collection de bibliothèques importées. Dans l'interface utilisateur graphique de MapForce, celles-ci correspondent aux entrées de la fenêtre **Gérer les bibliothèques**, ajoutées au niveau de l'application.

## Signature

LibraryImports : LibraryImports

## Erreurs

Code d'erreur	Description		
1200	L'objet n'est plus valide.		
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.		

## 16.7.1.1.1.10 MajorVersion

Obtient le numéro de la version majeure de MapForce. La version est calculée commençant en 1998 et est incrémentée de un chaque année. Par exemple, la version majeure est « 18 » pour la release 2016.

## Signature

MajorVersion : Long

### Erreurs

Code d'erreur	Description		
1000	L'objet d'application n'est plus valide.		
1001	Adresse invalide a été spécifiée pour le paramètre de renvoi.		

## 16.7.1.1.1.11 MinorVersion

Le numéro de la version mineure du produit, par ex., 2 pour 2006 R2 SP1.

## Signature

MinorVersion : Long

## Erreurs

Code d'erreur	Description	
1000	L'objet d'application n'est plus valide.	
1001	Adresse invalide a été spécifiée pour le paramètre de renvoi.	

## 16.7.1.1.1.12 Nom

Le nom de l'application.

## Signature

Nom :	String						
-------	--------	--	--	--	--	--	--

# Erreurs

Code d'erreur	Description		
1000	L'objet d'application n'est plus valide.		
1001	Adresse invalide a été spécifiée pour le paramètre de renvoi.		
# 16.7.1.1.1.13 Options

Cette propriété donne accès aux options qui configurent le code de génération.

### Signature

Options : Options

### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.1.14 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Application

### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.1.15 ServicePackVersion

Le numéro de version du Service Pack, par exemple 1 pour 2016 R2 SP1.

### Signature

ServicePackVersion : Long

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.1.16 Statut

Le statut de l'application. Il s'agit d'une des valeurs de l'énumération ENUMApplicationStatus.

#### Signature

Statut : ENUMApplicationStatus 1397

#### Erreurs

Code d'erreur	Description
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.1.17 Visible

**True** si MapForce est affiché à l'écran (cependant, cela pourrait être couvert par d'autres applications ou être iconisé).

**False** si MapForce est masqué. La valeur par défaut pour MapForce lorsqu'il est automatiquement démarré suite à une requête depuis le serveur Automation Application est false. Dans tous les autres cas, la propriété est initialisée par **true**.

On peut dire qu'une instance d'application qui est visible est contrôlée par l'utilisateur (et éventuellement par des clients connectés par le biais de l'interface automation). Elle ne s'arrêtera que si elle reçoit une requête explicite de l'utilisateur. Pour arrêter une instance d'application, définissez sa visibilité à false et effacez toutes les références de cette instance à l'intérieur de votre programme. L'instance d'application s'arrêtera automatiquement quand aucun autre client COM n'a de références y afférentes.

### Signature

Visible : Booléen

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.

Code d'erreur	Description
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.1.1.18 WindowHandle

Extrait Window Handle de l'application.

### Signature

WindowHandle : Long

#### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.1.2 Méthodes

# 16.7.1.1.2.1 HighlightSerializedMarker

Utilisez cette méthode pour marquer un emplacement dans un fichier de mappage qui a été précédemment sérialisé. Si le document correspondant n'a pas encore été chargé, il sera chargé en premier. Voir Document.GenerateCodeEx pour une méthode d'extraction d'un marqueur sérialisé.

#### Signature

```
HighlightSerializedMarker(dans i_strSerializedMarker:String) -> Void
```

#### Paramètres

Nom	Туре	Description
i_strSerializedMarker	String	L'objet ErrorMarker pour mettre en évidence. Utiliser ErrorMaker.Serialized pour obtenir cette valeur.

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1007	Le string passé dans <b>i_strSerializedMarker</b> n'est pas reconnu comme marqueur MapForce sérialisé.
1008	Le marqueur pointe vers un emplacement qui n'est plus valide.

# 16.7.1.1.2.2 NewDocument

Crée un nouveau document vide. Le nouveau document ouvert devient le ActiveDocument. Cette méthode est une forme raccourcie de Documents.NewDocument.

### Signature

NewDocument() -> Document

### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.2.3 NewProject

Crée un nouveau document vide. Le projet actuel est fermé. Le nouveau projet est accessible sous ActiveProject.

#### Signature

NewProject() -> Projet

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.2.4 NewWebServiceProject

Crée un nouveau projet Web Service vide. Le nouveau projet est accessible sous ActiveProject. Cette méthode est uniquement disponible dans MapForce Enterprise Edition.

### Signature

NewWebServiceProject() -> Project

### Erreurs

Code d'erreur	Description
1004	Erreur en créant un nouveau projet.
1005	Mauvaise édition de MapForce.

# 16.7.1.1.2.5 OpenDocument

Charge un fichier de document précédemment enregistré et poursuit le travail dans le fichier. Le nouveau document ouvert devient le ActiveDocument. Cette méthode est une forme raccourcie de Documents.OpenDocument.

### Signature

OpenDocument(dans i\_strFileName:String) -> Document

### Paramètres

Nom	Туре	Description
i_strFileName	String	Le chemin du document à ouvrir.

#### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.2.6 OpenProject

Ouvre un projet existant de Mapforce (\*.mfp). Le projet actuel est fermé. Le projet récemment ouvert est accessible sous ActiveProject.

### Signature

```
OpenProject(in i_strFileName:String) -> Project
```

### Paramètres

Nom	Туре	Description
i_strFileName	String	Le chemin du projet à ouvrir.

#### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1002	Le nom de fichier fourni n'est pas valide.

# 16.7.1.1.2.7 OpenURL

Charge un fichier de document précédemment enregistré depuis un emplacement URL. Permet la fourniture du nom d'utilisateur et du mot de passe.

#### Signature

```
OpenURL(in strURL:String, in strUser:String, in strPassword:String) -> Void
```

### Paramètres

Nom	Туре	Description
strURL	String	L'URL depuis lequel le document doit être chargé.
strUser	String	Le nom d'utilisateur requis pour accéder l'URL.
strPassword	String	Le mot de passe requis pour accéder l'URL.

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.

Code d'erreur	Description
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1002	L'URL fourni n'est pas valide.
1006	Erreur en ouvrant le fichier URL.

# 16.7.1.1.2.8 Quitter

Se déconnecte de MapForce pour permettre à l'application de se fermer. Appeler cette méthode est optionnel puisque MapForce assure le suivi de connexions COM externes et reconnaît automatiquement une déconnexion. Pour plus d'information sur un arrêt automatique, voir la propriété Visible.

# Signature

Quit() -> Void

### Erreurs

Code d'erreur	Description
1000	L'objet d'application n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.1.3 Événements

# 16.7.1.1.3.1 OnDocumentOpened

Cet événement est déclenché quand un document existant ou un nouveau document est ouvert. L'événement de fermeture correspondant est Document.OnDocumentClosed.

### Signature

```
OnDocumentOpened(in i_ipDocument:Document) : Annuler
```

# 16.7.1.1.3.2 OnProjectOpened

Cet événement est déclenché quand un document existant ou un nouveau document est chargé dans l'application. L'événement de fermeture correspondant est Document.OnDocumentClosed.

#### Signature

```
OnProjectOpened(in i_ipProject:Project) : Annuler
```

### 16.7.1.1.3.3 OnShutdown

Cet événement est déclenché quand l'application est en cours d'arrêt.

#### Signature

OnShutdown : Annuler

# 16.7.1.2 AppOutputLine

Représente une ligne de message. Contrairement à ErrorMarker, sa structure est plus détaillée et contient une collection de lignes enfant, formant pour cette raison une arborescence de lignes de message.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Accès de ligne :

- GetLineSeverity
- GetLineSymbol
- GetLineText
- GetLineTextEx
- GetLineTextWithChildren
- GetLineTextWithChildrenEx

Une seule AppOutputLine consiste en une ou plusieurs sous-lignes. Accès de sous-ligne :

• GetLineCount

Une sous-ligne consiste en une ou plusieurs cellules. Accès de cellule :

- GetCellCountInLine
- GetCellIcon
- GetCellSymbol
- GetCellText
- GetCellTextDecoration
- GetIsCellText

En dessous d'une AppOutputLine, il peut y avoir zéro, un ou plus de lignes enfant, qui elles-mêmes sont de type AppOutputLine, et forment donc une structure arborescente.

Accès de lignes enfant :

• ChildLines

# Propriétés

Nom	Description
Application <sup>1270</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
ChildLines <sup>1270</sup>	Lecture seule. Renvoie une collection de lignes enfant directes de la ligne actuelle.
Parent <sup>(127)</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

# Méthodes

Nom	Description
GetCellCountInLine <sup>127)</sup>	Obtient le nombre de cellules dans la sous-ligne indiquée par nLine dans la AppOutputLineactuelle.
GetCellIcon <sup>1272</sup>	Obtient l'icône de la cellule indiquée par nCell dans la sous- ligne actuelle AppOutputLine indiquée par nLine
GetCellSymbol <sup>(1272)</sup>	Obtient le symbole de la cellule indiquée par nCell dans la sous-ligne actuelle AppOutputLine indiquée par nLine
GetCellText <sup>1273</sup>	Obtient le texte de la cellule indiquée par nCell dans la sous- ligne actuelle AppOutputLine indiquée par nLine
GetCellTextDecoration <sup>1273</sup>	Obtient la décoration de la cellule de texte indiquée par nCell dans la sous-ligne actuelle AppOutputLineindiquée par nLine. Il peut s'agir d'une des valeurs ENUMAppOutputLine_TextDecoration.
GetIsCellText <sup>1274</sup>	Renvoie true si la cellule indiquée par nCell dans la sous-ligne actuelle AppOutputLine indiquée par nLine est une cellule de texte.
GetLineCount <sup>1275</sup>	Obtient le nombre de sous-lignes dont la ligne actuelle est composée.
GetLineSeverity <sup>1275</sup>	Obtient la gravité de la ligne. Il peut s'agir d'une des valeurs ENUMAppOutputLine_Severity.
GetLineSymbol 1275	Obtient le symbole assigné à toute la ligne.
GetLineText 1276	Obtient les contenus de la ligne en tant que texte.

Nom	Description
GetLineTextEx <sup>1276</sup>	Obtient les contenus de la ligne en tant que texte utilisant la partie spécifiée et les séparateurs de lignes.
GetLineTextWithChildren <sup>1277</sup>	Obtient les contenus de la ligne, y compris toutes les lignes descendantes et enfant en tant que texte.
GetLineTextWithChildrenEx	Obtient les contenus de la ligne, y compris toutes les lignes descendantes et enfant en tant que texte utilisant la partie, la ligne, l'onglet et les séparateurs d'item spécifiés.

# 16.7.1.2.1 Propriétés

## 16.7.1.2.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

#### Erreurs

Code d'erreur :	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.2.1.2 ChildLines

Renvoie une collection de lignes enfant directes de la ligne actuelle.

#### Signature

ChildLines : AppOutputLines

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.2.1.3 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : AppOutputLines

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2 Méthodes

# 16.7.1.2.2.1 GetCellCountInLine

Obtient le nombre de cellules dans la sous-ligne indiquée par nLine dans la AppOutputLineactuelle.

### Signature

```
GetCellCountInLine(in nLine:Long) -> Long
```

# Paramètres

Nom	Туре	Description
nLine	Long	Spécifie l'index de la ligne basé sur zéro.

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.2.2.2 GetCellIcon (obsolete)

Obtient l'icône de la cellule indiquée par nCell dans la sous-ligne actuelle AppOutputLine indiquée par nLine

### Signature

```
GetCellIcon(in nLine:Long, in nCell:Long) -> Long
```

### Paramètres

Nom	Туре	Description
nLine	Long	
nCell	Long	

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.3 GetCellSymbol

Obtient le symbole de la cellule indiquée par nCell dans la sous-ligne actuelle AppOutputLine indiquée par nLine

### Signature

```
GetCellSymbol(in nLine:Long, in nCell:Long) -> AppOutputLineSymbol
```

#### Paramètres

Nom	Туре	Description
nLine	Long	Spécifie l'index de la ligne basé sur zéro.
nCell	Long	Spécifie l'index de la cellule basé sur zéro.

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.4 GetCellText

Obtient le texte de la cellule indiquée par nCell dans la sous-ligne actuelle AppOutputLine indiquée par nLine

### Signature

```
GetCellText(in nLine:Long, in nCell:Long) -> String
```

### Paramètres

Nom	Туре	Description
nLine	Long	Spécifie l'index de la ligne basé sur zéro.
nCell	Long	Spécifie l'index de la cellule basé sur zéro.

#### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.5 GetCellTextDecoration

Obtient la décoration de la cellule de texte indiquée par nCell dans la sous-ligne actuelle AppOutputLineindiquée par nLine. Il peut s'agir d'une des valeurs ENUMAppOutputLine\_TextDecoration.

### Signature

GetCellTextDecoration(in nLine:Long, in nCell:Long) -> Long

# Paramètres

Nom	Туре	Description
nLine	Long	Spécifie l'index de la ligne basé sur zéro.
nCell	Long	Spécifie l'index de la cellule basé sur zéro.

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.6 GetIsCellText

Renvoie true si la cellule indiquée par nCell dans la sous-ligne actuelle AppOutputLine indiquée par nLine est une cellule de texte.

### Signature

```
GetIsCellText(in nLine:Long, in nCell:Long) -> Boolean
```

# Paramètres

Nom	Туре	Description
nLine	Long	Spécifie l'index de la ligne basé sur zéro.
nCell	Long	Spécifie l'index de la cellule basé sur zéro.

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.2.2.7 GetLineCount

Obtient le nombre de sous-lignes dont la ligne actuelle est composée.

### Signature

```
GetLineCount() -> Long
```

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.8 GetLineSeverity

Obtient la gravité de la ligne. Il peut s'agir d'une des valeurs ENUMAppOutputLine\_Severity.

### Signature

GetLineSeverity() -> Long

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.9 GetLineSymbol

Obtient le symbole assigné à toute la ligne.

### Signature

GetLineSymbol() -> AppOutputLineSymbol

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.10 GetLineText

Obtient les contenus de la ligne en tant que texte.

### Signature

GetLineText() -> String

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.11 GetLineTextEx

Obtient les contenus de la ligne en tant que texte utilisant la partie spécifiée et les séparateurs de lignes.

### Signature

```
GetLineTextEx(in psTextPartSeperator:String, in psLineSeperator:String) -> String
```

### Paramètres

Nom	Туре	Description
psTextPartSeperator	String	
psLineSeperator	String	

Code d'erreur	Description
4100	L'objet n'est plus valide.

Code d'erreur	Description
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.12 GetLineTextWithChildren

Obtient les contenus de la ligne, y compris toutes les lignes descendantes et enfant en tant que texte.

### Signature

```
GetLineTextWithChildren() -> String
```

### Erreurs

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.2.2.13 GetLineTextWithChildrenEx

Obtient les contenus de la ligne, y compris toutes les lignes descendantes et enfant en tant que texte utilisant la partie, la ligne, l'onglet et les séparateurs d'item spécifiés.

### Signature

```
GetLineTextWithChildrenEx(in psPartSep:String, in psLineSep:String, in psTabSep:String,
in psItemSep:String) -> String
```

### Paramètres

Nom	Туре	Description
psPartSep	String	
psLineSep	String	
psTabSep	String	
psitemSep	String	

Code d'erreur	Description
4100	L'objet n'est plus valide.
4101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.3 AppOutputLines

Représente une collection de lignes de message AppOutputLine.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Itération à travers la collection :

- Compter
- Item

### Propriétés

Nom	Description
Application <sup>1279</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Compter <sup>(1279</sup>	Lecture seule. Extrait le nombre de lignes dans la collection.
<u>Item</u> <sup>1280</sup>	Lecture seule. Extrait la ligne à l'index n de la collection. Les index commencent par 1.
Parent <sup>(280</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

# 16.7.1.3.1 Propriétés

# 16.7.1.3.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

#### Erreurs

Code d'erreur :	Description
4000	L'objet n'est plus valide.
4001	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.3.1.2 Count

Extrait le nombre de lignes dans la collection.

# Signature

Count : Entier

Code d'erreur	Description
4000	L'objet n'est plus valide.
4001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.3.1.3 Item

Extrait la ligne à l'index n de la collection. Les index commencent par 1.

### Signature

Item(in n:Integer) : AppOutputLine

### Erreurs

Code d'erreur	Description
4000	L'objet n'est plus valide.
4001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.3.1.4 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : AppOutputLine

#### Erreurs

Code d'erreur	Description
4000	L'objet n'est plus valide.
4001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.4 AppOutputLineSymbol

Un AppOutputLineSymbol représente un lien dans une ligne de message AppOutputLine sur laquelle vous pouvez cliquer dans la fenêtre des messages de MapForce. Il est appliqué à une cellule d'une AppOutputLine ou à toute la ligne elle-même.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Accès aux méthodes AppOutputLineSymbol :

• GetSymbolHREF

- GetSymbolID
- IsSymbolHREF

# Propriétés

Nom	Description
Application <sup>1281</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Parent <sup>1282</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

### Méthodes

Nom	Description
GetSymbolHREF <sup>1282</sup>	Si le symbole est de type URL, il renvoie l'URL en tant que string.
GetSymbolID <sup>1283</sup>	Obtient l'ID du symbole.
IsSymbolHREF <sup>1283</sup>	Renvoie true si le symbole est de type URL.

# 16.7.1.4.1 Propriétés

# 16.7.1.4.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

Code d'erreur :	Description
4200	L'objet n'est plus valide.
4201	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.4.1.2 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Application

### Erreurs

Code d'erreur	Description
4200	L'objet n'est plus valide.
4201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.4.2 Méthodes

# 16.7.1.4.2.1 GetSymbolHREF

Si le symbole est de type URL, il renvoie l'URL en tant que string.

### Signature

GetSymbolHREF() -> String

Code d'erreur	Description
4200	L'objet n'est plus valide.
4201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.4.2.2 GetSymbolID

Obtient l'ID du symbole.

### Signature

GetSymbolID() -> Long

### Erreurs

Code d'erreur	Description
4200	L'objet n'est plus valide.
4201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.4.2.3 IsSymbolHREF

Renvoie true si le symbole est de type URL.

### Signature

IsSymbolHREF() -> Boolean

### Erreurs

Code d'erreur	Description
4200	L'objet n'est plus valide.
4201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5 Composant

Un composant représente un composant de MapForce.

Pour accéder au contrôle, utilisez les propriétés Application et Parent .

#### Propriétés de composant :

- HasIncomingConnections
- HasOutgoingConnections
- CanChangeInputInstanceFile
- CanChangeOutputInstanceFile
- ComponentName.
- ID

- IsParameterInputRequired
- IsParameterSequence
- Nom
- Aperçu
- Schéma
- Sous-type
- Type

#### Propriétés relatives à l'instance :

- InputInstanceFile
- OutputInstanceFile

#### Datapoints :

• GetRootDatapoint

#### Exécution :

• GenerateOutput

# Propriétés

Nom	Description
Application <sup>1286</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
<u>CanChangeInputInstanceFile</u> <sup>(287)</sup>	Lecture seule. Indique si le nom de fichier de l'instance de sortie peut être modifié.
	Renvoie false si le composant contient un nœud filename et ce nœud a une connexion de son côté gauche (entrée), autrement il renvoie true. Si le composant ne contient pas de nœud filename, false est renvoyé.
CanChangeOutputInstanceFile <sup>(287</sup>	Lecture seule. Indique si le nom de fichier de l'instance de sortie peut être modifié.
	Renvoie false si le composant contient un nœud filename et ce nœud a une connexion de son côté gauche (entrée), autrement il renvoie true. Si le composant ne contient pas de nœud filename, false est renvoyé.
ComponentName <sup>1288</sup>	Obtient ou définit le nom de composant.
HasIncomingConnections <sup>1288</sup>	Lecture seule. Indique si le composant contient des connexions entrantes (de son côté gauche) ne comprenant pas le nœud filename. Une

Nom	Description
	connexion entrante sur le nœud de connexion n'a pas d'effet sur la valeur renvoyée.
HasOutgoingConnections <sup>1289</sup>	Lecture seule. Indique si le composant contient des connexions sortantes (de son côté droit).
1289 1D	Lecture seule. Extrait l'ID de composant.
InputInstanceFile <sup>1289</sup>	Obtient ou définit le fichier de l'instance d'entrée du composant.
IsParameterInputRequired <sup>1230</sup>	Obtient ou définit si le composant de paramètre d'entrée requiert une connexion ingoing sur le composant d'appel de la fonction définie par l'utilisateur dans lequel se trouve le composant de paramètre d'entrée. Cette propriété ne fonctionne que pour les composants qui sont des composants de paramètres d'entrée.
IsParameterSequence <sup>1290</sup>	Obtient ou définit si le composant de paramètre d'entrée ou de sortie prend en charge les séquences. Cette propriété fonctionne uniquement pour les composants qui sont des composants de paramètre d'entrée ou de sortie.
<u>Nom</u> (291	Lecture seule. Obtient le nom de composant.
OutputInstanceFile <sup>1291</sup>	Obtient ou définit le fichier d'instance de sortie du composant.
	Tenter d'accéder à OutputInstanceFile d'un composant via l'API ne renvoie aucune donnée si le « fichier » connecteur du composant a été connecté à un autre item dans le mappage.
Parent <sup>(1292</sup>	Lecture seule. L'objet parent selon le modèle d'objet.
Aperçu <sup>(292</sup>	Obtient ou définit si le composant est le composant preview actuel.
	Cette propriété ne fonctionne que pour les composants qui sont des composants cibles dans le mappage principal du document. Uniquement un composant cible dans le mappage principal peut être le composant preview à tout moment.
	Lorsque vous définissez cette propriété, il est uniquement possible de la définir en tant que true. Ceci définira aussi implicitement la propriété Preview de tous les autres composants comme false.
	Il n'existe qu'un seul composant cible unique dans le mappage principal, il s'agit également du composant preview.
Schéma <sup>(23)</sup>	Lecture seule.

Nom	Description
	Extrait le nom de fichier du schéma du composant.
Sous-type <sup>1293</sup>	Lecture seule. Extrait le sous-type du composant.
<u>Type</u> 1293	Lecture seule. Extrait le type de composant.
UsageKind <sup>1294</sup>	Lecture seule. Extrait le type d'utilisation du composant.

### Méthodes

Nom	Description
GenerateOutput <sup>1293</sup>	Génère le(s) fichier(s) de sortie défini(s) dans le mappage pour le composant actuel uniquement, en utilisant un langage de mappage interne de MapForce. Le(s) nom(s) du/des fichier(s) de sortie est/sont défini(s) comme propriété du composant actuel qui est l'item de sortie dans le mappage pour le processus de génération.
GetRootDatapoint <sup>1235</sup>	Obtient un datapoint racine du côté gauche (entrée) ou droit (sortie) du composant. Pour accéder aux enfants et descendants, l'objet Datapoint fournit des méthodes supplémentaires.

# 16.7.1.5.1 Propriétés

# 16.7.1.5.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

```
Application : Application
```

Code d'erreur :	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.5.1.2 CanChangeInputInstanceFile

Indique si le nom de fichier de l'instance de sortie peut être modifié.

Renvoie false si le composant contient un nœud filename et ce nœud a une connexion de son côté gauche (entrée), autrement il renvoie true. Si le composant ne contient pas de nœud filename, false est renvoyé.

### Signature

CanChangeInputInstanceFile : Booléen

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.5.1.3 CanChangeOutputInstanceFile

Indique si le nom de fichier de l'instance de sortie peut être modifié.

Renvoie false si le composant contient un nœud filename et ce nœud a une connexion de son côté gauche (entrée), autrement il renvoie true.

Si le composant ne contient pas de nœud filename, false est renvoyé.

# Signature

CanChangeOutputInstanceFile : Booléen

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.5.1.4 ComponentName

Obtient ou définit le nom de composant.

### Signature

ComponentName : String

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1246	Le composant ne prend pas en charge la définition de son nom.
1247	Nom de composant invalide.

# 16.7.1.5.1.5 HasIncomingConnections

Indique si le composant contient des connexions entrantes (de son côté gauche) ne comprenant pas le nœud filename. Une connexion entrante sur le nœud de connexion n'a pas d'effet sur la valeur renvoyée.

# Signature

HasIncomingConnections : Booléenne

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5.1.6 HasOutgoingConnections

Indique si le composant contient des connexions sortantes (de son côté droit).

### Signature

HasOutgoingConnections : Booléenne

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5.1.7 ID

Extrait l'ID de composant.

### Signature

ID : Long

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5.1.8 InputInstanceFile

Obtient ou définit le fichier de l'instance d'entrée du composant.

### Signature

InputInstanceFile : String

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5.1.9 IsParameterInputRequired

Obtient ou définit si le composant de paramètre d'entrée requiert une connexion ingoing sur le composant d'appel de la fonction définie par l'utilisateur dans lequel se trouve le composant de paramètre d'entrée. Cette propriété ne fonctionne que pour les composants qui sont des composants de paramètres d'entrée.

### Signature

IsParameterInputRequired : Booléenne

#### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1232	Cette opération ne fonctionne que pour un composant de paramètre d'entrée.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.

### 16.7.1.5.1.10 IsParameterSequence

Obtient ou définit si le composant de paramètre d'entrée ou de sortie prend en charge les séquences. Cette propriété fonctionne uniquement pour les composants qui sont des composants de paramètre d'entrée ou de sortie.

### Signature

IsParameterSequence : Booléenne

Code d'erreur	Description
1200	L'objet n'est plus valide.

Code d'erreur	Description
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1233	Cette opération ne fonctionne que pour un composant de paramètre d'entrée ou de sortie.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.

# 16.7.1.5.1.11 Nom

Obtient le nom de composant.

### Signature

Nom : String

### Erreurs

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

# 16.7.1.5.1.12 OutputInstanceFile

Obtient ou définit le fichier d'instance de sortie du composant.

Tenter d'accéder à OutputInstanceFile d'un composant via l'API ne renvoie aucune donnée si le « fichier » connecteur du composant a été connecté à un autre item dans le mappage.

### Signature

OutputInstanceFile : String

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.5.1.13 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Mappage

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.5.1.14 Preview

Obtient ou définit si le composant est le composant preview actuel.

Cette propriété ne fonctionne que pour les composants qui sont des composants cibles dans le mappage principal du document. Uniquement un composant cible dans le mappage principal peut être le composant preview à tout moment.

Lorsque vous définissez cette propriété, il est uniquement possible de la définir en tant que true. Ceci définira aussi implicitement la propriété Preview de tous les autres composants comme false.

Il n'existe qu'un seul composant cible unique dans le mappage principal, il s'agit également du composant preview.

#### Signature

Preview : Booléenne

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1234	Seul un composant cible dans le mappage principal peut être défini comme composant preview .
1235	Un composant ne peut pas être défini comme composant non-preview. Définir un autre composant comme composant preview à la place.

# 16.7.1.5.1.15 Schéma

Extrait le nom de fichier du schéma du composant.

### Signature

Schéma : String

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5.1.16 SubType

Extrait le sous-type du composant.

### Signature

SubType : ENUMComponentSubType

### Erreurs

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

# 16.7.1.5.1.17 Type

Extrait le type de composant.

### Signature

Type : ENUMComponentType<sup>1400</sup>

### 16.7.1.5.1.18 UsageKind

Extrait le type d'utilisation du composant.

### Signature

UsageKind : ENUMComponentUsageKind [401]

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.5.2 Méthodes

### 16.7.1.5.2.1 GenerateOutput

Génère le(s) fichier(s) de sortie défini(s) dans le mappage pour le composant actuel uniquement, en utilisant un langage de mappage interne de MapForce. Le(s) nom(s) du/des fichier(s) de sortie est/sont défini(s) comme propriété du composant actuel qui est l'item de sortie dans le mappage pour le processus de génération.

### Signature

```
GenerateOutput(out pbError:Boolean) -> AppOutputLines
```

### Paramètres

Nom	Туре	Description
pbError	Booléenne	Ceci est un paramètre output-only. Vous recevrez une valeur uniquement si le langage d'appel prend en charge les paramètres de sortie. Si ce n'est pas le cas, la valeur que vous transférez ici restera inchangée quand la fonction est terminée.

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1248	Générer la sortie est uniquement pris en charge quand l'interface utilisateur graphique est visible.

# 16.7.1.5.2.2 GetRootDatapoint

Obtient un datapoint racine du côté gauche (entrée) ou droit (sortie) du composant. Pour accéder aux enfants et descendants, l'objet Datapoint fournit des méthodes supplémentaires.

### Signature

GetRootDatapoint(in side: ENUMComponentDatapointSide 400, in strNamespace: String, in strLocalName: String, in strParameterName: String) -> Datapoint

### Paramètres

Nom	Туре	Description
side	ENUMComponentDatapointSide <sup>440</sup>	Le paramètre side indique si un datapoint d'entrée ou de sortie d'un composant doit être extrait.
strName space	String	Le nom d'espace spécifié et nom local indiquent le nom spécifique du nœud dont le datapoint doit être extrait. Pour les composants avec une information structurelle telle que les composants de schéma, vous allez devoir fournir un espace de nom ensemble avec le nom local, ou vous pouvez uniquement transférer un string vide pour l'espace de nom.
		Les composants basés sur fichier comme le composant de schéma contiennent un nœud spécial dans leur racine, le nœud de filename. Là, GetRootDatapoint peut uniquement trouver le nœud de filename. Vous allez devoir transférer l'espace de nom

Nom	Туре	Description
		"http://www.altova.com/mapfo rce" et le nom local "FileInstance" pour extraire le datapoint de ce nœud.
strLocalName	String	Voir ci-dessus.
strParameterName	String	Le nom du paramètre spécifié devrait être un string vide, sauf si le composant en question est un composant d'appel de fonction. Puisqu'une fonction définie par l'utilisateur peut contenir des paramètres d'entrée ou de sortie de la même structure, le composant d'appel de fonction appelant cette fonction définie par l'utilisateur peut avoir plus d'un nœud racine avec un espace de nom et nom local identiques. Ils seront alors différents uniquement par leurs noms de paramètre, qui sont en fait des noms de composants de paramètre correspondants dans le mappage de fonction définie par l'utilisateur lui-même. Par contre, il n'est pas obligatoire de spécifier le nom de paramètre. Dans ce cas, la méthode retournera le premier datapoint racine correspondant à l'espace

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1248	Datapoint pas trouvé.
# 16.7.1.6 Composants

Représente une collection d'objets de composant objects.

Pour accéder au modèle d'objet, utilisez les propriétés suivantes :

- Application
- Parent

Pour itérer à travers la collection :

- Compter
- Item

### Propriétés

Nom	Description
Application <sup>1297</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Compter <sup>1293</sup>	Lecture seule. Extrait le nombre de composants dans la collection.
Item <sup>1298</sup>	Lecture seule. Extrait le composant à l'index n de la collection. Les index commencent par 1.
Parent <sup>(1299)</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

### 16.7.1.6.1 Propriétés

### 16.7.1.6.1.1 Application

Extrait l'objet de l'application de premier niveau.

#### Signature

Application : Application

Code d'erreur :	Description
1200	L'objet n'est plus valide.

Code d'erreur :	Description
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

### 16.7.1.6.1.2 Count

Extrait le nombre de composants dans la collection.

### Signature

Count : Entier

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.6.1.3 Item

Extrait le composant à l'index n de la collection. Les index commencent par 1.

### Signature

Item(in n:Integer) : Composant

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.6.1.4 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Mappage

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.7 Connexion

Un objet de connexion représente un connecteur entre deux composants.

Pour accéder au modèle d'objet, utilisez les propriétés suivantes :

- Application
- Parent

Pour obetnir ou définir le type de connexion, utilisez ConnectionType.

### Propriétés

Nom	Description
Application <sup>1300</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
ConnectionType <sup>1300</sup>	Obtient ou définit le type de connexion.
Parent <sup>1301</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

## 16.7.1.7.1 Propriétés

### 16.7.1.7.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

#### Erreurs

Code d'erreur :	Description
2100	L'objet n'est plus valide.
2101	L'adresse invalide pour le paramètre de retour a été spécifiée.

## 16.7.1.7.1.2 ConnectionType

Obtient ou définit le type de connexion.

### Signature

ConnectionType : ENUMConnectionType [40]

Code d'erreur	Description
2100	L'objet n'est plus valide.
2101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
2102	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
2103	Échec de la modification du type de connexion.

### 16.7.1.7.1.3 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Mappage

### Erreurs

Code d'erreur	Description
2100	L'objet n'est plus valide.
2101	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.8 Datapoint

Un objet Datapoint représente une icône d'entrée ou de sortie d'un composant.

## Propriétés

Nom	Description
Application <sup>1302</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Parent <sup>1302</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

### Méthodes

Nom	Description
GetChild <sup>1302</sup>	Recherche un datapoint enfant direct du datapoint actuel, par le biais de l'espace de nom et le nom local.
	Si un composant de schéma a des éléments qui contiennent du contenu mixte, chacun affiche un nœud enfant additionnel, le soi-disant nœud <b>text()</b> . Pour extraire un datapoint d'un nœud <b>text()</b> , vous devrez transférer un string vide dans strNamespace de même que "#text" dans strLocalName et eSearchDatapointElement dans searchFlags.

### 16.7.1.8.1 Propriétés

#### 16.7.1.8.1.1 Application

Extrait l'objet de l'application de premier niveau.

#### Signature

Application : Application

#### Erreurs

Code d'erreur :	Description	
2000	L'objet n'est plus valide.	
2001	L'adresse invalide pour le paramètre de retour a été spécifiée.	

### 16.7.1.8.1.2 Parent

L'objet parent selon le modèle d'objet.

#### Signature

Parent : Composant

#### Erreurs

Code d'erreur	Description	
2000	L'objet n'est plus valide.	
2001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

### 16.7.1.8.2 Méthodes

### 16.7.1.8.2.1 GetChild

Recherche un datapoint enfant direct du datapoint actuel, par le biais de l'espace de nom et le nom local.

Si un composant de schéma a des éléments qui contiennent du contenu mixte, chacun affiche un nœud enfant additionnel, le soi-disant nœud **text()**. Pour extraire un datapoint d'un nœud **text()**, vous devrez transférer un

string vide dans strNamespace de même que "#text" dans strLocalName et eSearchDatapointElement dans searchFlags.

### Signature

```
GetChild(in strNamespace:String, in strLocalName:String, in searchFlags:ENUMSearchDatapointFlags<sup>[403]</sup>) -> Datapoint
```

### Paramètres

Nom	Туре	Description
strNamespace	String	L'espace de nom du datapoint enfant direct.
strLocalName	String	Le nom du datapoint enfant direct.
searchFlags	ENUMSearchDatapointFlags	Les indicateurs recherche peuvent être transférés comme une combinaison de valeurs (combiné utilisant binaire OR) de l'énumération ENUMSearchDatapointFlags.

### Erreurs

Code d'erreur	Description
2000	L'objet n'est plus valide.
2001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
2002	Datapoint pas trouvé.

# 16.7.1.9 Document

Un objet de Document représente un document MapForce (un fichier MFD chargé). Un document contient un mappage principal et zéro ou plus de mappages définis par l'utilisateur local.

Pour accéder au modèle d'objet, utilisez les propriétés suivantes :

- Application
- Parent

Pour la gestion du fichier, utilisez :

- Activer
- Fermer
- FullName

- Nom
- Chemin d'accès
- Enregistré
- Enregistrer
- SaveAs

#### Pour la gestion du mappage, utilisez :

- MainMapping
- Mappages
- CreateUserDefinedFunction

#### Pour la gestion du composant, utilisez :

• FindComponentByID

#### Pour la génération de code, utilisez :

- OutputSettings\_ApplicationName
- JavaSettings\_BasePackageName
- GenerateCHashCode
- GenerateCodeEx
- GenerateCppCode
- GenerateJavaCode
- GenerateXQuery
- GenerateXSLT
- GenerateXSLT2
- GenerateXSLT3
- HighlightSerializedMarker

#### Pour l'exécution de mappage, utilisez :

- GenerateOutput
- GenerateOutputEx

#### Accès d'affichage :

• MapForceView

#### Obsolète :

• OutputSettings\_Encoding

#### Propriétés

Nom	Description
Application <sup>1308</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
FullName <sup>1308</sup>	Chemin et nom du fichier du document.
JavaSettings_BasePackageName	Définit ou extrait le nom du pack de base utilisé lors de la génération du code Java. Dans l'interface utilisateur graphique

Nom	Description
	de MapForce, ce paramètre est disponible dans la boîte de dialogue des <b>Paramètres de nmappage</b> (clic droit sur le mappage et sélectionnez les <b>Mapping Settings</b> depuis le menu contextuel).
LibraryImports <sup>1309</sup>	Lecture seule. Obtient une collection de bibliothèques importées. Dans l'interface utilisateur graphique de MapForce, celles-ci correspondent aux entrées de la fenêtre <b>Gérer les</b> <b>bibliothèques</b> , ajoutées au niveau du document.
MainMapping <sup>1309</sup>	Lecture seule. Extrait le mappage principal du document.
MapForceView <sup>(131)</sup>	Lecture seule. Cette propriété donne accès à la fonctionnalité spécifique à l'affichage de MapForce.
Mappages <sup>(131)</sup>	Lecture seule. Retourne une collection de mappages contenue dans le document.
<u>Nom</u> <sup>(310)</sup>	Lecture seule. Nom du fichier document sans chemin de fichier.
OutputSettings_ApplicationName	Définit ou extrait le nom de l'application disponible dans la boîte de dialogue <b>Paramètres de mappage</b> (pour afficher cette boîte de dialogue dans MapForce, clic droit sur le mappage et sélectionnez <b>Paramètres de mappage</b> depuis le menu contextuel).
OutputSettings_Encoding <sup>1311</sup>	Cette propriété n'est plus prise en charge. Les paramètres d'encodage de sortie n'existent plus. Les composants ont des paramètres d'encodage de sortie individuels.
Parent <sup>(312)</sup>	Lecture seule. L'objet parent selon le modèle d'objet.
Chemin d'accès <sup>1312</sup>	Lecture seule. Le chemin du fichier de document sans nom.
Enregistré <sup>1312</sup>	Lecture seule. <b>True</b> si le document n'a pas été modifié depuis la dernière opération de sauvegarde, <b>false</b> , si le contraire.

## Méthodes

Nom	Description
Activate <sup>1313</sup>	Rend ce document actif.

Nom	Description
Fermer <sup>(313)</sup>	Ferme le document sans l'enregistrer.
CreateUserDefinedFunction (314	Crée une fonction définie par l'utilisateur dans le document actuel.
FindComponentByID <sup>1315</sup>	Cherche dans le document entier, aussi dans tous ses mappages, pour le composant avec l'ID spécifiée.
GenerateCHashCode	Génère le code C# qui exécutera le mappage. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
GenerateCodeEx <sup>1316</sup>	Génère le code qui exécutera le mappage. Le paramètre <b>i_nLanguage</b> spécifie le langage cible. La méthode renvoie un objet qui peut être utilisé pour énumérer tous les messages créés par le générateur de code. Il s'agit des mêmes messages qui sont affichés dans la fenêtre de messages de MapForce.
GenerateCppCode <sup>1316</sup>	Génère le code C++ qui exécutera le mappage. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
GenerateJavaCode <sup>1317</sup>	Génère le code Java qui exécutera le mappage. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
GenerateOutput <sup>1317</sup>	Génère tous les fichiers de sortie définis dans le mappage utilisant un langage de mappage interne de MapForce. Les noms des fichiers de sortie sont définis comme propriétés des éléments de sortie dans le mappage.
	<b>Note :</b> Cette méthode peut uniquement être utilisée quand la fenêtre principale de MapForce (exécutée comme serveur COM) est visible, ou si elle est incorporée avec une interface utilisateur graphique. Si la méthode est appelée tandis que MapForce n'est pas visible, alors l'erreur apparaîtra.
GenerateOutputEx <sup>1318</sup>	Génère tous les fichiers de sortie définis dans le mappage utilisant un langage de mappage interne de MapForce. Les noms des fichiers de sortie sont définis comme propriétés des éléments de sortie dans le mappage. Cette méthode est identique à GenerateOutput, à l'exception de sa valeur de renvoi contenant les messages résultants, avertissements et erreurs organisés en tant qu'arborescences de AppOutputLines.
	<b>Note :</b> Cette méthode peut uniquement être utilisée quand la fenêtre principale de MapForce (exécutée comme serveur COM) est visible, ou si elle est incorporée avec une interface utilisateur graphique. Si la méthode est appelée tandis que MapForce n'est pas visible, alors l'erreur apparaîtra.

Nom	Description
GenerateXQuery <sup>1318</sup>	Génère du code de mappage comme XQuery. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
GenerateXSLT 1319	Génère du code de mappage en tant que XSLT. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
GenerateXSLT2 <sup>1319</sup>	Génère le code de mappage comme XSLT2. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
GenerateXSLT3 <sup>1319</sup>	Génère un code de mappage XSLT 3.0. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.
HighlightSerializedMarker	Utilisez cette méthode pour marquer un emplacement dans un fichier de mappage qui a été précédemment sérialisé. Si le document correspondant n'est pas déjà chargé, il sera chargé d'abord. Voir GenerateCodeEx pour une méthode d'extraction d'un marqueur sérialisé.
Enregistrer <sup>1321</sup>	Enregistre le document vers le fichier défini par Document.FullName.
SaveAs <sup>(1321)</sup>	Enregistre le document vers le nom de fichier spécifique, et définit Document.FullName vers cette valeur si l'opération Enregistrer a réussi.

# Événements

Nom	Description
OnDocumentClosed <sup>1322</sup>	Cet événement est déclenché lorsqu'un document est fermé. L'objet de document passé au gestionnaire d'événements ne devrait pas être accédé. L'événement ouvert correspondant est Application.OnDocumentOpened.
OnModifiedFlagChanged <sup>[322</sup>	Cet événement est déclenché lorsque le statut de modification d'un document change.

### 16.7.1.9.1 Propriétés

### 16.7.1.9.1.1 Application

Extrait l'objet de l'application de premier niveau.

#### Signature

Application : Application

#### Erreurs

Code d'erreur :	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

### 16.7.1.9.1.2 FullName

Chemin et nom du fichier du document.

#### Signature

FullName : String

### Erreurs

Error code	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide a été spécifiée pour le paramètre de renvoi.	

### 16.7.1.9.1.3 JavaSettings\_BasePackageName

Définit ou extrait le nom du pack de base utilisé lors de la génération du code Java. Dans l'interface utilisateur graphique de MapForce, ce paramètre est disponible dans la boîte de dialogue des **Paramètres de nmappage** (clic droit sur le mappage et sélectionnez les **Mapping Settings** depuis le menu contextuel).

#### Signature

JavaSettings\_BasePackageName : String

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.9.1.4 LibraryImports

Obtient une collection de bibliothèques importées. Dans l'interface utilisateur graphique de MapForce, celles-ci correspondent aux entrées depuis la fenêtre **Gérer les biblitohèques**, ajoutées au niveau du document.

### Signature

LibraryImports : LibraryImports

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.9.1.5 MainMapping

Extrait le mappage principal du document.

### Signature

```
MainMapping : Mappage
```

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.9.1.6 MapForceView

Cette propriété donne accès à la fonctionnalité spécifique à l'affichage de MapForce.

#### Signature

MapForceView : MapForceView

#### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.9.1.7 Mappages

Retourne une collection de mappages contenue dans le document.

#### Signature

Mappages : Mappages

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.9.1.8 Nom

Nom du fichier du document sans chemin de fichier.

#### Signature

Nom : String

#### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.9.1.9 OutputSettings\_ApplicationName

Définit ou extrait le nom de l'application disponible dans la boîte de dialogue **Paramètres de mappage** (pour afficher cette boîte de dialogue dans MapForce, clic droit sur le mappage et sélectionnez **Paramètres de mappage** depuis le menu contextuel).

### Signature

OutputSettings\_ApplicationName : String

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.9.1.10 OutputSettings\_Encoding (obsolete)

Cette propriété n'est plus prise en charge. Les paramètres d'encodage de sortie n'existent plus. Les composants ont des paramètres d'encodage de sortie individuels.

### Signature

OutputSettings\_Encoding : String

### 16.7.1.9.1.11 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Documents

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.9.1.12 Chemin

Le chemin du fichier de document sans nom.

#### Signature

Chemin : String

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.9.1.13 Enregistré

True si le document n'a pas été modifié depuis la dernière opération de sauvegarde, false, si le contraire.

#### Signature

Enregistré : Booléenne

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.9.2 Méthodes

## 16.7.1.9.2.1 Activate

Rend ce document actif.

### Signature

Activate() -> Void

### Erreurs

Code d'erreur :	Description
1200	L'objet n'est plus valide.

## 16.7.1.9.2.2 Fermer

Ferme le document sans l'enregistrer.

### Signature

```
Fermer() -> Résilier
```

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

### 16.7.1.9.2.3 CreateUserDefinedFunction

Crée une fonction définie par l'utilisateur dans le document actuel.

### Signature

CreateUserDefinedFunction(dans strFunctionName:String, dans strLibraryName:String, dans strSyntax:String, dans strDetails:String, dans bInlinedUse:Boolean) -> Mapping

### Paramètres

Nom	Туре	Description
strFunctionName	String	Le nom de la fonction.
strLibraryName	String	Le nom de la bibliothèque à laquelle cette fonction appartient.
strSyntax	String	Un string qui décrit la syntaxe de cette fonction (à des fins d'information uniquement).
strDetails	String	Une description de cette fonction.
bInlinedUse	Booléen	Indicateur booléen qui spécifie si la fonction a une utilisation inlined.

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1208	Échec de la création d'une fonction définie par l'utilisateur.
1209	Il n'est pas autorisé de modifier le document. Il est en lecture seule.

### 16.7.1.9.2.4 FindComponentByID

Cherche dans le document entier, aussi dans tous ses mappages, pour le composant avec l'ID spécifiée.

### Signature

FindComponentByID(in nID:Unsigned Long) -> Composant

### Paramètres

Nom	Туре	Description
nID	Unsigned Long	L'ID du composant à rechercher.

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide a été spécifiée pour le paramètre de renvoi.

## 16.7.1.9.2.5 GenerateCHashCode

Génère le code C# qui exécutera le mappage. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

### Signature

GenerateCHashCode() -> Void

Code d'erreur	Description
1200	Cet objet n'est plus valide.
1201	L'adresse invalide a été spécifiée pour le paramètre de renvoi.
1205	Erreur pendant la génération de code.

### 16.7.1.9.2.6 GenerateCodeEx

Génère le code qui exécutera le mappage. Le paramètre **i\_nLanguage** spécifie le langage cible. La méthode renvoie un objet qui peut être utilisé pour énumérer tous les messages créés par le générateur de code. Il s'agit des mêmes messages qui sont affichés dans la fenêtre de messages de MapForce.

#### Signature

### Paramètres

Nom	Туре	Description
i_nLanguage	ENUMProgrammingLanguage <sup>1402</sup>	Spécifie le langage de génération de code cible.

#### Errors

Error code	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide a été spécifiée pour le paramètre de renvoi.
1205	Erreur pendant la génération de code

### 16.7.1.9.2.7 GenerateCppCode

Génère le code C++ qui exécutera le mappage. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

#### Signature

```
GenerateCppCode() -> Void
```

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1205	Erreur pendant la génération de code

### 16.7.1.9.2.8 GenerateJavaCode

Génère le code Java qui exécutera le mappage. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

### Signature

GenerateJavaCode() -> Void

#### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1205	Erreur pendant la génération de code

### 16.7.1.9.2.9 GenerateOutput

Génère tous les fichiers de sortie définis dans le mappage utilisant un langage de mappage interne de MapForce. Les noms des fichiers de sortie sont définis comme propriétés des éléments de sortie dans le mappage.

**Note :** Cette méthode peut uniquement être utilisée quand la fenêtre principale de MapForce (exécutée comme serveur COM) est visible, ou si elle est incorporée avec une interface utilisateur graphique. Si la méthode est appelée tandis que MapForce n'est pas visible, alors l'erreur apparaîtra.

#### Signature

```
GenerateOutput() -> Void
```

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1206	Erreur durant l'exécution de l'algorithme de mappage.
1210	Générer la sortie est uniquement pris en charge quand l'interface utilisateur graphique est visible.

### 16.7.1.9.2.10 GenerateOutputEx

Génère tous les fichiers de sortie définis dans le mappage utilisant un langage de mappage interne de MapForce. Les noms des fichiers de sortie sont définis comme propriétés des éléments de sortie dans le mappage. Cette méthode est identique à GenerateOutput, à l'exception de sa valeur de renvoi contenant les messages résultants, avertissements et erreurs organisés en tant qu'arborescences de AppOutputLines.

**Note :** Cette méthode peut uniquement être utilisée quand la fenêtre principale de MapForce (exécutée comme serveur COM) est visible, ou si elle est incorporée avec une interface utilisateur graphique. Si la méthode est appelée tandis que MapForce n'est pas visible, alors l'erreur apparaîtra.

#### Signature

GenerateOutputEx() -> AppOutputLines

#### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1206	Erreur durant l'exécution de l'algorithme de mappage.
1210	Générer la sortie est uniquement pris en charge quand l'interface utilisateur graphique est visible.

### 16.7.1.9.2.11 GenerateXQuery

Génère du code de mappage comme XQuery. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

#### Signature

```
GenerateXQuery() -> Void
```

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	
1204	Erreur durant la génération de code XSLT/XSLT2/XQuery.	

### 16.7.1.9.2.12 GenerateXSLT

Génère du code de mappage en tant que XSLT. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

### Signature

GenerateXSLT() -> Void

#### Erreurs

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	
1204	Erreur durant la génération de code XSLT/XSLT2/XQuery.	

### 16.7.1.9.2.13 GenerateXSLT2

Génère le code de mappage comme XSLT2. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

### Signature

```
GenerateXSLT2() -> Void
```

#### Erreurs

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	
1204	Erreur durant la génération de code XSLT/XSLT2/XQuery.	

### 16.7.1.9.2.14 GenerateXSLT3

Génère le code de mappage comme XSLT 3.0. Utilise les propriétés définies dans Application.Options pour configurer la génération de code.

### Signature

GenerateXSLT3() -> **Void** 

### Errors

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	
1204	Erreur durant la génération de code XSLT/XSLT2/XSLT3/XQuery.	

### 16.7.1.9.2.15 HighlightSerializedMarker

Utilisez cette méthode pour marquer un emplacement dans un fichier de mappage qui a été précédemment sérialisé. Si le document correspondant n'est pas déjà chargé, il sera chargé d'abord. Voir GenerateCodeEx pour une méthode d'extraction d'un marqueur sérialisé.

### Signature

HighlightSerializedMarker(dans i\_strSerializedMarker:String) -> Void

#### Paramètres

Nom	Туре	Description
i_strSerializedMarker	String	L'objet ErrorMarker pour mettre en évidence. Utiliser ErrorMaker.Serialized pour obtenir cette valeur.

Code d'erreur	Description
1000	L'objet n'est plus valide.
1001	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1007	Le string passé dans i_strSerializedMarker n'est pas reconnu marqueur MapForce sérialisé.
1008	Le marqueur pointe vers un emplacement qui n'est plus valide.

### 16.7.1.9.2.16 Enregistrer

Enregistre le document vers le fichier défini par Document.FullName.

### Signature

```
Enregistrer() -> Void
```

### Erreurs

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

### 16.7.1.9.2.17 SaveAs

Enregistre le document vers le nom de fichier spécifique, et définit Document.FullName vers cette valeur si l'opération Enregistrer a réussi.

### Signature

```
SaveAs(in i_strFileName:String) -> Void
```

### Paramètres

Nom	Туре	Description
i_strFileName	String	Spécifie le chemin où le document doit être enregistré.

Code d'erreur	Description	
1200	L'objet n'est plus valide.	
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

## 16.7.1.9.3 Événements

#### 16.7.1.9.3.1 OnDocumentClosed

Cet événement est déclenché lorsqu'un document est fermé. L'objet de document passé au gestionnaire d'événements ne devrait pas être accédé. L'événement ouvert correspondant est Application.OnDocumentOpened.

#### Signature

OnDocumentClosed(dans i\_ipDocument:Document) : Annuler

### 16.7.1.9.3.2 OnModifiedFlagChanged

Cet événement est déclenché lorsque le statut de modification d'un document change.

#### Signature

```
OnModifiedFlagChanged(dans i_bIsModified:Boolean) : Annuler
```

## 16.7.1.10 Documents

Représente une collection d'objets de document objects.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Ouvrir et créer des mappages :

- OpenDocument
- NewDocument

Itération à travers la collection :

- Compter
- Item
- ActiveDocument

#### Propriétés

Nom	Description
ActiveDocument <sup>1323</sup>	Lecture seule.

Nom	Description
	Extrait le document actif. Si aucun document n'est ouvert, null est retourné.
Application <sup>1324</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Compter <sup>1324</sup>	Lecture seule. Extrait le nombre de documents dans la collection.
Item <sup>1324</sup>	Lecture seule. Extrait le document à l'index $n$ de la collection. Les index commencent par 1.
Parent <sup>1325</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

### Méthodes

Nom	Description
NewDocument <sup>1325</sup>	Crée un nouveau document, l'ajoute à la fin de la collection, et rend le document actif.
OpenDocument <sup>1326</sup>	Ouvre un document de mappage existant (*.mfd). Ajoute un document nouvellement ouvert à la fin de la collection et rend le document actif.

# 16.7.1.10.1 Propriétés

## 16.7.1.10.1.1 ActiveDocument

Extrait le document actif. Si aucun document n'est ouvert, null est retourné.

### Signature

```
ActiveDocument : Document
```

Code d'erreur :	Description
1600	L'objet n'est plus valide.
1601	L'adresse invalide pour le paramètre de retour a été spécifiée.

### 16.7.1.10.1.2 Application

Extrait l'objet de l'application de premier niveau.

#### Signature

Application : Application

#### Erreurs

Code d'erreur :	Description
1600	L'objet n'est plus valide.
1601	L'adresse invalide pour le paramètre de retour a été spécifiée.

### 16.7.1.10.1.3 Count

Extrait le nombre de documents dans la collection.

#### Signature

Count : Entier

### Erreurs

Code d'erreur	Description
1600	L'objet n'est plus valide.
1601	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.10.1.4 Item

Extrait le document à l'index n de la collection. Les index commencent par 1.

#### Signature

Item(in n:Integer) : Document

### Erreurs

Code d'erreur	Description	
1600	L'objet n'est plus valide.	
1601	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

### 16.7.1.10.1.5 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Application

### Erreurs

Code d'erreur	Description
1600	L'objet n'est plus valide.
1601	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.10.2 Méthodes

### 16.7.1.10.2.1 NewDocument

Crée un nouveau document, l'ajoute à la fin de la collection, et rend le document actif.

### Signature

NewDocument() -> Document

Code d'erreur	Description
1600	L'objet n'est plus valide.
1601	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

#### 16.7.1.10.2.2 OpenDocument

Ouvre un document de mappage existant (\*.mfd). Ajoute un document nouvellement ouvert à la fin de la collection et rend le document actif.

#### Signature

OpenDocument(dans strPath :String) -> Document

#### Paramètres

Nom	Туре	Description
strPath	String	Le chemin du fichier de mappage.

#### Erreurs

Code d'erreur	Description
1600	L'objet n'est plus valide.
1601	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.11 ErrorMarker

Représente une ligne de message simple. Contrairement aux objets AppOutputLine, les marqueurs d'erreur n'ont pas de structure hiérarchique.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Accès à l'information du message :

- DocumentFileName
- ErrorLevel
- Mettre en surbrillance
- Sérialisation
- Texte

### Propriétés

Nom	Description
Application <sup>1327</sup>	Propriétés Extrait l'objet de premier niveau de l'application.

Nom	Description
DocumentFileName <sup>1328</sup>	Propriétés Extrait le nom du fichier de mappage avec lequel le marqueur d'erreur est associé.
ErrorLevel <sup>1328</sup>	Propriétés Extrait la gravité de l'erreur.
Parent <sup>1329</sup>	Propriétés L'objet parent conformément au modèle d'objet.
<u>Sérialisation</u> <sup>1329</sup>	Propriétés Sérialiser le marqueur d'erreur dans un string. Utiliser ce string dans des appels vers Application.HighlightSerializedMarker OU Document.HighlightSerializedMarker pour mettre en surbrillance l'item marqué dans le mappage. Ce string peut être persistant et utilisé dans d'autres instanciations de MapForce ou son contrôle.
Texte <sup>(132)</sup>	Propriétés Extrait le texte de message.

### Méthodes

Nom	Description
Mettre en surbrillance <sup>1330</sup>	Met en surbrillance l'item avec lequel le marqueur d'erreur est associé. Si le document correspondant n'est pas ouvert, il sera ouvert.

# 16.7.1.11.1 Propriétés

## 16.7.1.11.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

Code d'erreur :	Description
1900	L'objet n'est plus valide.

Code d'erreur :	Description
1901	L'adresse invalide pour le paramètre de retour a été spécifiée.

### 16.7.1.11.1.2 DocumentFileName

Extrait le nom du fichier de mappage avec lequel le marqueur d'erreur est associé.

## Signature

DocumentFileName : String

### Erreurs

Code d'erreur	Description
1900	L'objet n'est plus valide.
1901	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.11.1.3 ErrorLevel

Extrait la gravité de l'erreur.

### Signature

ErrorLevel : ENUMCodeGenErrorLevel

Code d'erreur	Description
1900	L'objet n'est plus valide.
1901	L'adresse invalide a été spécifiée pour le paramètre de renvoi.

### 16.7.1.11.1.4 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : ErrorMarkers

### Erreurs

Code d'erreur	Description
1900	L'objet n'est plus valide.
1901	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.11.1.5 Sérialisation

Sérialiser le marqueur d'erreur dans un string. Utilisez ce string dans des appels à

Application.HighlightSerializedMarker OU Document.HighlightSerializedMarker pour surligner l'item marqué dans le mappage. Le string peut être persistant et utilisé dans d'autres instanciations de MapForce ou son contrôle.

### Signature

Sérialisation : String

### Erreurs

Code d'erreur	Description
1900	L'objet n'est plus valide.
1901	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.11.1.6 Texte

Extrait le texte de message.

### Signature

Texte : String

### Erreurs

Code d'erreur	Description
1900	L'objet n'est plus valide.
1901	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.11.2 Méthodes

### 16.7.1.11.2.1 Mise en surbrillance

Met en surbrillance l'item avec lequel le marqueur d'erreur est associé. Si le document correspondant n'est pas ouvert, il sera ouvert.

#### Signature

Highlight() -> Void

### Erreurs

Code d'erreur	Description
1900	L'objet n'est plus valide.
1901	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1008	Le marqueur pointe vers un emplacement qui n'est plus valide.

# 16.7.1.12 ErrorMarkers

Represents a collection of ErrorMarker objects.

Properties to navigate the object model:

- Application
- Parent

Iterating through the collection:

- Count
- Item

### Properties

Name	Description
Application <sup>1331</sup>	Read-only. Retrieves the application's top-level object.
Count 1331	Read-only. Retrieves the number of error markers in the collection.
<u>Item</u> <sup>1332</sup>	Read-only. Retrieves the error marker at index $\mathbf{n}$ from the collection. Indices start with 1.
Parent <sup>1332</sup>	Read-only. The parent object according to the object model.

## 16.7.1.12.1 Propriétés

### 16.7.1.12.1.1 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

### Erreurs

Code d'erreur	Description
1800	L'objet n'est plus valide.
1801	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.12.1.2 Count

Extrait le nombre de marqueurs d'erreurs dans la collection.

### Signature

Count : Entier

#### Erreurs

Code d'erreur	Description
1800	L'objet n'est plus valide.
1801	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.12.1.3 Item

Extrait le marqueur d'erreur à l'index n de la collection. Les indices démarrent avec 1.

#### Signature

Item(in n:Integer) : ErrorMarker

#### Erreurs

Code d'erreur	Description
1800	L'objet n'est plus valide.
1801	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

### 16.7.1.12.1.4 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Application

### Erreurs

Code d'erreur	Description
1800	L'objet n'est plus valide.
1801	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.13 LibraryImport

Une importation LibraryImport représente un fichier bibliothèque importé (une entrée depuis la fenêtre Gérer les bibliothèques).
# Propriétés

Nom	Description
Application <sup>1334</sup>	Lecture seule. Extrait l'objet de premier niveau de l'application.
Parent <sup>1334</sup>	Lecture seule. Extrait l'objet parent conformément au modèle d'objet.
Chemin <sup>1335</sup>	Propriétés Obtient le chemin de la bibliothèque importée.
SaveRelativePath	Lorsque vous enregistrez le document, cette propriété spécifie si le chemin d'accès à la bibliothèque devrait être enregistré en tant qu'absolu ou relatif. Si <b>true</b> , le chemin d'accès à la bibliothèque sera relatif au document. Quand <b>false</b> , le chemin d'accès à la bibliothèque sera absolu.
	Ne vous fiez pas à cette propriété pour déterminer si le chemin est absolu ou relatif puisque le chemin peut avoir changé (soit depuis l'interface utilisateur ou via l'API), parce que le document a été chargé depuis le fichier .mfd.
	Si vous définissez la propriété (soit via API, soit via l'interface utilisateur), la fenêtre « Gérer les bibliothèques » affiche immédiatement le statut correct du chemin dans l'interface utilisateur. En interne, toutefois, le Chemin de l'objet ImportedLibrary ne sera pas modifié jusqu'à l'enregistrement du document.
	Les bibliothèques importées globalement ne peuvent pas être enregistrées avec un chemin relatif. Seules les bibliothèques importées au niveau du document peuvent être enregistrées.

## 16.7.1.13.1 Propriétés

## 16.7.1.13.1.1 Application

Extrait l'objet de l'application de premier niveau.

## Signature

Application : Application

#### Erreurs

Code d'erreur	Description
2500	L'objet n'est plus valide.
2501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.13.1.2 Parent

Extrait l'objet parent conformément au modèle d'objet.

## Signature

Parent : LibraryImports

Code d'erreur	Description
2500	L'objet n'est plus valide.
2501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.13.1.3 Chemin

Obtient le chemin de la bibliothèque importée.

## Signature

Chemin : String

## Erreurs

Code d'erreur	Description
2500	L'objet n'est plus valide.
2501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.13.1.4 SaveRelativePath

Lorsque vous enregistrez le document, cette propriété spécifie si le chemin d'accès à la bibliothèque devrait être enregistré en tant qu'absolu ou relatif. Si **true**, le chemin d'accès à la bibliothèque sera relatif au document. Quand **false**, le chemin d'accès à la bibliothèque sera absolu.

Ne vous fiez pas à cette propriété pour déterminer si le chemin est absolu ou relatif puisque le chemin peut avoir changé (soit depuis l'interface utilisateur ou via l'API), parce que le document a été chargé depuis le fichier .mfd.

Si vous définissez la propriété (soit via API, soit via l'interface utilisateur), la fenêtre « Gérer les bibliothèques » affiche immédiatement le statut correct du chemin dans l'interface utilisateur. En interne, toutefois, le Chemin de l'objet ImportedLibrary ne sera pas modifié jusqu'à l'enregistrement du document.

Les bibliothèques importées globalement ne peuvent pas être enregistrées avec un chemin relatif. Seules les bibliothèques importées au niveau du document peuvent être enregistrées.

## Signature

SaveRelativePath : Booléen

Code d'erreur	Description
2500	L'objet n'est plus valide.
2501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
2502	Les importations globales ne peuvent pas être enregistrées avec un chemin relatif.

# 16.7.1.14 LibraryImports

Représente une collection de bibliothèques importées (objets LibraryImport). Utilisez les propriétés Application et Parent pour parcourir le modèle d'objet. Utilisez les propriétés Count et Item pour itérer à travers la collection. Vous pouvez obtenir cette collection comme suit :

- Localement (au niveau du document), par le biais de la propriété Document.LibraryImports
- Globalement (au niveau de l'application), par le biais de la propriété Application.LibraryImports.

Si vous obtenez la collection LibraryImports depuis l'objet de l'application, la propriété Parent de la collection sera nulle.

## Propriétés

Nom	Description
Application <sup>1337</sup>	Lecture seule. Obtient l'objet de l'application de premier niveau.
Compter <sup>1337</sup>	Lecture seule. Obtient le compte d'objets LibraryImport dans cette collection.
<u>Item</u> <sup>1338</sup>	Lecture seule. Extrait une entrée de bibliothèque à l'index <i>n</i> de cette collection. L'index est basé sur 1.
Parent <sup>1333</sup>	Lecture seule. Obtient le document parent pour les importations de la bibliothèque locale. Si vous obtenez la collection LibraryImports depuis l'objet de l'application, la propriété Parent de la collection sera nulle.

#### Méthodes

Nom	Description	
Ajouter	Ajoute une nouvelle bibliothèque à cet objet LibraryImports. La nouvelle bibliothèque aura le chemin fourni par le paramètre i_strFileName.	
Chercher	Renvoie une référence bibliothèque en fonction du chemin vers le fichier bibliothèque.	
Supprimer <sup>1396</sup>	Supprime une référence bibliothèque depuis la fenêtre <b>Gérer les</b> <b>Bibliothèques</b> .	

# 16.7.1.14.1 Propriétés

## 16.7.1.14.1.1 Application

Obtient l'objet de l'application de premier niveau.

## Signature

Application : Application

#### Erreurs

Code d'erreur	Description
2400	L'objet n'est plus valide.
2401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.14.1.2 Count

Obtient le compte d'objets LibraryImport dans cette collection.

## Signature

Count : Entier

Code d'erreur	Description
2400	L'objet n'est plus valide.
2401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.14.1.3 Item

Extrait une entrée de bibliothèque à l'index *n* de cette collection. L'index est basé sur 1.

### Signature

Item(in n:Integer) : LibraryImport

### Erreurs

Code d'erreur	Description
2400	L'objet n'est plus valide.
2401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.14.1.4 Parent

Obtient le document parent pour les importations de la bibliothèque locale. Si vous obtenez la collection LibraryImports depuis l'objet de l'application, la propriété Parent de la collection sera nulle.

#### Signature

Parent : Document

#### Erreurs

Code d'erreur	Description
2400	L'objet n'est plus valide.
2401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.14.2 Méthodes

## 16.7.1.14.2.1 Add (2)

Adds a new library to this LibraryImports object. The new library will have the path supplied by the **i\_strFileName** parameter.

#### Signature

Add(in i\_strFileName:String) -> LibraryImport

## Parameters

Name	Туре	Description
i_strFileName	String	Specifies the path of the library file. This path can be either absolute or relative to the mapping, depending on the state in which it was passed to the object. When the document is saved, the path will be made relative if the LibraryImport.SaveRelativePa th flag is <b>true</b> ; otherwise, it will be made absolute.

# Errors

Error code	Description
2400	The object is no longer valid.
2401	Invalid address for the return parameter was specified.
2402	Adding library file failed.

# 16.7.1.14.2.2 Find (2)

Returns a library reference given the path to the library file.

## Signature

```
Find(in i_strFileName:String) -> LibraryImport
```

## Parameters

Name	Туре	Description
i_strFileName	String	The path of the library file to search for. For locally imported libraries, you can specify either the absolute or the relative path to the library file (unlike the Remove method, which requires the exact path).
		path must always be absolute

Name	Туре	Description
		(since globally imported libraries cannot have a relative path).

## Errors

Error code	Description
2400	The object is no longer valid.
2401	Invalid address for the return parameter was specified.

## 16.7.1.14.2.3 Remove (2)

Removes a library reference from the Manage Libraries window.

## Signature

Remove(in i\_strFileName:String) -> Void

## Parameters

Name	Туре	Description
i_strFileName	String	The path of the library file to remove. Note that the path must reflect exactly the current (most recent) state of the LibraryImport object. Remember that the path may be either relative or absolute, and it may have changed if you saved the document, depending on the LibraryImport.SaveRelativePath flag. Therefore, if the LibraryImport object currently contains a relative path, then you should supply a relative path as value of this parameter. Otherwise, the library is not found and the Remove method fails.
		The above applies only for locally imported libraries. For globally imported libraries, the path must always be absolute (since

Name	Туре	Description
		globally imported libraries cannot have a relative path)

### Errors

Error code	Description
2400	The object is no longer valid.

# 16.7.1.15 MapForceView

Représente l'affichage actuel dans l'onglet de mappage de MapForce pour un document. Un document a exactement un affichage MapForceView, qui affiche le mappage actuellement actif.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

#### Activation et propriétés d'affichage :

- Activé
- ShowItemTypes
- ShowLibraryInFunctionHeader
- HighlightMyConnections
- HighlightMyConnectionsRecursivly

#### Propriétés relatives au mappage :

- ActiveMapping
- ActiveMappingName

#### Ajouter des éléments :

- InsertWSDLCall
- InsertXMLFile
- InsertXMLSchema
- InsertXMLSchemaWithSample

## Propriétés

Nom	Description
Activé <sup>1343</sup>	Utilisez cette propriété pour demander si la vue de mappage est la vue active, ou définissez cette vue en tant que vue active.

Nom	Description
ActiveMapping <sup>1343</sup>	Obitent ou définit le mappage actif actuel dans le document auquel ce MapForceView appartient.
ActiveMappingName <sup>1344</sup>	Obtient ou définit le mappage actif actuel par le nom dans le document auquel ce MapForceView appartient.
Application <sup>1344</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
HighlightMyConnections <sup>1344</sup>	Cette propriété définit si uniquement les connexions depuis les éléments sélectionnés doivent être mises en surbrillance.
HighlightMyConnectionsRecursively	Cette propriété définit si uniquement les connexions venant directement ou indirectement des éléments sélectionnés doivent être mis en surbrillance.
Parent <sup>1357</sup>	Lecture seule. L'objet parent selon le modèle d'objet.
ShowItemTypes <sup>1345</sup>	Cette propriété définit si les types d'items devraient être affichés dans le diagramme de mappage.
ShowLibraryInFunctionHeader <sup>1346</sup>	Cette propriété définit si le nom de la bibliothèque de fonction devrait faire partie de noms de fonction.

## Méthodes

Nom	Description
InsertWSDLCall <sup>1346</sup>	Ajoute un nouveau composant d'appel WSDL au mappage.
InsertXMLFile <sup>1347</sup>	MapForceView.InsertXMLFile <b>est obsolète.Utilisez</b> Mapping.InsertXMLFile <b>à la place</b> .
InsertXMLSchema <sup>1347</sup>	MapForceView.InsertXMLSchema <b>est obsolète.Utilisez</b> Mapping.InsertXMLSchema <b>à la place</b> .
InsertXMLSchemaWithSample	MapForceView.InsertXMLSchemaWithSample est obsolète. Utilisez Mapping.InsertXMLFile à la place. Notez que Mapping.InsertXMLFile ne requiert pas de paramètre pour passer l'élément racine. L'élément racine est automatiquement définit en tant que nom d'élément de la racine du fichier XML.

# 16.7.1.15.1 Propriétés

## 16.7.1.15.1.1 Activé

Utilisez cette propriété pour demander si la vue de mappage est la vue active, ou définissez cette vue en tant que vue active.

## Signature

Active : Booléen			
------------------	--	--	--

### Erreurs

Code d'erreur :	Description
1300	L'objet n'est plus valide.
1301	L'adresse invalide pour le paramètre de retour a été spécifiée.

## 16.7.1.15.1.2 ActiveMapping

Obtient ou définit le mappage actif actuel dans le document auquel ce MapForceView appartient.

## Signature

ActiveMapping : Mappage

Code d'erreur :	Description	
1300	L'objet n'est plus valide.	
1301	L'adresse invalide pour le paramètre de retour a été spécifiée.	

## 16.7.1.15.1.3 ActiveMappingName

Obtient ou définit le mappage actif actuel par le nom dans le document auquel ce MapForceView appartient.

### Signature

ActiveMappingName : String

### Erreurs

Code d'erreur :	Description
1300	L'objet n'est plus valide.
1301	L'adresse invalide pour le paramètre de retour a été spécifiée.

## 16.7.1.15.1.4 Application

Extrait l'objet de l'application de premier niveau.

### Signature

Application : Application

### Erreurs

Code d'erreur	Description	
1300	L'objet n'est plus valide.	
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

## 16.7.1.15.1.5 HighlightMyConnections

Cette propriété définit si uniquement les connexions depuis les éléments sélectionnés doivent être mises en surbrillance.

#### Signature

HighlightMyConnections : Booléenne

## Erreurs

Code d'erreur	Description	
1300	L'objet n'est plus valide.	
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

## 16.7.1.15.1.6 *HighlightMyConnectionsRecursively*

Cette propriété définit si uniquement les connexions venant directement ou indirectement des éléments sélectionnés doivent être mis en surbrillance.

## Signature

HighlightMyConnectionsRecursively : Booléenne

## Erreurs

Code d'erreur	Description
1300	L'objet n'est plus valide.
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.15.1.7 ShowItemTypes

Cette propriété définit si les types d'items devraient être affichés dans le diagramme de mappage.

## Signature

```
ShowItemTypes : Booléen
```

Code d'erreur	Description	
1300	L'objet n'est plus valide.	
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

## 16.7.1.15.1.8 ShowLibraryInFunctionHeader

Cette propriété définit si le nom de la bibliothèque de fonction devrait faire partie de noms de fonction.

### Signature

ShowLibraryInFunctionHeader : Booléen

### Erreurs

Code d'erreur	Description
1300	L'objet n'est plus valide.
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.15.2 Méthodes

## 16.7.1.15.2.1 InsertWSDLCall

Ajoute un nouveau composant d'appel WSDL au mappage.

#### Signature

```
InsertWSDLCall(dans i_strWSDLFileName :String) -> Void
```

## Paramètres

Nom	Туре	Description
i_strWSDLFileName	String	Spécifie le chemin du fichier WSDL pour ajouter au mappage.

Code d'erreur	Description
1300	L'objet n'est plus valide.
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.15.2.2 InsertXMLFile (obsolete)

MapForceView.InsertXMLFile est obsolète. Utilisez Mapping.InsertXMLFile à la place.

### Signature

InsertXMLFile(in i\_strFileName:String, dans i\_strXMLRootName:String) -> Void

### Paramètres

Nom	Туре	Description
i_strFileName	String	
i_strXMLRootName	String	

## 16.7.1.15.2.3 InsertXMLSchema (obsolete)

MapForceView.InsertXMLSchema est obsolète. Utilisez Mapping.InsertXMLSchema à la place.

#### Signature

InsertXMLSchema(in i\_strSchemaFileName:String, in i\_strXMLRootName:String) -> Void

#### Paramètres

Nom	Туре	Description
i_strSchemaFileName	String	
i_strXMLRootName	String	

## 16.7.1.15.2.4 InsertXMLSchemaWithSample (obsolete)

MapForceView.InsertXMLSchemaWithSample est obsolète. Utilisez Mapping.InsertXMLFile à la place. Notez que Mapping.InsertXMLFile ne requiert pas de paramètre pour passer l'élément racine. L'élément racine est automatiquement définit en tant que nom d'élément de la racine du fichier XML.

#### Signature

```
InsertXMLSchemaWithSample(in i_strSchemaFileName:String, in i_strXMLExampleFile:String,
in i_strXMLRootName:String) -> Void
```

## Paramètres

Nom	Туре	Description
i_strSchemaFileName	String	
i_strXMLExampleFile	String	
i_strXMLRootName	String	

# 16.7.1.16 Mappage

Un objet de mappage représente un mappage dans un document, soit le mappage principal, soit le mappage de fonction définie par l'utilisateur.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

#### Propriétés au mappage :

- IsMainMapping
- Nom

Composants dans le mappage :

• Composants

#### Ajouter des éléments :

- CreateConnection
- InsertFunctionCall
- InsertXMLFile
- InsertXMLSchema
- InsertXMLSchemaInputParameter
- InsertXMLSchemaOutputParameter

## Propriétés

Nom	Description
Application <sup>(359</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Composants <sup>1350</sup>	Lecture seule. Renvoie une collection de tous les composants dans le mappage actuel.
IsMainMapping <sup>1351</sup>	Lecture seule.

Nom	Description
	Indique si le mappage actuel est le mappage principal du document dans lequel le mappage se trouve.
	<b>True</b> signifie qu'il s'agit du mappage principal. <b>False</b> signifie qu'il s'agit d'une fonction définie par l'utilisateur (UDF).
<u>Nom</u> <sup>(35)</sup>	Lecture seule. Le nom du mappage ou de la fonction définie par l'utilisateur (UDF).
Parent <sup>1365</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

## Méthodes

Nom	Description
CreateConnection <sup>1351</sup>	Crée une connexion entre les deux datapoints fournis (DatapointFrom & DatapointTo).
	La connexion ne sera pas établie si DatapointFrom n'est pas un datapoint output-side, si DatapointTo n'est pas un datapoint input-side, ou si une connexion entre ces deux datapoints existe déjà.
InsertFunctionCall <sup>1352</sup>	Insère un composant d'appel de fonction dans le mappage actuel.
	Les noms de bibliothèque et de fonction spécifiés indiquent la fonction ou la fonction définie par l'utilisateur à appeler.
InsertXMLFile <sup>1353</sup>	Ajoute un nouveau compostant de schéma XML au mappage.
	La structure interne du composant est déterminée par le schéma référencé dans le fichier XML spécifié (i_strFileName) ou, si le fichier XML ne référence pas de fichier de schéma, par le fichier de schéma spécifié séparément (i_strSchemaFileName).
	Si le fichier XML a une référence de fichier de schéma, alors le paramètre i_strSchemaFileName est ignoré.
	L'élément racine du fichier XML sera utilisé dans le composant.
	Le fichier XML spécifié est utilisé comme échantillon d'entrée pour évaluer le mappage.
InsertXMLSchema <sup>1354</sup>	Ajoute un nouveau compostant de schéma XML au mappage.

Nom	Description
	La structure interne du composant est déterminée par le fichier de schéma spécifié dans le premier paramètre.
	Le deuxième paramètre définit l'élément racine de ce schéma s'il existe plus d'un candidat.
	Si l'élément racine passé est un string vide et plus de candidats sont disponibles, une boîte de dialogue <b>Sélectionner</b> <b>l'élément racine</b> apparaîtra si MapForce est visible. Si MapForce est invisible, aucune boîte de dialogue n'apparaîtra et seule une erreur est retournée.
	Aucun échantillon d'entrée XML n'est assigné à ce composant.
InsertXMLSchemaInputParameter <sup>(1355</sup>	Insère un composant de paramètre d'entrée du schéma XML dans le mappage actuel.
	Le mappage actuel doit être une fonction définie par l'utilisateur. Tenter de l'insérer (le paramètre d'entrée de schéma) dans le mappage principal échouera.
InsertXMLSchemaOutputParameter <sup>(356</sup>	Insère un composant de paramètre de sortie de schéma XML dans le mappage actuel.
	Le mappage actuel doit être une fonction définie par l'utilisateur. Tenter de l'insérer (le paramètre de sortie du schéma) dans le mappage principal échouera.

# 16.7.1.16.1 Propriétés

## 16.7.1.16.1.1 Composants

Renvoie une collection de tous les composants dans le mappage actuel.

## Signature

Components : Composants

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.16.1.2 IsMainMapping

Indique si le mappage actuel est le mappage principal du document dans lequel le mappage se trouve.

**True** signifie qu'il s'agit du mappage principal. **False** signifie qu'il s'agit d'une fonction définie par l'utilisateur (UDF).

### Signature

IsMainMapping : Booléenne

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.16.1.3 Nom

Le nom du mappage ou de la fonction définie par l'utilisateur (UDF).

#### Signature

Nom : String

## Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.16.2 Méthodes

## 16.7.1.16.2.1 CreateConnection

Crée une connexion entre les deux datapoints fournis (DatapointFrom & DatapointTo).

La connexion ne sera pas établie si DatapointFrom n'est pas un datapoint output-side, si DatapointTo n'est pas un datapoint input-side, ou si une connexion entre ces deux datapoints existe déjà.

## Signature

CreateConnection(dans DatapointFrom:Datapoint, dans DatapointTo:Datapoint) -> Connection

### Paramètres

Nom	Туре	Description
DatapointFrom	Datapoint	Le datapoint duquel la connexion doit être créée.
DatapointTo	Datapoint	Le datapoint de destination.

#### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
1241	Échec de la création de la connexion.

## 16.7.1.16.2.2 InsertFunctionCall

Insère un composant d'appel de fonction dans le mappage actuel.

Les noms de bibliothèque et de fonction spécifiés indiquent la fonction ou la fonction définie par l'utilisateur à appeler.

#### Signature

InsertFunctionCall(dans strFunctionName:String, dans strLibraryName:String) -> Component

## Paramètres

Nom	Туре	Description
strFunctionName	String	Le nom de la fonction à être insérée.
strLibraryName	String	Le nom de la bibliothèque de la fonction à insérer

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
1242	Échec lors de la création du composant d'appel de la fonction.

## 16.7.1.16.2.3 InsertXMLFile

Ajoute un nouveau compostant de schéma XML au mappage.

La structure interne du composant est déterminée par le schéma référencé dans le fichier XML spécifié (i\_strFileName) ou, si le fichier XML ne référence pas de fichier de schéma, par le fichier de schéma spécifié séparément (i\_strSchemaFileName).

Si le fichier XML a une référence de fichier de schéma, alors le paramètre i\_strSchemaFileName est ignoré.

L'élément racine du fichier XML sera utilisé dans le composant.

Le fichier XML spécifié est utilisé comme échantillon d'entrée pour évaluer le mappage.

#### Signature

```
InsertXMLFile(in i_strFileName:String, dans i_strSchemaFileName:String) -> Component
```

## Paramètres

Nom	Туре	Description
i_strFileName	String	Le chemin de l'instance du fichier XML à ajouter.
i_strSchemaFileName	String	Le chemin du fichier de la Définition de schéma XML à ajouter.

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

Code d'erreur	Description
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
1244	Échec de la création du composant.

## 16.7.1.16.2.4 InsertXMLSchema

Ajoute un nouveau compostant de schéma XML au mappage.

La structure interne du composant est déterminée par le fichier de schéma spécifié dans le premier paramètre.

Le deuxième paramètre définit l'élément racine de ce schéma s'il existe plus d'un candidat.

Si l'élément racine passé est un string vide et plus de candidats sont disponibles, une boîte de dialogue **Sélectionner l'élément racine** apparaîtra si MapForce est visible. Si MapForce est invisible, aucune boîte de dialogue n'apparaîtra et seule une erreur est retournée.

Aucun échantillon d'entrée XML n'est assigné à ce composant.

### Signature

```
InsertXMLSchema(dans i_strSchemaFileName:String, dans i_strXMLRootName:String) ->
Component
```

#### Paramètres

Nom	Туре	Description
i_strSchemaFileName	String	Le chemin du fichier de la Définition de schéma XML à ajouter.
i_strXMLRootName	String	L'élément racine du schéma (si applicable quand le schéma a plus d'un élément racine).

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
1244	Échec de la création du composant.

## 16.7.1.16.2.5 InsertXMLSchemaInputParameter

Insère un composant de paramètre d'entrée du schéma XML dans le mappage actuel.

Le mappage actuel doit être une fonction définie par l'utilisateur. Tenter de l'insérer (le paramètre d'entrée de schéma) dans le mappage principal échouera.

## Signature

InsertXMLSchemaInputParameter(dans strParamName:String, in strSchemaFileName:String, dans
strXMLRootElementName:String) -> Component

## Paramètres

Nom	Туре	Description
strParamName	String	Le nom du composant de paramètre d'entrée à créer.
strSchemaFileName	String	Le chemin du fichier de la Définition de schéma XML à ajouter.
strXMLRootElementName	String	L'élément racine du schéma (si applicable quand le schéma a plus d'un élément racine). Si l'élément racine passé est un string vide et plus de candidats sont disponibles, un dialogue <b>Sélectionner l'élément racine</b> apparaîtra si MapForce est visible. Si MapForce est invisible, aucune boîte de dialogue n'apparaîtra et seule une erreur est retournée.

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
1243	Échec de la création du composant de paramètre.
1245	Cette opération n'est pas prise en charge pour le mappage principal.

## 16.7.1.16.2.6 InsertXMLSchemaOutputParameter

Insère un composant de paramètre de sortie de schéma XML dans le mappage actuel.

Le mappage actuel doit être une fonction définie par l'utilisateur. Tenter de l'insérer (le paramètre de sortie du schéma) dans le mappage principal échouera.

## Signature

InsertXMLSchemaOutputParameter(dans strParamName:String, in strSchemaFileName:String, in
strXMLRootElementName:String) -> Component

## **Paramètres**

Nom	Туре	Description
strParamName	String	Le nom du composant de paramètre de sortie à créer.
strSchemaFileName	String	Le chemin du fichier de la Définition de schéma XML à ajouter.
strXMLRootElementName	String	L'élément racine du schéma (si applicable quand le schéma a plus d'un élément racine). Si l'élément racine passé est un string vide et plus de candidats sont disponibles, un dialogue <b>Sélectionner l'élément racine</b> apparaîtra si MapForce est visible. Si MapForce est invisible, aucune boîte de dialogue n'apparaîtra et seule une erreur est retournée.

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1240	Il n'est pas autorisé de modifier le document. Il est en lecture seule.
1243	Échec de la création du composant de paramètre.
1245	Cette opération n'est pas prise en charge pour le mappage principal.

# 16.7.1.17 Mappages

Représente une collection d'objets de mappage.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Itération à travers la collection :

- Compter
- Item

## Propriétés

Nom	Description
Application <sup>1371</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
Compter <sup>1371</sup>	Lecture seule. Extrait le nombre de mappages dans la collection.
Item (1372)	Lecture seule. Extrait le mappage à l'index n de la collection. Les index commencent par 1.
Parent <sup>1375</sup>	Lecture seule. L'objet parent selon le modèle d'objet.

## 16.7.1.17.1 Propriétés

## 16.7.1.17.1.1 Parent

L'objet parent selon le modèle d'objet.

## Signature

Parent : Document

Code d'erreur	Description
1300	L'objet n'est plus valide.

Code d'erreur	Description
1301	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.18 Options

Cet objet donne accès à toutes les options disponibles dans le dialogue Outils | Options.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

#### Options générales :

- ShowLogoOnPrint
- ShowLogoOnStartup
- UseGradientBackground

#### Options pour la génération de code :

- DefaultOutputEncoding
- DefaultOutputByteOrder
- DefaultOutputByteOrderMark
- XSLTDefaultOutputDirectory
- CodeDefaultOutputDirectory
- CPPSettings\_DOMType
- CPPSettings\_GenerateVC6ProjectFile
- CppSettings\_GenerateVSProjectFile
- CPPSettings\_LibraryType
- CPPSettings\_UseMFC
- CSharpSettings\_ProjectType

## Propriétés

Nom	Description
Application <sup>1386</sup>	Lecture seule. Extrait l'objet de l'application de premier niveau.
CodeDefaultOutputDirectory <sup>(360)</sup>	Spécifie le répertoire cible où les fichiers générés par Document.GenerateCppCode, Document.GenerateJavaCode et Document.GenerateCHashCode sont placés.
CPPSettings_DOMType <sup>1360</sup>	Spécifie le type de DOM utilisé par Document.GenerateCppCode.
CPPSettings_GenerateVC6ProjectFile	Spécifie si les fichiers de projet VisualC++ 6.0 devraient être générés par Document.GenerateCppCode.
CppSettings_GenerateVSProjectFile	Spécifie la version de Visual Studio dans laquelle les fichiers de projet devraient être générés par Document.GenerateCppCode.

Nom	Description
CPPSettings_LibraryType	Spécifie le type de bibliothèque utilisé par Document.GenerateCppCode.
CPPSettings_UseMFC <sup>1362</sup>	Spécifie si la prise en charge MFC devrait être utilisée par C++ code généré par Document.GenerateCppCode.
CSharpSettings_ProjectType	Spécifie le type de projet C# utilisé par Document.GenerateCHashCode.
DefaultOutputByteOrder <sup>1363</sup>	Ordre d'octet pour l'encodage de fichier utilisé pour les fichiers de sortie.
DefaultOutputByteOrderMark <sup>1363</sup>	Indique si une marque d'ordre d'octet (BOM) doit être incluse dans l'encodage de fichier des fichiers de sortie.
DefaultOutputEncoding <sup>1364</sup>	Encodage de fichier utilisé pour les fichiers de sortie.
GenerateWrapperClasses <sup>1364</sup>	Indique si les classes de wrapper doivent également être générées lorsqu'elles génèrent du code.
JavaSettings_ApacheAxisVersion	Cette propriété est obsolète.
Parent <sup>1389</sup>	Lecture seule. L'objet parent selon le modèle d'objet.
ShowLogoOnPrint <sup>1365</sup>	Afficher ou masquer le logo de MapForce sur des sorties imprimées.
ShowLogoOnStartup <sup>1365</sup>	Afficher ou masquer le logo de MapForce sur démarrage de l'application.
UseGradientBackground <sup>1366</sup>	Définit ou extrait le mode de couleur d'arrière-plan pour une fenêtre de mappage.
XSLTDefaultOutputDirectory <sup>(366)</sup>	Spécifie le répertoire cible où les fichiers générés par Document.GenerateXSLT sont placés.

# 16.7.1.18.1 Propriétés

# 16.7.1.18.1.1 Application

Extrait l'objet de l'application de premier niveau.

## Signature

Application : Application

### Erreurs

Code d'erreur :	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

## 16.7.1.18.1.2 CodeDefaultOutputDirectory

Spécifie le répertoire cible où les fichiers générés par Document.GenerateCppCode, Document.GenerateJavaCode et Document.GenerateCHashCode sont placés.

## Signature

CodeDefaultOutputDirectory : String

### Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.3 CPPSettings\_DOMType

Spécifie le type de DOM utilisé par Document.GenerateCppCode.

## Signature

```
CPPSettings_DOMType : ENUMDOMType
```

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1402	La valeur de paramètre est en dehors de la plage.
1403	La valeur de paramètre n'est plus disponible.

## 16.7.1.18.1.4 CPPSettings\_GenerateVC6ProjectFile (obsolete)

Spécifie si les fichiers de projet VisualC++ 6.0 devraient être générés par Document.GenerateCppCode.

## Signature

CPPSettings\_GenerateVC6ProjectFile : Booléen

## Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1402	La valeur de paramètre est en dehors de la plage.
1403	La valeur de paramètre n'est plus disponible.

## 16.7.1.18.1.5 CppSettings\_GenerateVSProjectFile

Spécifie la version de Visual Studio dans laquelle les fichiers de projet devraient être générés par Document.GenerateCppCode.

## Signature

CppSettings\_GenerateVSProjectFile : <u>ENUMProjectType</u><sup>1403</sup>

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1402	La valeur de paramètre est en dehors de la plage.
1403	La valeur de paramètre n'est plus disponible.

## 16.7.1.18.1.6 CPPSettings\_LibraryType

Spécifie le type de bibliothèque utilisé par Document.GenerateCppCode.

### Signature

```
CPPSettings_LibraryType : ENUMLibType<sup>1402</sup>
```

## Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.7 CPPSettings\_UseMFC

Spécifie si la prise en charge MFC devrait être utilisée par C++ code généré par Document.GenerateCppCode.

## Signature

CPPSettings\_UseMFC : Booléen

#### Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.8 CSharpSettings\_ProjectType

Spécifie le type de projet C# utilisé par Document.GenerateCHashCode.

#### Signature

CSharpSettings\_ProjectType : ENUMProjectType<sup>1403</sup>

Code d'erreur	Description
1400	L'objet n'est plus valide.

Code d'erreur	Description
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1402	La valeur de paramètre est en dehors de la plage.
1403	La valeur de paramètre n'est plus disponible.

## 16.7.1.18.1.9 DefaultOutputByteOrder

Ordre d'octet pour l'encodage utilisé pour les fichiers de sortie.

### Signature

DefaultOutputByteOrder : String

## Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.10 DefaultOutputByteOrderMark

Indique si une marque d'ordre d'octet (BOM) doit être incluse dans l'encodage de fichier des fichiers de sortie.

## Signature

DefaultOutputByteOrderMark : Booléen

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.11 DefaultOutputEncoding

Encodage de fichier utilisé pour les fichiers de sortie.

## Signature

DefaultOutputEncoding : String

## Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.12 GenerateWrapperClasses

Indique si les classes de wrapper doivent également être générées lorsqu'elles génèrent du code.

## Signature

GenerateWrapperClasses : Booléenne

#### Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.13 JavaSettings\_ApacheAxisVersion (obsolete)

Cette propriété est obsolète.

## Signature

JavaSettings\_ApacheAxisVersion : ENUMApacheAxisVersion

## 16.7.1.18.1.14 Parent

L'objet parent selon le modèle d'objet.

## Signature

Parent : Document

## Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.15 ShowLogoOnPrint

Afficher ou masquer le logo de MapForce sur des sorties imprimées.

## Signature

ShowLogoOnPrint : Booléen

## Erreurs

Code d'erreur	Description	
1400	L'objet n'est plus valide.	
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

## 16.7.1.18.1.16 ShowLogoOnStartup

Afficher ou masquer le logo de MapForce sur démarrage de l'application.

## Signature

ShowLogoOnStartup : Booléen

#### Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.17 UseGradientBackground

Définit ou extrait le mode de couleur d'arrière-plan pour une fenêtre de mappage.

### Signature

UseGradientBackground : Booléen

#### Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

## 16.7.1.18.1.18 XSLTDefaultOutputDirectory

Spécifie le répertoire cible où les fichiers générés par Document.GenerateXSLT sont placés.

## Signature

```
XSLTDefaultOutputDirectory : String
```

## Erreurs

Code d'erreur	Description	
1400	L'objet n'est plus valide.	
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.	

# 16.7.1.19 Projet

Un objet de Projet représente un projet et son arborescence d'éléments de projet dans MapForce.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Gestion de fichier :

- FullName
- Nom
- Chemin
- Enregistré
- Enregistrer
- Fermer

Navigation de l'arborescence de projet :

- Compter
- Item
- \_NewEnum

Manipulation de l'arborescence de projet :

- AddActiveFile
- AddFile
- InsertWebService (uniquement Enterprise edition)
- CreateFolder

#### Génération de code :

- Output\_Folder
- Output\_Language
- Output\_TextEncoding
- Java\_BasePackageName
- GenerateCode
- GenerateCodeEx
- GenerateCodeIn
- GenerateCodeInEx

Pour des exemples sur comment utiliser les propriétés et méthodes recensées ci-dessous, voir <u>Exemple</u> : <u>Tâches de projet</u> . Notez que pour réaliser des opérations qui impliquent les services Web, MapForce Enterprise edition est requis.

## Propriétés

Nom	Description
_NewEnum <sup>1370</sup>	Propriétés Cette propriété prend en charge une énumération standard spécifique au langage.
Application	Lecture seule. Extrait l'objet de l'application de premier niveau.
Compter <sup>(1387)</sup>	Lecture seule.

Nom	Description
	Extrait le nombre d'enfants de l'item racine du projet. Pour des exemples, voir Item ou _NewEnum
FullName <sup>1372</sup>	Chemin et nom du fichier de projet.
Item (138)	Lecture seule. Retourne l'enfant à la position <i>n</i> de la racine du projet. L'index est basé sur 1 (le premier index est 1). L'index le plus grand valide est Count. Pour une alternative, voir _NewEnum.
Java_BasePackageName	Définit ou obtient le nom du package de base des packages Java qui seront générés. Cette propriété est utilisée uniquement lorsque le code Java est généré.
<u>Nom</u> <sup>(373)</sup>	Lecture seule. Nom du fichier de projet sans chemin de fichier.
Output_Folder 1373	Définit ou obtient le dossier de sortie par défaut utilisé avec GenerateCode et GenerateCodeIn. Les éléments de projet peuvent écraser cette valeur dans leur propriété CodeGenSettings_OutputFolder, quand CodeGenSettings_UseDefault est définie à false.
Output_Language 1374	Définit ou obtient le langage par défaut pour la génération de code en utilisant GenerateCode. Les éléments de projet peuvent écraser cette valeur dans leur propriété CodeGenSettings_OutputLanguage quand CodeGenSettings_UseDefault est défini à false.
Output_TextEncoding <sup>1374</sup>	Définit ou obtient l'encodage de texte utilisé lorsque le code basé sur XML est généré.
Parent	Lecture seule. L'objet parent selon le modèle d'objet.
Chemin <sup>1375</sup>	Lecture seule. Chemin du fichier de projet sans nom.
Enregistré <sup>1375</sup>	Lecture seule. <b>True</b> si le document n'a pas été modifié depuis la dernière opération de sauvegarde, <b>false</b> , si le contraire.

## Méthodes

Nom	Description
AddActiveFile 1376	Ajoute le document actuellement ouvert au dossier de mappage de la racine du projet.
AddFile <sup>(377)</sup>	Ajoute le document spécifié au dossier de mappage de la racine du projet.
Nom	Description
---------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Fermer <sup>(1377)</sup>	Ferme le projet sans l'enregistrer.
CreateFolder <sup>1378</sup>	Crée un nouveau dossier en tant qu'enfant de l'item racine du projet.
GenerateCode <sup>1378</sup>	Génère le code pour tous les items de projet du projet. Le langage de code et l'emplacement de sortie sont déterminés par des propriétés du projet et des items de projet.
GenerateCodeEx	Génère le code pour tous les items de projet du projet. Le langage de code et l'emplacement de sortie sont déterminés par des propriétés du projet et des items de projet. Un objet qui peut être utilisé pour itérer à travers tous les messages publiés par le processus de générateur de code est retourné. Ces messages sont les mêmes que ceux affichés dans la fenêtre de messages de MapForce.
<u>GenerateCodeIn</u> <sup>1379</sup>	Génère le code pour tous les items de projet du projet dans le langage spécifié. L'emplacement de sortie est déterminé par les propriétés du projet et les items de projet.
GenerateCodeInEx <sup>1379</sup>	Génère le code pour tous les items de projet du projet dans le langage spécifié. L'emplacement de sortie est déterminé par les propriétés du projet et les items de projet. Renvoie un objet qui peut être utilisé pour itérer à travers tous les messages émis par le processus de génération de code. Ces messages sont les mêmes que ceux affichés dans la fenêtre <b>Messages</b> de MapForce.
InsertWebService <sup>1380</sup>	Insère un nouveau projet de service Web dans le dossier de service Web du projet. Si <b>i_bGenerateMappings</b> est true, les documents de mappage initial pour tous les ports sont générés automatiquement.
Enregistrer <sup>1381</sup>	Enregistre le projet vers le fichier défini par FullName.

# Événements

Nom	Description
OnProjectClosed <sup>1381</sup>	Cet événement est déclenché quand le projet est fermé. L'objet du projet passé dans le gestionnaire d'événement ne devrait pas être accédé. L'événement ouvert correspondant est Application.OnProjectOpened.

### 16.7.1.19.1 Propriétés

### 16.7.1.19.1.1 \_NewEnum

Cette propriété prend en charge une énumération standard spécifique au langage.

#### Signature

|--|

#### Erreurs

Code d'erreur :	Description
1500	L'objet n'est plus valide.

#### Exemples

```
// ------
// JScript sample - enumeration of a project's project items.
function AllChildrenOfProjectRoot()
{
    objProject = objMapForce.ActiveProject;
    if ( objProject != null )
        {
        for ( objProjectIter = new Enumerator(objProject); ! objProjectIter.atEnd(); objProjectIt
        er.moveNext() )
            {
            objProjectItem = objProjectItemIter.item();
            // do something with project item here
        }
    }
}
```

```
// ------
// JScript sample - iterate all project items, depth first.
function IterateProjectItemsRec(objProjectItemIter)
{
    while ( ! objProjectItemIter.atEnd() )
    {
        objProjectItem = objProjectItemIter.item();
        // do something with project item here
        IterateProjectItemsRec( new Enumerator(objProjectItem) );
        objProjectItemIter.moveNext();
    }
```

```
}
function IterateAllProjectItems()
{
    objProject = objMapForce.ActiveProject;
    if ( objProject != null )
    {
        IterateProjectItemsRec( new Enumerator(objProject) );
    }
}
```

# 16.7.1.19.1.2 Application

Extrait l'objet de l'application de premier niveau.

# Signature

Application : Application

# Erreurs

Code d'erreur :	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.19.1.3 Count

Extrait le nombre de mappages dans la collection.

# Signature

Count : Entier

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.4 FullName

Chemin et nom du fichier de projet.

### Signature

FullName : String

### Erreurs

Code d'erreur	Description
1500	Cet objet n'est plus valide.
1501	L'adresse invalide a été spécifiée pour le paramètre de renvoi.

# 16.7.1.19.1.5 Item

Extrait le mappage à l'index n de la collection. Les index commencent par 1.

#### Signature

Item(in n:Integer) : Mappage

### Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.6 Java\_BasePackageName

Définit ou obtient le nom du package de base des packages Java qui seront générés. Cette propriété est utilisée uniquement lorsque le code Java est généré.

#### Signature

Java\_BasePackageName : String

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	Nom du pack invalide spécifié. L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.7 Nom

Nom du fichier de projet sans chemin de fichier.

# Signature

Nom : String

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.8 Output\_Folder

Définit ou obtient le dossier de sortie par défaut utilisé avec GenerateCode et GenerateCodeIn. Les éléments de projet peuvent écraser cette valeur dans leur propriété CodeGenSettings\_OutputFolder, quand CodeGenSettings\_UseDefault est définie à false.

# Signature

Output\_Folder : String

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	Nom de dossier invalide spécifié. L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.9 Output\_Language

Définit ou obtient le langage par défaut pour la génération de code en utilisant GenerateCode. Les éléments de projet peuvent écraser cette valeur dans leur propriété CodeGenSettings\_OutputLanguage quand CodeGenSettings\_UseDefault est défini à false.

# Signature

	(102
Output Language :	ENTIMProgrammingLanguage
oucpuc_hanguage	<u>Hitorit i Oğrammiriği aliğadığa</u>

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	Le langage invalide est spécifié. L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.10 Output\_TextEncoding

Définit ou obtient l'encodage de texte utilisé lorsque le code basé sur XML est généré.

### Signature

Output\_TextEncoding : String

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	L'encodage de texte invalide est spécifié. L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.11 Parent

L'objet parent selon le modèle d'objet.

## Signature

Parent : Document

# Erreurs

Code d'erreur	Description
1200	L'objet n'est plus valide.
1201	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.12 Path

Chemin du fichier de projet sans nom.

## Signature

Path : String

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.1.13 Enregistré

True si le document n'a pas été modifié depuis la dernière opération de sauvegarde, false, si le contraire.

### Signature

Enregistré : Booléenne

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.19.2 Méthodes

# 16.7.1.19.2.1 AddActiveFile

Ajoute le document actuellement ouvert au dossier de mappage de la racine du projet.

# Signature

AddActiveFile() -> **ProjectItem** 

Code d'erreur :	Description
1500	L'objet n'est plus valide.
1501	L'adresse invalide pour le paramètre de retour a été spécifiée.
1503	Aucun document valide n'est disponible.
1504	Un nom de chemin doit être attribué aux documents actifs avant qu'il ne puisse être ajouté au projet.
1705	Le mappage n'a pas pu être attribué au projet. Peut-être est-il déjà contenu dans le dossier cible.

# 16.7.1.19.2.2 AddFile

Ajoute le document spécifié au dossier de mappage de la racine du projet.

## Signature

```
AddFile(in i_strFileName:String) -> ProjectItem
```

# Paramètres

Nom	Туре	Description
i_strFileName	String	Spécifie le chemin d'accès du document à ajouter.

## Erreurs

Code d'erreur :	Description
1500	L'objet n'est plus valide.
1501	L'adresse invalide pour le paramètre de retour a été spécifiée.
1705	Le mappage n'a pas pu être attribué au projet. Le fichier n'existe pas ou n'est pas un mappage de MapForce. Peut-être que le fichier est déjà attribué au dossier cible.

# 16.7.1.19.2.3 Fermer

Ferme le projet sans l'enregistrer.

# Signature

```
Fermer() -> Résilier
```

Code d'erreur	Description
1500	L'objet n'est plus valide.

# 16.7.1.19.2.4 CreateFolder

Crée un nouveau dossier en tant qu'enfant de l'item racine du projet.

### Signature

CreateFolder(dans i\_strFileName:String) -> ProjectItem

### Paramètres

Nom	Туре	Description
i_strFolderName	String	Le nom du dossier à créer.

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	Le nom du dossier invalide ou l'adresse invalide pour le paramètre de renvoi a été spécifié.

# 16.7.1.19.2.5 GenerateCode

Génère le code pour tous les items de projet du projet. Le langage de code et l'emplacement de sortie sont déterminés par des propriétés du projet et des items de projet.

#### Signature

GenerateCode() -> **Void** 

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1706	Erreur pendant la génération de code.

# 16.7.1.19.2.6 GenerateCodeEx

Génère le code pour tous les items de projet du projet. Le langage de code et l'emplacement de sortie sont déterminés par des propriétés du projet et des items de projet. Un objet qui peut être utilisé pour itérer à travers

tous les messages publiés par le processus de générateur de code est retourné. Ces messages sont les mêmes que ceux affichés dans la fenêtre de messages de MapForce.

# Signature

GenerateCodeEx() -> ErrorMarkers

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	Adresse invalide a été spécifiée pour le paramètre de renvoi.
1706	Erreur pendant la génération de code.

# 16.7.1.19.2.7 GenerateCodeIn

Génère le code pour tous les items de projet du projet dans le langage spécifié. L'emplacement de sortie est déterminé par les propriétés du projet et de l'élément de projet..

## Signature

GenerateCodeIn(in i\_nLanguage:ENUMProgrammingLanguage<sup>1402</sup>) -> Void

### Paramètres

Nom	Туре	Description
i_nLanguage	ENUMProgrammingLanguage <sup>1402</sup>	Spécifie le langage de programmation dans lequel le code devrait être généré.

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1706	Erreur pendant la génération de code.

# 16.7.1.19.2.8 GenerateCodeInEx

Génère le code pour tous les items de projet du projet dans le langage spécifié. L'emplacement de sortie est déterminé par les propriétés du projet et de l'élément de projet. Renvoie un objet qui peut être utilisé pour itérer

à travers tous les messages émis par le processus de génération de code. Ces messages sont les mêmes que ceux affichés dans la fenêtre **Messages** de MapForce.

### Signature

GenerateCodeInEx(in i\_nLanguage:ENUMProgrammingLanguage<sup>402</sup>) -> ErrorMarkers

### Paramètres

Nom	Туре	Description
i_nLanguage	ENUMProgrammingLanguage <sup>1402</sup>	Spécifie le langage de programmation dans lequel le code devrait être généré.

#### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1706	Erreur pendant la génération de code.

### 16.7.1.19.2.9 InsertWebService

Insère un nouveau projet de service Web dans le dossier de service Web du projet. Si **i\_bGenerateMappings** est true, les documents de mappage initial pour tous les ports sont générés automatiquement.

#### Signature

```
InsertWebService(in i_strWSDLFile:String, in i_strService:String, dans i_strPort:String,
in i_bGenerateMappings:Boolean) -> ProjectItem
```

#### Paramètres

Nom	Туре	Description
i_strWSDLFile	String	Spécifie le chemin du fichier WSDL à ajouter.
i_strService	String	Spécifie le nom du service Web à ajouter.
i_strPort	String	Spécifie le port du service Web à ajouter.

Nom	Туре	Description
i_bGenerateMappings	Booléenne	Si ce paramètre est true, les documents de mappage initial pour tous les ports sont générés automatiquement.

Code d'erreur	Description
1500	L'objet n'est plus valide.
1501	Le fichier WDSL ne peut pas être trouvé ou est invalide. Le service ou les noms de port sont invalides. L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1503	L'opération n'est pas prise en charge par l'édition actuelle.

# 16.7.1.19.2.10 Enregistrer

Enregistre le projet vers le fichier défini par FullName.

## Signature

Enregistrer() -> Void

### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.
1502	Ne peut pas enregistrer le fichier.

# 16.7.1.19.3 Événements

# 16.7.1.19.3.1 OnProjectClosed

Cet événement est déclenché quand le projet est fermé. L'objet du projet passé dans le gestionnaire d'événement ne devrait pas être accédé. L'événement ouvert correspondant est Application.OnDocumentOpened.

### Signature

OnProjectClosed(in i\_ipProject:Project) : Annuler

# 16.7.1.20 ProjectItem

Un objet ProjectItem représente un élément dans une arborescence de projet.

Propriétés pour parcourir le modèle d'objet :

- Application
- Parent

Navigation de l'arborescence de projet :

- Compter
- Item
- \_NewEnum

Propriétés d'élément de projet :

- Genre
- Nom
- WSDLFile (uniquement disponible pour les éléments de projet du service Web)
- QualifiedName (uniquement disponible pour les éléments de projet du service Web)

Manipulation de l'arborescence de projet :

- AddActiveFile (uniquement disponible pour les éléments de dossier)
- AddFile (uniquement disponible pour les éléments de dossier)
- CreateFolder (uniquement disponible pour les éléments de dossier)
- CreateMappingForProject (uniquement disponible pour les opérations du service Web)
- Supprimer

Accès au document :

• Ouvrir (uniquement disponible pour les éléments de mappage et opérations du service Web)

#### Code-generation :

- CodeGenSettings\_UseDefault
- CodeGenSettings\_OutputFolder
- CodeGenSettings\_Language
- GenerateCode
- GenerateCodeEx
- GenerateCodeIn
- GenerateCodeInEx

Pour des exemples sur comment utiliser les propriétés et méthodes recensées ci-dessous, voir <u>Exemple :</u> <u>Tâches de projet</u><sup>[24]</sup>. Pour des opérations avec les services Web, MapForce Enterprise edition est requis.

#### Propriétés

Nom	Description
_NewEnum <sup>1385</sup>	Lecture seule.

Nom	Description
	Cette propriété prend en charge une énumération standard spécifique au langage. Pour des exemples, voir Project.Item OU ProjectNewEnum.
Application	Lecture seule. Extrait l'objet de l'application de premier niveau.
CodeGenSettings_Language	Obtient ou définit le langage à utiliser avec GenerateCode ou Project.GenerateCode. Cette propriété est consultée uniquement si CodeGenSettings_UseDefault est défini comme false.
CodeGenSettings_OutputFolder	Obtient ou définit le répertoire de sortie à utiliser avec GenerateCode, GenerateCodeIn, Project.GenerateCode OU Project.GenerateCodeIn. Cette propriété est consultée uniquement si CodeGenSettings_UseDefault est défini comme false.
CodeGenSettings_UseDefault	Obtient ou définit si le répertoire de sortie et le langage de code sont utilisés tels que définis soit par (a) les dossiers parents, ou (b) la racine du projet. Cette propriété est utilisée avec des appels pour GenerateCode, GenerateCodeIn, Project.GenerateCode et Project.GenerateCodeIn. Si la propriété est définie en tant que false, les valeurs de CodeGenSettings_OutputFolder et CodeGenSettings_Language sont utilisées pour générer le code pour cet item de projet.
Compter	Lecture seule. Extrait le nombre d'enfants de l'item de ce projet. Voir aussi Item. Pour des exemples, voir Project.Item ou ProjectNewEnum.
Item	Lecture seule. Retourne l'enfant à la position n de cet item de projet. L'index est basé sur 1 (le premier index est 1). L'index valide le plus grand est ProjectItem.Count. Pour une alternative, voir ProjectItem.NewEnum. Pour des exemples, voir Project.Item OU Project.NewEnum.
Genre	Propriétés Extrait le genre d'élément de projet. La disponibilité de quelques propriétés et l'applicabilité de certaines méthodes sont restreintes aux genres spécifiques d'éléments de projet. La description de toutes les méthodes et propriétés contient des informations sur ces restrictions.
Nom 1388	Extrait ou définit le nom d'un élément de projet. Le nom de la plupart des éléments est en lecture seule. Les exceptions sont les dossiers créés par l'utilisateur, dont les noms peuvent être altérés après la création.

Nom	Description
Parent	Lecture seule. Extrait le projet dont cet item est un enfant. A la même effect en tant que Application.ActiveProject.
QualifiedName <sup>1389</sup>	Propriétés Extrait le nom qualifié d'un élément de service Web.
WSDLFile <sup>1390</sup>	Propriétés Extrait le nom du fichier du fichier WSDL définissant le service Web qui héberge l'item de projet actuel.

# Méthodes

Nom	Description
AddActiveFile <sup>1390</sup>	Ajoute le document actuellement actif à cet item de projet s'il s'agit d'un enfant valide. Autrement, le document est ajouté au dossier de mappage de la racine de projet.
AddFile <sup>1391</sup>	Ajoute le document spécifié à cet item de projet s'il s'agit d'un enfant valide. Autrement, le document est ajouté au dossier de mappage de la racine de projet.
CreateFolder <sup>1392</sup>	Crée un nouveau dossier en tant qu'enfant de cet item de projet.
CreateMappingForProject	Crée un document de mappage initial pour une opération de service Web et l'enregistre dans <b>i_strFileName</b> . Lorsque vous utilisez Project.InsertWebService, vous pouvez définir l'indicateur <b>i_bGenerateMappings</b> de telle façon que MapForce puisse générer automatiquement les mappages initiaux pour tous les ports.
<u>GenerateCode</u> <sup>1393</sup>	Génère le code pour tous les items de projet et leurs enfants. Le langage de code et l'emplacement de sortie sont déterminés par CodeGenSettings_UseDefault, CodeGenSettings_Language and CodeGenSettings_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.
GenerateCodeEx <sup>1333</sup>	Génère le code pour tous les items de projet et leurs enfants. Le langage de code et l'emplacement de sortie sont déterminés par CodeGenSettings_UseDefault, CodeGenSettings_Language et CodeGenSettings_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.
GenerateCodeIn	Génère le code pour l'élément de projet et ses enfants dans le langage spécifié. L'emplacement de sortie est déterminé par CodeGenSettings_UseDefault et CodeGenSettings_OutputFolder. Les enfants de cet item de

Nom	Description
	projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.
GenerateCodeInEx	Génère le code pour l'élément de projet et ses enfants dans le langage spécifié. L'emplacement de sortie est déterminé par CodeGenSettings_UseDefault et CodeGenSettings_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code. Un objet qui peut être utilisé pour itérer à travers tous les messages publiés par le processus de génération de code est retourné. Ces messages sont les mêmes que ceux affichés dans la fenêtre de messages de MapForce.
Ouvrir <sup>(1393)</sup>	Ouvre l'élément de projet comme document ou rend le document correspondant actif, s'il est déjà ouvert. L'élément de projet doit être un mappage MapForce ou, pour Enterprise edition uniquement, une opération de service Web.
Supprimer <sup>1396</sup>	Supprimer cet élément de projet et tous ses enfants depuis l'arborescence du projet.

# Événements

Nom	Description
OnModifiedFlagChanged <sup>1397</sup>	Apparaît lorsque le statut de modification ProjectItem change.
OnProjectClosed <sup>1397</sup>	Cet événement est déclenché quand le projet est fermé. L'objet du projet passé dans le gestionnaire d'événement ne devrait pas être accédé. L'événement ouvert correspondant est Application.OnProjectOpened.

# 16.7.1.20.1 Propriétés

# 16.7.1.20.1.1 \_NewEnum

Cette propriété prend en charge une énumération standard spécifique au langage. Pour des exemples, voir Project.Item OU Project.\_NewEnum.

# Signature

\_NewEnum : IUnknown

Code d'erreur :	Description
1700	L'objet n'est plus valide.

# 16.7.1.20.1.2 Application

Extrait l'objet de l'application de premier niveau.

# Signature

Application : Application

#### Erreurs

Code d'erreur :	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de retour a été spécifiée.

# 16.7.1.20.1.3 CodeGenSettings\_Language

Obtient ou définit le langage à utiliser avec GenerateCode ou Project.GenerateCode. Cette propriété est consultée uniquement si CodeGenSettings\_UseDefault est défini comme false.

### Signature

CodeGenSettings\_Language : ENUMProgrammingLanguage

#### Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	Le langage invalide ou l'adresse invalide pour le paramètre de renvoi a été spécifié.

# 16.7.1.20.1.4 CodeGenSettings\_OutputFolder

Obtient ou définit le répertoire de sortie à utiliser avec GenerateCode, GenerateCodeIn, Project.GenerateCode OU Project.GenerateCodeIn. Cette propriété est consultée uniquement si CodeGenSettings\_UseDefault est défini comme false.

## Signature

CodeGenSettings\_OutputFolder : String

## Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	Le langage invalide ou l'adresse invalide pour le paramètre de renvoi a été spécifié.

# 16.7.1.20.1.5 CodeGenSettings\_UseDefault

Obtient ou définit si le répertoire de sortie et le langage de code sont utilisés tels que définis soit par (a) les dossiers parents, ou (b) la racine du projet. Cette propriété est utilisée avec des appels pour GenerateCode, GenerateCodeIn, Project.GenerateCode et Project.GenerateCodeIn. Si la propriété est définie en tant que false, les valeurs de CodeGenSettings\_OutputFolder et CodeGenSettings\_Language sont utilisées pour générer le code pour cet item de projet.

# Signature

CodeGenSettings\_UseDefault : Booléen

### Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.20.1.6 Count

Extrait le nombre d'enfants de l'item racine du projet. Pour des exemples, voir Item ou \_NewEnum

### Signature

Count : Entier		
	Count : Entier	Count : Entier

Code d'erreur	Description
1500	L'objet n'est plus valide.

### 16.7.1.20.1.7 Item

Retourne l'enfant à la position *n* de la racine du projet. L'index est basé sur 1 (le premier index est 1). L'index le plus grand valide est Count. Pour une alternative, voir \_NewEnum.

#### Signature

Item(in n:Integer) : ProjectItem

#### Erreurs

Code d'erreur	Description
1500	L'objet n'est plus valide.

#### Exemples

```
// -----
// JScript code snippet - énumérer les enfants en utilisant Count et Item.
for( nItemIndex = 1; nItemIndex <= objProject.Count; nItemIndex++ )
{
    objProjectItem = objProject.Item(nItemIndex);
    // faire quelque chose avec l'item de projet ici
}</pre>
```

# 16.7.1.20.1.8 Genre

Extrait le genre d'élément de projet. La disponibilité de quelques propriétés et l'applicabilité de certaines méthodes sont restreintes aux genres spécifiques d'éléments de projet. La description de toutes les méthodes et propriétés contient des informations sur ces restrictions.

#### Signature

Genre : ENUMProjectItemType

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.20.1.9 Nom

Extrait ou définit le nom d'un élément de projet. Le nom de la plupart des éléments est en lecture seule. Les exceptions sont les dossiers créés par l'utilisateur, dont les noms peuvent être altérés après la création.

# Signature

Nom : String

#### Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1702	L'élément de projet ne permet pas d'altérer son nom.

# 16.7.1.20.1.10 Parent

L'objet parent selon le modèle d'objet.

### Signature

Parent : Application

### Erreurs

Code d'erreur	Description
1400	L'objet n'est plus valide.
1401	L'adresse invalide pour le paramètre de renvoi a été spécifiée.

# 16.7.1.20.1.11 QualifiedName

Extrait le nom qualifié d'un élément de service Web.

#### Signature

QualifiedName : String

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1702	L'élément de projet ne fait pas partie du service Web.

# 16.7.1.20.1.12 WSDLFile

Extrait le nom du fichier du fichier WSDL définissant le service Web qui héberge l'item de projet actuel.

### Signature

WSDLFile : String

# Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1702	L'élément de projet ne fait pas partie du service Web.

# 16.7.1.20.2 Méthodes

# 16.7.1.20.2.1 AddActiveFile

Ajoute le document actuellement actif à cet item de projet s'il s'agit d'un enfant valide. Autrement, le document est ajouté au dossier de mappage de la racine de projet.

### Signature

AddActiveFile() -> ProjectItem

Code d'erreur :	Description
1700	L'objet n'est plus valide.

Code d'erreur :	Description
1701	Le nom du fichier est vide. L'adresse invalide pour le paramètre de retour a été spécifiée.
1703	Aucun document valide n'est disponible.
1704	Un nom de chemin doit être attribué aux documents actifs avant qu'il ne puisse être ajouté au projet.
1705	Le mappage n'a pas pu être attribué au projet. Le fichier n'existe pas ou n'est pas un mappage de MapForce. Peut-être que le fichier est déjà attribué au dossier cible.

# 16.7.1.20.2.2 AddFile

Ajoute le document spécifié à cet item de projet s'il s'agit d'un enfant valide. Autrement, le document est ajouté au dossier de mappage de la racine de projet.

# Signature

```
AddFile(dans i_strFileName:String) -> ProjectItem
```

# Paramètres

Nom	Туре	Description
i_strFilePath	String	Le chemin du document à ajouter.

Code d'erreur :	Description
1700	L'objet n'est plus valide.
1701	Le nom du fichier est vide. L'adresse invalide pour le paramètre de retour a été spécifiée.
1705	Le mappage n'a pas pu être attribué au projet. Le fichier n'existe pas ou n'est pas un mappage de MapForce. Peut-être que le fichier est déjà attribué au dossier cible.

# 16.7.1.20.2.3 CreateFolder

Crée un nouveau dossier en tant qu'enfant de cet item de projet.

### Signature

CreateFolder(dans i\_strFileName:String) -> ProjectItem

### Paramètres

Nom	Туре	Description
i_strFolderName	String	Le nom du dossier à créer.

### Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	Le nom du dossier invalide ou l'adresse invalide pour le paramètre de renvoi a été spécifié.
1702	L'item de projet ne prend pas en charge les enfants.

# 16.7.1.20.2.4 CreateMappingForProject

Crée un document de mappage initial pour une opération de service Web et l'enregistre dans **i\_strFileName**. Lorsque vous utilisez Project.InsertWebService, vous pouvez définir l'indicateur **i\_bGenerateMappings** de telle façon que MapForce puisse générer automatiquement les mappages initiaux pour tous les ports.

#### Signature

```
CreateMappingForProject(dans i_strFileName:String) -> ProjectItem
```

### Paramètres

Nom	Туре	Description
i_strFileName	String	Spécifie le chemin dans lequel le mappage devrait être enregistré.

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1707	Ne peut pas créer de nouveau mappage. L'item de projet ne prend pas en charge la création automatique de mappages initiaux ou un mappage existe déjà.
1708	L'opération n'est pas prise en charge dans l'édition actuelle.

# 16.7.1.20.2.5 GenerateCode

Génère le code pour tous les items de projet et leurs enfants. Le langage de code et l'emplacement de sortie sont déterminés par CodeGenSettings\_UseDefault, CodeGenSettings\_Language and CodeGenSettings\_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.

# Signature

GenerateCode() -> **Void** 

### Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1706	Erreur pendant la génération de code.

# 16.7.1.20.2.6 GenerateCodeEx

Génère le code pour tous les items de projet et leurs enfants. Le langage de code et l'emplacement de sortie sont déterminés par CodeGenSettings\_UseDefault, CodeGenSettings\_Language and CodeGenSettings\_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.

### Signature

GenerateCodeEx() -> ErrorMarkers

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1706	Erreur pendant la génération de code.

# 16.7.1.20.2.7 GenerateCodeIn

Génère le code pour l'élément de projet et ses enfants dans le langage spécifié. L'emplacement de sortie est déterminé par CodeGenSettings\_UseDefault et CodeGenSettings\_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.

### Signature

GenerateCodeIn(**dans i\_nLanguage**: ENUMProgrammingLanguage (1402) -> Void

### Paramètres

Nom	Туре	Description
i_nLanguage	ENUMProgrammingLanguage <sup>1402</sup>	Spécifie le langage de programmation pour la génération de code.

# Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	Le langage invalide est spécifié.
1706	Erreur pendant la génération de code.

# 16.7.1.20.2.8 GenerateCodeInEx

Génère le code pour l'élément de projet et ses enfants dans le langage spécifié. L'emplacement de sortie est déterminé par CodeGenSettings\_UseDefault et CodeGenSettings\_OutputFolder. Les enfants de cet item de projet peuvent avoir leurs propres paramètres de propriété associés à la génération-code.

Un objet qui peut être utilisé pour itérer à travers tous les messages publiés par le processus de génération de code est retourné. Ces messages sont les mêmes que ceux affichés dans la fenêtre de messages de MapForce.

# Signature

GenerateCodeInEx(in i\_nLanguage:ENUMProgrammingLanguage<sup>1402</sup>) -> ErrorMarkers

### Paramètres

Nom	Туре	Description
i_nLanguage	ENUMProgrammingLanguage <sup>1402</sup>	Spécifie le langage de programmation pour la génération de code.

### Erreurs

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	Le langage invalide spécifié ou l'adresse invalide pour le paramètre de renvoi a été spécifié.
1706	Erreur pendant la génération de code.

# 16.7.1.20.2.9 Ouvrir

Ouvre l'élément de projet comme document ou rend le document correspondant actif, s'il est déjà ouvert. L'élément de projet doit être un mappage MapForce ou, pour Enterprise edition uniquement, une opération de service Web.

### Signature

Ouvrir() -> Document

Code d'erreur	Description
1700	L'objet n'est plus valide.
1701	L'adresse invalide pour le paramètre de renvoi a été spécifiée.
1702	L'élément de projet ne se réfère pas à un fichier de mappage MapForce.
1708	L'opération n'est pas prise en charge dans l'édition actuelle.

# 16.7.1.20.2.10 Supprimer

Supprime une référence bibliothèque depuis la fenêtre Gérer les Bibliothèques.

# Signature

Remove(in i\_strFileName:String) -> Void

# Paramètres

Nom	Туре	Description
i_strFileName	String	Le chemin du fichier bibliothèque à supprimer. Veuillez noter que le chemin doit refléter exactement le statut actuel (le plus récent) de l'objet LibraryImport. Rappelez-vous que le chemin peut être relatif ou absolu, et il peut avoir changé si vous avez enregistré le document, dépendant de l'indicateur LibraryImport.SaveRelativePath. Pour cette raison, si l'objet LibraryImport contient actuellement un chemin relatif, alors vous devriez fournir un chemin relatif comme valeur de ce paramètre. Autrement, la bibliothèque n'est pas trouvée et la méthode Supprimer échoue.
		Ce qui est expliqué ci-dessus ne s'applique que pour les bibliothèques importées localement. En ce qui concerne les bibliothèques importées globalement, le chemin doit toujours être absolu (puisque les bibliothèques importées globalement ne peuvent pas avoir de chemin relatif)

Code d'erreur	Description
2400	L'objet n'est plus valide.

# 16.7.1.20.3 Événements

# 16.7.1.20.3.1 OnModifiedFlagChanged

Apparaît lorsque le statut de modification ProjectItem change.

### Signature

OnModifiedFlagChanged(dans i\_bIsModified:Boolean) : Annuler

# 16.7.1.20.3.2 OnProjectClosed

Cet événement est déclenché quand le projet est fermé. L'objet du projet passé dans le gestionnaire d'événement ne devrait pas être accédé. L'événement ouvert correspondant est Application.OnProjectOpened.

### Signature

```
OnProjectClosed(in i_ipProject:Project) : Annuler
```

# 16.7.2 Énumérations

# 16.7.2.1 ENUMApacheAxisVersion (obsolete)

Ce type d'énumération est obsolète.

### Membres

eApacheAxisVersion\_Axis = 1

```
eApacheAxisVersion_Axis2 = 2
```

# 16.7.2.2 ENUMApplicationStatus

Valeurs d'énumération pour indiquer le statut de l'application.

### Membres

```
eApplicationRunning = 0
```

eApplicationAfterLicenseCheck = 1

eApplicationBeforeLicenseCheck = 2

eApplicationConcurrentLicenseCheckFailed = 3

```
eApplicationProcessingCommandLine = 4
```

# 16.7.2.3 ENUMAppOutputLine\_Severity

Valeurs d'énumération pour identifier la gravité de AppOutputLine.

# Membres

eSeverity\_Undefined = -1  $eSeverity_Info = 0$ eSeverity\_Warning = 1 eSeverity\_Error = 2 eSeverity\_CriticalError = 3 eSeverity\_Success = 4 eSeverity\_Summary = 5  $eSeverity_Progress = 6$ eSeverity\_DataEdit = 7 eSeverity\_ParserInfo = 8 eSeverity\_PossibleInconsistencyWarning = 9 eSeverity\_Message = 10 eSeverity\_Document = 11 eSeverity\_Rest = 12 eSeverity\_NoSelect = 13 eSeverity\_Select = 14 eSeverity\_Autoinsertion = 15 eSeverity\_GlobalResources\_DefaultWarning = 16 eSeverity\_XPath\_Styles\_Changed = 17 eSeverity\_XPath\_Styles\_Unchanged = 18 eSeverity\_XPath\_Styles\_Skipped = 19 eSeverity\_XPath\_ComboBox\_Values\_Changed = 20 eSeverity\_XPath\_ComboBox\_Values\_Unchanged = 21 eSeverity\_XPath\_ComboBox\_Values\_Skipped = 22 eSeverity\_XPath\_Assertions\_Changed = 23 eSeverity\_XPath\_Assertions\_Unchanged = 24 eSeverity\_XPath\_Assertions\_Skipped = 25

# 16.7.2.4 ENUMAppOutputLine\_TextDecoration

Valeurs d'énumération de différents types de décoration de texte d'un AppOutputLine.

# Membres

eTextDecorationDefault = 0 eTextDecorationBold = 1 eTextDecorationDebugValues = 2 eTextDecorationDB\_ObjectName = 3 eTextDecorationDB\_ObjectLink = 4 eTextDecorationDB\_ObjectKind = 5 eTextDecorationDB\_TimeoutValue = 6 eTextDecorationFind\_MatchingString = 7 eTextDecorationValidation\_Speclink = 8 eTextDecorationValidation\_ErrorPosition = 9 eTextDecorationValidation\_UnkownParam = 10

# 16.7.2.5 ENUMCodeGenErrorLevel

Valeurs d'énumération pour identifier la gravité des messages de génération de code.

# Membres

eCodeGenErrorLevel\_Information = 0

 $eCodeGenErrorLevel_Warning = 1$ 

 $eCodeGenErrorLevel_Error = 2$ 

 $eCodeGenErrorLevel_Undefined = 3$ 

# 16.7.2.6 ENUMComponentDatapointSide

Valeurs d'énumération pour indiquer le côté d'un datapoint sur son composant. Voir également Component.GetRootDatapoint.

#### Membres

eDatapointSideInput = 0

eDatapointSideOutput = 1

# 16.7.2.7 ENUMComponentSubType

Valeurs d'énumération pour indiquer les sous-types de composant.

### Membres

eComponentSubType\_None = 0

eComponentSubType\_Text\_EDI = 1

eComponentSubType\_Text\_Flex = 2

eComponentSubType\_Text\_CSVFLF = 3

# 16.7.2.8 ENUMComponentType

Valeurs d'énumération pour indiquer les types de composant.

### Membres

- eComponentType\_Unknown = 0
- eComponentType\_XML = 1
- $eComponentType_DB = 2$
- $eComponentType_Text = 3$
- eComponentType\_Excel = 4
- eComponentType\_WSDL = 5
- eComponentType\_XBRL = 6
- eComponentType\_Input = 7
- eComponentType\_JSON = 8

# 16.7.2.9 ENUMComponentUsageKind

Valeurs d'énumération pour indiquer le type d'utilisation du composant indiqué.

# Membres

- eComponentUsageKind\_Unknown = 0
- eComponentUsageKind\_Instance = 1
- eComponentUsageKind\_Input = 2
- eComponentUsageKind\_Output = 3
- eComponentUsageKind\_Variable = 4
- eComponentUsageKind\_String = 5

# 16.7.2.10 ENUMConnectionType

Valeurs d'énumération pour indiquer le type de connexion. Voir également Connection.ConnectionType.

# Membres

eConnectionTypeTargetDriven = 0 eConnectionTypeSourceDriven = 1 eConnectionTypeCopyAll = 2

# 16.7.2.11 ENUMDOMType

Valeurs d'énumération pour spécifier le type de DOM utilisé par le code de mappage C++ généré.

NOTE : La valeur eDOMType\_xerces est obsolète. eDOMType\_xerces3 indique l'utilisation Xerces 3.x. Obsolète dans ce contexte signifie que cette valeur n'est pas prise en charge et ne devrait pas être utilisée.

### Membres

eDOMType\_xerces = 1 (obsolete) eDOMType xerces3= 2

eDOMType\_msxml6= 3

# 16.7.2.12 ENUMLibType

Valeurs d'énumération pour spécifier le type de bibliothèque utilisé par le code de mappage C++.

### Membres

```
eLibType_static = 0
eLibType_dll = 1
```

# 16.7.2.13 ENUMProgrammingLanguage

Enumeration values to select a programming language.

### Members

```
eUndefinedLanguage = -1
eJava = 0
eCpp = 1
eCSharp = 2
eXSLT = 3
eXSLT2 = 4
eXQuery = 5
```

# 16.7.2.14 ENUMProjectItemType

Énumération pour identifier les différents types d'items de projet qui peuvent être des enfants de Project ou du dossier similaire ProjectItems. Voir également ProjectItem.Kind.

### Membres

- eProjectItemType\_MappingFolder = 0
- eProjectItemType\_Mapping = 1
- eProjectItemType\_WebServiceFolder = 2
- eProjectItemType\_WebServiceRoot = 3
- eProjectItemType\_WebServiceService = 4
- eProjectItemType\_WebServicePort = 5

eProjectItemType\_WebServiceOperation = 6 eProjectItemType\_ExternalFolder = 7 eProjectItemType\_LibraryFolder = 8 eProjectItemType\_ResourceFolder = 9 eProjectItemType\_VirtualFolder = 10 eProjectItemType\_Count = 11 eProjectItemType\_Invalid = -1

# 16.7.2.15 ENUMProjectType

Enumeration values to select a project type for generated C# and C++ mapping code.

# Members

- eVisualStudio2010Project = 6
- eVisualStudio2013Project = 7
- eVisualStudio2015Project = 8
- eVisualStudio2017Project = 9
- eVisualStudio2019Project = 10

# 16.7.2.16 ENUMSearchDatapointFlags

Valeurs d'énumération utilisées comme bits indicateurs ; à être utilisé comme combinaison d'indicateurs lors de la recherche d'un datapoint. Voir aussi GetChild.

# Membres

- eSearchDatapointElement = 1
- eSearchDatapointAttribute = 2

# 16.7.2.17 ENUMViewMode

Valeurs d'énumération pour sélectionner une vue MapForce.

# Membres

eMapForceView = 0

eXSLView = 1

eOutputView = 2
# 17 ActiveX Integration

The MapForce user interface and the functionality described in this section can be integrated into custom applications that can consume ActiveX controls. ActiveX technology enables a wide variety of languages to be used for integration, such as C++, C#, VB.NET, HTML. (Note that ActiveX components integrated in HTML must be run with Microsoft Internet Explorer versions and platforms that support ActiveX). All components are full OLE Controls. Integration into Java is provided through wrapper classes.

To integrate the ActiveX controls into your custom code, the MapForce Integration Package must be installed (see <u>https://www.altova.com/components/download</u>). Ensure that you install MapForce first, and then the MapForce Integration Package. Other prerequisites apply, depending on language and platform (see <u>Prerequisites</u><sup>1406</sup>).

You can flexibly choose between two different levels of integration: application level and document level.

Integration at application level means embedding the complete interface of MapForce (including its menus, toolbars, panes, etc) as an ActiveX control into your custom application. For example, in the most simple scenario, your custom application could consist of only one form that embeds the MapForce graphical user interface. This approach is easier to implement than integration at document level but may not be suitable if you need flexibility to configure the MapForce graphical user interface according to your custom requirements.

Integration at document level means embedding MapForce into your own application piece-by-piece. This includes implementing not only the main MapForce control but also the main document editor window, and, optionally, any additional windows. This approach provides greater flexibility to configure the GUI, but requires advanced interaction with ActiveX controls in your language of choice.

The sections Integration at the Application Level<sup>[410]</sup> and Integration at Document Level<sup>[413]</sup> describe the key steps at these respective levels. The <u>ActiveX Integration Examples</u><sup>[417]</sup> section provides examples in C#, HTML, and Java. Looking through these examples will help you to make the right decisions quickly. The <u>Object</u> <u>Reference</u><sup>[453]</sup> section describes all COM objects that can be used for integration, together with their properties and methods.

For information about using MapForce as a Visual Studio plug-in, see <u>MapForce in Visual Studio</u><sup>(00)</sup>.

## 17.1 **Prerequisites**

To integrate the MapForce ActiveX control into a custom application, the following must be installed on your computer:

- MapForce
- The MapForce Integration Package, available for download at <u>https://www.altova.com/components/download</u>

To integrate the 64-bit ActiveX control, install the 64-bit versions of MapForce and MapForce Integration Package. For applications developed under Microsoft .NET platform with Visual Studio, both the 32-bit and 64bit versions of MapForce and MapForce Integration Package must be installed, as explained below.

### Microsoft .NET (C#, VB.NET) with Visual Studio

To integrate the MapForce ActiveX control into a 32-bit application developed under Microsoft .NET, the following must be installed on your computer:

- Microsoft .NET Framework 4.0 or later
- Visual Studio 2012/2013/2015/2017/2019/2022
- MapForce 32-bit and MapForce Integration Package 32-bit
- The ActiveX controls must be added to the Visual Studio toolbox (see <u>Adding the ActiveX Controls to</u> the <u>Toolbox</u><sup>[400]</sup>).

If you want to integrate the 64-bit ActiveX control, the following prerequisites apply in addition to the ones above:

- MapForce 32-bit and MapForce Integration Package 32-bit must still be installed (this is required to provide the 32-bit ActiveX control to the Visual Studio designer, since Visual Studio runs on 32-bit)
- MapForce 64-bit and MapForce Integration Package 64-bit must be installed (provides the actual 64-bit ActiveX control to your custom application at runtime)
- In Visual Studio, create a 64-bit build configuration and build your application using this configuration.
   For an example, see <u>Running the Sample C# Solution<sup>[417]</sup></u>.

#### Java

To integrate the MapForce ActiveX control into Java application using the Eclipse development environment, the following must be installed on your computer:

- Java Runtime Environment (JRE) or Java Development Kit (JDK) 7 or later
- Eclipse
- MapForce and MapForce Integration Package
- **Note:** To run the 64-bit version of the MapForce ActiveX control, use a 64-bit version of Eclipse, as well as the 64-bit version of MapForce and the MapForce Integration Package.

#### MapForce integration and deployment on client computers

If you create a .NET application and intend to distribute it to other clients, you will need to install the following on the client computer(s):

- MapForce •
- The MapForce Integration PackageThe custom integration code or application.

# 17.2 Adding the ActiveX Controls to the Toolbox

To use the MapForce ActiveX controls in an application developed with Visual Studio, the controls must first be added to the Visual Studio Toolbox, as follows:

- 1. On the Tools menu of Visual Studio, click Choose Toolbox Items.
- 2. On the **COM Components** tab, select the check boxes next to the MapForceControl, MapForceControl Document, and MapForceControl Placeholder.

In case the controls above are not available, follow the steps below:

- On the COM Components tab, click Browse, and select the MapForceControl.ocx file from the MapForce installation folder. Remember that the MapForce Integration Package must be installed; otherwise, this file is not available, see <u>Prerequisites</u><sup>(406)</sup>.
- 2. If prompted to restart Visual Studio with elevated permissions, click **Restart under different** credentials.

Microso	oft Visual Studio	×
+	This task requires the application to have elevated permissions.	
	Why is using the Administrator or other account necessary?	
	Restart under different credentials Saves the current changes and then restarts Microsoft Visual Studio. You will be prompted to change your user account.	
	Cancel the task and return to Microsoft Visual Studio	
🕑 V	iew error <u>i</u> nformation Can	cel

If the steps above were successful, the MapForce ActiveX controls become available in the Visual Studio Toolbox.

Toolbo	ĸ	$\bullet \Box \times$			
Search	Toolbox	<del>،</del> م			
⊳ All W	/indows Forms				
▷ Com	mon Controls				
▷ Cont	ainers				
🔺 Men	us & Toolbars				
k	Pointer				
<u>*</u> =	ContextMenuStrip				
	MenuStrip				
	🕳 StatusStrip				
:51	🔤 ToolStrip				
	ToolStripContainer				
(2)	MapForceControl				
$\odot$	MapForceControl Doc	ument			
	MapForceControl Plac	eHolder			
Data					
▷ Components					
▷ Printing					
▷ Dialogs					
WPF Interoperability					
▷ Gene	> General				

**Note:** For an application-level integration, only the **MapForceControl** ActiveX control is used (see <u>Integration</u> <u>at Application Level</u><sup>(410)</sup>). The **MapForceControl Document** and **MapForceControl Placeholder** controls are used for document-level integration (see <u>Integration at Document Level</u><sup>(413)</sup>).

# **17.3** Integration at Application Level

Integration at application level allows you to embed the complete interface of MapForce into a window of your application. With this type of integration, you get the whole user interface of MapForce, including all menus, toolbars, the status bar, document windows, and helper windows. Customization of the application's user interface is restricted to what MapForce provides. This includes rearrangement and resizing of helper windows and customization of menus and toolbars.

The only ActiveX control you need to integrate is <u>MapForceControl</u><sup>[456]</sup>. Do not instantiate or access <u>MapForceControlDocument</u><sup>[469]</sup> or <u>MapForceControlPlaceHolder</u><sup>[470]</sup> ActiveX controls when integrating at application-level.

If you have any initialization to do or if you want to automate some behaviour of MapForce, use the properties, methods, and events described for <u>MapForceControl</u><sup>[457]</sup>. Consider using <u>MapForceControl.Application</u><sup>[457]</sup> for more complex access to MapForce functionality.

For an example that shows how the MapForce application can be embedded in an HTML page, see <u>HTML</u> <u>Integration at Application Level</u><sup>(423)</sup>.

In C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the MapForce ActiveX controls at application level are as follows:

- 1. Check that all prerequisites are met (see <u>Prerequisites</u><sup>1406</sup>).
- 2. Create a new Visual Studio Windows Forms project with a new empty form.
- 3. If you have not done that already, add the ActiveX controls to the toolbox (see <u>Adding the ActiveX</u> <u>Controls to the Toolbox</u><sup>[403</sup>).
- 4. Drag the **MapForceControl** from the toolbox onto your new form.
- 5. Select the **MapForceControl** on the form, and, in the Properties window, set the **IntegrationLevel** property to **ICActiveXIntegrationOnApplicationLevel**.

Properties 🔹 🗖 🗙						
axMapForceControl AxM	axMapForceControl AxMapForceControlLib.AxMapForceControl					
🗄 🛃 🖗 🖉						
	A					
(Name)	axMapForceControl					
AccessibleDescription						
AccessibleName						
AccessibleRole	Default					
AllowDrop	False					
Anchor	Top, Left					
Appearance	0					
BorderStyle	0 - None					
CausesValidation	True					
ContextMenuStrip	(none)					
Dock	None					
EnableUserPrompts	True					
GenerateMember	True					
IntegrationLevel	ICActiveXIntegrationOnApplicationLevel					
E Location	280, 8					
Locked	False					
	3, 3, 3, 3					

- 6. Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:
  - a. Right-click the solution in Visual Studio, and select Configuration Manager.
  - b. Under Active solution platform, select New... and then select the x86 or x64 configuration (in this example, x86).

New Solution Platform				
Type or select the new <u>p</u> latform:				
x86 🗸				
Copy <u>s</u> ettings from:				
Any CPU 👻				
✓ Create new project platforms				
OK Cancel				

You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64).

# 17.4 Integration at Document Level

Compared to integration at application level, integration at document level is a more complex, yet more flexible way to embed MapForce functionality into your application by means of ActiveX controls. With this approach, your code can access selectively the following parts of the MapForce user interface:

- Document editing window
- Project window
- Libraries window
- Overview window
- Messages window

As mentioned in <u>Integration at Application Level</u><sup>[410]</sup>, for an ActiveX integration at application level, only one control is required, namely the **MapForceControl**. However, for an ActiveX integration at document level, MapForce functionality is provided by the following ActiveX controls:

- MapForceControl
   <sup>1456</sup>
- <u>MapForceControl Document</u><sup>1464</sup>
- MapForceControl Placeholder<sup>1470</sup>

These controls are supplied by the **MapForceControl.ocx** file available in the application installation folder of MapForce. When you develop the ActiveX integration with Visual Studio, you will need to add these controls to the Visual Studio toolbox (see <u>Adding the ActiveX Controls to the Toolbox</u><sup>(408)</sup>).

The basic steps to integrate the ActiveX controls at document level into your application are as follows:

- First, instantiate MapForceControl in your application. Instantiating this control is mandatory; it enables support for the MapForceControl Document and MapForceControl Placeholder controls mentioned above. It is important to set the <u>IntegrationLevel</u> [459] property to ICActiveXIntegrationOnDocumentLevel (or "1"). To hide the control from the user, set its Visible property to False.
- **Note:** When integrating at document level, do not use the **Open** method of the **MapForceControl**; this might lead to unexpected results. Use the corresponding open methods of **MapForceControl Document** and **MapForceControl PlaceHolder** instead.
  - 2. Create at least one instance of **MapForceControl Document** in your application. This control supplies the document editing window of MapForce to your application and can be instantiated multiple times if necessary.

Use the method **Open** to load any existing file. To access document-related functionality, use the **Path** and **Save** or methods and properties accessible via the property **Document**.

- Note: The control does not support a read-only mode. The value of the property ReadOnly is ignored.
  - 3. Optionally, add to your application the **MapForceControl Placeholder** control for each additional window (other than the document window) that must be available to your application.

Instances of **MapForceControl PlaceHolder** allow you to selectively embed additional windows of MapForce into your application. The window kind (for example, Project window) is defined by the

property **PlaceholderWindowID**. Therefore, to set the window kind, set the property **PlaceholderWindowID**. For valid window identifiers, see <u>MapForceControlPlaceholderWindow</u><sup>[473]</sup>.

#### Note: Use only one MapForceControl PlaceHolder for each window identifier.

For placeholder controls that select the MapForce project window, additional methods are available. Use **OpenProject** to load a MapForce project. Use the property Project and the methods and properties from the MapForce automation interface to perform any other project related operations.

For example, in C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the MapForce ActiveX controls at document level could be similar to those listed below. Note that your application may be more complex if necessary; however, the instructions below are important to understand the minimum requirements for an ActiveX integration at document level.

- 1. Create a new Visual Studio Windows Forms project with a new empty form.
- 2. If you have not done that already, add the ActiveX controls to the toolbox (see <u>Adding the ActiveX</u> <u>Controls to the Toolbox</u><sup>(403)</sup>).
- 3. Drag the <u>MapForceControl</u><sup>[456]</sup> from the toolbox onto your new form.
- Set the IntegrationLevel property of the MapForceControl to ICActiveXIntegrationOnDocumentLevel, and the Visible property to False. You can do this either from code or from the Properties window.
- Drag the <u>MapForceControl Document</u> <sup>(MB)</sup> from the toolbox onto the form. This control provides the main document window of MapForce to your application, so you may need to resize it to a reasonable size for a document.
- 6. Optionally, add one or more <u>MapForceControl Placeholder</u><sup>(470)</sup> controls to the form (one for each additional window type that your application needs, for example, the **Project** window). You will typically want to place such additional placeholder controls either below or to the right or left of the main document control, for example:



- 7. Set the **PlaceholderWindowID** property of each **MapForceControl Placeholder** control to a valid window identifier. For the list of valid values, see <u>MapForceControlPlaceholderWindow</u><sup>(473)</sup>.
- 8. Add commands to your application (at minimum, you will need to open, save and close documents), as shown below.

## Querying MapForce Commands

When you integrate at document level, no MapForce menu or toolbar is available to your application. Instead, you can retrieve the required commands, view their status, and execute them programmatically, as follows:

- To retrieve all available commands, use the CommandsList <sup>[457]</sup> property of the **MapForceControl**.
- To retrieve commands organized according to their menu structure, use the MainMenu<sup>1450</sup> property.
- To retrieve commands organized by the toolbar in which they appear, use the **Toolbars** to property.
- To send commands to MapForce, use the **Exec**<sup>[460]</sup> method.
- To query if a command is currently enabled or disabled, use the <u>QueryStatus</u><sup>[46]</sup> method.

This enables you to flexibly integrate MapForce commands into your application's menus and toolbars.

Your installation of MapForce also provides you with command label images used within MapForce. See the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your MapForce installation for icons in GIF format. The file names correspond to the command names as they are listed in the <u>Command Reference</u> section.

#### General considerations

To automate the behaviour of MapForce, use the properties, methods, and events described for the <u>MapForceControl</u><sup>[453]</sup>, <u>MapForceControl Document</u><sup>[463]</sup>, and <u>MapForceControl Placeholder</u><sup>[470]</sup>.

For more complex access to MapForce functionality, consider using the following properties:

- MapForceControl.Application<sup>[457</sup>
- MapForceControlDocument.Document
   <sup>1465</sup>
- MapForceControlPlaceHolder.Project (1471)

These properties give you access to the MapForce automation interface (MapForceAPI)

**Note:** To open a document, always use <u>MapForceControlDocument.Open</u><sup>[467]</sup> or <u>MapForceControlDocument.New</u><sup>[466]</sup> on the appropriate document control. To open a project, always use <u>MapForceControlPlaceHolder.OpenProject</u><sup>[47]</sup> on a placeholder control embedding a MapForce project window.

For examples that show how to instantiate and access the necessary controls in different programming environments, see <u>ActiveX Integration Examples</u><sup>[417]</sup>.

# 17.5 ActiveX Integration Examples

This section contains examples of MapForce document-level integration using different container environments and programming languages. Source code for all examples is available in the folder <a href="https://www.applicationFolder-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexamples-lexam

# 17.5.1 C#

A basic ActiveX integration example solution for C# and Visual Studio is available in the folder <ApplicationFolder>\Examples\ActiveX\C#. Before you compile the source code and run the sample, make sure that all prerequisites are met (see <u>Running the Sample C# Solution</u><sup>[41]</sup>).

## 17.5.1.1 Running the Sample C# Solution

The sample Visual Studio solution available in the folder **<ApplicationFolder>\Examples\ActiveX\C#** illustrates how to consume the MapForce ActiveX controls. Before attempting to build and run this solution, note the following steps:

### Step 1: Check the prerequisites

Visual Studio 2010 or later is required to open the sample solution. For the complete list of prerequisites, see <u>Prerequisites</u><sup>[406]</sup>.

### Step 2: Copy the sample to a directory where you have write permissions

To avoid running Visual Studio as an Administrator, copy the source code to a directory where you have write permissions, instead of running it from the default location.

## Step 3: Check and set all required control properties

The sample application contains one instance of <u>MapForceControlDocument</u><sup>[463]</sup> and several instances of <u>MapForceControlPlaceHolder</u><sup>[470]</sup> controls. Double-check that the following properties of these controls are set as shown in the table below:

Control name	Property	Property value
axMapForceControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLe vel
axMapForceControlLibrary	PlaceholderWindowID	0
axMapForceControlOutput	PlaceholderWindowID	2
axMapForceControlPreview	PlaceholderWindowID	1

Here is how you can view or set the properties of an ActiveX control:

1. Open the **MDIMain.cs** form in the designer window.

**Note:** On 64-bit Windows, it may be necessary to change the build configuration of the Visual Studio solution to "x86" **before** opening the designer window. If you need to build the sample as a 64-bit application, see <u>Prerequisites</u><sup>[406]</sup>.



2. Open the **Document Outline** window of Visual Studio (On the **View** menu, click **Other Windows** | **Document Outline**).



3. Click an ActiveX control in the **Document Outline** window, and edit its required property in the **Properties** window, for example:

Pr	Properties 🝷 🗖 🗙				
ах	MapForceControl AxMapForceC	ontrolLib.AxMapForceControl			
	🛃 🔁 F 🖉				
	(Name)	axMapForceControl			
	AccessibleDescription				
	AccessibleName				
	AccessibleRole	Default			
	AllowDrop	False			
	Anchor	Top, Left			
	Appearance	0			
	BorderStyle	0 - None			
	CausesValidation	True			
	ContextMenuStrip	(none)			
	Dock	None			
	EnableUserPrompts	True			
	GenerateMember	True			
	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel 🗸			
Ŧ	Location	280, 8 👻			

IntegrationLevel

#### Step 4: Set the build platform

- Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:
  - a. Right-click the solution in Visual Studio, and select Configuration Manager.
  - b. Under Active solution platform, select New... and then select the x86 or x64 configuration (in this example, x86).

New Solution Platform
Type or select the new <u>p</u> latform:
×86 🗸
Copy <u>s</u> ettings from:
Any CPU 👻
✓ Create new project platforms
OK Cancel

You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64); otherwise, runtime errors might occur.

On running the sample, the main MDI Frame window is displayed. Use **File | Open** to open a mapping file (for example, MarketingExpenses.mfd, which is in the MapForce examples folder). The file is loaded and displayed in a new document child window:

MapForceControl								
File Edit Insert Component	Connection	Function Ou	put Debug	View	Tools	Help	Window	
F core								
- aggregate functions		C:\User	\altova\Docu	ments\			×	
avo	result = avo(						<u> </u>	
count	result = cour							
max	result = maxi							
max-string	result = maxi	ExpRep	rt					
min	result = min(	🛛 🗋 File: I	xpReport.xm	File/Stri	ing [>			
min-string	result = min(	🔁 🖯 exp	ense-report	expense-	repol>			
string-join	result = strin	> = d	tailed		¢	C= ""		
sum	result = sumi	>   = ci	rrency		¢			
		> = to	tal-sum		¢	_ /		
and a conversion functions		> ⊕ () P	rson		-		<del>/</del>	
boolean	result = book	>  −-O	irst		- I			
format-date	result = form	>  −-O	.ast		- P-			
format-dateTime	result = form	>  O	Title				- <u>-</u>	
format-number	result = form							
format-time	result = form	Mapping	DB Query	Out	put 🤅	🗿 HTML	e e	
number	result = numi							
string	result = strin							
file path functions								
get-fileext	extension = (							
get-folder	folder = get-t							
main-mfd-filepath	filepath = ma							
mfd-filepath	filepath = mf							
remove-fileext	result-filepat							
remove-folder	filename = re							
replace-fileext	result-filepat							
resolve-filepath	result-filepat					<u>¥</u> <u>×</u>		
generator functions auto-number	result = auto				5/4/3/2			
logical functions	-							
Search for function	•	-						
Add/Remove Libraries	V/////////////////////////////////////		///////////////////////////////////////				-	

After you load the document, you can execute commands against the active document using the menu. Context menus are also available. You can also load additional documents. Save any modifications using the **File | Save** command.

## 17.5.1.2 Retrieving Command Information

The MapForceControl gives access to all commands of MapForce through its CommandsList, MainMenu, and Toolbars properties. The example project available in the folder <**ApplicationFolder**>**ExamplesActiveXC#** uses the MainMenu property to create the MapForce menu structure dynamically.

The code that gets the menu commands can be found in the MDIMain method in MDIMain.cs file:

```
public MDIMain()
{
```

}

//	
// Get the MainMenu property of the control and cre	eate the menu structure from it.
InsertMenuStructure(mainMenu, objCommand);	concror.mainmenu/

In the code listing above, mainMenu is the existing static menu of the main MDI Frame window. If you open the **MDIMain.cs** form in the Visual Studio Designer, you will notice that this menu contains two menu items: **File** and **Window**.

М	DIMain.cs	[Design] ⊣⊨ ×	
	🖳 Map	ForceControl	
	<u>F</u> ile	<u>W</u> indow	
			MapForceX Manager Control

MDIMain.cs

The method InsertMenuStructure takes as parameters the mainMenu and the objCommand objects (the former is the existing static menu, while the latter contains the full menu structure retrieved from the MapForce ActiveX control). The retrieved MapForce menu structure is then merged into the existing static menu. Note that the menus **File**, **Project**, and **Window** are not added dynamically. This is intentional, because these menus deal with actively open documents, and they would require code which is beyond the scope of this example. The basic file management commands (create, open, save, bring into focus) are handled by the existing static menus **File** and **Window**. All other menus are inserted dynamically based on the information taken from the MainMenu property of the ActiveX control. The new menus are inserted after "File" but before "Window", i.e. starting at menu index 1.

The method InsertMenuStructure iterates through all top-level menus found in MapForceCommand object and adds a new menu item for each. Since each top-level menu has its own child menu items, a call to the method InsertMenuCommand takes place for each encountered child menu item. Furthermore, since each child menu item can have its own children menu items, and so on, the InsertMenuCommand method recurses into itself until no more child menu items exist.

The commands added dynamically are instances of the class CustomMenuItem, which is defined in CustomMenuItem.cs. This class is derived from System.Windows.Forms.MenuItem class and has an additional member to store the MapForce command ID.

```
public class CustomMenuItem : System.Windows.Forms.MenuItem
{
     public int m_MapForceCmdID;
}
```

All dynamically added commands (except those that are containers for other commands) get the same event handler AltovaMenuItem\_Click which does the processing of the command:

```
private void AltovaMenuItem_Click(object sender, EventArgs e)
{
    if(sender.GetType() == System.Type.GetType("MapForceApplication.CustomMenuItem"))
    {
        CustomMenuItem customItem = (CustomMenuItem)sender;
        ProcessCommand(customItem.m_MapForceCmdID);
    }
}
```

If the command is a container for other commands (that is, if it has child commands), it gets the event handler AltovaSubMenu\_Popup. This handler queries the status of each child command and enables or disables it as required. This ensures that each command is enabled only when that is meaningful (for example, the **File** | **Save** menu item should be disabled if there is no active document open).

The method ProcessCommand delegates the execution either to the MapForceControl itself or to any active MapForce document loaded in a MapForceControlDocument control. This is necessary because the MapForceControl has no way to know which document is currently active in the hosting application.

```
private void ProcessCommand(int nID)
{
    MapForceDoc docMapForce = GetCurrentMapForceDoc();
    if(docMapForce != null)
        docMapForce.axMapForceControlDoc.Exec(nID);
    else
        axMapForceControl.Exec(nID);
}
```

## 17.5.1.3 Handling Events

Because all events in the MapForce library are based on connection points, you can use the C# delegate mechanism to provide the custom event handlers. You will always find a complete list of events on the property page of each control of the MapForce library. The image below shows the events of the main MapForceControl:

Properties 👻 🗖 🗙				
axMapForceControl AxMapForceControlLib.AxMapForceControl				
🗄 🛃 🖗 🗲 🖉				
		<u>_</u>		
CausesValidationChanged				
ClientSizeChanged				
ContextMenuStripChanged				
ControlAdded				
ControlRemoved				
DockChanged				
Enter				
Leave				
LocationChanged				
MarginChanged				
MouseCaptureChanged				
Move				
OnCloseEditingWindow				
OnComboBoxSelectionUpdate				
OnContextChanged				
OnDocumentOpened				
OnFileChangedAlert				
OnFileExternalChange	axMapForceControl_OnFileExternalChange	~		
OnGetLastActiveEditor				
OnGetPluginManagerControl		-		

As you can see, the example project only overrides the OnFileExternalChange event. The creation of the C# delegate is done for you by the C# Framework. All you need to do is fill in the empty event handler.

For example, the handler implementation shown below turns off any file reloading and displays a message box to inform the user that a file loaded by the MapForceControl has been changed from outside:

```
private void axMapForceControl_OnFileExternalChange(object sender,
AxMapForceControlLib._DMapForceControlEvents_OnFileExternalChangeEvent e)
{
    MessageBox.Show("Attention: The file " + e.strPath + " has been changed from
    outside\nbut reloading is turned off in the sample application!");
    // This turns off any file reloading:
    e.varRet = false;
}
```

## 17.5.2 HTML

The code listings in this section show how to integrate the MapForceControl at application level and document level. Source code for all examples is available in the folder <a href="https://www.applicationFolder>Lexamples/ActiveX/HTML">https://www.applicationFolder>Lexamples/ActiveX/HTML</a> of your MapForce installation.

**Note:** ActiveX controls in an HTML page are supported only by Internet Explorer when it runs as a 32-bit application. When Internet Explorer 10 or 11 runs in 64-bit mode, it does not load ActiveX controls. The default browser security settings will normally block ActiveX, so you will need to explicitly allow blocked content to run on the page when prompted by Internet Explorer.

## 17.5.2.1 HTML Integration at Application Level

This example shows a simple integration of the MapForce control at application-level into a HTML page. The integration is described in the following sections:

- Instantiate a MapForceControl in HTML code.
- Implement buttons to load documents and automate code-generation tasks.
- Define actions for some application events.

The code for this example is available at the following location in your MapForce installation: <ApplicationFolder>\Examples\ActiveX\HTML\MapForceActiveX\_ApplicationLevel.htm.

### 17.5.2.1.1 Instantiate the Control

The HTML Object tag is used to create an instance of the MapForceControl. The Classid is that of MapForceControl. Width and height specify the window size. No additional parameters are necessary, since application-level is the default.

```
<OBJECT id="objMapForceControl"
Classid="clsid:A38637E9-5759-4456-A167-F01160CC22C1"
width="800"
height="500"
VIEWASTEXT>
</OBJECT>
```

### 17.5.2.1.2 Add Button to Open Default Document

As a simple example of how to automate some tasks, we add a button to the page:

```
<input type="button" value="Open Marketing Expenses" onclick="BtnOpenMEFile()">
```

When clicked, a predefined document will be opened in the MapForceControl. The MakeAbsolutePath method creates an absolute path using the location of the script as a base path.

```
function BtnOpenMEFile()
{
    var strPath = MakeAbsolutePath("MarketingExpenses.mfd");
    var objDoc = objMapForceControl.Open(strPath );
    if (objDoc == null)
        alert("Unable to locate MarketingExpenses.mfd at: " + objMapForceControl.BaseHref);
}
```

#### 17.5.2.1.3 Add Buttons for Code Generation

Although code-generation for the active document is available via menus, we want to have buttons that will generate code without asking the user for the location of the output. The method is similar to that used in the previous section.

First come the buttons:

```
<input type="button" value="Generate XSLT" onclick="BtnGenerate( 0 )">
<input type="button" value="Generate Java" onclick="BtnGenerate( 1 )">
<input type="button" value="Generate C++" onclick="BtnGenerate( 2 )">
<input type="button" value="Generate C++" onclick="BtnGenerate( 3 )">
```

Then we provide the script that will generate the code into sub-folders of the currently defined default output folders.

```
// ----
// generate code for active document into language-specific subfolders of
// the current default output dicrectory. No user intercation necessary.
function BtnGenerate(languageID)
   // get top-level object of automation interface
  var objApp = objMapForceControl.Application;
  // get the active document
  var objDocument = objApp.ActiveDocument;
  // retrive object to set the generation output path
  var objOptions = objApp.Options;
   if ( objDocument == null )
     alert( "no active document found" );
   else
      objOptions.XSLTDefaultOutputDirectory = objOptions.CodeDefaultOutputDirectory =
GetDefaultOutputDirectory();
      if (languageID == 0)
      {
         objOptions.XSLTDefaultOutputDirectory = objOptions.XSLTDefaultOutputDirectory +
"\\XSLTGen";
```

```
objDocument .GenerateXSLT();
      }
      else if (languageID == 1)
      ł
         objOptions.CodeDefaultOutputDirectory = objOptions.CodeDefaultOutputDirectory +
"/JavaCode";
        objDocument .GenerateJavaCode();
      }
      else if (languageID == 2)
      {
        objOptions.CodeDefaultOutputDirectory = objOptions.CodeDefaultOutputDirectory +
"/CPPCode";
        objDocument .GenerateCppCode();
      }
      else if (languageID == 3)
      {
        objOptions.CodeDefaultOutputDirectory = objOptions.CodeDefaultOutputDirectory +
"/CSharpCode";
        objDocument .GenerateCHashCode();
      }
  }
}
```

### 17.5.2.1.4 Connect to Custom Events

The example implements two event callbacks for MapForceControl custom events to show the principle:

## 17.5.2.2 HTML Integration at Document Level

This example shows an integration of the MapForce control at document-level into a HTML page. The following topics are covered:

- Instantiate a MapForceControl ActiveX control object in HTML code
- Instantiate a MapForceControlDocument ActiveX control to allow editing a MapForce file
- Instantiate one MapForceControlPlaceHolder for a MapForceControl project window
- Instantiate one MapForceControlPlaceHolder to alternatively host one of the MapForce helper windows

- Create a simple custom toolbar for some heavy-used MapForce commands
- Add some more buttons that use the COM automation interface of MapForce
- Use event handlers to update command buttons

This example is available in its entirety in the file MapForceActiveX\_DocumentLevel.htm within the <ApplicationFolder>\Examples\ActiveX\HTML\ folder of your MapForce installation.

### 17.5.2.2.1 Instantiate the MapForceControl

The HTML OBJECT tag is used to create an instance of the MapForceControl. The Classid is that of MapForceControl. Width and height are set to 0 since we use this control as manager control without use for its user interface. The integration level is specified as a parameter within the OBJECT tag.

### 17.5.2.2.2 Create Editor Window

The HTML OBJECT tag is used to embed an editing window. The additional custom parameter specifies that the control is to be initialized with a new empty document.

## 17.5.2.2.3 Create Project Window

The HTML OBJECT tag is used to create a MapForceControlPlaceHolder window. The parameter defines the placeholder to show the MapForce project window.

#### 17.5.2.2.4 Create Placeholder for Helper Windows

The **MapForceControlPlaceHolder** control is required to host an application helper window, see also Integration at Document Level<sup>1418</sup>. In the code listing below, the HTML object tag is used to instantiate a control that will host the **Libraries** window by default (PlaceholderWindowID is 0).

The example HTML page includes a few buttons which call the BtnHelperWindow method when clicked. The BtnHelperWindow method reassigns the PlaceholderWindowID of the control, and thus cause the ActiveX object to display a different helper window.

```
// specify which of the helper windows shall be shown in the placeholder control.
function BtnHelperWindow(i_ePlaceholderWindowID)
{
    objPlaceholderWindow.PlaceholderWindowID = i_ePlaceholderWindowID;
}
```

For the list of possible values of PlaceholderWindowID, See <u>MapForceControlPlaceholderWindow</u><sup>[473]</sup>.

### 17.5.2.2.5 Create a Custom Toolbar

The example HTML page also includes a custom toolbar (intended as a replica of the MapForce menu). The custom toolbar consists of buttons with images of MapForce commands, for example:

The names of button images correspond to the command ID numbers, see <u>Command Reference</u><sup>1449</sup>. On clicking the button, the corresponding command ID is sent to the main control and executed:

```
// perform any command specified by cmdID.
// command routing includes application, active document and view.
function BtnDoCommand(cmdID)
{
    objMapForceX.Exec( cmdID );
    msgtext.innerText = "Command " + cmdID + " performed.";
}
```

#### 17.5.2.2.6 Create More Buttons

In the example, we add some more buttons to show some automation code.

The corresponding JavaScript looks like this:

```
// ---
// open a new empty document in the specified document control window.
function BtnNewFile(objDocCtrl)
{
  objDocCtrl.OpenDocument("");
   objDocCtrl.setActive();
}
// -----
// Saves the current file in the specified document control window.
function BtnSaveFile(objDocCtrl)
ł
   if(objDocCtrl.Path.length > 0)
objDocCtrl.SaveDocument();
  else
if(strPath.value.length > 0)
{
   objDocCtrl.Path = strPath.value;
  objDocCtrl.SaveDocument();
}
else
{
   alert("Please set path for the document first!");
```

```
strPath.focus();
}
   }
   objDocCtrl.setActive();
}
                        _____
// ----
// open a document in the specified document control window.
function BtnOpenMEFile(objDocCtrl)
{
  // do not use MapForceX.Application.OpenDocument(...) to open a document,
  // since then MapForceControl wouldn't know a control window to show
  // the document in. Instead:
  var strPath = MakeAbsolutePath("MarketingExpenses.mfd");
  var objDoc = objDocCtrl.OpenDocument(strPath);
  if (objDoc != null)
   {
objDocCtrl.setActive();
msgtext.innerText = "Opened mapping: " + strPath;
   }
  else
alert("Unable to open " + strPath);
}
```

### 17.5.2.2.7 Create Event Handler to Update Button Status

Availability of a command may vary with every mouse click or keystroke. The custom event OnUpdateCmdUI of MapForceControl gives us an opportunity to update the enabled/disabled state of buttons associated with MapForce commands. The method <u>MapForceControl.QueryStatus</u><sup>[46]</sup> is used to query whether a command is enabled or not.

```
<!-- custom event 'OnUpdateCmdUI" of MapForceControl object -->
function objMapForceX::OnUpdateCmdUI()
{
  if ( document.readyState == "complete" )// 'complete'
   {
     // update status of buttons
     // set activity status of simulated toolbar
     GenerateXSLT.disabled = ! (objDoc1.QueryStatus(32360) & 0x02); // not enabled
     GenerateJava.disabled = ! (objDoc1.QueryStatus(32358) & 0x02); // not enabled
     GenerateCpp.disabled = ! (objDoc1.QueryStatus(32356) & 0x02); // not enabled
     GenerateCSharp.disabled = ! (objDoc1.QueryStatus(32357) & 0x02); // not enabled
     btnInsertXML.disabled = ! (objDoc1.QueryStatus(32393) & 0x02);
     btnInsertDB.disabled = ! (objDoc1.QueryStatus(32389) & 0x02);
     btnInsertEDI.disabled = ! (objDoc1.QueryStatus(32390) & 0x02);
     btnInsertText.disabled = ! (objDoc1.QueryStatus(32392) & 0x02);
     btnInsertConstant.disabled = ! (objDoc1.QueryStatus(32388) & 0x02);
```

```
btnInsertFilter.disabled = ! (objDocl.QueryStatus(32391) & 0x02);
btnInsertIFELSE.disabled = ! (objDocl.QueryStatus(32394) & 0x02);
btnInsertException.disabled = ! (objDocl.QueryStatus(32311) & 0x02);
btnFuncUserDef.disabled = ! (objDocl.QueryStatus(32380) & 0x02);
btnFuncUserDefSel.disabled = ! (objDocl.QueryStatus(32381) & 0x02);
btnFuncSettings.disabled = ! (objDocl.QueryStatus(32387) & 0x02);
btnInsertInput.disabled = ! (objDocl.QueryStatus(32387) & 0x02);
btnGenXSLT.disabled = ! (objDocl.QueryStatus(32360) & 0x02);
btnGenXSLT2.disabled = ! (objDocl.QueryStatus(32361) & 0x02);
btnGenXQuery.disabled = ! (objDocl.QueryStatus(32359) & 0x02);
btnGenCPP.disabled = ! (objDocl.QueryStatus(32359) & 0x02);
btnGenCPP.disabled = ! (objDocl.QueryStatus(32357) & 0x02);
btnGenCSharp.disabled = ! (objDocl.QueryStatus(32357) & 0x02);
btnGenJava.disabled = ! (objDocl.QueryStatus(32358) & 0x02);
```

## 17.5.3 Java

}

MapForce ActiveX components can be accessed from Java code. Java integration is provided by the libraries listed below. These libraries are available in the folder <applicationFolder>\Examples\JavaAPI of your MapForce installation, after you have installed both MapForce and the MapForce Integration Package (see also <a href="Prerequisites">Prerequisites</a>

- AltovaAutomation.dll: a JNI wrapper for Altova automation servers (in case of the 32-bit installation of MapForce)
- AltovaAutomation\_x64.dll: a JNI wrapper for Altova automation servers (in case of the 64-bit installation of MapForce)
- AltovaAutomation.jar: Java classes to access Altova automation servers
- MapForceActiveX.jar: Java classes that wrap the MapForce ActiveX interface
- MapForceActiveX\_JavaDoc.zip: a Javadoc file containing help documentation for the Java interface
- **Note:** In order to use the Java ActiveX integration, the .dll and .jar files must be included in the Java class search path.

#### Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details, see Example Java Project<sup>[433]</sup>.

#### Rules for mapping the ActiveX Control names to Java

For the documentation of ActiveX controls, see <u>Object Reference</u><sup>1439</sup>. Note that the object naming conventions are slightly different in Java compared to other languages. Namely, the rules for mapping between the ActiveX controls and the Java wrapper are as follows:

#### Classes and class names

For every component of the MapForce ActiveX interface a Java class exists with the name of the component.

#### 1432 ActiveX Integration

#### • Method names

Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with get and set can be used. If a property does not support write-access, no setter method is available. Example: For the IntegrationLevel property of the MapForceControl, the Java methods getIntegrationLevel and setIntegrationLevel are available.

#### • Enumerations

For every enumeration defined in the ActiveX interface, a Java enumeration is defined with the same name and values.

#### • Events and event handlers

For every interface in the automation interface that supports events, a Java interface with the same name plus 'Event' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus 'DefaultHandler'. For example:

MapForceControl: Java class to access the application

MapForceControlEvents: Events interface for the MapForceControl

MapForceControlEventsDefaultHandler: Default handler for MapForceControlEvents

### Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Changes in Java class
MapForceControlDocument, method New	Renamed to newDocument
MapForceControlDocument, method OpenDocument	Removed. Use the Open method
MapForceControlDocument, method NewDocument	Removed. Use the newDocument method
MapForceControlDocument, method SaveDocument	Removed. Use the Save method

### This section

This section shows how some basic MapForce ActiveX functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project<sup>1433</sup>
- <u>Creating the ActiveX Controls</u>
   <sup>1435</sup>
- Loading Data in the Controls [1435]
- <u>Basic Event Handling</u><sup>1436</sup>
- Menus<sup>1436</sup>
- <u>UI Update Event Handling</u><sup>1438</sup>
- Creating a MapForce Mapping Table<sup>[139]</sup>

# 17.5.3.1 Example Java Project

The MapForce installation package contains an example Java project, located in the ActiveX Examples folder of the application folder: <a href="https://www.applicationFolder-comples/activeX/Java/">https://www.applicationFolder-comples/activeX/Java/</a>.

The Java example shows how to integrate the MapForceControl in a common desktop application created with Java. You can test it directly from the command line using the batch file BuildAndRun.bat, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

### File list

The Java examples folder contains all the files required to run the example project. These files are listed below:

.classpath	Eclipse project helper file
.project	Eclipse project file
AltovaAutomation.dll	Java-COM bridge: DLL part (for the 32-bit installation)
AltovaAutomation_x64.dll	Java-COM bridge: DLL part (for the 64-bit installation)
AltovaAutomation.jar	Java-COM bridge: Java library part
BuildAndRun.bat	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
MapForceActiveX.jar	Java classes of the MapForce ActiveX control
MapForceActiveX_JavaDoc.zip	Javadoc file containing help documentation for the Java API
MapForceContainer.java	Java example source code
MapForceContainerEventHandler.java	Java example source code
MapForceTable.java	Java example source code

#### What the example does

The example places one MapForce document editor window, the MapForce project window, the MapForce library window and the MapForce validation window in an AWT frame window. It reads out the main menu defined for MapForce and creates an AWT menu with the same structure. You can use this menu or the project window to open and work with files in the document editor.

You can modify the example in any way you like.

The following specific features are described in code listings:

- <u>Creating the ActiveX Controls</u><sup>[433]</sup>: Starts MapForce, which is registered as an automation server, or activates MapForce if it is already running.
- <u>Loading Data in the Controls</u><sup>[435]</sup>: Locates one of the example documents installed with MapForce and opens it.

- <u>Basic Event Handling</u><sup>[1439</sup>: Changes the view of all open documents to Text View. The code also shows how to <u>ite</u>rate through open documents.
- Menus<sup>[1439</sup>: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- <u>UI Update Event Handling</u><sup>[143]</sup>: Shows how to handle MapForce events.
- <u>Creating a MapForce Mapping Table</u><sup>[439</sup>: Shows how to create a MapForce mapping table and prepare it for modal activation.

#### Updating the path to the Examples folder

Before running the provided sample, you may need to edit the **MapForceContainer.java** file. Namely, check that the following path refers to the actual folder where the MapForce example files are stored on your operating system:

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\Documents\\Altova\
\MapForce2023\\MapForceExamples\\";
```

#### Running the example from the command line

To run the example from the command line:

- 1. Check that all prerequisites are met (see <u>Prerequisites</u><sup>1406</sup>).
- 2. Open a command prompt window, change the current directory to the sample Java project folder, and type:

buildAndRun.bat "<Path-to-the-Java-bin-folder>"

3. Press Enter.

The Java source in MapForceContainer. java will be compiled and then executed.

#### Compiling and running the example in Eclipse

To import the sample Java project into Eclipse:

- 1. Check that all prerequisites are met (see <u>Prerequisites</u><sup>1406</sup>).
- 2. On the **File** menu, click **Import**.
- 3. Select **Existing Projects into Workspace**, and browse for the Eclipse project file located at <ApplicationFolder>\Examples\ActiveX\Java\. Since you may not have write-access in this folder, it is recommended to select the **Copy projects into workspace** check box on the Import dialog box.

To run the example application, right-click the project in Package Explorer and select the command **Run as |** Java Application.

Help for Java API classes is available through comments in code as well as the Javadoc view of Eclipse. To enable the Javadoc view in Eclipse, select the menu command **Window | Show View | JavaDoc**.

## 17.5.3.2 Creating the ActiveX Controls

The code listing below show how ActiveX controls can be created. The constructors will create the Java wrapper objects. Adding these Canvas-derived objects to a panel or to a frame will trigger the creation of the wrapped ActiveX object.

```
/**
01
02
     * MapForce manager control - always needed
     */
03
04
    public static MapForceControl
                                          mapForceControl = null;
05
     /**
06
     * MapForceDocument editing control
07
     */
08
09
    public static MapForceControlDocument
                                               mapForceDocument = null;
10
     /**
11
     * Tool windows - MapForce place-holder controls
12
     * /
13
14
    private static MapForceControlPlaceHolder
                                                 mapForceProjectToolWindow = null;
    private static MapForceControlPlaceHolder
                                                 mapForceValidationToolWindow = null;
15
    private static MapForceControlPlaceHolder mapForceLibraryToolWindow = null;
16
17
18
    // Create the MapForce ActiveX control; the parameter determines that we want
    // to place document controls and place-holder controls individually.
19
    // It gives us full control over the menu, as well.
        mapForceControl = new MapForceControl(
20
        ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue(), false );
21
22
        mapForceDocument = new MapForceControlDocument();
23
        frame.add( mapForceDocument, BorderLayout.CENTER );
24
25
26
    // Create a project window and open the sample project in it
27
        mapForceProjectToolWindow = new MapForceControlPlaceHolder(
        MapForceControlPlaceholderWindow.MapForceXProjectWindow.getValue(),
        strExamplesFolder + "MapForceExamples.mfp" ) ;
        mapForceProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );
28
```

## 17.5.3.3 Loading Data in the Controls

The code listing below show how data can be loaded in the ActiveX controls.

## 17.5.3.4 Basic Event Handling

The code listing below shows how basic events can be handled. When calling the MapForceControl's open method, or when trying to open a file via the menu or Project tree, the onOpenedOrFocused event is sent to the attached event handler. The basic handling for this event is opening the file by calling the MapForceDocumentControl's open method.

```
01
         // Open the Marketing file when button is pressed
02
         btnMarkExp.addActionListener( new ActionListener() {
03
           public void actionPerformed(ActionEvent e) {
04
             try {
05
               // Instruct the Document control to open the file - avoid calling the open
method of MapForceControl (see help)
06
               mapForceDocument.open( strExamplesFolder + "MarketingExpenses.mfd" );
07
               mapForceDocument.requestFocusInWindow();
80
             } catch (AutomationException e1) {
09
               e1.printStackTrace();
10
11
           }
12
         });
13
         public void onOpenedOrFocused( String i_strFileName, boolean
i_bOpenWithThisControl, boolean i_bFileAlreadyOpened ) throws AutomationException
14
     {
15
       // Handle the New/Open events coming from the Project tree or from the menus
16
       if ( !i_bFileAlreadyOpened )
17
       ł
         // This is basically an SDI interface, so open the file in the already existing
18
document control
19
        try {
20
          MapForceContainer.mapForceDocument.open( i_strFileName );
21
          MapForceContainer.mapForceDocument.requestFocusInWindow();
22
         } catch (Exception e) {
           e.printStackTrace();
23
24
25
       }
26
     }
```

## 17.5.3.5 Menus

The code listing below shows how menu items can be created. Each MapForceCommand object gets a corresponding MenuItem object, with the ActionCommand set to the ID of the command. The actions generated by all menu items are handled by the same function, which can perform specific handlings (like reinterpreting the closing mechanism) or can delegate the execution to the MapForceControl object by calling its exec method. The menuMap object that is filled during menu creation is used later (see section UI Update Event Handling<sup>4489</sup>).

```
01
02 // Load the file menu when the button is pressed
03 btnMenu.addActionListener( new ActionListener() {
04 public void actionPerformed(ActionEvent e) {
05 try {
06 // Create the menubar that will be attached to the frame
```

```
07
               MenuBar mb = new MenuBar();
               // Load the main menu's first item - the File menu
08
09
               MapForceCommand xmlSpyMenu =
mapForceControl.getMainMenu().getSubCommands().getItem( 0 );
10
               // Create Java menu items from the Commands objects
               Menu fileMenu = new Menu();
11
12
               handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
13
               fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
14
               mb.add( fileMenu );
15
               frame.setMenuBar( mb );
16
               frame.validate();
17
             } catch (AutomationException el) {
18
               e1.printStackTrace();
19
             }
20
             // Disable the button when the action has been performed
21
             ((AbstractButton) e.getSource()).setEnabled( false );
22
           }
23
         });
     /**
24
      \ast Populates a menu with the commands and submenus contained in an MapForceCommands
25
object
26
27
     public void fillMenu(Menu newMenu, MapForceCommands mapForceMenu) throws
AutomationException
28
     {
29
       // For each command/submenu in the mapForceMenu
30
       for ( int i = 0 ; i < mapForceMenu.getCount() ; ++i )</pre>
31
       ł
32
         MapForceCommand mapForceCommand = mapForceMenu.getItem( i );
33
         if ( mapForceCommand.getIsSeparator() )
34
          newMenu.addSeparator();
35
         else
36
37
           MapForceCommands subCommands = mapForceCommand.getSubCommands();
           // Is it a command (leaf), or a submenu?
38
           if ( subCommands.isNull() || subCommands.getCount() == 0 )
39
40
41
             // Command -> add it to the menu, set its ActionCommand to its ID and store it
in the menuMap
42
            MenuItem mi = new MenuItem( mapForceCommand.getLabel().replace( "&", "" ) );
43
            mi.setActionCommand( "" + mapForceCommand.getID() );
44
            mi.addActionListener( this );
45
            newMenu.add( mi );
46
            menuMap.put( mapForceCommand.getID(), mi );
           }
47
48
           else
49
           {
             // Submenu -> create submenu and repeat recursively
50
51
             Menu newSubMenu = new Menu();
52
             fillMenu( newSubMenu, subCommands );
53
             newSubMenu.setLabel( mapForceCommand.getLabel().replace( "&", "" ) );
54
             newMenu.add( newSubMenu );
55
           3
56
         }
       }
57
     }
58
     /**
59
60
      * Action handler for the menu items
```

```
61
      * Called when the user selects a menu item; the item's action command corresponds to
the command table for MapForce
     */
62
63
    public void actionPerformed( ActionEvent e )
64
65
       try
66
       {
67
         int iCmd = Integer.parseInt( e.getActionCommand() );
68
         // Handle explicitly the Close commands
69
         switch ( iCmd )
70
         {
                             // Close
71
           case 57602:
          case 34050:
72
                             // Close All
73
             MapForceContainer.initMapForceDocument();
74
             break;
75
          default:
76
             MapForceContainer.mapForceControl.exec( iCmd );
77
             break;
        }
78
79
       }
80
       catch ( Exception ex )
81
      {
82
         ex.printStackTrace();
83
       }
84
85
     }
```

## 17.5.3.6 UI Update Event Handling

The code listing below shows how a UI-Update event handler can be created.

```
01 /**
      * Call-back from the MapForceControl.
02
     * Called to enable/disable commands
03
     */
04
05
    @Override
06
    public void onUpdateCmdUI() throws AutomationException
07
    {
80
      // A command should be enabled if the result of queryStatus contains the Supported
(1) and Enabled (2) flags
09
     for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )
10
      pair.getValue().setEnabled( MapForceContainer.mapForceControl.queryStatus( pair.getKe
y()) > 2);
11
    }
12 /**
13
     * Call-back from the MapForceControl.
     * Usually called while enabling/disabling commands due to UI updates
14
     * /
15
16
    @Override
     public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
17
18
     {
19
       try {
20
         return
MapForceContainer.mapForceDocument.getDocument().getFullName().equalsIgnoreCase( i_strFileP
ath );
```

## 17.5.3.7 Listing the Properties of a MapForce Mapping

The listing below shows how a Mapping object in MapForce can be loaded as a table and prepared for modal activation.

```
01 //access MapForce Java-COM bridge
02 import com.altova.automation.MapForce.*;
03 import com.altova.automation.MapForce.Component;
04 import com.altova.automation.MapForce.Enums.ENUMComponentUsageKind;
05
06 //access AWT and Swing components
07 import java.awt.*;
08 import javax.swing.*;
09 import javax.swing.table.*;
10
11
12 /**
13 * A simple example of a table control loading the structure from a Mapping object.
14 * The class receives an Mapping object, loads its components in a JTable, and prepares
15 * for modal activation.
16
17
   * Feel free to modify and extend this sample.
18
19 * @author Altova GmbH
20 */
21 class MapForceTable extends JDialog
22 {
    /**
23
     * The table control
24
     * /
25
    private JTable myTable;
26
27
28
    /**
29
     * Constructor that prepares the modal dialog containing the filled table control
     * @param mapping The data to be displayed in the table
30
31
     * @param parent Parent frame
32
     */
    public MapForceTable( Mapping mapping, Frame parent )
33
34
    {
35
      // Construct the modal dialog
     super( parent, "MapForce component table", true );
36
37
      // Build up the tree
38
      fillTable( mapping );
39
      // Arrange controls in the dialog
40
      setContentPane( new JScrollPane( myTable ) );
41
     }
42
     /**
43
44
     * Loads the components of a Mapping object in the table
45
      * @param mapping Source data
```

```
46
      * /
47
     private void fillTable( Mapping mapping)
48
     {
49
       try
50
       {
51
         // count how many Instance components do we have
52
         int size = 0;
         for (Component comp : mapping.getComponents())
53
54
          if ( comp.getUsageKind() ==
ENUMComponentUsageKind.eComponentUsageKind_Instance )
55
             ++size;
56
57
         // Prepare data
        final String[] columnNames = { "Component", "Has inputs", "Has outputs", "Input
58
file", "Output file", "Schema" };
59
        final Object[][] data = new Object[size ][ 7 ] ;
60
         int index = 0 ;
61
         for (Component comp : mapping.getComponents())
62
          if ( comp.getUsageKind() ==
ENUMComponentUsageKind.eComponentUsageKind_Instance )
63
          {
64
             int i = 0;
65
             data[ index ][ i++ ] = comp.getName() ;
66
             data[ index ][ i++ ] = new Boolean( comp.getHasIncomingConnections() );
67
             data[ index ][ i++ ] = new Boolean( comp.getHasOutgoingConnections() );
68
             data[ index ][ i++ ] = comp.getInputInstanceFile();
69
             data[ index ][ i++ ] = comp.getOutputInstanceFile();
70
             data[ index++ ][ i ] = comp.getSchema() ;
71
           }
72
73
         // Set up table
         myTable = new JTable( new AbstractTableModel() {
74
75
             public String getColumnName(int col) { return columnNames[col]; }
76
             public int getRowCount() { return data.length; }
77
             public int getColumnCount() { return columnNames.length; }
78
             public Object getValueAt(int row, int col) { return data[row][col]; }
79
             public boolean isCellEditable(int row, int col) { return false; }
80
             public Class getColumnClass(int c) { return getValueAt(0, c).getClass(); }
81
         });
82
83
         // Set width
84
         for( index = 0 ; index < columnNames.length ; ++index )</pre>
85
          myTable.getColumnModel().getColumn( index ).setMinWidth( 80 );
86
         myTable.getColumnModel().getColumn( 5 ).setMinWidth( 400 );
87
       }
88
      catch (Exception e)
89
       ł
90
         e.printStackTrace();
91
       }
92
     }
93
94 }
```
# 17.5.4 VB.NET

Source code which illustrates integration of MapForceControl into a VB.NET application can be found in the folder <ApplicationFolder>\Examples\ActiveX\VB.NET of your MapForce installation. The solution consists of three windows, as follows:

- 1. MainWindow.vb the main document window, which also includes a basic application menu.
- LibraryWindow.vb the Library window. The contents of this window is populated by a Placeholder control which has the PlaceholderWindowID property set to 0 (this value instructs the control to display specifically the Library window).

LibraryWindow			
	Properties		- <b>□</b> ×
	AxMapForceControlPlace	Holder1 AxMapForceControlLib.	AxMapForceContr 👻
	E 🛃 🖓 ۶ 👂		
	⊞ MinimumSize     ■	0, 0	<b>A</b>
	Modifiers	Friend	
	Padding	0, 0, 0, 0	
	PlaceholderWindowlD	0	
	H Size	251, 446	
	Tabindex	U	•
	PlaceholderWindowID		
~			
Search for function 🔹			
Add/Remove Libraries			

 OutputWindow.vb - the Messages (Output) window. The contents of this window is populated by a Placeholder control which has the PlaceholderWindowID property set to 2 (this value instructs the control to display specifically the Output window).

	OutputV	Vindow		<b>×</b> ,
	25X			
<ul> <li>Y6/5/4/3/2/</li> </ul>				< >
S				
	Properties		- <b>-</b> ×	
	MapForceValidation AxMapFor	rceControlLib.AxMapForceC	ontrolPlaceHolder 👻	
	₽ <b>₽ ₽ ₽</b>			
	MinimumSize	0, 0	▲	
	Modifiers	Friend		
	Padding	0, 0, <mark>0</mark> , 0		
	PlaceholderWindowlD	2	-	
	⊡ Size	761, 133		
	TabIndex	0	-	
	PlaceholderWindowID			

Before attempting to build and run this solution, note the following steps:

#### Step 1: Check the prerequisites

For the list of prerequisites, see <u>Prerequisites</u><sup>1406</sup>.

### Step 2: Copy the sample to a directory where you have write permissions

To avoid running Visual Studio as an Administrator, copy the source code to a directory where you have write permissions, instead of running it from the default location.

#### Step 3: Set the build platform

- Create a build platform configuration that matches the platform under which you want to build (x86, x64). Here is how you can create the build configuration:
  - a. Right-click the solution in Visual Studio, and select Configuration Manager.
  - b. Under Active solution platform, select New... and then select the x86 or x64 configuration (in this example, x86).

New Solution Platform
Type or select the new <u>p</u> latform:
x86 🗸
Copy <u>s</u> ettings from:
Any CPU 👻
☑ Create new project platforms
OK Cancel

You are now ready to build and run the solution in Visual Studio. Remember to build using the configuration that matches your target platform (x86, x64); otherwise, runtime errors might occur.

# 17.6 Command Reference

This section lists the names and identifiers of all menu commands that are available within MapForce. Every sub-section lists the commands from the corresponding top-level menu of MapForce. The command tables are organized as follows:

- The "Menu Item" column shows the command's menu text as it appears in MapForce, to make it easier for you to identify the functionality behind the command.
- The "Command Name" column specifies the string that can be used to get an icon with the same name from **ActiveX\Images** folder of the MapForce installation directory.
- The "ID" column shows the numeric identifier of the column that must be supplied as argument to methods which execute or query this command.

To execute a command, use the <u>MapForceControl.Exec</u><sup>[460]</sup> or the <u>MapForceControlDocument.Exec</u><sup>[460]</sup> methods. To query the status of a command, use the <u>MapForceControl.QueryStatus</u><sup>[461]</sup> or <u>MapForceControlDocument.QueryStatus</u><sup>[467]</sup> methods.

Depending on the edition of MapForce you have installed, some of these commands might not be supported.

## 17.6.1 "File" Menu

Menu item	Command name	ID
New	ID_FILE_NEW	57600
Open	ID_FILE_OPEN	57601
Save	ID_FILE_SAVE	57603
Save As	ID_FILE_SAVE_AS	57604
Save All	ID_FILE_SAVEALL	32377
Reload	IDC_FILE_RELOAD	32467
Close	ID_WINDOW_CLOSE	32453
Close All	ID_WINDOW_CLOSEALL	32454
Print	ID_FILE_PRINT	57607
Print Preview	ID_FILE_PRINT_PREVIEW	57609
Print Setup	ID_FILE_PRINT_SETUP	57606
Validate Mapping	ID_MAPPING_VALIDATE	32347
Mapping Settings	ID_MAPPING_SETTINGS	32396
Generate Code in Selected Language	ID_FILE_GENERATE_SELECTED_CODE	32362

The "File" menu has the following commands:

Menu item	Command name	ID
XSLT 1.0	ID_FILE_GENERATEXSLT	32360
XSLT 2.0	ID_FILE_GENERATEXSLT2	32361
XQuery	ID_FILE_GENERATEXQUERY	32359
Java	ID_FILE_GENERATEJAVACODE	32358
C# (Sharp)	ID_FILE_GENERATECSCODE	32357
C++	ID_FILE_GENERATECPPCODE	32356
Compile to MapForce Server Execution File	ID_FILE_CREATE_SERVER_EXECUTION_FIL E	32517
Deploy to FlowForce Server	ID_FILE_DEPLOY_MAPPING	32506
Generate Documentation	ID_FILE_GENERATE_DOCUMENTATION	32468
Recent File	ID_FILE_MRU_FILE1	57616
Exit	ID_APP_EXIT	57665

# 17.6.2 "Edit" Menu

The "Edit" menu has the following commands:

Menu item	Command name	ID
Undo	ID_EDIT_UNDO	57643
Redo	ID_EDIT_REDO	57644
Find	ID_EDIT_FIND	57636
Find Next	ID_EDIT_FINDNEXT	32349
Find Previous	ID_EDIT_FINDPREV	32350
Cut	ID_EDIT_CUT	57635
Сору	ID_EDIT_COPY	57634
Paste	ID_EDIT_PASTE	57637
Delete	ID_EDIT_CLEAR	57632
Select All	ID_EDIT_SELECT_ALL	57642

# 17.6.3 "Insert" Menu

The "Insert" menu has the following commands:

Menu item	Command name	ID
XML Schema/File	ID_INSERT_XSD	32393
Database	ID_INSERT_DATABASE	32389
EDI	ID_INSERT_EDI	32390
Text File	ID_INSERT_TXT	32392
Web Service Function	ID_INSERT_WEBSERVICE_FUNCTION	32319
Excel 2007+ File	ID_INSERT_EXCEL	32376
XBRL Document	ID_INSERT_XBRL	32469
JSON Schema/File	ID_INSERT_JSON	32531
Insert Input	ID_FUNCTION_INSERT_INPUT	32383
Insert Output	ID_FUNCTION_INSERT_OUTPUT	32402
Constant	ID_INSERT_CONSTANT	32388
Variable	ID_INSERT_VARIABLE	32500
Join	ID_INSERT_JOIN	32581
Sort: Nodes/Rows	ID_INSERT_SORT	32444
Filter: Nodes/Rows	ID_INSERT_FILTER	32391
SQL-WHERE/ORDER	ID_INSERT_SQLWHERE_CONDITION	32351
Value-Map	ID_INSERT_VALUEMAP	32354
IF-Else Condition	ID_INSRT_CONDITION	32394
Exception	ID_INSERT_EXCEPTION	32311

# 17.6.4 "Project" Menu

The "Project" menu has the following commands:

Menu item	Command name	ID
Reload Project	ID_PROJECT_RELOAD	32476

Menu item	Command name	ID
Close Project	ID_FILE_CLOSEPROJECT	32355
Save Project	ID_FILE_SAVEPROJECT	32378
Add Files to Project	ID_PROJECT_ADDFILESTOPROJECT	32420
Add Active File to Project	ID_PROJECT_ADDACTIVEFILETOPROJECT	32419
Create Folder	ID_PROJECT_CREATE_FOLDER	32310
Open Mapping	ID_PROJECT_OPEN_MAPPING	32307
Create Mapping for Operation	ID_PROJECT_CREATE_MAPPING_FOR_OPE RATION	32399
Add Mapping File for Operation	ID_PROJECT_ADD_MAPPING	32309
Insert Web Service	ID_PROJECT_INSERT_WEBSERVICE	32306
Open File in XMLSpy	ID_PROJECT_OPEN_IN_XMLSPY	32305
Generate Code for Entire Project	ID_PROJECT_GENERATE_ALL	32303
XSLT 1.0	ID_PROJECT_GENERATEXSLTCODE_ENTIRE	32408
XSLT 2.0	ID_PROJECT_GENERATEXSLT2CODE_ENTIR E	32409
XQuery	ID_PROJECT_GENERATEXQUERYCODE_EN TIRE	32410
Java	ID_PROJECT_GENERATEJAVACODE_ENTIR E	32411
C# (Sharp)	ID_PROJECT_GENERATECSCODE_ENTIRE	32412
C++	ID_PROJECT_GENERATECPPCODE_ENTIRE	32413
Properties	ID_PROJECT_PROPERTIES	32404
Recent Project	ID_FILE_MRU_PROJECT1	32364

# 17.6.5 "Component" Menu

The "Component" menu has the following commands:

Menu item	Command name	ID
Change Root Element	ID_COMPONENT_CHANGEROOTELEMENT	32334
Edit Schema Definition in XMLSpy	ID_COMPONENT_EDIT_SCHEMA	32337

Menu item	Command name	ID
Edit FlexText Configuration	ID_COMPONENT_EDIT_MFT	32301
Add/Remove/Edit Database Objects	ID_COMPONENT_SELECTTABLES	32346
Create Mapping to EDI X12 997	ID_COMPONENT_CREATE_MAPPING_TO_99 7	32483
Create Mapping to EDI X12 999	ID_COMPONENT_CREATE_MAPPING_TO_99 9	32484
Refresh	IDC_COMMAND_REFRESH_COMPONENT	32373
Add Duplicate Input Before	ID_COMPONENT_CREATE_DUPLICATE_ICO N_BEFORE	32503
Add Duplicate Input After	ID_COMPONENT_CREATE_DUPLICATE_ICO N	32335
Remove Duplicate	ID_COMPONENT_REMOVE_DUPLICATE_ICO N	32339
Add Comment Before	ID_COMPONENT_ADD_COMMENT_BEFORE	32518
Add Comment After	ID_COMPONENT_ADD_COMMENT_AFTER	32519
Add Processing Instruction Before	ID_COMPONENT_ADD_PI_BEFORE	32520
Add Processing Instruction After	ID_COMPONENT_ADD_PI_AFTER	32521
Edit Processing Instruction Name	ID_COMPONENT_EDIT_PI	32524
Delete Comment/Processing Instruction	ID_COMPONENT_REMOVE_COMMENT_PI	32522
Write Content as CDATA Section	ID_COMPONENT_TOGGLE_CDATA	32525
Database Table Actions	ID_POPUP_DATABASETABLEACTIONS	32400
Query Database	ID_QUERY_DATABASE	32341
Align Tree Left	ID_COMPONENT_LEFTALIGNTREE	32338
Align Tree Right	ID_COMPONENT_RIGHTALIGNTREE	32340
Properties	ID_COMPONENT_PROPERTIES	32336

# 17.6.6 "Connection" Menu

The "Connection" menu has the following commands:

Menu item	Command name	ID
Auto Connect Matching Children	ID_CONNECTION_AUTOCONNECTCHILDREN	32342

Menu item	Command name	ID
Settings for Connect Matching Children	ID_CONNECTION_SETTINGS	32344
Connect Matching Children	ID_CONNECTION_MAPCHILDELEMENTS	32343
Target Driven (Standard)	ID_POPUP_NORMALCONNECTION	32401
Copy-All (Copy Child Items)	ID_POPUP_NORMALWITHCHILDREN_CONNE CTION	32460
Source Driven (Mixed Content)	ID_POPUP_ORDERBYSOURCECONNECTION	32403
Properties	ID_POPUP_CONNECTION_SETTINGS	32398

# 17.6.7 "Function" Menu

The "Function" menu has the following commands:

Menu item	Command name	ID
Create User-Defined Function	ID_FUNCTION_CREATE_EMPTY	32380
Create User-Defined Function from Selection	ID_FUNCTION_CREATE_FROM_SELECTION	32381
Function Settings	ID_FUNCTION_SETTINGS	32387
Remove Function	ID_FUNCTION_REMOVE	32385
Insert Input	ID_FUNCTION_INSERT_INPUT	32383
Insert Output	ID_FUNCTION_INSERT_OUTPUT	32402

# 17.6.8 "Output" Menu

The "Output" menu has the following commands:

Menu item	Command name	ID
XSLT 1.0	ID_SELECT_LANGUAGE_XSLT	32433
XSLT 2.0	ID_SELECT_LANGUAGE_XSLT2	32434
XQuery	ID_SELECT_LANGUAGE_XQUERY	32432
Java	ID_SELECT_LANGUAGE_JAVA	32431
C# (Sharp)	ID_SELECT_LANGUAGE_CSHARP	32430
C++	ID_SELECT_LANGUAGE_CPP	32429

Menu item	Command name	ID
Built-In Execution Engine	ID_SELECT_LANGUAGE_BUILTIN	32490
Validate Output File	ID_XML_VALIDATE	32458
Save Output File	IDC_FILE_SAVEGENERATEDOUTPUT	32321
Save All Output Files	IDC_FILE_SAVEALLGENERATEDOUTPUT	32374
Regenerate Output	ID_REGENERATE_PREVIEW_OUTPUT	32480
Run SQL-Script	ID_TRANSFORM_RUN_SQL	32442
Insert/Remove Bookmark	ID_TOGGLE_BOOKMARK	32317
Next Bookmark	ID_GOTONEXTBOOKMARK	32315
Previous Bookmark	ID_GOTOPREVBOOKMARK	32314
Remove All Bookmarks	ID_REMOVEALLBOOKMARKS	32313
Pretty-Print XML Text	ID_PRETTY_PRINT_OUTPUT	32363
Text View Settings	ID_TEXTVIEWSETTINGSDIALOG	32472

# 17.6.9 "Debug" Menu

The "Debug" menu has the following commands:

Menu item	Command name	ID
Start Debugging	ID_DEBUG_START	32540
Stop Debugging	ID_DEBUG_STOP	32541
Step Into	ID_DEBUG_STEP_INTO	32545
Step Over	ID_DEBUG_STEP_OVER	32551
Step Out	ID_DEBUG_STEP_OUT	32552
Minimal Step	ID_DEBUG_STEP_NEXT_TRACE	32554

## 17.6.10 "View" Menu

The "View" menu has the following commands:

Menu item	Command name	ID
Show Annotations	ID_SHOW_ANNOTATION	32435

Menu item	Command name	ID
Show Types	ID_SHOW_TYPES	32437
Show Library in Function Header	ID_VIEW_SHOWLIBRARYINFUNCTIONHEAD ER	32448
Show Tips	ID_SHOW_TIPS	32436
XBRL Display Options	ID_VIEW_XBRL_DISPLAY_OPTIONS	32473
Show Selected Component Connectors	ID_VIEW_AUTOHIGHLIGHTCOMPONENTCON NECTIONS	32443
Show Connectors from Source to Target	ID_VIEW_RECURSIVEAUTOHIGHLIGHT	32447
Zoom	ID_VIEW_ZOOM	32451
Back	ID_CMD_BACK	32479
Forward	ID_CMD_FORWARD	32478
Status Bar	ID_VIEW_STATUS_BAR	59393
Library Window	ID_VIEW_LIBRARY_WINDOW	32445
Messages	ID_VIEW_VALIDATION_OUTPUT	32450
Overview	ID_VIEW_OVERVIEW_WINDOW	32446
Project Window	ID_VIEW_PROJECT_WINDOW	32302
Values	ID_DEBUG_VIEW_VALUES_WINDOW	32544
Context	ID_DEBUG_VIEW_CONTEXT_WINDOW	32546
Breakpoints	ID_DEBUG_VIEW_DEBUGPOINTS_WINDOW	32547

# 17.6.11 "Tools" Menu

The "Tools" menu has the following commands:

Menu item	Command name	ID
Global Resources	IDC_GLOBALRESOURCES	37401
<plugin loaded="" not=""></plugin>	IDC_GLOBALRESOURCES_SUBMENUENTR Y1	37408
Create Reversed Mapping	ID_CREATE_REVERSED_MAPPING	32489
Customize	IDC_APP_TOOLS_CUSTOMIZE	32959
Options	ID_TOOLS_OPTIONS	32441

# 17.6.12 "Window" Menu

The "Window" menu has the following commands:

Menu item	Command name	ID
Cascade	ID_WINDOW_CASCADE	57650
Tile Horizontal	ID_WINDOW_TILE_HORZ	57651
Tile Vertical	ID_WINDOW_TILE_VERT	57652

# 17.6.13 "Help" Menu

The "Help" menu has the following commands:

Menu item	Command name	ID
Table of Contents	IDC_HELP_CONTENTS	32966
Index	IDC_HELP_INDEX	32967
Search	IDC_HELP_SEARCH	32969
Software Activation	IDC_ACTIVATION	32970
Order Form	IDC_OPEN_ORDER_PAGE	32971
Registration	IDC_REGISTRATION	32972
Check for Updates	IDC_CHECK_FOR_UPDATES	32973
MapForce Product Comparison	IDC_PRODUCT_COMPARISON	32955
Support Center	IDC_OPEN_SUPPORT_PAGE	32961
FAQ on the Web	IDC_OPEN_FAQ_PAGE	32962
Download Components and Free Tools	IDC_OPEN_COMPONENTS_PAGE	32963
MapForce on the Internet	IDC_OPEN_HOME_PAGE	32964
MapForce Training	IDC_OPEN_TRAINING_PAGE	32965
About MapForce	ID_APP_ABOUT	57664

# 17.7 Object Reference

#### Objects:

MapForceCommand<sup>[4433</sup> MapForceCommands<sup>[4453</sup> MapForceControl<sup>1456</sup> MapForceControlDocument<sup>[463]</sup> MapForceControlPlaceHolder<sup>[470]</sup>

To give access to standard MapForce functionality, objects of the **MapForce automation interface** can be accessed as well. See <u>MapForceControl.Application</u><sup>(457)</sup>, <u>MapForceControlDocument.Document</u><sup>(466)</sup> and <u>MapForceControlPlaceHolder.Project</u><sup>(477)</sup> for more information.

# 17.7.1 MapForceCommand

### **Properties:**

ID <sup>1454</sup>
Label <sup>1454</sup>
Name <sup>1454</sup>
IsSeparator <sup>1454</sup>
ToolTip <sup>1455</sup>
StatusText <sup>1455</sup>
Accelerator <sup>1453</sup>
SubCommands <sup>1455</sup>

#### Description:

A command object can be one of the following: an executable command, a command container (for example, a menu, submenu, or toolbar), or a menu separator. To determine what kind of information is stored in the current Command object, query its ID, IsSeparator, and SubCommands properties, as follows.

The Command object is	When
An executable command	<ul> <li>ID is greater than zero</li> <li>IsSeparator is false</li> <li>SubCommands is empty</li> </ul>
A command container	<ul> <li>ID is zero</li> <li>IsSeparator is false</li> <li>SubCommands contains a collection of Command objects.</li> </ul>
Separator	<ul> <li>ID is zero</li> <li>IsSeparator is true</li> </ul>

## 17.7.1.1 Accelerator

Property: Accelerator as string

Returns the accelerator key defined for the command. If the command has no accelerator key assigned, this property returns the empty string. The string representation of the accelerator key has the following format:

```
[ALT+][CTRL+][SHIFT+]key
```

Where key is converted using the Windows Platform SDK function GetKeyNameText.

## 17.7.1.2 ID

Property: ID as long

#### Description:

This property gets the unique identifier of the command. A command's ID is required to execute the command (using <u>Exec</u><sup>1460</sup>) or query its status (using <u>QueryStatus</u><sup>1460</sup>). If the command is a container for other commands (for example, a top-level menu), or a separator, the ID is 0.

## 17.7.1.3 IsSeparator

**Property:** IsSeparator as boolean

#### Description:

The property returns true if the command object is a menu separator; false otherwise. See also Command<sup>[1453]</sup>.

## 17.7.1.4 Label

#### Property: Label as string

#### Description:

This property gets the text of the command as it is displayed in the graphical user interface of MapForce. If the command is a separator, "Label" is an empty string. This property may also return an empty string for some toolbar commands that do not have any GUI text associated with them.

## 17.7.1.5 Name

#### Property: Name as string

#### Description:

This property gets the unique name of the command. This value can be used to get the icon file of the command, where it is available. The available icon files can be found in the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your MapForce installation.

## 17.7.1.6 StatusText

#### Property: Label as string

#### Description:

The status text is the text shown in the status bar of MapForce when the command is selected. It applies only to command objects that are not separators or containers of other commands; otherwise, the property is an empty string.

## 17.7.1.7 SubCommands

Property: SubCommands as Commands<sup>1455</sup>

#### Description:

The SubCommands property gets the collection of <u>Command</u><sup>(45)</sup> objects that are sub-commands of the current command. The property is applicable only to commands that are containers for other commands (menus, submenus, or toolbars). Such container commands have the ID set to 0, and the IsSeparator property set to false.

## 17.7.1.8 ToolTip

#### Property: ToolTip as string

#### Description:

This property gets the text that is shown as a tool-tip for each command. If the command does not have a tooltip text, the property returns an empty string.

## 17.7.2 MapForceCommands

Properties: Count<sup>1455</sup> Item<sup>1456</sup>

#### Description:

Collection of <u>Command</u><sup>[1453</sup> objects to get access to command labels and IDs of the MapForceControl. Those commands can be executed with the Exec<sup>[1460</sup> method and their status can be queried with <u>QueryStatus</u><sup>[1461</sup>].

## 17.7.2.1 Count

Property: Count as long

#### Description:

Number of Command<sup>[453]</sup> objects on this level of the collection.

## 17.7.2.2 Item

Property: Item (n as long) as Command [1453]

#### Description:

Gets the command with the index n in this collection. Index is 1-based.

## 17.7.3 MapForceControl

Properties: IntegrationLevel <sup>(458)</sup> Appearance <sup>(457)</sup> Application <sup>(457)</sup> BorderStyle <sup>(457)</sup> CommandsList <sup>(457)</sup> EnableUserPrompts <sup>(458)</sup> MainMenu <sup>(458)</sup> Toolbars <sup>(459)</sup>

#### Methods:

Open<sup>1460</sup> Exec<sup>1460</sup> QueryStatus<sup>1461</sup>

Events: OnUpdateCmdUI OnOpenedOrFocused OnCloseEditingWindow OnFileChangedAlert OnContextChanged OnDocumentOpened OnValidationWindowUpdated (463) (464) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (465) (

This object is a complete ActiveX control and should only be visible if the MapForce library is used in the Application Level mode.

CLSID: A38637E9-5759-4456-A167-F01160CC22C1 ProgID: Altova.MapForceControl

## 17.7.3.1 Properties

The following properties are defined:

IntegrationLevel<sup>(453)</sup> EnableUserPrompts<sup>(453)</sup> Appearance<sup>(457)</sup> BorderStyle<sup>(457)</sup>

Command related properties:

CommandsList MainMenu<sup>1458</sup> Toolbars<sup>1459</sup>

Access to MapForceAPI: Application<sup>1457</sup>

## 17.7.3.1.1 Appearance

**Property:** Appearance as short

Dispatch Id: -520

#### **Description:**

A value not equal to 0 displays a client edge around the control. Default value is 0.

## 17.7.3.1.2 Application

**Property:** Application as Application

#### Dispatch Id: 1

#### Description:

The Application property gives access to the Application object of the complete MapForce automation server API. The property is read-only.

## 17.7.3.1.3 BorderStyle

**Property:** BorderStyle as short

#### Dispatch Id: -504

Description:

A value of 1 displays the control with a thin border. Default value is 0.

## 17.7.3.1.4 CommandsList

**Property:** CommandList as <u>Commands</u><sup>1455</sup> (read-only)

#### Dispatch Id: 1004

#### Description:

This property returns a flat list of all commands defined available with MapForceControl. To get commands organized according to their menu structure, use <u>MainMenu</u><sup>[459]</sup>. To get toolbar commands, use <u>Toolbars</u><sup>[459]</sup>.

```
public void GetAllMapForceCommands()
{
    // Get all commands from the MapForce ActiveX control assigned to the current form
    MapForceControlLib.MapForceCommands commands = this.axMapForceControll.CommandList;
    // Iterate through all commands
    for (int i = 0; i < commands.Count; i++)
    {
        // Get each command by index and output it to the console
        MapForceControlLib.MapForceCommand cmd = axMapForceControll.CommandList[i];
        Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}</pre>
```

C# example

### 17.7.3.1.5 EnableUserPrompts

Property: EnableUserPrompts as boolean

#### Dispatch Id: 1006

#### Description:

Setting this property to false, disables user prompts in the control. The default value is true.

#### 17.7.3.1.6 IntegrationLevel

**Property:** IntegrationLevel **as** ICActiveXIntegrationLevel<sup>[472]</sup>

#### Dispatch Id: 1000

#### Description:

The IntegrationLevel property determines the operation mode of the control. See also Integration at Application Level 410 and Integration at Document Level 410 for more information.

Note: It is important to set this property immediately after the creation of the MapForceControl object.

#### 17.7.3.1.7 MainMenu

**Property:** MainMenu as <u>Command</u><sup>[453]</sup> (read-only)

#### Dispatch Id: 1003

#### Description:

This property provides information about the structure and commands available in the MapForceControl main menu, as a Command object. The Command object contains all available submenus of MapForce (for example "File", "Edit", "View" etc.). To access the submenu objects, use the SubCommands property of the MainMenu property. Each submenu is also a Command object. For each submenu, you can then further iterate through their

SubCommands property in order to get their corresponding child commands and separators (this technique may be used, for example, to create the application menu programmatically). Note that some menu commands act as containers ("parents") for other menu commands, in which case they also have a SubCommands property. To get the structure of all menu commands programmatically, you will need a recursive function, as illustrated for C# in <u>Retrieving Command Information</u><sup>(420)</sup>.

```
public void GetMapForceMenus()
{
    // Get the main menu from the MapForce ActiveX control assigned to the current form
    MapForceControlLib.MapForceCommand mainMenu = this.axMapForceControll.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
        MapForceControlLib.MapForceCommand menu = mainMenu.SubCommands[i];
        Console.WriteLine("{0} menu has {1} children items (including separators)",
        menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}</pre>
```

C# example

### 17.7.3.1.8 Toolbars

**Property:** Toolbars as <u>Commands</u><sup>[455]</sup> (read-only)

#### Dispatch Id: 1005

#### Description:

This property provides information about the structure of MapForceControl toolbars, as a Command object. The Command object contains all available toolbars of MapForce. To access the toolbars, use the SubCommands property of the Toolbars property. Each toolbar is also a Command object. For each toolbar, you can then further iterate through their SubCommands property in order to get their commands (this technique may be used, for example, to create the application's toolbars programmatically).

```
public void GetMapForceToolbars()
{
    // Get the application toolbars from the MapForce ActiveX control assigned to the
    current form
    MapForceControlLib.MapForceCommands toolbars = this.axMapForceControll.Toolbars;

    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)
    {
        MapForceControlLib.MapForceCommand toolbar = toolbars[i];
        Console.WriteLine();
        Console.WriteLine("The toolbar \"{0}\" has the following commands:",
        toolbar.Label);
        // Iterate through all commands of this toolbar</pre>
```

```
for (int j = 0; j < toolbar.SubCommands.Count; j++)
{
    MapForceControlLib.MapForceCommand cmd = toolbar.SubCommands[j];
    // Output only command objects that are not separators
    if (!cmd.IsSeparator)
    {
        Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
        ""));
    }
    }
}</pre>
```

C# example

## 17.7.3.2 Methods

The following methods are defined:

Open<sup>1460</sup> Exec<sup>1460</sup> QueryStatus<sup>1461</sup>

### 17.7.3.2.1 Exec

Method: Exec (nCmdID as long) as boolean

#### Dispatch Id: 6

#### Description:

This method calls the MapForce command with the ID nCmdID. If the command can be executed, the method returns true. To get a list of all available commands, use <u>CommandsList</u><sup>[457]</sup>. To retrieve the status of any command, use <u>QueryStatus</u><sup>[46]</sup>.

### 17.7.3.2.2 Open

Method: Open (strFilePath as string) as boolean

#### Dispatch Id: 5

#### Description:

The result of the method depends on the extension passed in the argument strFilePath. If the file extension is .sps, a new document is opened. If the file extension is .svp, the corresponding project is opened. If a different file extension is passed into the method, the control tries to load the file as a new component into the active document.

Do not use this method to load documents or projects when using the control in document-level integration mode. Instead, use <u>MapForceControlDocument.Open</u><sup>(46)</sup> and <u>MapForceControlPlaceHolder.OpenProject</u><sup>(47)</sup>.

## 17.7.3.2.3 QueryStatus

Method: QueryStatus (nCmdID as long) as long

#### Dispatch Id: 7

#### Description:

QueryStatus returns the enabled/disabled and checked/unchecked status of the command specified by nCmdID. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if QueryStatus returns 0 the command ID is not recognized as a valid MapForce command. If QueryStatus returns a value of 1 or 5, the command is disabled.

## 17.7.3.3 Events

The MapForceControl ActiveX control provides the following connection point events:

OnUpdateCmdUI<sup>[463]</sup> OnOpenedOrFocused<sup>[463]</sup> OnCloseEditingWindow<sup>[46]</sup> OnFileChangedAlert<sup>[462]</sup> OnContextChanged<sup>[462]</sup>

<u>OnDocumentOpened</u><sup>1462</sup> <u>OnValidationWindowUpdated</u><sup>1463</sup>

## 17.7.3.3.1 OnCloseEditingWindow

**Event:** OnCloseEditingWindow (i\_strFilePath as String) as boolean

#### Dispatch Id: 1002

#### Description:

This event is triggered when MapForce needs to close an already open document. As an answer to this event, clients should close the editor window associated with *i\_strFilePath*. Returning *true* from this event indicates that the client has closed the document. Clients can return *false* if no specific handling is required and MapForceControl should try to close the editor and destroy the associated document control.

### 17.7.3.3.2 OnContextChanged

**Event:** OnContextChanged (i\_strContextName as String, i\_bActive as bool) as bool

#### Dispatch Id: 1004

**Description:** This event is not used in MapForce

### 17.7.3.3.3 OnDocumentOpened

**Event:** OnDocumentOpened (objDocument as Document)

#### Dispatch Id: 1

#### Description:

This event is triggered whenever a document is opened. The argument objDocument is a Document object from the MapForce automation interface and can be used to query for more details about the document, or perform additional operations. When integrating on document-level, it is often better to use the event MapForceControlDocument.OnDocumentOpened [469] instead.

## 17.7.3.3.4 OnFileChangedAlert

**Event:** OnFileChangedAlert (i\_strFilePath as String) as bool

#### Dispatch Id: 1001

#### Description:

This event is triggered when a file loaded with MapForceControl is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if MapForce should handle it in its customary way, i.e. prompting the user for reload.

### 17.7.3.3.5 OnLicenseProblem

**Event:** OnLicenseProblem (i\_strLicenseProblemText as String)

#### Dispatch Id: 1005

#### Description:

This event is triggered when MapForceControl detects that no valid license is available for this control. In case of restricted user licenses this can happen some time after the control has been initialized. Integrators should use this event to disable access to this control's functionality. After returning from this event, the control will block access to its functionality (e.g. show empty windows in its controls and return errors on requests).

### 17.7.3.3.6 OnOpenedOrFocused

**Event:** OnOpenedOrFocused (i\_strFilePath as String, i\_bOpenWithThisControl as bool)

#### Dispatch Id: 1000

#### Description:

When integrating at application level, this event informs clients that a document has been opened, or made active by MapForce.

When integrating at document level, this event instructs the client to open the file *i\_strFilePath* in a document window. If the file is already open, the corresponding document window should be made the active window.

if i\_bOpenWithThisControl is true, the document must be opened with MapForceControl, since internal access is required. Otherwise, the file can be opened with different editors.

## 17.7.3.3.7 OnToolWindowUpdated

**Event:** OnToolWindowUpdated(pToolWnd as long )

Dispatch Id: 1006

#### Description:

This event is triggered when the tool window is updated.

### 17.7.3.3.8 OnUpdateCmdUI

**Event:** OnUpdateCmdUI ()

#### Dispatch Id: 1003

#### Description:

Called frequently to give integrators a good opportunity to check status of MapForce commands using <u>MapForceControl.QueryStatus</u><sup>[46]</sup>. Do not perform long operations in this callback.

### 17.7.3.3.9 OnValidationWindowUpdated

**Event:** OnValidationWindowUpdated ()

#### Dispatch Id: 3

#### Description:

This event is triggered whenever the validation output window is updated with new information.

## 17.7.4 MapForceControlDocument

#### Properties:

Appearance<sup>[465]</sup> BorderStyle<sup>[465]</sup> Document<sup>[465]</sup> IsModified<sup>[465]</sup> Path<sup>[465]</sup> ReadOnly<sup>[466]</sup>

#### Methods:

Exec<sup>1466</sup> <u>New</u><sup>1467</sup> <u>Open</u><sup>1467</sup> <u>QueryStatus</u><sup>1467</sup> <u>Reload</u><sup>1467</sup> <u>Save</u><sup>1467</sup> <u>SaveAs</u><sup>1468</sup>

#### Events:

OnDocumentOpened<sup>[469]</sup> OnDocumentClosed<sup>[469]</sup> OnModifiedFlagChanged<sup>[469]</sup> OnContextChanged<sup>[469]</sup> OnFileChangedAlert<sup>[469]</sup> OnActivate<sup>[469]</sup>

If the MapForceControl is integrated in the Document Level mode each document is displayed in an own object of type MapForceControlDocument. The MapForceControlDocument contains only one document at the time but can be reused to display different files one after another.

This object is a complete ActiveX control.

CLSID: DFBB0871-DAFE-4502-BB66-08CEB7DF5255 ProgID: Altova.MapForceControlDocument

## 17.7.4.1 Properties

The following properties are defined:

ReadOnly<sup>1466</sup> IsModified<sup>1465</sup> Path<sup>1465</sup> Appearance<sup>1465</sup> BorderStyle<sup>1465</sup>

Access to MapForceAPI:

### 17.7.4.1.1 Appearance

**Property:** Appearance as short

#### Dispatch Id: -520

#### Description:

A value not equal to 0 displays a client edge around the document control. Default value is 0.

## 17.7.4.1.2 BorderStyle

**Property:** BorderStyle as short

#### Dispatch Id: -504

**Description:** A value of 1 displays the control with a thin border. Default value is 0.

## 17.7.4.1.3 Document

Property: Document as Document

#### Dispatch Id: 1

#### Description:

The Document property gives access to the Document object of the MapForce automation server API. This interface provides additional functionality which can be used with the document loaded in the control. The property is read-only.

## 17.7.4.1.4 IsModified

Property: IsModified as boolean (read-only)

#### Dispatch Id: 1006

#### Description:

IsModified is *true* if the document content has changed since the last open, reload or save operation. It is *false*, otherwise.

## 17.7.4.1.5 Path

Property: Path as string

Sets or gets the full path name of the document loaded into the control.

## 17.7.4.1.6 ReadOnly

**Property:** ReadOnly as boolean

Dispatch Id: 1007

#### Description:

Using this property you can turn on and off the read-only mode of the document. If ReadOnly is true it is not possible to do any modifications.

## 17.7.4.2 Methods

The following methods are defined:

Document handling: <u>New</u> <u>Open</u> <u>467</u> <u>Reload</u> <u>467</u> <u>Save</u> <u>467</u> <u>Save</u> <u>467</u> <u>Save</u> <u>466</u>

Command Handling: <u>Exec</u><sup>1466</sup> <u>QueryStatus</u><sup>1467</sup>

### 17.7.4.2.1 Exec

Method: Exec (nCmdID as long) as boolean

Dispatch Id: 8

#### Description:

Exec calls the MapForce command with the ID nCmdID. If the command can be executed, the method returns true. This method should be called only if there is currently an active document available in the application.

To get commands organized according to their menu structure, use the <u>MainMenu</u> froperty of MapForceControl. To get toolbar commands, use the <u>Toolbars</u> froperty of the MapForceControl.

#### 17.7.4.2.2 New

Method: New () as boolean

This method initializes a new document inside the control.

## 17.7.4.2.3 Open

Method: Open (strFileName as string) as boolean

Dispatch Id: 1001

Description:

Open loads the file strFileName as the new document into the control.

## 17.7.4.2.4 QueryStatus

Method: QueryStatus (nCmdID as long) as long

#### Dispatch Id: 9

#### Description:

QueryStatus returns the enabled/disabled and checked/unchecked status of the command specified by nCmdID. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if <code>QueryStatus</code> returns 0 the command ID is not recognized as a valid MapForce command. If <code>QueryStatus</code> returns a value of 1 or 5 the command is disabled. The client should call the <code>QueryStatus</code> method of the document control if there is currently an active document available in the application.

## 17.7.4.2.5 Reload

Method: Reload () as boolean

Dispatch Id: 1002

**Description:** Reload updates the document content from the file system.

## 17.7.4.2.6 Save

Method: Save () as boolean

Save saves the current document at the location  $Path^{(466)}$ .

17.7.4.2.7 SaveAs

Method: SaveAs (strFileName as string) as boolean

Dispatch Id: 1004

**Description:** SaveAs sets Path<sup>465</sup> to *strFileName* and then saves the document to this location.

## 17.7.4.3 Events

The MapForceControlDocument ActiveX control provides following connection point events:

OnDocumentOpened OnDocumentClosed OnModifiedFlagChanged OnContextChanged OnFileChangedAlert OnActivate OnSetEditorTitle

## 17.7.4.3.1 OnActivate

**Event:** OnActivate ()

Dispatch Id: 1005

#### Description:

This event is triggered when the document control is activated, has the focus, and is ready for user input.

### 17.7.4.3.2 OnContextChanged

Event: OnContextChanged (i\_strContextName as String, i\_bActive as bool) as bool

Dispatch Id: 1004

Description: None

### 17.7.4.3.3 OnDocumentClosed

**Event:** OnDocumentClosed (objDocument as Document)

#### Dispatch Id: 1001

#### Description:

This event is triggered whenever the document loaded into this control is closed. The argument objDocument is a Document object from the MapForce automation interface and should be used with care.

## 17.7.4.3.4 OnDocumentOpened

**Event:** OnDocumentOpened (objDocument as Document)

#### Dispatch Id: 1000

#### Description:

This event is triggered whenever a document is opened in this control. The argument objDocument is a Document object from the MapForce automation interface, and can be used to query for more details about the document, or perform additional operations.

## 17.7.4.3.5 OnDocumentSaveAs

**Event:** OnContextDocumentSaveAs (i\_strFileName as String)

#### Dispatch Id: 1007

#### Description:

This event is triggered when this document gets internally saved under a new name.

### 17.7.4.3.6 OnFileChangedAlert

**Event:** OnFileChangedAlert () as bool

#### Dispatch Id: 1003

#### Description:

This event is triggered when the file loaded into this document control is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if MapForce should handle it in its customary way, i.e. prompting the user for reload.

## 17.7.4.3.7 OnModifiedFlagChanged

**Event:** OnModifiedFlagChanged (i\_bIsModified as boolean)

#### Dispatch Id: 1002

#### Description:

This event gets triggered whenever the document changes between modified and unmodified state. The parameter *i\_blsModifed* is *true* if the document contents differs from the original content, and *false*, otherwise.

## 17.7.4.3.8 OnSetEditorTitle

*Event:* OnSetEditorTitle ()

Dispatch Id: 1006

#### Description:

This event is being raised when the contained document is being internally renamed.

# 17.7.5 MapForceControlPlaceHolder

Properties available for all kinds of placeholder windows: <u>PlaceholderWindowID</u><sup>[47]</sup>

Properties for project placeholder window: <u>Project</u><sup>[47]</sup>

#### Methods for project placeholder window: <u>OpenProject</u> <u>CloseProject</u> [47]

The MapForceControlPlaceHolder control is used to show the additional MapForce windows like Overview, Library or Project window. It is used like any other ActiveX control and can be placed anywhere in the client application.

CLSID: FDEC3B04-05F2-427d-988C-F03A85DE53C2 ProgID: Altova.MapForceControlPlaceHolder

## 17.7.5.1 Properties

The following properties are defined:

<u>PlaceholderWindowID</u><sup>1471</sup>

Access to MapForceAPI: Project<sup>1471</sup>

## 17.7.5.1.1 Label

Property: Label as String (read-only)

Dispatch Id: 1001

#### Description:

This property gives access to the title of the placeholder. The property is read-only.

### 17.7.5.1.2 PlaceholderWindowID

**Property:** PlaceholderWindowID as <u>MapForceControlPlaceholderWindow</u><sup>[473]</sup>

#### Dispatch Id: 1

#### Description:

This property specifies which MapForce window should be displayed in the client area of the control. The PlaceholderWindowID can be set at any time to any valid value of the <u>MapForceControlPlaceholderWindow</u><sup>(473)</sup> enumeration. The control changes its state immediately and shows the new MapForce window.

### 17.7.5.1.3 Project

Project as Project (read-only)

#### Dispatch Id: 2

#### Description:

The Project property gives access to the Project object of the MapForce automation server API. This interface provides additional functionality which can be used with the project loaded into the control. The property will return a valid project interface only if the placeholder window has <u>PlaceholderWindowID</u> with a value of MapForceXProjectWindow (=3). The property is read-only.

## 17.7.5.2 Methods

The following method is defined:

<u>OpenProject</u><sup>(47)</sup> CloseProject<sup>(47)</sup>

## 17.7.5.2.1 OpenProject

Method: OpenProject (strFileName as string) as boolean Dispatch ld: 3 Description:

OpenProject loads the file strFileName as the new project into the control. The method will fail if the placeholder window has a PlaceholderWindowID<sup>[47]</sup> different to MapForceXProjectWindow (=3).

## 17.7.5.2.2 CloseProject

Method: CloseProject ()

CloseProject closes the project loaded by the control. The method will fail if the placeholder window has a PlaceholderWindowID<sup>[47]</sup> different to MapForceXProjectWindow (=3).

## 17.7.5.3 Events

The MapForceControlPlaceholder ActiveX control provides following connection point events:

OnModifiedFlagChanged<sup>1472</sup>

#### 17.7.5.3.1 OnModifiedFlagChanged

**Event:** OnModifiedFlagChanged (i\_blsModified as boolean)

Dispatch Id: 1

#### Description:

This event gets triggered only for placeholder controls with a <u>PlaceholderWindowID</u><sup>[47]</sup> of MapForceXProjectWindow (=3). The event is fired whenever the project content changes between modified and unmodified state. The parameter *i\_blsModifed* is *true* if the project contents differs from the original content, and *false*, otherwise.

### 17.7.5.3.2 OnSetLabel

**Event:** OnSetLabel(i\_strNewLabel as string)

Dispatch Id: 1000

**Description:** Raised when the title of the placeholder window is changed.

## 17.7.6 Enumerations

The following enumerations are defined:

```
ICActiveXIntegrationLevel<sup>1472</sup>
MapForceControlPlaceholderWindow<sup>1473</sup>
```

## 17.7.6.1 ICActiveXIntegrationLevel

Possible values for the IntegrationLevel 458 property of the MapForceControl.

ICActiveXIntegrationOnApplicationLevel = 0
ICActiveXIntegrationOnDocumentLevel = 1

# 17.7.6.2 MapForceControlPlaceholderWindow

This enumeration contains the list of the supported additional MapForce windows.

MapForceXNoWindow	=	-1
MapForceXLibraryWindow	=	0
MapForceXOverviewWindow	=	1
MapForceXValidationWindow	=	2
MapForceXProjectWindow	=	3
MapForceXDebuggerValuesWindow	=	4
MapForceXDebuggerContextWindow	=	5
MapForceXDebuggerPointsWindow	=	6

#### 18 **Annexes**

Ces annexes contiennent une information technique sur MapForce, ses aspects techniques et la licence. Elle fournit également une liste de termes clés spécifiques à MapForce et aux produits y afférents. La section est organisée en sous-sections comme suit :

- Notes de prise en charge •
- Informations processeur Données techniques •
- ٠
- Information de licence<sup>1568</sup> ٠

# 18.1 Notes de prise en charge

MapForce® est une application Windows de 32/64-bit exécutée sur les systèmes d'exploitation suivants :

- Windows 7 SP1 avec mise à jour de la plateforme, Windows 8, Windows 10, Windows 11
- Windows Server 2008 R2 SP1 avec mise à jour de la plateforme ou plus récent

La prise en charge 64-bit est disponible pour les éditions Enterprise et Professional.

MapForce est disponible en option en tant que plug-in pour les environnements de développement intégrés suivants :

- Visual Studio 2012/2013/2015/2017/2019/2022. Voir MapForce Plug-in pour Visual Studio
- Eclipse 2022-09, 2022-06, 2022-03, 2021-12. Voir <u>MapForce Plug-in pour Eclipse</u><sup>1004</sup>.

MapForce peut être intégré aux produits Microsoft Office comme suit :

- Il peut mapper des données de ou vers les bases de données Access. Pour des versions prises en charge, voir <u>Bases de données et MapForce</u><sup>156</sup>
- Il peut générer de la documentation de mappage dans Word 2000 ou versions ultérieures. Voir <u>Générer</u> et personnaliser la documentation de mappage<sup>916</sup>.

## **18.1.1 Sources et cibles prises en charge**

Lorsque vous modifiez le langage de transformation d'un mappage MapForce, certaines fonctions peuvent ne pas être prises en charge par ce langage spécifique. La table suivante résume la compatibilité de formats de mappage et langages de transformation dans **MapForce Professional Edition**.

Remarques :

• *Built-in* signifie que vous pouvez exécuter le mappage en cliquant sur l'onglet **Sortie** dans MapForce, ou l'exécuter avec MapForce Server.

Format de mappage		XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
XML <sup>1</sup>		۲	۲	۲	۲	۲	۲	۲	۲
CSV et texte						۲	۲	۲	۲
Fichiers binaires									۲
Bases de donnée s <sup>2</sup>	ADO					۲	۲		۲
	ADO.NET						۲		۲
	JDBC							۲	۲

Format de mappage		XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
	Native SQLite								۲
	Native PostgreS QL								۲
	ODBC					۲	۲	۲	۲

Notes de bas de page :

- 1. XML avec un traitement de signatures numériques est uniquement pris en charge par MapForce Enterprise Edition en utilisant BUILT-IN en tant que langage de transformation.
- 2. Les limites s'appliquent au type de base de données et à l'environnement cible. Pour plus d'informations, voir <u>Mappages de base de données dans des environnements d'exécution divers</u><sup>157</sup>.

# **18.1.2** Fonctions prises en charge dans le code généré

La table suivante recense les fonctions pertinentes pour la génération de code et l'étendue de la prise en charge dans chaque langage dans **MapForce Professional Edition**.

Fonction	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
Fournir des paramètres au mappage <sup>477</sup>	۲	۲	۲	۲	۲	۲	۲	۲
Fournir les noms de fichier d'entrée dynamiquement depuis le mappage	۲	۲	۲	۲	۲	۲	۲	۲
Fournir des noms de fichier de caractère générique en tant qu'entrée de mappage <sup>633</sup> 1		۲	۲	۲	۲	۲	۲	۲
<u>Générer les noms</u> <u>de fichier de sortie</u> <u>dynamiquement</u> <u>depuis le</u> <u>mappage</u>		۲	۲		۲	۲	۲	۲
Fonction	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
-----------------------------------------------------------------------------	-------------	-------------	-------------	--------	-----	----	------	--------------
Retourner des valeurs de string depuis le mappage	۲	۲	۲		۲	۲	۲	۲
Variables 492		۲	۲	۲	۲	۲	۲	۲
<u>Composants de</u> <u>tri</u> <sup>517</sup>		۲	۲	۲				۲
Fonctions de regroupement 540		۲	۲		۲	۲	۲	۲
Filtres 523	۲	۲	۲	۲	۲	۲	۲	۲
Composants Join <sup>505</sup>								۲
Composants Value- Map <sup>530</sup>	۲	۲	۲	۲	۲	۲	۲	۲
Valeurs par défaut et fonctions de nœud <sup>560</sup>								۲
Exceptions de mappage <sup>547</sup>		۲	۲	۲	۲	۲	۲	۲
Parsage de string et sérialisation <sup>904</sup> <sup>2</sup>								۲
<u>Noms de nœuds</u> dynamiques <sup>(851)</sup>		۲	۲	۲	۲	۲	۲	۲
Insertions bulk de base de données								۲
Base de données SQL SELECT sans paramètres d'entrée					۲	۲	۲	۲
Base de données SQL SELECT avec paramètres d'entrée <sup>382</sup>								۲
Procédures stockées dans								۲

Fonction	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
base de données <sup>396</sup>								
Gestion d'exception de base de données 423 3					۲	۲	۲	۲
Traçage de base de données et logging d'erreurs 435								•
<u>Générer des fichiers</u> <u>d'exécution</u> <u>MapForce</u> <u>Server</u> <sup>957</sup>								۲
<u>Déployer des</u> mappages sur <u>FlowForce</u> Server <sup>960</sup>								۲
Lire des données depuis des fichiers binaires <sup>471</sup>								•
Écrire des données depuis des fichiers binaires <sup>471</sup>								۲

Notes de bas de page :

- 1. XSLT 2.0, XSLT 3.0 et XQuery utilisent la fonction fn:collection. La mise en place dans les moteurs Altova XSLT 2.0, XSLT 3.0 et XQuery résolvent des caractères génériques. D'autres moteurs peuvent se comporter différemment.
- 2. Pour JSON, le parsage et la sérialisation sont en plus pris en charge dans et C#.
- La gestion d'exception de base de données est possible quand le langage de mappage est pris en charge par le pilote de base de données connecté actuellement, tel qu'indiqué dans la table précédente.

# **18.2** Information des moteurs

Cette section contient des informations concernant les fonctions spécifiques à la mise en place du Valideur XML Altova XML, Altova XSLT 1.0 Engine, Altova XSLT 2.0 Engine et Altova XQuery Engine.

# **18.2.1** Informations concernant le moteur XSLT et XQuery

Les moteurs XSLT et XQuery de MapForce 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 MapForce.

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 :

- <u>XSLT 1.0<sup>1479</sup></u>
- XSLT 2.0<sup>1480</sup>
- <u>XQuery 1.0<sup>1482</sup></u>

# 18.2.1.1 XSLT 1.0

Le moteur XSLT 1.0 de MapForce est conforme aux <u>Recommandations XSLT 1.0 du 16 novembre 1999</u> et aux <u>Recommandations XPath 1.0 du 16 novembre 1999</u> du World Wide Web Consortium (W3C's). Veuillez noter les informations suivantes concernant l'implémentation.

## Notes concernant l'implémentation

Lorsque l'attribut method de xs1: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, .

# 18.2.1.2 XSLT 2.0

Cette section :

- <u>Conformité du moteur</u><sup>1480</sup>
- <u>Rétrocompatibilité</u><sup>1480</sup>
- Espaces de nom
   <sup>1480</sup>
- <u>Compatibilité avec le schéma</u><sup>1431</sup>
- <u>Comportement spécifique à l'implémentation</u><sup>(48)</sup>

## Conformité

Le moteur XSLT 2.0 de MapForce est conforme aux <u>Recommandations XSLT 2.0 du 23 janvier 2007</u> et aux <u>Recommandations XPath 2.0 du 14 décembre 2010</u> du World Wide Web Consortium (W3C's).

## Rétrocompatibilité

Le moteur XSLT 2.0 est rétrocompatible. Le seul moment où la compatibilité rétroactive du moteur XSLT 2.0 prend effet est si vous utilisez le moteur XSLT 2.0 pour traiter une feuille de style XSLT 1.0. Veuillez noter qu'il peut y avoir des différences dans les sorties produites par le moteur XSLT 1.0 et la rétrocompatibilité du moteur XSLT 2.0.

# Espaces de nom

Votre feuille de style XSLT 2.0 devrait déclarer les espaces de noms suivants afin que vous puissiez utiliser les constructeurs de type et les fonctions disponibles dans XSLT 2.0. Les préfixes indiqués ci-dessous sont utilisés de manière conventionnelle ; vous pourriez utiliser les préfixes alternatifs si vous le souhaitez.

Nom d'espace de nom	Préfixe	Espace de nom URI
Types de schéma XML	xs:	http://www.w3.org/2001/XMLSchema
Fonctions XPath 2.0	fn:	http://www.w3.org/2005/xpath-functions

Généralement ces espaces de nom seront déclarés sur l'élément xsl:stylesheet ou xsl:transform, tel que montré dans la liste suivante :

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"</pre>
```

# </xsl:stylesheet>

Veuillez noter les points suivants :

- Le moteur XSLT 2.0 utilise l'espace de nom XPath 2.0 et Fonctions XQuery 1.0 (recensées dans la table ci-dessus) en tant que son **espace de noms de fonctions par défaut**. Ainsi, vous pouvez utiliser les fonctions XPath 2.0 et XSLT 2.0 dans votre feuille de style sans aucun préfixe. Si vous déclarez l'espace de nom des fonctions XPath 2.0 dans votre feuille de style avec un préfixe, vous pourrez utiliser en plus le préfixe attribué dans la déclaration.
- Lors de l'utilisation des constructeurs de types et des types provenant de l'espace de nom du Schéma XML, le préfixe utilisé dans la déclaration d'espace de nom doit être utilisé lors de l'appel du constructeur de type (par exemple, xs:date).
- Certaines fonctions XPath 2.0 portent le même nom que les types de données du schéma XML. Par exemple, pour les fonctions XPath fn:string et fn:boolean, il existe des types de données du schéma XML portant le même nom local : xs:string et xs:boolean. Donc si vous décidez d'utiliser l'expression XPath string('Hello'), l'expression évalue en tant que fn:string('Hello')et non pas en tant que xs:string('Hello').

# Compatibilité avec le schéma

Le moteur XSLT 2.0 est compatible avec le schéma. Vous pouvez ainsi utiliser des types de schéma définis par l'utilisateur et l'instruction xsl:validate.

# Comportement spécifique à l'implémentation

Ci-dessous, vous trouverez une description de la gestion du moteur XSLT 2.0 des aspects spécifiques à l'implémentation du comportement de certaines fonctions XSLT 2.0.

#### xsl:result-document

Les encodages pris en charge en supplément sont (les codes spécifiques à Altova) : x-base16tobinary et x-base64tobinary.

#### function-available

La fonction teste la disponibilité des fonctions in-scope (XSLT, XPath, et fonctions d'extension).

#### unparsed-text

L'attribut href accepte (i) les chemins relatifs pour les fichiers dans le dossier de base-uri, et (ii) les chemins absolus avec ou sans le protocole file://. De plus, les encodages pris en charge supplémentaires sont (les codes spécifiques Altova) : x-binarytobase16 et x-binarytobase64.

#### unparsed-text-available

L'attribut href accepte (i) les chemins relatifs pour les fichiers dans le dossier de base-uri, et (ii) les chemins absolus avec ou sans le protocole file://. De plus, les encodages pris en charge supplémentaires sont (les codes spécifiques Altova) : x-binarytobase16 et x-binarytobase64.

**Note :** les valeurs d'encodage suivantes, qui ont été mises en place dans les versions plus anciennes du produit prédécesseur de RaptorXML, AltovaXML, sont désormais obsolètes : basel6tobinary, base64tobinary, binarytobasel6 et binarytobase64.

# 18.2.1.3 XQuery 1.0

#### Cette section :

- <u>Conformité du moteur</u><sup>1482</sup>
- <u>Compatibilité du schéma (Schema awareness)</u><sup>1482</sup>
- Encodage<sup>1482</sup>
- Espaces de nom<sup>1480</sup>
- XML source et validation<sup>1483</sup>
- <u>Contrôle de type statique et dynamique</u><sup>1484</sup>
- <u>Modules bibliothèque</u><sup>1484</sup>
- Modules externes
   <sup>1484</sup>
- <u>Collations</u><sup>1484</sup>
- Précision des données numériques<sup>1485</sup>
- Prise en charge des instructions XQuery<sup>1485</sup>

#### Conformité

Le moteur XQuery 1.0 de MapForce est conforme à la <u>Recommandation XQuery 1.0 du 14 décembre 2010</u> du World Wide Web Consortium (W3C's). Le standard XQuery accorde un pouvoir discrétionnaire concernant la mise en place de nombreuses fonctions. Ci-dessous, vous trouverez une liste expliquant comment le moteur XQuery 1.0 implémente ces fonctions.

#### Compatibilité avec le schéma

Le moteur XQuery 1.0 est schema-aware.

#### Encodage

Les encodages de caractères UTF-8 et UTF-16 sont pris en charge.

#### Espaces de nom

Les URI d'espace de nom suivant et leurs liaisons associées sont prédéfinies.

Nom d'espaces de nom	Préfixe	URI Espace de noms
Types de schéma XML	xs:	http://www.w3.org/2001/XMLSchema
Instance de schéma	xsi:	http://www.w3.org/2001/XMLSchema-instance
Fonctions intégrées	fn:	http://www.w3.org/2005/xpath-functions
Fonctions locales	local:	http://www.w3.org/2005/xquery-local-functions

Veuillez noter les points suivants :

- Le moteur XQuery 1.0 Engine reconnaît les préfixes recensés ci-dessus comme étant liés aux espaces de noms correspondants.
- Étant donné que l'espace de noms des fonctions intégrées recensé ci-dessus est l'espace de noms des fonctions par défaut dans XQuery, le préfixe fn: ne doit pas nécessairement être utilisé lorsque des fonctions intégrées sont invoquées (par exemple, string("Hello") appellera la fonction fn:string). Néanmoins, le préfixe fn: peut être utilisé pour appeler une fonction intégrée sans avoir à déclarer l'espace de noms dans le prologue query (par exemple : fn:string("Hello")).
- Vous pouvez changer l'espace de noms des fonctions par défaut en déclarant l'expression default function namespace dans le prologue de de requête.
- En cas d'utilisation des types depuis l'espace de noms du Schéma XML, le préfixe xs: peut être utilisé sans devoir déclarer explicitement les espace de noms et lier ces préfixes dans le prologue de requête. (Exemple: xs:date and xs:yearMonthDuration.) Si vous souhaitez utiliser d'autres préfixes pour l'espace de noms du schéma XML, cela doit être déclaré explicitement dans le prologue de requête. (Exemple: declare namespace alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04").)
- Veuillez noter que les types de données untypedAtomic, dayTimeDuration, et yearMonthDuration ont été déplacés, avec les CR de 23 January 2007, depuis l'espace de noms des Types de données XPath vers l'espace de noms du schéma XML, donc : xs:yearMonthDuration.

Si des espaces de noms pour les fonctions, les constructeurs de type, les tests de nœud, etc. sont mal attribués, une erreur est rapportée. Veuillez noter, néanmoins, que certaines fonctions portent le même nom que les types de données de schéma, par ex. fn:string et fn:boolean. (Les deux xs:string et xs:boolean sont définis.) Le préfixe d'espace de noms détermine si la fonction ou le constructeur de type est utilisé.

# XML document de source et validation

Les documents XML utilisés dans l'exécution d'un document XQuery avec le moteur XQuery 1.0 doit être bien formé. Néanmoins, ils ne doivent pas être valides conformément à un schéma XML. Si le fichier n'est pas valide, le fichier invalide est chargé sans information de schéma. Si le fichier XML est associé avec un schéma interne et est valide conformément à ce schéma, l'information de validation post-schéma sera générée pour les données XML et sera utilisée pour l'évaluation de requête.

#### Contrôle de type statique et dynamique

La phase d'analyse statique contrôle les aspects de la requête comme la syntaxe, si des références externes existent (par ex. pour les modules), si des fonctions et des variables invoquées sont définies, etc. Si une erreur est détectée dans la phase de l'analyse statique, elle sera rapportée et l'exécution sera stoppée.

Le contrôle de type dynamique est effectué lors de l'exécution, lorsque la requête est réellement exécutée. Si un type est incompatible avec les exigences d'une opération, une erreur sera rapportée. Par exemple, l'expression xs:string("1") + 1 retourne une erreur parce que l'opération d'édition ne peut pas être effectuée sur un opérande de type xs:string.

#### Modules de bibliothèque

Les modules de Bibliothèque stockent les fonctions et les variables de manière à ce qu'elles puissent être réutilisées. Le moteur XQuery 1.0 prend en charge des modules qui sont stockés **dans un seul fichier XQuery externe**. Un tel fichier de module doit contenir une déclaration module dans son prologue, qui associe un espace de noms cible. Voici un module d'exemple :

```
module namespace libns="urn:module-library";
declare variable $libns:company := "Altova";
declare function libns:webaddress() { "http://www.altova.com" };
```

Toutes les fonctions et les variables déclarées dans le module font partie de l'espace de noms associé au module. Celui-ci est utilisé en l'important dans un fichier XQuery avec l'instruction import module se trouvant dans le prologue de requête. L'instruction import module importe uniquement les fonctions et les variables déclarées directement dans le fichier de module de bibliothèque. Comme suit :

```
import module namespace modlib = "urn:module-library" at "modulefilename.xq";
if ($modlib:company = "Altova")
then modlib:webaddress()
else error("No match found.")
```

# Fonctions externes

Les fonctions externes ne sont pas prises en charge, c.à.d. dans les expressions utilisant le mot-clé external, comme dans :

declare function hoo(\$param as xs:integer) as xs:string **external**;

#### Collations

La collation par défaut est la collation de point de code Unicode, qui compare les chaînes sur la base de leur point de code Unicode. Les autres collations prises en charge sont les <u>collations ICU</u> recensées <u>ici</u><sup>[485]</sup>. Pour utiliser une collation spécifique, fournir son URI tel que donné dans la <u>liste des collations prises en charge</u><sup>[485]</sup>. 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.

# **18.2.2 Fonctions XSLT et XPath/XQuery**

Cette section réunit les fonctions d'extension Altova et d'autres fonctions d'extension qui peuvent être utilisées dans les expressions XPath et/ou XQuery. Les fonctions d'extension Altova peuvent être utilisées avec les moteurs XSLT et XQuery d'Altova, et elles offrent des fonctions supplémentaires à celles disponibles dans les bibliothèques de fonctions définies dans les standards W3C.

## Points généraux

Les points généraux suivants devraient être notés : Normaltrue

- Normaltrue
- Les fonctions provenant des librairies des fonctions essentielles définies dans les spécifications W3C peuvent être appelées sans un préfixe. Cela est dû au fait que les moteurs XSLT et XQuery lisent les fonctions non préfixées comme faisant partie d'un espace de nom des fonctions par défaut qui est celui spécifié dans les spécifications des fonctions XPath/XQuery <a href="http://www.w3.org/2005/xpath-functions">http://www.w3.org/2005/xpath-functions</a>. 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 <u>collations ICU</u> regroupées ci-dessous : pour utiliser une collation spécifique, il faut fournir son URI tel que spécifié dans la table ci-dessous. Toute comparaison de chaîne, y compris en ce qui concerne les fonctions max et min, sera effectuée conformément à la collation spécifiée. Si l'option de collation n'est pas spécifiée, la collation de point de code Unicode par défaut sera utilisée.

Langue	URI
da: Danois	da_DK
de: Allemand	de_AT, de_BE, de_CH, de_DE, de_LI, de_LU
en: Anglais	en_AS, en_AU, en_BB, en_BE, en_BM, en_BW, en_BZ, en_CA, en_GB, en_GU, en_HK, en_IE, en_IN, en_JM, en_MH, en_MP, en_MT, en_MU, en_NA, en_NZ, en_PH, en_PK, en_SG, en_TT, en_UM, en_US, en_VI, en_ZA, en_ZW
es: Espagnol	es_419, es_AR, es_BO, es_CL, es_CO, es_CR, es_DO, es_EC, es_ES, es_GQ, es_GT, es_HN, es_MX, es_NI, es_PA, es_PE, es_PR, es_PY, es_SV, es_US, es_UY, es_VE
fr: Français	<pre>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</pre>
it: Italien	it_CH, it_IT
ja: <b>Japonais</b>	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

#### <u>Axe du nom d'espace</u>

L'axe du nom d'espace est devenu obsolète dans XPath 2.0. Néanmoins, l'utilisation de l'axe du nom d'espace est prise en charge. Pour accéder aux information de l'espace de nom, avec des mécanismes XPath 2.0, utilisez les fonctions in-scope-prefixes(), namespace-uri() et namespace-uri-for-prefix().

# 18.2.2.1 Fonctions d'extension Altova

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova, http://www.altova.com/xslt-extensions**, et sont indiquées dans cette section par le préfixe **altova:**, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Les fonctions définies dans les spécifications de Fonctions XPath/XQuery de W3C peuvent être utilisées dans : (i) les expressions XPath dans un contexte XSLT, et (ii) dans les expressions XQuery dans un document XQuery. Dans cette documentation, nous indiquons les fonctions à utiliser dans le contexte précédent (XPath dans XSLT) avec un symbole **xP** et les appelons fonctions XPath ; les fonctions qui peuvent être utilisées dans le contexte à venir (XQuery) sont indiquées avec un symbole **xQ** ; elles fonctionnent en tant que fonctions XQuery. Les spécifications XSLT de W3C —pas les spécifications de Fonctions XPath/XQuery —définissent également les fonctions qui peuvent être utilisées dans des expressions XPath dans des documents XSLT. Ces fonctions sont marquées avec un symbole **xSLT** et sont appelées fonctions XSLT. Les versions XPath/XQuery et XSLT dans lesquelles une fonction peut être utilisée sont indiquées dans la description de la fonction (*voir symboles ci-dessous*). Les fonctions provenant des librairies de fonction XPath/XQuery et XSLT sont recensées dans un préfixe. Les fonctions d'extension provenant d'autres librairies, comme les fonctions d'extension Altova, sont regroupés avec un préfixe.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

# Fonctions XSLT<sup>1488</sup>

Les fonctions XSLT peuvent uniquement être utilisées dans des expressions XPath dans un contexte XSLT (comme les fonctions current-group() ou key() de XSLT 2.0). Ces fonctions ne sont pas prévues pour, et ne fonctionneront pas dans un contexte non-XSLT (par exemple, dans un contexte XQuery). Veuillez noter que les fonctions XSLT pour XBRL peuvent être utilisées uniquement avec les éditions des produits Altova qui ont une prise en charge XBRL.

# Fonctions XPath/XQuery

Les fonctions XPath/XQuery peuvent être utilisées tous les deux dans les expressions XPath dans les contextes XSLT et dans les expressions XQuery :

• Date/Heure<sup>1491</sup>

- <u>Géolocalisation</u><sup>1508</sup>
- Liée à l'image<sup>1518</sup>
- <u>Numérique</u><sup>1522</sup>
- <u>Séquence</u><sup>1544</sup>
- Chaîne<sup>1553</sup>
- Divers 1559

# 18.2.2.1.1 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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

## Fonctions standard

distinct-nodes [altova:]

altova:distinct-nodes(*node()*\*) asnode()\* XSLT1 XSLT2 XSLT3 Prend un ensemble d'un ou de plusieurs nœuds en tant que son entrée et retourne le même ensemble moins les nœuds avec des valeurs dupliquées. La comparaison s'effectue en utilisant la fonction XPath/XQuery fn:deep-equal.

<u>Exemples</u>

- 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 : <u>altova:evaluate('//Name[1]'</u>) retourne les contenues du premier élément Name dans le document. Veuillez noter que l'expression //Name[1] est passée en tant que chaîne en l'enfermant dans des guillemets simples.

La fonction altova:evaluate peut prendre des arguments supplémentaires en option. Ces arguments sont les valeurs des variables in-scope qui portent les noms p1, p2, p3... pN. Veuillez noter les points suivants concernant l'utilisation : (i) Les variables doivent être définies avec les noms de la formule pX, lorsque x est un entier ; (ii) les arguments de la fonction altova:evaluate (*voir signature ci-dessus*), à partir du deuxième argument, fournissent les valeurs de la variables, avec la séquence des arguments correspondant à la séquence des variables classées numériquement : p1 à pN: le deuxième argument sera la valeur de la variable p1, le troisième argument celui de la variable p2, etc. ; (iii) Les valeurs de variable doivent être de type item\*.

```
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 "//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.



## 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.
```

# [<u>Haut</u><sup>1488</sup>]

# 18.2.2.1.2 Fonctions XPath/XQuery : Date et heure

Les fonctions d'extension date/heure d'Altova peuvent être utilisées dans les expressions XPath et XQuery et fournissent des fonctions supplémentaires pour le traitement des données contenues en tant que les types de données de date et d'heures variés de XML Schema. Les fonctions dans cette section peuvent être utilisées avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova. Ils sont disponibles dans les contextes XPath/XQuery.

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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section 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
  - <u>Ajouter une durée à xs:dateTime et retourner xs:dateTime</u>
     <sup>1492</sup>
  - Ajouter une durée à xs:date et retourner xs:date
  - Ajouter une durée à xs:time et retourner à xs:time<sup>1496</sup>
  - Formater et récupérer des durées<sup>1495</sup>
  - Supprimer des fuseaux horaires de fonctions qui génèrent des date/heures actuels<sup>[497]</sup>
  - Retourne un jour de la semaine en tant qu'un entier à partir de la date
  - <u>Retourne le numéro de la semaine en tant qu'un entier à partir de la date</u>
  - <u>Construire le type de date, d'heure ou de durée à partir des composants lexicaux de chaque type</u>
  - <u>Construire le type de date, dateHeure ou heure à partir de l'entrée de chaîne 1603</u>
  - Fonctions liées à l'âge<sup>1505</sup>
  - Fonctions Epoch time (heure Unix)<sup>1506</sup>
- Recensées par ordre alphabétique

altova:add-days-to-date<sup>1494</sup> altova:add-days-to-dateTime<sup>1492</sup>

altova:add-hours-to-dateTime<sup>1492</sup> altova:add-hours-to-time<sup>149</sup> altova:add-minutes-to-dateTime<sup>1492</sup> altova:add-minutes-to-time altova:add-months-to-date<sup>1494</sup> altova:add-months-to-dateTime<sup>1492</sup> altova:add-seconds-to-dateTime<sup>1492</sup> altova:add-seconds-to-time altova:add-years-to-date<sup>1494</sup> altova:add\_vears-to-dateTime<sup>1492</sup> altova:age<sup>1505</sup> altova:age-details<sup>1505</sup> altova:build-date<sup>1502</sup> altova:build-duration<sup>1502</sup> altova:build-time<sup>150</sup> <u>altova:current-dateTime-no-TZ</u><sup>1497</sup> altova:current-date-no-TZ<sup>149</sup> altova:current-time-no-TZ<sup>1497</sup> altova:date-no-TZ<sup>149</sup> altova:dateTime-from-epoch<sup>1506</sup> altova:dateTime-from-epoch-no-TZ<sup>1506</sup> 149 altova:dateTime-no-TZ altova:days-in-month<sup>1498</sup> <u>altova:epoch-from-da</u>teTime<sup>1506</sup> altova:hours-from-dateTimeDuration-accumulated altova:minutes-from-dateTimeDuration-accumulated<sup>[498</sup> altova:seconds-from-dateTimeDuration-accumulated<sup>1498</sup> altova:format-duration altova:parse-date 1503 altova:parse-dateTime<sup>1503</sup> altova:parse-duration<sup>1495</sup> altova:parse-time altova:time-no-TZ<sup>1497</sup> altova:weekday-from-date<sup>1500</sup> altova:weekday-from-dateTime altova:weeknumber-from-date altova:weeknumber-from-dateTime<sup>1501</sup>

[ <u>Haut</u><sup>1491</sup> ]

# 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.ss. Il s'agit d'une concaténation des formats xs:date et xs:time séparés par la lettre T. Un suffixe de fuseau horaire +01:00 (par exemple) (+01:00, par exemple) est optionnel.

#### add-years-to-dateTime [altova:]

altova:add-years-to-dateTime(DateTime as xs:dateTime, Years as xs:integer)
asxs:dateTime XP3.1 XQ3.1

Ajoute une durée en années à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre d'années à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

#### <u>Exemples</u>

- 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.

<u>Exemples</u>

- 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.

<u>Exemples</u>

- 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.

<u>Exemples</u>

- altova:add-hours-to-dateTime(xs:dateTime("2014-01-15T13:00:00"), 10) retourne 2014-01-15T23:00:00
- altova:add-hours-to-dateTime(xs:dateTime("2014-01-15T13:00:00"), -8) retourne 2014-01-15T05:00:00

add-minutes-to-dateTime [altova:]

```
altova:add-minutes-to-dateTime(DateTime as xs:dateTime, Minutes as xs:integer)
asxs:dateTime XP3.1 XQ3.1
```

Ajoute une durée en minutes à xs:dateTime (voir exemples ci-dessous). Le deuxième argument est le nombre of minutes à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

Exemples

- altova:add-minutes-to-dateTime(xs:dateTime("2014-01-15T14:10:00"), 45) retourne 2014-01-15T14:55:00
- altova:add-minutes-to-dateTime(xs:dateTime("2014-01-15T14:10:00"), -5) retourne 2014-01-15T14:05:00

#### add-seconds-to-dateTime [altova:]

altova:add-seconds-to-dateTime(DateTime as xs:dateTime, Seconds as xs:integer)
asxs:dateTime XP3.1 XQ3.1

Ajoute une durée en secondes à xs:dateTime (*voir exemples ci-dessous*). Le deuxième argument est le nombre de secondes à être ajouté à xs:dateTime fourni en tant que le premier argument. Le résultat est de type xs:dateTime.

- <u>Exemples</u>
  - 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



#### 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) refourne 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. <u>Exemples</u>

```
• 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



#### 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 XO3.1

Formate une durée qui est soumise en tant que le premier argument, selon une chaîne d'image soumise en tant que le second argument. La sortie est une chaîne de texte formatée conformément à la chaîne d'image.

- Exemples
  - altova:format-duration(xs:duration("P2DT2H53M11.7S"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") retourne "Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"
  - altova:format-duration(xs:duration("P3M2DT2H53M11.7S"), "Months:[M01] Days:[D01] Hours: [H01] Minutes: [m01]") retourne "Months: 03 Days: 02 Hours: 02 Minutes: 53"
- parse-duration [altova:]

altova:parse-duration(InputString as xs:string, Picture as xs:string) asxs:duration XP3.1 XO3.1

Prend un patterned string en tant que le premier argument et une chaîne image en tant que le second argument. La chaîne d'entrée est parsée sur la base de la chaîne d'image et une xs:duration est retournée.

Exemples

- altova:parse-duration("Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"), "Days: [D01] Hours: [H01] Minutes: [m01] Seconds: [s01] Fractions: [f0]") retourne "P2DT2H53M11.7S"
- altova:parse-duration("Months:03 Days:02 Hours:02 Minutes:53 Seconds:11

```
Fractions:7", "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") retourne
"P3M2DT2H53M"
```



#### 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.

<u>Exemples</u>

- 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.

<u>Exemples</u>

- 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



## 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

 $\label{eq:cetter} \begin{array}{l} \mbox{Cette fonction ne prend aucun argument. Elle supprime la partie fuseau horaire de {\tt current-dateTime()} \\ \mbox{(qui est la date-heure actuelle selon l'horloge système) et retourne une valeurxs:dateTime.} \end{array}$ 

<u>Exemples</u>

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.

- <u>Exemples</u>
  - 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.

<u>Exemples</u>

• altova:date-no-TZ(xs:date("2014-01-15+01:00"))) retourne 2014-01-15

dateTime-no-TZ [altova:]

altova:dateTime-no-TZ(InputDateTime as xs:dateTime) asxs:dateTime XP3.1 XQ3.1 Cette fonction prend un argument xs:dateTime, en supprime la partie fuseau horaire, et retourne une valeur xs:dateTime. Veuillez noter que ni la date ni l'heure n'est modifiée.

#### Exemples

```
• altova:dateTime-no-TZ(xs:date("2014-01-15T14:00:00+01:00")) retourne 2014-01-
15T14:00:00
```

time-no-TZ [altova:]

```
• altova:time-no-TZ(xs:time("14:00:00+01:00"))) retourne 14:00:00
```



#### Retourne le nombre de jours, d'heures, de minutes, de secondes des durées XP3.1 XQ3.1

Ces fonctions retournent le nombre de jours dans un mois, et le nombre d'heures, de minutes et de secondes, respectivement depuis les durées.

days-in-month [altova:]

altova:days-in-month(Year as xs:integer, Month as xs:integer) asxs:integer XP3.1 XQ3.1 Retourne le nombre de jours dans le mois spécifié. Le mois est spécifié avec les arguments Year et Month.

<u>Exemples</u>

- altova:days-in-month(2018, 10) retourne 31
- altova:days-in-month(2018, 2) retourne 28
- altova:days-in-month(2020, 2) retourne 29

#### hours-from-dayTimeDuration-accumulated

#### altova:hours-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) asxs:integer XP3.1 XQ3.1

Retourne le nombre total d'heures dans la durée soumise par l'argument DayAndTime (qui est de type xs:duration). Les heures dans les composants Day et Time sont additionnés pour donner un résultat qui est un entier. Le décompte d'une nouvelle heure dure uniquement 60 minutes complètes. Des durées négatives entraînent une valeur d'heures négative.

Exemples

- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5D")) retourne 120, qui est le nombre total d'heures dans 5 jours.
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H")) retourne 122, qui est le nombre total d'heures dans 5 jours plus 2 heures.
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H60M")) retourne

Information des moteurs 1499

123, qui est le nombre total d'heures dans 5 jours plus 2 heures et 60 mins.

- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H119M")) refourne 123, qui est le nombre total d'heures dans 5 jours plus 2 heures et 119 mins.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H120M")) refourne 124,qui est le nombre total d'heures dans 5 jours plus 2 heures et 120 mins.
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("-P5DT2H")) retourne -122

#### minutes-from-dayTimeDuration-accumulated

altova:minutes-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) asxs:integer XP3.1 XQ3.1

Retourne le nombre total de minutes dans la durée soumise par l'argument DayAndTime (qui est de type xs:duration). Les minutes dans les composants Day et Time sont additionnés pour donner un résultat qui est un entier. Des durées négatives entraînent une valeur de minute négative.

<u>Exemples</u>

- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT60M")) retourne 60
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT1H")) retourne 60, qui est le nombre total de minutes dans une heure.
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT1H40M")) refourne 100
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("P1D")) retourne 1440, qui est le nombre total de minutes dans un jour.
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("-P1DT60M")) retourne 1500
- seconds-from-dayTimeDuration-accumulated

# altova:seconds-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) asxs:integer XP3.1 XQ3.1

Retourne le nombre total de minutes dans la durée soumise par l'argument DayAndTime (qui est de type xs:duration). Les minutes dans les composants Day et Time sont additionnés pour donner un résultat qui est un entier. Des durées négatives entraînent une valeur de minute négative .

<u>Exemples</u>

- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1M")) retourne 60, qui est le nombre total de secondes dans une minute.
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1H")) retourne 3600, qui est le nombre total de secondes dans une heure.
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1H2M")) retOurne 3720
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("P1D")) retourne 86400, qui est le nombre total de secondes dans un jour.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("-P1DT1M")) **retourne** 86460

### Retourne le jour de la semaine à partir de xs:dateTime ou xs:date xP3.1 xQ3.1

Ces fonctions retournent le jour de la semaine (en tant qu'entier) depuis xs:dateTime ou xs:date. Les jours de la semaine sont numérotés (format américain) de 1 à 7, avec Sunday=1. Dans le format européen, la semaine commence par Lundi (=1). Dans le format américain elle commence par Sunday=1. Configurer en utilisant l'entier 0 et où un entier est accepté pour indiquer le format.

#### weekday-from-dateTime [altova:]

altova:weekday-from-dateTime(DateTime as xs:dateTime) asxs:integer XP3.1 XQ3.1 Prend une date-avec-heure en tant que son seul argument et retourne le jour de la semaine de cette date sous forme d'un entier. Les jours de la semaine sont numérotés en commençant avec Sunday=1. Si le format européen est requis (où Monday=1), utiliser l'autre signature de cette fonction (voir signature suivante ci-dessous).

Exemples

• **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00")) retourne 2, ce qui indique un lundi.

# altova:weekday-from-dateTime(DateTime as xs:dateTime, Format as xs:integer) asxs:integer XP3.1 XQ3.1

Prend une date-avec-heure en tant que son premier argument et retourne le jour de la semaine de cette date sous forme d'un entier. Si le second argument (entier) est 0, les jours de la semaine sont numérotés de 1 à 7 en commençant avec Sunday=1. Si le second argument est un entier différent de 0, alors Monday=1. S'il n'y a pas de second argument, la fonction est lue comme possédant l'autre signature de cette fonction (*voir signature précédente*).

∃ <u>Exemples</u>

- 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).

- <u>Exemples</u>
  - 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.

#### Retourne le nombre de la semaine à partir de xs:dateTime ou xs:date xP2 xQ1 xP3.1 xQ3.1

Ces fonctions retournent le numéro de la semaine (en tant qu'un entier) depuis xs:dateTime or xs:date. La numérotation des semaines est disponible dans les formats de calendrier US, ISO/European et Islamiques. La numérotation des semaines est différente dans ces formats de calendrier parce que la semaine est considérée démarrer avec un jour différent selon le format (dimanche pour le format US, lundi pour le format ISO/Européen, et samedi dans le format islamique).

#### 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"), 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.

[<u>Haut</u><sup>1491</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 premier, second et troisième arguments sont respectivement l'année, le mois et la date. Ils sont combinés pour construire une valeur de type xs:date. Les valeurs de l'entier doivent se situer dans le cadre de la plage correcte de cette partie de la date. Par exemple, le deuxième argument (pour la partie du mois) ne devrait pas être supérieur à 12.

Exemples

- altova:build-date(2014, 2, 03) retourne 2014-02-03
- build-time [altova:]

altova:build-time(Hours as xs:integer, Minutes as xs:integer, Seconds as xs:integer) asxs:time XP3.1 XQ3.1

Les premiers, seconds et troisièmes arguments sont, respectivement, les valeurs d'heure (0 to 23), de minutes (0 to 59) et de secondes (0 to 59). Ils sont combinés pour construire une valeur de type xs:time. Les valeurs des entiers doivent se trouver dans le cadre de la plage correcte de cette partir de temps en particulier. Par exemple, le deuxième argument (Minutes) ne devrait pas être supérieur à 59. Pour ajouter une partie fuseau horaire à la valeur, utiliser l'autre signature de cette fonction (*voir signature suivante*).

• 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.

• altova:build-time(23, 4, 57, '+1') retourne 23:04:57+01:00

#### build-duration [altova:]

altova:build-duration(Years as xs:integer, Months as xs:integer) asxs:yearMonthDuration XP3.1 XQ3.1

Prend deux arguments pour construire une valeur de type xs:yearMonthDuration. Les premiers arguments fournissent la partie Years de la valeur de durée, alors que le deuxième argument fournie la partie Months. Si le deuxième argument (Months) est supérieur ou égale à 12, alors l'entier est divisé par 12; le quotient est ajouté au premier argument pour fournir la partie Years de la valeur de durée alors que le reste (de la division) fournit la partie Months. Pour construire une durée de type xs:dayTimeDuration., voir la signature suivante.

#### <u>Exemples</u>

- 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 utilisé pour le calcul de la valeur de durée totale. Pour construire une durée de type xs:yearMonthDuration., voir la signature précédente.

<u>Exemples</u>

- 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

[ <u>Haut</u><sup>1491</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.

- D Jour
- M Mois
- Y Année

Le modèle dans DatePattern doit correspondre au modèle dans Date. Puisque la sortie est de type xs:date, la sortie aura toujours le format lexical YYYY-MM-DD.

Exemples

- altova:parse-date(xs:string("09-12-2014"), "[D]-[M]-[Y]") retourne 2014-12-09
- altova:parse-date(xs:string("09-12-2014"), "[M]-[D]-[Y]") retourne 2014-09-12
- altova:parse-date("06/03/2014", "[M]/[D]/[Y]") retourne 2014-06-03
- altova:parse-date("06 03 2014", "[M] [D] [Y]") retourne 2014-06-03
- altova:parse-date("6 3 2014", "[M] [D] [Y]") retourne 2014-06-03

parse-dateTime [altova:]

# altova:parse-dateTime(DateTime as xs:string, DateTimePattern as xs:string) asxs:dateTime XP2 XQ1 XP3.1 XQ3.1

Retourne la chaîne d'entrée DateTime en tant que valeur xs:dateTime. Le deuxième argument DateTimePattern spécife le modèle (séquence des composants) de la chaîne d'entrée. DateTimePattern est décrit avec les spécificateurs de composants regroupés ci-dessous et avec les séparateurs de composant qui peuvent être n'importe quel caractère. Voir les exemples ci-dessous.

D	Date
м	Mois
Y	Année
н	Heure
m	Minutes
S	Secondes

Le modèle dans DateTimePattern doit correspondre au modèle dans DateTime. Puisque la sortie est de type xs:dateTime, la sortie aura toujours le format lexical YYYY-MM-DDTHH:mm:ss.

#### <u>Exemples</u>

- 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.

- н Heure
- m minutes s secondes

Le modèle dans TimePattern doit correspondre au modèle dans Time. Puisque la sortie est de type xs:time, la sortie aura toujours le format lexical HH:mm:ss.

Exemples

- altova:parse-time(xs:string("13:56:24"), "[H]:[m]:[s]") retourne 13:56:24
- altova:parse-time("13-56-24", "[H]-[m]") retourne 13:56:00
- **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



#### 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.

<u>Exemples</u>

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.

<u>Exemples</u>

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.

- <u>Exemples</u>
  - 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)



#### Fonctions Epoch time (heure Unix) xP3.1 xQ3.1

Epoch time est un système horaire utilisé dans les systèmes Unix. Il définit tout moment donné comme étant le nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. Ces fonctions Epoch time convertissent les valeurs xs:datetime en valeurs Epoch time et vice versa.

dateTime-from-epoch [altova:]

altova:dateTime-from-epoch(Epoch as xs:decimal as xs:dateTime XP3.1 XQ3.1 Epoch time est un système horaire utilisé sur les systèmes Unix. Il définit tout moment donné comme étant l nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. La fonction dateTimefrom-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 [altova:]

altova:dateTime-from-epoch(Epoch as xs:decimal as xs:dateTime XP3.1 XQ3.1 Epoch time est un système horaire utilisé sur les systèmes Unix. Il définit tout moment donné comme étant l nombre de secondes écoulées depuis 00:00:00 UTC le 1er janvier 1970. La fonction dateTimefrom-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 dateTimefrom-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

# 18.2.2.1.3 Fonctions XPath/XQuery : Géolocalisation

Les fonctions d'extension de géolocalisation XPath/XQuery suivantes sont prises en charge dans la version actuelle de MapForce 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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette le comportement de certa

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

```
    format-geolocation [altova:]
```

```
altova:format-geolocation(Latitude as xs:decimal, Longitude as xs:decimal,
GeolocationOutputStringFormat as xs:integer) asxs:string
XP3.1 X03.1
```

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 <u>image-exif-data</u><sup>1518</sup> et les attributs de métadonnées Exif peuvent être utilisés pour fournie les chaînes d'entrée.

```
Exemples
```

1

2

3

4

- 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"
- 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).

Degrés, minutes, secondes décimales, avec orientation suffixée (N/S, E/O) D°M'S.SS"N/S D°M'S.SS"E/W <u>Exemple</u>: 33°55'11.11"N 22°44'66.66"W

```
Degrés décimaux, avec orientation suffixée (N/S, E/O)
D.DDN/S D.DDE/W
<u>Exemple</u>: 33.33N 22.22W
```

Degrés, minutes, secondes décimales, avec signe préfixé (+/-); le signe plus (n/ɛ) est optionnel +/-Dºм's.ss" +/-Dºm's.ss" <u>Exemple</u>: 33º55'11.11" -22º44'66.66"

Degrés décimaux, avec signe préfixé (+/-); le signe plus (N/E) est optionnel +/-D.DD +/-D.DD <u>Exemple</u>: 33.33 -22.22

#### Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut Geolocation personnalisable depuis les onglets standard de métadonnées Exif. Geolocation est une concaténation de quatre onglets Exif : GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

GPSLa	atitude	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
33 51	L 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

#### parse-geolocation [altova:]

altova:parse-geolocation(GeolocationInputString as xs:string) asxs:decimal+ XP3.1 XQ3.1 Parse l'argument GeolocationInputString fourni et retourne la latitude et longitude de géolocalisation (dans cet ordre) en séquence deux items décimaux xs:decimal. Les formats dans lesquels la chaîne d'entrée de géolocalisation peut être fournie sont recensés ci-dessous.

**Note :** La fonction <u>image-exif-data</u><sup>[518]</sup> et l'attribut <u>@Geolocation</u><sup>[518]</sup> de métadonnées Exif peuvent être utilisés pour fournir la chaîne d'entrée de géolocalisation (*voir exemple ci-dessous*).

#### Exemples

- altova:parse-geolocation("33.33 -22.22") retourne la séquence de deux xs:decimals (33.33, 22.22)
- altova:parse-geolocation("48°51'29.6""N 24°17'40.2""") retourne la séquence de deux xs:decimals (48.858222222222, 24.2945)
- altova:parse-geolocation('48°51''29.6"N 24°17''40.2"') retourne la séquence de deux xs:decimals (48.858222222222, 24.2945)
- altova:parse-geolocation( image-exifdata(//MyImages/Image20141130.01)/@Geolocation )retourne une séquence de deux xs:decimals

#### E 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"
   <u>Exemple</u>: 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'
   <u>Exemple</u>: +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
   <u>Exemple</u>: 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	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
33 51 21.91	S	151 13 11.73	Е	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 <u>image-exif-data</u><sup>(518)</sup> et l'attribut <u>@Geolocation</u><sup>(518)</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""") retourne xs:decimal 4183.08132372392
- E 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"
   <u>Exemple</u>: 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'
   <u>Exemple</u>: +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	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
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 <u>image-exif-data</u><sup>(518)</sup> et l'attribut <u>@Geolocation</u><sup>(518)</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""") 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 ( $\mathbb{N}$  à s). Les valeurs de longitude varient de +180 à -180 ( $\mathbb{E}$  à  $\mathbb{N}$ ).

**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
 <u>Exemple</u>: 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"
   <u>Exemple</u>: 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'
   <u>Exemple</u>: +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
   <u>Exemple</u>: 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	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
33 51 21.91	S	151 13 11.73	Е	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 <u>image-exif-data</u><sup>(518)</sup> et l'attribut <u>@Geolocation</u><sup>(518)</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")) refourne true()
- altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24")) retourne true()
- altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48°51'29.6""N 24°17'40.2""")) retourne true()
- E 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"
   <u>Exemple</u>: 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
   <u>Exemple</u>: 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'
   <u>Exemple</u>: +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

```
<u>Exemple</u>: 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	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
33 51 21.91	S	151 13 11.73	Е	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 <u>image-exif-data</u><sup>[518]</sup> et l'attribut <u>@Geolocation</u><sup>[518]</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""") 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"
   <u>Exemple</u>: 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'
   <u>Exemple</u>: +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
   <u>Exemple</u>: 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	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
33 51 21.91	S	151 13 11.73	Е	33°51'21.91"S 151° 13'11.73"E

[<u>Top</u><sup>1508</sup>]

## 18.2.2.1.4 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 MapForce 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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

suggested-image-file-extension [altova:]

altova:suggested-image-file-extension(Base64String as string) asstring? XP3.1 XQ3.1 Prend le code Base64 d'un fichier d'image en tant que son argument et retourne l'extension de fichier de l'image comme enregistré dans le codage Base64 de l'image. La valeur retournée est une suggestion basée sur l'information du type d'image disponible dans le codage. Si cette information n'est pas disponible, une chaîne vide est retournée. Cette fonction est utile si vous souhaitez enregistrer une image Base64 en tant que fichier et que vous souhaitez extraire dynamiquement une extension de fichier appropriée.

- Exemples
  - altova:suggested-image-file-extension(/MyImages/MobilePhone/Image20141130.01) retourne 'jpg'
  - altova:suggested-image-file-extension(\$XML1/Staff/Person/@photo) retourne ''

Dans les exemples ci-dessus, les nœuds fournis en tant qu'arguments de la fonction sont assumés contenir une image codée Base64. Le premier exemple extrait jpg en tant que type et extension de fichier. Dans le second exemple, le codage Base64 soumis ne fournit pas une information de fichier d'extension utile.

image-exif-data [altova:]

altova:image-exif-data(Base64BinaryString as string) aselement? XP3.1 XQ3.1

Prend une image codée Base64 en tant que son argument et retourne un élément appelé **Exif** qui contient les métadonnées Exif de l'image. Celles-ci sont créées en tant que paires attribute-value pairs de l'élément Exif. Les noms d'attribut sont les onglets de données Exif trouvés dans le codage Base64. La liste des onglets des spécifications Exif est indiquée ci-dessous. Si un onglet spécifique à un distributeur est présent dans les données Exif, cet onglet et sa valeur seront aussi retournés en tant que paire attribute-value. Outre les onglets de métadonnées Exif standard (*voir la liste ci-dessous*), des paires attribute-value spécifiques à Altova sont également générées. Ces attributs Exif Altova sont recensés ci-dessous.

#### Exemples

- Pour accéder à n'importe quel attribut, utiliser la fonction comme suit : image-exif-data(//MyImages/Image20141130.01)/@GPSLatitude image-exif-data(//MyImages/Image20141130.01)/@Geolocation
- Pour accéder à tous les attributs, utiliser la fonction comme suit: image-exif-data(//MyImages/Image20141130.01)/@\*
- Pour accéder au nom de tous les attributs, utiliser l'expression suivante : for \$i in image-exif-data(//MyImages/Image20141130.01)/@\* return name(\$i) Cela est utile pour trouver les noms des attributs retournés par la fonction.

#### Attribut Altova Exif : Géolocalisation

La machine Altova XPath/XQuery génère l'attribut Geolocation personnalisable depuis les onglets standard de métadonnées Exif. Geolocation est une concaténation de quatre onglets Exif : GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, avec des unités ajoutées (voir table ci-dessous).

GPSLatitude	GPSLatitudeRe f	GPSLongitude	GPSLongitudeRe f	Geolocation
33 51 21.91	S	151 13 11.73	Е	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 cidessous. 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



# 18.2.2.1.5 Fonctions XPath/XQuery : Numérique

Les fonctions d'extension numériques d'Altova peuvent être utilisées dans des expressions XPath et XQuery et proposent des fonctions supplémentaires pour le traitement des données. Les fonctions dans cette section peuvent être utilisée avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova. Ils sont disponibles dans des contextes XPath/XQuery.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des

fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova**, <u>http://www.altova.com/xslt-extensions</u>, et sont indiquées dans cette section par le préfixe <u>altova</u>; qui est présupposé être lié à cet espace de nom. Veuillez noter que, en ce qui concerne les versions futures de votre produit, la prise en charge d'une fonction peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

# Fonctions de numérotation automatique

generate-auto-number [altova:]

altova:generate-auto-number(ID as xs:string, StartsWith as xs:double, Increment as xs:double, ResetOnChange as xs:string) as xs:integer XP1 XP2 XQ1 XP3.1 XQ3.1 Génère un numéro à chaque fois que la fonction est appelée. Le premier numéro, qui est généré la première fois que la fonction est appelée, est spécifié par l'argument StartsWith. Chaque appel subséquent vers la fonction génère un nouveau numéro, ce numéro est augmenté au-dessus du numéro précédemment généré par la valeur spécifiée dans l'argument Increment. En effet, la fonction altova:generate-auto-number Crée un compteur comportant un nom spécifié par l'argument ID, et dont le compteur est augmenté à chaque fois que la fonction précédent, 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.

• 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.



## 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.

- <u>Exemples</u>
  - 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') refourne 15
  - altova:hex-string-to-integer('G') retourne une erreur
  - altova:hex-string-to-integer('10') retourne 16
  - altova:hex-string-to-integer('01') retourne 1
  - altova:hex-string-to-integer('20') retourne 32
  - altova:hex-string-to-integer('21') retourne 33
  - altova:hex-string-to-integer('5A') retourne 90
  - altova:hex-string-to-integer('USA') retourne une erreur
- integer-to-hex-string [altova:]

altova:integer-to-hex-string(Integer as xs:integer) asxs:string XP3.1 XQ3.1
 Prend un argument d'entier et retourne son équivalent de Base-16 en tant que chaîne.
 <u>Exemples</u>

- altova:integer-to-hex-string(1) retourne '1'
- altova:integer-to-hex-string(9) retourne '9'
- altova:integer-to-hex-string(10) retourne 'A'
- altova:integer-to-hex-string(11) retourne 'B'
- altova:integer-to-hex-string(15) retourne 'F'
- altova:integer-to-hex-string(16) retourne '10'
- altova:integer-to-hex-string(32) retourne '20'
- altova:integer-to-hex-string(33) retourne '21'
- altova:integer-to-hex-string(90) retourne '5A'



# Fonctions de formatage de numéro



# 18.2.2.1.6 Fonctions XPath/XQuery : Schéma

Les fonctions d'extension Altova recensées ci-dessous retournent l'information concernant le schéma. Cidessous, vous trouverez les descriptions des fonctions, ainsi que des (i) exemples et (ii) une liste des composants de schéma et de leurs propriétés respectives. Elles peuvent être utilisées avec les moteurs **XPath 3.0** et **XQuery 3.0** d'Altova et sont disponibles dans des contextes XPath/XQuery.

#### Information de schéma depuis les documents de schéma

La fonction altova:schema détient deux arguments : un avec zéro arguments et l'autre avec deux arguments. La fonction à zéro argument retourne l'ensemble du schéma. Ensuite, à partir de là, vous pouvez naviguer dans le schéma pour localiser les composants de schéma que vous souhaitez. La fonction à deux arguments retourne un type de composant spécifique qui est identifié par son QName. Dans les deux cas, la valeur de retour est un fonction. Pour naviguer dans le composant retourné, vous devez sélectionner une propriété de ce composant spécifique. Si la propriété est un item non atomique (c'est à dire, s'il s'agit d'un composant), vous pouvez aller plus loin en choisissant une propriété de ce composant. Si la propriété sélectionnée est un item atomique, la valeur de l'item est retournée et vous ne pouvez pas aller plus loin.

**Note:** Dans les expressions XQuery, le schéma doit être importé explicitement. Dans les expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple, dans XSLT avec l'instruction xslt:import.

#### Information de schéma depuis les nœuds XML

La fonction altova:type soumet le nœud à un document XML et retoune l'information de type du nœud depuis le PSVI.

Note concernant le nommage de fonctions et de l'applicabilité de la langue

Les fonctions d'extension Altova peuvent être utilisées dans les expressions XPath/XQuery. Elles fournissent des fonctions supplémentaires aux fonctions d'ores et déjà disponibles dans la librairie standard des fonctions XPath, XQuery et XSLT. Les fonctions d'extension Altova se trouvent dans **l'espace de nom des fonctions d'extension Altova**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section paul étre interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

#### Schéma (zéro arguments)

altova:schema() as (function(xs:string) as item()\*)? XP3.1 XQ3.1 Retourne le composant de schema en entier. Vous pouvez donc aller plus loin dans le composant de schema en sélectionnant une des propriétés du composant de schema.

- Si cette propriété est un composant, vous pouvez aller encore plus en loin en sélectionnant une des propriétés de ce composant. Vous pouvez renouveler cette étape en allant plus loin dans le schéma.
- Si le composant est une valeur atomique, celle-ci sera retournée et vous ne pourrez pas aller plus loin.

Les propriétés du composant de schema sont :

```
"type definitions"
"attribute declarations"
"element declarations"
"attribute group definitions"
"model group definitions"
"notation declarations"
"identity-constraint definitions"
```

Les propriétés de tous les types de composant (à part schema) sont regroupées ci-dessous.

**Note:** Dans des expressions XQuery, le schéma doit être importé explicitement. Dans des expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple dans XSLT avec l'instruction xslt:import.

#### Exemples

- import schema "" at "C:\Test\ExpReport.xsd"; for \$typedef in altova:schema() ("type definitions") return \$typedef ("name") retourne les noms de tous les Types simples ou Types complexes dans le schéma
- import schema "" at "C:\Test\ExpReport.xsd"; altova:schema() ("type definitions")[1]("name") retourne le nom du premier de tous les Types simples ou Types complexes dans le schéma

#### Composants et leurs propriétés

Assertion

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Assertion"
test	XPath Property Record	

#### Déclaration d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Declaration"
name	string	Local name of the attribute
target namespace	string	Namespace URI of the attribute
type definition	Type simple or Type complexe	
scope	A function with properties ("class":"Scope", "variety": "global" or "local", "parent": the containing Type complexe or Attribute Group)	
value constraint	If present, a function with properties ("class": "Value Constraint", "variety": "fixed" or "default", "value": atomic value, "lexical form": string. Note that the "value" property is not available for namespace-sensitive types	
inheritable	boolean	

## ■ Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
name	string	Local name of the attribute group
target namespace	string	Namespace URI of the attribute group
attribute uses	Sequence of (Attribute Use)	
attribute wildcard	Optional Attribute Wildcard	

## Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
required	boolean	true if the attribute is required, false if optional
value constraint	See Attribute Declaration	
inheritable	boolean	

## Caractère générique

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"

namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict" "lax" "skip")	

# ■ Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
base type definition	Complex Type Definition	
final	Sequence of strings ("restriction" "extension")	
context	Empty sequence (not implemented)	
derivation method	string ("restriction" "extension")	
abstract	boolean	
attribute uses	Sequence of Attribute Use	
attribute wildcard	Optional Attribute Wildcard	
content type	function with properties: ("class":"Content Type", "variety":string ("element- only" "empty" "mixed" "simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
prohibited substitutions	Sequence of strings ("restriction" "extension")	
assertions	Sequence of Assertion	

## Déclaration d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"

name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
type definition	Type simple or Type complexe	
type table	function with properties ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Type simple or Type complexe)	
scope	function with properties ("class":"Scope", "variety": ("global" "local"), "parent": optional Type complexe)	
value constraint	see Attribute Declaration	
nillable	boolean	
identity-constraint definitions	Sequence of Identity Constraint	
substitution group affiliations	Sequence of Element Declaration	
substitution group exclusions	Sequence of strings ("restriction" "extension")	
disallowed substitutions	Sequence of strings ("restriction" "extension" "substitution")	
abstract	boolean	

# Garactère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict" "lax" "skip")	

## Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	The name of the facet, for example "minLength" or "enumeration"

value	depends on facet	The value of the facet
fixed	boolean	
typed-value	For the enumeration facet only, array(xs:anyAtomicType*)	An array containing the enumeration values, each of which may in general be a sequence of atomic values. (Note: for the enumeration facet, the "value" property is a sequence of strings, regardless of the actual type)

#### Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
name	string	Local name of the constraint
target namespace	string	Namespace URI of the constraint
identity-constraint category	string ("key" "unique" "keyRef")	
selector	XPath Property Record	
fields	Sequence of XPath Property Record	
referenced key	(For keyRef only): Identity Constraint	The corresponding key constraint

#### Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Groupe de modèle"
compositor	string ("sequence" "choice" "all")	
particles	Séquence de particule	

## Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de groupe de modèle"
name	string	Nom local du groupe de modèle
target namespace	string	URI d'espace du groupe de modèle
model group	Groupe de modèle	

#### Notation

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Déclaration de notation"
name	string	Nom local de la notation

target namespace	string	URI d'espace de nom de la notation
system identifier	anyURI	
public identifier	string	

## Particule

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Particule"
min occurs	entier	
max occurs	entier ou string("unbounded")	
term	Déclaration d'élément, Caractère générique d'élément ou ModelGroup	

## Type simple

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de type simple"
name	string	Nom local du type (vide si anonyme)
target namespace	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction" "extension" "list" "unio n")	
context	composant contenant	
base type definition	Type simple	
facets	Séquence de Facette	
fundamental facets	Séquence vide (pas implémentée)	
variety	string ("atomic" "list" "union")	
primitive type definition	Type simple	
item type definition	uniquement pour les types de liste) Type simple	
member type definitions	(uniquement pour les types d'union) Séquence de Type simple	

#### ■ Alternative de type

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type Alternative"

test	XPath Property Record	
type definition	Type simple ou Type complexe	

XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
namespace bindings	Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

Schéma (deux arguments)

altova:schema(ComponentKind as xs:string, Name as xs:QName) as (function(xs:string) as item()\*)? XP3.1 XQ3.1

Retourne le type du composant qui est spécifié dans le premier argument qui a un nom identique à celui fourni dans le second argument. Vous pouvez donc aller plus loin en sélectionnant une des propriétés du composant.

- Si cette propriété est un composant, vous pouvez aller encore plus en profondeur en sélectionnant une des propriétés de ce composant. Vous pouvez renouveler cette étape en allant plus loin dans le schéma.
- Si le composant est une valeur atomique, celle-ci sera retournée et vous ne pourrez pas aller plus loin.

**Note:** Dans des expressions XQuery, le schéma doit être importé explicitement. Dans des expressions XPath, le schéma doit avoir été importé dans l'environnement de traitement, par exemple dans XSLT avec l'instruction xslt:import.

#### Exemples

```
• import schema "" at "C:\Test\ExpReport.xsd";
altova:schema("element declaration", xs:QName("OrgChart"))("type definition")
("content type")("particles")[3]!.("term")("kind")
```

retourne la propriété kind du terme du troisième composant de particles. Ce composant de particules est un descendant de la déclaration d'élément ayant un QName de OrgChart

```
• import schema "" at "C:\Test\ExpReport.xsd";
let $typedef := altova:schema("type definition", xs:QName("emailType"))
for $facet in $typedef ("facets")
return [$facet ("kind"), $facet("value")]
retourne, pour chaque facet de chaque composant emailType, un array contenant le type et la
valeur de cette facette
```

#### Composants et leurs propriétés

#### Assertion

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Assertion"
test	XPath Property Record	

#### Déclaration d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Declaration"
name	string	Local name of the attribute
target namespace	string	Namespace URI of the attribute
type definition	Type simple or Type complexe	
scope	A function with properties ("class":"Scope", "variety": "global" or "local", "parent": the containing Type complexe or Attribute Group)	
value constraint	If present, a function with properties ("class": "Value Constraint", "variety": "fixed" or "default", "value": atomic value, "lexical form": string. Note that the "value" property is not available for namespace-sensitive types	
inheritable	boolean	

## Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
name	string	Local name of the attribute group
target namespace	string	Namespace URI of the attribute group
attribute uses	Sequence of (Attribute Use)	
attribute wildcard	Optional Attribute Wildcard	

#### Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
required	boolean	true if the attribute is required, false if optional

value constraint	See Attribute Declaration	
inheritable	boolean	

## Caractère générique

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict" "lax" "skip")	

# ■ Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
base type definition	Complex Type Definition	
final	Sequence of strings ("restriction" "extension")	
context	Empty sequence (not implemented)	
derivation method	string ("restriction" "extension")	
abstract	boolean	
attribute uses	Sequence of Attribute Use	
attribute wildcard	Optional Attribute Wildcard	
content type	function with properties: ("class":"Content Type", "variety":string ("element- only" "empty" "mixed" "simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
prohibited	Sequence of strings	

substitutions	("restriction" "extension")	
assertions	Sequence of Assertion	

## Déclaration d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
type definition	Type simple or Type complexe	
type table	function with properties ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Type simple or Type complexe)	
scope	function with properties ("class":"Scope", "variety": ("global" "local"), "parent": optional Type complexe)	
value constraint	see Attribute Declaration	
nillable	boolean	
identity-constraint definitions	Sequence of Identity Constraint	
substitution group affiliations	Sequence of Element Declaration	
substitution group exclusions	Sequence of strings ("restriction" "extension")	
disallowed substitutions	Sequence of strings ("restriction" "extension" "substitution")	
abstract	boolean	

# Garactère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict" "lax" "skip")	

#### E Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	The name of the facet, for example "minLength" or "enumeration"
value	depends on facet	The value of the facet
fixed	boolean	
typed-value	For the enumeration facet only, array(xs:anyAtomicType*)	An array containing the enumeration values, each of which may in general be a sequence of atomic values. (Note: for the enumeration facet, the "value" property is a sequence of strings, regardless of the actual type)

#### Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
name	string	Local name of the constraint
target namespace	string	Namespace URI of the constraint
identity-constraint category	string ("key" "unique" "keyRef")	
selector	XPath Property Record	
fields	Sequence of XPath Property Record	
referenced key	(For keyRef only): Identity Constraint	The corresponding key constraint

## Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Groupe de modèle"
compositor	string ("sequence" "choice" "all")	
particles	Séquence de particule	

## Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de groupe de modèle"
name	string	Nom local du groupe de modèle
target namespace	string	URI d'espace du groupe de modèle

model group	Groupe de modèle	
-------------	------------------	--

#### Notation

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Déclaration de notation"
name	string	Nom local de la notation
target namespace	string	URI d'espace de nom de la notation
system identifier	anyURI	
public identifier	string	

## Particule

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Particule"
min occurs	entier	
max occurs	entier ou string("unbounded")	
term	Déclaration d'élément, Caractère générique d'élément ou ModelGroup	

# Type simple

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de type simple"
name	string	Nom local du type (vide si anonyme)
target namespace	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction" "extension" "list" "unio n")	
context	composant contenant	
base type definition	Type simple	
facets	Séquence de Facette	
fundamental facets	Séquence vide (pas implémentée)	
variety	string ("atomic" "list" "union")	
primitive type definition	Type simple	
item type definition	uniquement pour les types de liste) Type simple	

member type	(uniquement pour les types d'union)
definitions	Séquence de Type simple

#### Alternative de type

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type Alternative"
test	XPath Property Record	
type definition	Type simple ou Type complexe	

#### XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
namespace bindings	Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

#### • Туре

altova:type(Node as item?) as (function(xs:string) as item()\*)? XP3.1 XQ3.1 La fonction altova:type soumet un nœud d'élément ou d'attribut d'un document XML et document et retourne l'information du type du nœud depuis le PSVI.

Note: Le document XML doit avoir une déclaration de schéma afin que ce schéma puisse être référencé.

```
Exemples
```

```
• for $element in //Email
    let $type := altova:type($element)
    return $type
    retourne une fonction qui contient l'information du type du nœud
```

```
• for $element in //Email
let $type := altova:type($element)
return $type ("kind")
prend le composant du type du nœud (Type simple ou Type complexe) et retourne la valeur de la
propriété kind du composant
```

#### Composants et leurs propriétés

Assertion

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Assertion"
test	XPath Property Record	

Déclaration d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Declaration"
name	string	Local name of the attribute
target namespace	string	Namespace URI of the attribute
type definition	Type simple or Type complexe	
scope	A function with properties ("class":"Scope", "variety": "global" or "local", "parent": the containing Type complexe or Attribute Group)	
value constraint	If present, a function with properties ("class": "Value Constraint", "variety": "fixed" or "default", "value": atomic value, "lexical form": string. Note that the "value" property is not available for namespace-sensitive types	
inheritable	boolean	

Déclaration de groupe d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Group Definition"
name	string	Local name of the attribute group
target namespace	string	Namespace URI of the attribute group
attribute uses	Sequence of (Attribute Use)	
attribute wildcard	Optional Attribute Wildcard	

Utilisation d'attribut

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Attribute Use"
required	boolean	true if the attribute is required, false if optional
value constraint	See Attribute Declaration	
inheritable	boolean	

## Caractère générique

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict" "lax" "skip")	

# ■ Type complexe

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
base type definition	Complex Type Definition	
final	Sequence of strings ("restriction" "extension")	
context	Empty sequence (not implemented)	
derivation method	string ("restriction" "extension")	
abstract	boolean	
attribute uses	Sequence of Attribute Use	
attribute wildcard	Optional Attribute Wildcard	
content type	function with properties: ("class":"Content Type", "variety":string ("element- only" "empty" "mixed" "simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Type simple)	
prohibited substitutions	Sequence of strings ("restriction" "extension")	
assertions	Sequence of Assertion	

Déclaration d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Type complexe"
name	string	Local name of the type (empty if anonymous)
target namespace	string	Namespace URI of the type (empty if anonymous)
type definition	Type simple or Type complexe	
type table	function with properties ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Type simple or Type complexe)	
scope	function with properties ("class":"Scope", "variety": ("global" "local"), "parent": optional Type complexe)	
value constraint	see Attribute Declaration	
nillable	boolean	
identity-constraint definitions	Sequence of Identity Constraint	
substitution group affiliations	Sequence of Element Declaration	
substitution group exclusions	Sequence of strings ("restriction" "extension")	
disallowed substitutions	Sequence of strings ("restriction" "extension" "substitution")	
abstract	boolean	

# Garactère générique d'élément

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Wildcard"
namespace constraint	function with properties ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	string ("strict" "lax" "skip")	

## ■ Facette

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	The name of the facet, for

		example "minLength" or "enumeration"
value	depends on facet	The value of the facet
fixed	boolean	
typed-value	For the enumeration facet only, array(xs:anyAtomicType*)	An array containing the enumeration values, each of which may in general be a sequence of atomic values. (Note: for the enumeration facet, the "value" property is a sequence of strings, regardless of the actual type)

## Contrainte d'identité

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Identity-Constraint Definition"
name	string	Local name of the constraint
target namespace	string	Namespace URI of the constraint
identity-constraint category	string ("key" "unique" "keyRef")	
selector	XPath Property Record	
fields	Sequence of XPath Property Record	
referenced key	(For keyRef only): Identity Constraint	The corresponding key constraint

#### Groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Groupe de modèle"
compositor	string ("sequence" "choice" "all")	
particles	Séquence de particule	

#### Définition de groupe de modèle

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de groupe de modèle"
name	string	Nom local du groupe de modèle
target namespace	string	URI d'espace du groupe de modèle
model group	Groupe de modèle	

Notation

Nom de propriété Type de propriété	Valeur de propriété
------------------------------------	---------------------

kind	string	"Déclaration de notation"
name	string	Nom local de la notation
target namespace	string	URI d'espace de nom de la notation
system identifier	anyURI	
public identifier	string	

## Particule

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Particule"
min occurs	entier	
max occurs	entier ou string("unbounded")	
term	Déclaration d'élément, Caractère générique d'élément ou ModelGroup	

# Type simple

Nom de propriété	Type de propriété	Valeur de propriété
kind	string	"Définition de type simple"
name	string	Nom local du type (vide si anonyme)
target namespace	string	URI d'espace de noms du type (vide si anonyme)
final	Séquence de string("restriction" "extension" "list" "unio n")	
context	composant contenant	
base type definition	Type simple	
facets	Séquence de Facette	
fundamental facets	Séquence vide (pas implémentée)	
variety	string ("atomic" "list" "union")	
primitive type definition	Type simple	
item type definition	uniquement pour les types de liste) Type simple	
member type definitions	(uniquement pour les types d'union) Séquence de Type simple	

## Alternative de type

Nom de propriété	Type de propriété	Valeur de propriété
------------------	-------------------	---------------------

kind	string	"Type Alternative"
test	XPath Property Record	
type definition	Type simple ou Type complexe	

Second XPath Property Record

Nom de propriété	Type de propriété	Valeur de propriété
namespace bindings	Séquence des fonctions avec les propriétés ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	L'URI de base statique de l'expression XPath
expression	string	L'expression XPath en tant que string

# 18.2.2.1.7 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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section paul et préfixe <a href="http://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section paul et sont indiquées dans cette le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery	XQ1 XQ3.1

dans XQuery) :

attributes [altova:]

• **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.

<u>Exemples</u>

• **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.

<u>Exemples</u>

- 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") refourne 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)) refourne xs:string C:\Temp\Customers.xml

Si le fichier C:\Temp\Customers.xml n'existe pas et que http://www.altova.com/index.html existe :

• altova:find-first( ("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1) ) retourne xs:string http://www.altova.com/index.html

Si le fichier C:\Temp\Customers.xml n'existe pas, et que http://www.altova.com/index.html n'existe pas non plus :

• altova:find-first( ("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1) ) ne retourne aucun résultat

#### Notes à propos des exemples indiqués ci-dessus

- La fonction XPath 3.0, doc-available, prend un seul argument de chaîne, qui est utilisé en tant qu'URI, et retourne true si un nœud de document est trouvé à l'URI soumis. (Le document à l'URI soumis doit donc être un document XML.)
- La fonction doc-available peut être utilisée pour condition, le second argument de altova:find-first, parce qu'il ne prend qu'un seul argument (arité=1), parce qu'il prend un item() en tant qu'entrée (une chaîne qui est utilisée en tant qu'URI), et retourne une valeur booléenne.
- Veuillez noter que la fonction doc-available est uniquement référencée, elle n'est pas appelée. Le suffixe #1 qui y est attaché indique une fonction avec une arité de 1. Sous sa forme complète, doc-available#1 signifie simplement : *Utiliser la fonction doc-availabe() à l'arité=1, en l'y passant en tant que son seul argument, chacun à son tour, chacun des items dans la première séquence*. En résultat, chacun des deux chaînes sera passée à doc-available(), qui utilise la chaîne en tant qu'URI et teste si un nœud de document existe à l'URI. S'il en existe un, docavailable() évalue à true() et cette chaîne est retournée en tant que le résultat de la fonction altova:find-first. Note à propos de la fonction doc-available() : les chemins relatifs sont résolus relativement à l'URI de base actuel, qui est par défaut l'URI du document XML à partir duquel la fonction est chargée.

#### find-first-combination [altova:]

altova:find-first-combination((Seq-01 as item()\*), (Seq-02 as item()\*), (Condition( Seq-01-Item, Seq-02-Item as xs:boolean)) asitem()\* XP3.1 XQ3.1 Cette fonction prend trois arguments :

- Les deux premiers arguments, seq-01 et seq-02, sont des séquences d'un ou de plusieurs items de tout type de données.
- Le troisième argument, condition, est une référence à une fonction XPath qui prend deux arguments (a une arité de 2) et retourne un booléen.

Les items de seq-01 et seq-02 sont passés dans des paires ordonnées (un item de chaque séquence faisant une paire) en tant que les arguments de la fonction dans condition. Les paires sont classées comme suit :

If Seq-01 = X1, X2, X3 ... Xn And Seq-02 = Y1, Y2, Y3 ... Yn Then (X1 Y1), (X1 Y2), (X1 Y3) ... (X1 Yn), (X2 Y1), (X2 Y2) ... (Xn Yn)

La première paire ordonnée qui entraîne la fonction condition à évaluer à true() est retournée en tant que le résultat de altova:find-first-combination. Veuillez noter que : (i) si la fonction Condition itère par le biais des paires d'argument soumises et n'évalue pas une fois à true(), alors altova:findfirst-combination retournera Aucun résultat ; (ii) Le résultat de altova:find-first-combination sera toujours une paire d'items (de tout type de données) ou aucun item.

Exemples

- altova:find-first-combination(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) retourne la séquence de xs:integers (11, 21)
- altova:find-first-combination(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) retourne la séquence de f xs:integers (11, 22)
- altova:find-first-combination(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 34}) retourne la séquence de xs:integers (11, 23)

find-first-pair [altova:]

altova:find-first-pair((Seq-01 as item()\*), (Seq-02 as item()\*), (Condition( Seq-01-Item, Seq-02-Item as xs:boolean)) asitem()\* XP3.1 XQ3.1 Cette fonction prend trois arguments :

- Les deux premiers arguments, seq-01 et seq-02, sont des séquences d'un ou de plusieurs items de tout type de données.
- Le troisième argument, condition, est une référence à une fonction XPath qui prend deux arguments (a une arité de 2) et retourne un booléen.

Les items de seq-01 et seq-02 sont passés dans des paires ordonnées en tant que les arguments de la fonction dans condition. Les paires sont classées comme suit :

If Seq-01 = X1, X2, X3 ... Xn
And Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)

La première paire ordonnée qui cause la fonction condition à évaluer à true() est retournée en tant que le résultat de altova:find-first-pair. Veuillez noter que : (i) Si la fonction Condition itère par le biais des paires d'arguments soumis et n'évalue pas une seule fois à true(), alors altova:find-first-pair retournera *Aucun résultat*; (ii) Le résultat de altova:find-first-pair sera toujours une paire d'items (de tout type de données) ou aucun item.

Exemples

- altova:find-first-pair(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) retourne la séquence de xs:integers (11, 21)
- altova:find-first-pair(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) retourne Aucun résultat

Veuillez noter à partir des deux exemples ci-dessus que l'ordonnance des paires est : (11, 21) (12, 22) (13, 23)...(20, 30). C'est pourquoi le second exemple retourne *Aucun résultat* (parce qu'aucune paire ordonnée de donne une somme de 33).
#### find-first-pair-pos [altova:]

altova:find-first-pair-pos((Seq-01 as item()\*), (Seq-02 as item()\*), (Condition( Seq-01-Item, Seq-02-Item as xs:boolean)) asxs:integer XP3.1 XQ3.1 Cette fonction prend trois arguments :

- Les deux premiers arguments, seq-01 and seq-02, sont des séquences d'un ou de plusieurs items de tout type de données.
- Le troisième argument, condition, est une référence à une fonction XPath qui prend deux arguments (a une arité de 2) et retourne un booléen.

Les items de seq-01 et seq-02 sont passés dans des paires ordonnées en tant que les arguments de la fonction dans condition. Les paires sont classées comme suit :

If Seq-01 = X1, X2, X3 ... Xn
And Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)

La position d'index de la première paire ordonnée qui entraîne la fonction condition à évaluer à true() est retournée en tant que le résultat de altova:find-first-pair-pos. Veuillez noter que si la fonction condition itère par le biais des paires d'arguments soumises et n'évalue pas une seule fois à true(), alors altova:find-first-pair-pos retournera Aucun résultat.

#### Exemples

- altova:find-first-pair-pos(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) retourne 1
- altova:find-first-pair-pos(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) retourne Aucun résultat

Veuillez noter à partir des deux exemples ci-dessus que l'ordonnance des paires est : (11, 21) (12, 22) (13, 23)...(20, 30). dans le premier exemple, la première paire entraîne la fonction **Condition** à évaluer à true(), et donc sa position d'index dans la séquence, 1, est retournée. Le second exemple retourne *Aucun résultat* parce qu'aucune paire ne totalise pas une somme de 33.

find-first-pos [altova:]

altova:find-first-pos((Sequence as item()\*), (Condition( Sequence-Item as xs:boolean))
asxs:integer XP3.1 XQ3.1

Cette fonction prend deux arguments. Le premier argument est une séquence d'un ou de plusieurs items de tout type de données. Le second argument, Condition, est une référence à une fonction XPath qui prend un argument (a une arité de 1) et retourne une booléenne. Chaque item de sequence est soumis à son tour à la fonction référencée dans Condition. (*Rappel* : cette fonction prend un seul argument.) Le premier item sequence qui cause la fonction dans condition à évaluer à true() voit sa position index dans sequence retournée en tant que résultat de altova:find-first-pos, et l'itération stoppe.

Exemples

• altova:find-first-pos(5 to 10, function(\$a) {\$a mod 2 = 0}) retourne xs:integer 2 L'argument condition référence la fonction en ligne XPath 3.0, function(), qui déclare une fonction en ligne nommée \$a et puis la définit. Chaque item dans l'argument de sequence de altova:findfirst-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, docavailable#1 signifie simplement : *Utiliser la fonction doc-availabe() qui a arité=1, y passant, en tant que son argument simple, chacun à son tour, chaque item dans la première séquence*. En tant que résultat. chacune des deux chaînes sera passée à doc-available(), qui utilise la chaîne en tant qu'un URI et teste si un nœud de document existe à l'URI. Si c'est le cas, la fonction doc-available() évalue à true() et la position de l'index de cette chaîne dans la séquence est retournée en tant que le résultat de la fonction altova:find-first-pos. Note à *propos de la fonction doc-available() : les chemins relatifs sont résolus relativement à l'URI de base actuel, qui par défaut est l'URI du document XML à partir duquel la fonction est chargée.*

#### for-each-attribute-pair [altova:]

altova:for-each-attribute-pair(Seq1 as element()?, Seq2 as element()?, Function as function()) asitem()\* XP3.1 XQ3.1

Les premiers deux arguments identifient deux éléments, dont les attributs sont utilisés pour générer des paires d'attribut, dans laquelle un attribut d'une paire est obtenu depuis le premier élément et l'autre attribut est obtenu depuis le second élément. Les paires d'attribut sont sélectionnées sur le fait qu'ils présentent le même nom, et les paires sont classées par ordre alphabétique (sur leur nom) dans un

ensemble. Si, pour un attribut, aucun attribut correspondant n'existe dans l'autre élément, la paire sera "disjointe", ce qui signifie qu'elle consiste en un seul membre. L'item de fonction (troisième argument Function) est appliqué séparément à chaque paire dans la séquence des paires (joints et disjointe), résultant en une sortie qui est une séquence d'items.

Exemples

• altova:for-each-attribute-pair(/Example/Test-A, /Example/Test-B, function(\$a, \$b) {\$a+b}) retourne ...

```
(2, 4, 6) Si
<Test-A att1="1" att2="2" att3="3" />
<Test-B att1="1" att2="2" att3="3" />
(2, 4, 6) Si
<Test-A att2="2" att1="1" att3="3" />
<Test-B att3="3" att2="2" att1="1" />
(2, 6) Si
<Test-A att4="4" att1="1" att3="3" />
<Test-B att3="3" att2="2" att1="1" />
```

<Test-A att4="4" att1="1" att3="3" /><Test-B att3="3" att2="2" att1="1" />

<u>Note</u>: 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
```

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*).

• 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.

```
<u>Exemples</u>
```

• altova:for-each-matching-attribute-pair(/Example/Test-A, /Example/Test-B, function(\$a, \$b){\$a+b}) refourne ...

```
(2, 4, 6) Si
<Test-A att1="1" att2="2" att3="3" />
<Test-B att1="1" att2="2" att3="3" />
(2, 4, 6) Si
<Test-A att2="2" att1="1" att3="3" />
<Test-B att3="3" att2="2" att1="1" />
(2, 6) Si
<Test-A att4="4" att1="1" att3="3" />
<Test-B att3="3" att2="2" att3="1" />
```

• altova:for-each-matching-attribute-pair(/Example/Test-A, /Example/Test-B, concat#2) retourne ...

```
(11, 22, 33) si
<Test-A att1="1" att2="2" att3="3" />
<Test-B att1="1" att2="2" att3="3" />
(11, 33) si
<Test-A att4="4" att1="1" att3="3" />
<Test-B att3="3" att2="2" att1="1" />
```

substitute-empty [altova:]

```
altova:substitute-empty(FirstSequence as item()*, SecondSequence as item()) asitem()*
XP3.1 XQ3.1
Si FirstSequence est vide, retourne SecondSequence. Si FirstSequence n'est pas vide, retourne
FirstSequence.
```

<u>Exemples</u>

- altova:substitute-empty( (1,2,3), (4,5,6) ) retourne (1,2,3)
- altova:substitute-empty((), (4,5,6)) retourne(4,5,6)

#### 18.2.2.1.8 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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section 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.

<u>Exemples</u>

Si l'item contextuel est 1234ABCD:

- altova:char(2) retourne 2
- altova:char(5) retourne A
- **altova:char**(9) retourne la chaîne vide.
- **altova:char**(-2) retourne la chaîne vide.

altova:char(InputString as xs:string, Position as xs:integer) asxs:string XP3.1 XQ3.1 Retourne une chaîne contenant la caractère à la position spécifiée par l'argument Position, dans la chaîne soumise en tant que l'argument InputString. La chaîne de résultat sera vide si aucun caractère n'existe à l'index soumis par l'argument Position.

- <u>Exemples</u>
  - altova:char("2014-01-15", 5) retourne -
  - altova:char("USA", 1) retourne U
  - **altova:char**("USA", 10) retourne la chaîne vide.
  - **altova:char**("USA", -2) retourne la chaîne vide.

create-hash-from-string[altova:]

altova:create-hash-from-string(InputString as xs:string) asxs:string XP2 XQ1 XP3.1 XQ3.1 altova:create-hash-from-string(InputString as xs:string, HashAlgo as xs:string) asxs:string XP2 XQ1 XP3.1 XQ3.1

Génère un string de hashage depuis InputString en utilisant l'algorithme de hashage spécifié par l'argument HashAlgo. Les algorithmes de hashage suivants peuvent être spécifiés (en majuscule ou en minuscule) : MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512. Si le deuxième argument n'est pas spécifié (*voir la première signature ci-dessus*), l'algorithme de hashage SHA-256 sera utilisé.

<u>Exemples</u>

- 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.

<u>Exemples</u>

Si l'item de contexte est 1234ABCD :

- **altova:first-chars**(2) retourne 12
- **altova:first-chars**(5) retourne 1234A
- altova:first-chars(9) retourne 1234ABCD

altova:first-chars(InputString as xs:string, X-Number as xs:integer) asxs:string XP3.1 XQ3.1

Retourne une chaîne contenant le premier X-Number des caractères de la chaîne soumise en tant que l'argument InputString.

- Exemples
  - altova:first-chars("2014-01-15", 5) retourne 2014-
  - altova:first-chars("USA", 1) retourne U
- format-string [altova:]

altova:format-string(InputString as xs:string, FormatSequence as item()\*) asxs:string
XP3.1 XQ3.1

Le string d'entrée (premier argument) contient des paramètres de position (%1, %2, etc). Chaque paramètre est remplacé par l'item de string qui est situé dans la position correspondante dans la séquence de format (soumise en tant que le second argument). Donc le premier item dans la séquence de format remplace de paramètre de positionnement %1, le second item remplace %2, etc. La fonction retourne ce string formaté qui contient les remplacements. Si aucun string n'existe pour un paramètre de positionnement, alors le paramètre de positionnement lui-même est retourné. Cela se produit lorsque l'index d'un paramètre de positionnement est supérieur au nombre d'items dans la séquence de format.

```
Exemples
```

- altova:format-string('Hello %1, %2, %3', ('Jane','John','Joe')) retourne "Hello Jane, John, Joe"
- altova:format-string('Hello %1, %2, %3', ('Jane','John','Joe', 'Tom')) retourne "Hello Jane, John, Joe"
- altova:format-string('Hello %1, %2, %4', ('Jane','John','Joe', 'Tom')) retourne "Hello Jane, John, Tom"
- altova:format-string('Hello %1, %2, %4', ('Jane','John','Joe')) retourne "Hello Jane, John, %4"
- last-chars [altova:]

altova:last-chars(X-Number as xs:integer) asxs:string XP3.1 XQ3.1 Retourne une chaîne contenant le dernier X-Number de caractères de la chaîne obtenue en convertissant la valeur de l'item contextuel en xs:string.

- <u>Exemples</u>
  - Si l'item contextuel est 1234ABCD :
  - altova:last-chars(2) retourne CD
  - altova:last-chars(5) retourne 4ABCD

• **altova:last-chars**(9) **retourne** 1234ABCD

altova:last-chars(InputString as xs:string, X-Number as xs:integer) asxs:string XP3.1 XQ3.1

Retourne une chaîne contenant le dernier X-Number de caractères de la chaîne soumise en tant que l'argument InputString.

<u>Exemples</u>

- altova:last-chars("2014-01-15", 5) retourne 01-15
- altova:last-chars("USA", 10) retourne USA
- pad-string-left [altova:]

altova:pad-string-left(StringToPad as xs:string, StringLength as xs:integer, PadCharacter as xs:string) asxs:string XP3.1 XQ3.1

L'argument PadCharacter est un caractère unique. Il est bourré à la gauche de la chaîne pour augmenter le nombre de caractères dans StringToPad de manière à ce que ce nombre soit équivalent à la valeur d'entier de l'argument StringLength. L'argument StringLength peut avoir toute valeur d'entier (positive ou négative), mais le padding n'aura lieu que si la valeur de StringLength est supérieure au nombre de caractères dans StringToPad. Si StringToPad. comporte plus de caractères que la valeur de StringLength, alors StringToPad ne sera pas modifié.

- Exemples
  - altova:pad-string-left('AP', 1, 'Z') retourne 'AP'
  - altova:pad-string-left('AP', 2, 'Z') retourne 'AP'
  - altova:pad-string-left('AP', 3, 'Z') retourne 'ZAP'
  - altova:pad-string-left('AP', 4, 'Z') retourne 'ZZAP'
  - altova:pad-string-left('AP', -3, 'Z') retourne 'AP'
  - altova:pad-string-left('AP', 3, 'YZ') retourne une erreur pad-character-too-long
- pad-string-right [altova:]

altova:pad-string-right(StringToPad as xs:string, StringLength as xs:integer, PadCharacter as xs:string) asxs:string XP3.1 XQ3.1

L'argument PadCharacter est un caractère unique. Il est bourré à la droite de la chaîne pour augmenter le nombre de caractères dans StringToPad de manière à ce que ce nombre soit équivalent à la valeur d'entier de l'argument StringLength. L'argument StringLength peut avoir toute valeur d'entier (positive ou négative), mais le padding n'aura lieu que si la valeur de StringLength est supérieure au nombre de caractères dans StringToPad. Si StringToPad. comporte plus de caractères que la valeur de StringLength, alors StringToPad ne sera pas modifié.

Exemples

- altova:pad-string-right('AP', 1, 'Z') retourne 'AP'
- altova:pad-string-right('AP', 2, 'Z') retourne 'AP'
- altova:pad-string-right('AP', 3, 'Z') retourne 'APZ'
- altova:pad-string-right('AP', 4, 'Z') retourne 'APZZ'
- altova:pad-string-right('AP', -3, 'Z') retourne 'AP'
- altova:pad-string-right('AP', 3, 'YZ') retourne une erreur pad-character-too-long

repeat-string [altova:]

altova:repeat-string(InputString as xs:string, Repeats as xs:integer) asxs:string XP2 XQ1 XP3.1 XQ3.1

Génère une chaîne qui est composée du premier argument InputString répété Repeats nombre de fois.

• altova:repeat-string("Altova #", 3) retourne "Altova #Altova #Altova #"

substring-after-last [altova:]

altova:substring-after-last(MainString as xs:string, CheckString as xs:string)
asxs:string XP3.1 XQ3.1

Si CheckString est trouvé dans MainString, alors la sous-chaîne qui se produit après CheckString dans MainString est retournée. Si CheckString n'est pas trouvé dans MainString, la chaîne vide est retournée. Si CheckString est une chaîne vide, alors MainString est retourné dans sa totalité. S'il y a plus d'une survenance de CheckString dans MainString, alors la sous-chaîne après la dernière survenance de CheckString est retournée.

- <u>Exemples</u>
  - altova:substring-after-last('ABCDEFGH', 'B') retourne 'CDEFGH'
  - altova:substring-after-last('ABCDEFGH', 'BC') retourne 'DEFGH'
  - altova:substring-after-last('ABCDEFGH', 'BD') retourne ''
  - altova:substring-after-last('ABCDEFGH', 'Z') retourne ''
  - altova:substring-after-last('ABCDEFGH', '') retourne 'ABCDEFGH'
  - altova:substring-after-last('ABCD-ABCD', 'B') retourne 'CD'
  - altova:substring-after-last('ABCD-ABCD-ABCD', 'BCD') retourne ''
- substring-before-last [altova:]

altova:substring-before-last(MainString as xs:string, CheckString as xs:string)
asxs:string XP3.1 XQ3.1

Si CheckString est trouvé dans MainString, alors la sous-chaîne qui se produit avant CheckString dans MainString est retournée. Si CheckString n'est pas trouvé dans MainString, ou si CheckString est une chaîne vide, la chaîne vide est retournée. S'il y a plus d'une survenance de CheckString dans MainString, alors la sous-chaîne avant la dernière survenance de CheckString est retournée.

- <u>Exemples</u>
  - altova:substring-before-last('ABCDEFGH', 'B') retourne 'A'
  - altova:substring-before-last('ABCDEFGH', 'BC') retourne 'A'
  - altova:substring-before-last('ABCDEFGH', 'BD') retourne ''
  - altova:substring-before-last('ABCDEFGH', 'Z') retourne ''
  - altova:substring-before-last('ABCDEFGH', '') retourne ''
  - altova:substring-before-last('ABCD-ABCD', 'B') retourne 'ABCD-A'
  - altova:substring-before-last('ABCD-ABCD-ABCD', 'ABCD') retourne 'ABCD-ABCD-'
- substring-pos [altova:]

altova:substring-pos(StringToCheck as xs:string, StringToFind as xs:string)
asxs:integer XP3.1 XQ3.1

Retourne la position de caractère de la première occurrence de StringToFind dans la chaîne

StringToCheck. La position du caractère est retournée en tant qu'un entier. Le premier caractère de StringToCheck a la position 1. Si StringToFind ne se produit pas dans le cadre de StringToCheck, l'entier 0 est retourné. Pour contrôler la deuxième occurrence ou une occurrence ultérieure de StringToCheck, utiliser la signature suivante de cette fonction.

<u>Exemples</u>

- altova: substring-pos('Altova', 'to') retourne 3
- altova:substring-pos('Altova', 'tov') retourne 3
- altova:substring-pos('Altova', 'tv') retourne 0
- altova:substring-pos('AltovaAltova', 'to') retourne 3

altova:substring-pos(StringToCheck as xs:string, StringToFind as xs:string, Integer as xs:integer) asxs:integer XP3.1 XQ3.1

Retourne la position de caractère de StringToFind dans la chaîne, StringToCheck. La recherche de StringToFind commence à partir de la position de caractère indiquée par l'argument Integer ; la souschaîne du caractère avant cette position n'est pas recherchée. Néanmoins, l'entier retourné, est la position de la chaîne trouvée dans le cadre de la chaîne *entière*, StringToCheck. Cette signature est utile pour trouver la deuxième position ou une position ultérieure d'une chaîne qui se produit plusieurs fois avec StringToCheck. Si StringToFind ne se produit pas dans le cadre de StringToCheck, l'entier 0 est retourné.

Exemples

- altova:substring-pos('Altova', 'to', 1) retourne 3
- altova:substring-pos('Altova', 'to', 3) retourne 3
- altova:substring-pos('Altova', 'to', 4) retourne 0
- altova:substring-pos('Altova-Altova', 'to', 0) retourne 3
- **altova:substring-pos**('Altova-Altova', 'to', 4) retourne 10

trim-string [altova:]

altova:trim-string(InputString as xs:string) asxs:string XP3.1 XQ3.1 Cette fonction prend un argument xs:string, supprime tout espace blanc de tête et de fin et retourne une xs:string « nettoyée ».

<u>Exemples</u>

- altova:trim-string(" Hello World ") retourne "Hello World"
- altova:trim-string("Hello World ") retourne "Hello World"
- altova:trim-string(" Hello World") retourne "Hello World"
- altova:trim-string("Hello World") retourne "Hello World"
- altova:trim-string("Hello World") retourne "Hello World"

trim-string-left [altova:]

altova:trim-string-left(InputString as xs:string) asxs:string XP3.1 XQ3.1 Cette fonction prend un argument xs:string, supprime tout espace blanc de tête, et retourne une xs:string nettoyée à gauche.

Exemples

```
• altova:trim-string-left(" Hello World ") retourne "Hello World "
```

```
• altova:trim-string-left("Hello World ") retourne "Hello World "
```

```
• altova:trim-string-left(" Hello World") retourne "Hello World"
```

```
• altova:trim-string-left("Hello World") retourne "Hello World"
```

• altova:trim-string-left("Hello World") retourne "Hello World"

trim-string-right [altova:]

altova:trim-string-right(InputString as xs:string) asxs:string XP3.1 XQ3.1 Cette fonction prend un argument xs:string, supprime tout espace blanc de fin de ligne, et retourne une xs:string nettoyée à droite.

<u>Exemples</u>

- altova:trim-string-right(" Hello World ") retourne " Hello World"
- altova:trim-string-right("Hello World ") retourne "Hello World"
- altova:trim-string-right(" Hello World") retourne " Hello World"
- altova:trim-string-right("Hello World") retourne "Hello World"
- altova:trim-string-right("Hello World") refourne "Hello World"

#### 18.2.2.1.9 Fonctions XPath/XQuery : Divers

L'objectif général suivant des fonctions d'extension XPath/XQuery sont prises en charge dans la version actuelle de MapForce 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**, <a href="http://www.altova.com/xslt-extensions">http://www.altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova.com/xslt-extensions</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section par le préfixe <a href="https://www.altova.com/xslt-extensions">altova</a>, et sont indiquées dans cette section peut être interrompue et le comportement de certaines fonctions peut changer. Veuillez consulter la documentation lors des publications à venir pour plus d'informations concernant la prise en charge des fonctions d'extension Altova de cette version.

Fonctions XPath (utilisées dans les expressions XPath dans XSLT) :	XP1 XP2 XP3.1
Fonctions XSLT (utilisées dans les expressions XPath dans XSLT) :	XSLT1 XSLT2 XSLT3
Fonctions XQuery (utilisées dans les expressions XQuery dans XQuery) :	XQ1 XQ3.1

decode-string [altova:]

altova:decode-string(Input as xs:base64Binary) as xs:string XP3.1 XQ3.1

altova:decode-string(Input as xs:base64Binary, Encoding as xs:string) as xs:string XP3.1 XQ3.1

Décode l'entrée base64Binary soumise à un string en utilisant l'encodage spécifié. Si aucun codage n'est spécifié, l'encodage UTF-8 est utilisé. Les encodages suivants sont pris en charge: US-ASCII, ISO-8859-1, UTF-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4

- Exemples
  - altova:decode-string(\$XML1/MailData/Meta/b64B) retourne l'entrée base64Binary en tant qu'un string encodé en UTF-8
  - **altova:decode-string**(\$XML1/MailData/Meta/b64B, "UTF-8") retourne l'entrée base64Binary en tant qu'un string encodé en UTF-8
  - altova:decode-string(\$XML1/MailData/Meta/b64B, "ISO-8859-1") retourne l'entrée base64Binary en tant qu'un string encodé en ISO-8859-1
- get-temp-folder [altova:]

altova:get-temp-folder() as xs:string XP2 XQ1 XP3.1 XQ3.1 Cette fonction ne prend aucun argument. Elle retourne le chemin vers le dossier temporaire de l'utilisateur actuel.

- Exemples
  - altova:get-temp-folder() retournerait, sur une machine Windows, quelque chose du genre C: \Users\<UserName>\AppData\Local\Temp\ en tant que xs:string.
- generate-guid [altova:]

altova:generate-guid() asxs:string XP2 XQ1 XP3.1 XQ3.1 Génère un string GUID unique.

- Exemples
  - altova:generate-guid() retourne (par exemple) 85F971DA-17F3-4E4E-994E-99137873ACCD
- high-res-timer [altova:]

#### altova:high-res-timer() asxs:double XP3.1 XQ3.1

Retourne une valeur de minuteur haute résolution en secondes. Un minuteur de haute résolution, lorsqu'il est présent dans un système, permet des mesures temporelles de haute précision lorsque celles-ci sont requises (par exemples pour des animations et pour déterminer précisément l'heure d'exécution du code). Cette fonction fournit la résolution du minuteur haute résolution du système.

```
Exemples
```

- altova:high-res-timer() retourne quelque chose comme '1.16766146154566E6'
- parse-html [altova:]

```
altova:parse-html(HTMLText as xs:string) asnode() XP3.1 XQ3.1
L'argument HTMLText est un string qui contient le texte d'un document HTML. La fonction crée une
```

arborescence HTML depuis le string. Le string soumis peut contenir l'élément HTML ou pas. Dans tous les cas, l'élément racine de l'arborescence est un élément nommé HTML. Il est préférable de s'assurer que le code HTML dans le string soumis est un HTML valide.

Exemples

• altova:parse-html("<html><head/><body><h1>Header</h1></body></html>") crée une arborescence HTML depuis le string soumis

sleep[altova:]

altova:sleep(Millisecs as xs:integer) asempty-sequence() XP2 XQ1 XP3.1 XQ3.1 Suspend l'exécution de l'opération actuelle pour le nombre de millisecondes donné par l'argument Millisecs.

∃ <u>Exemples</u>

• altova:sleep(1000) suspend l'exécution de l'opération actuelle pour 1000 millisecondes.



### 18.2.2.2 Fonctions d'extension diverses

Il existe plusieurs types de fonctions prêtes à l'utilisation dans les langages de programmation comme Java et C# qui ne sont pas disponibles en tant que fonctions XQuery/XPath ou en tant que fonctions XSLT. Un bon exemple sont les fonctions mathématiques disponibles en Java, comme sin() et cos(). Si ces fonctions étaient disponibles aux designers des feuilles de style XSLT et des requêtes XQuery, cela augmenterait le champ d'application des feuilles de style et des requêtes et faciliterait considérablement les tâches des créateurs de feuilles de style. Les moteurs XSLT et XQuery utilisés dans un grand nombre de produits Altova prennent en charge l'utilisation des fonctions d'extension dans Java et .NET, et pour les <u>scripts MSXSL pour XSLT</u><sup>(169)</sup>. Cette section décrit comment utiliser les fonctions d'extension et les scripts MSXSL dans vos feuilles de scripts XSLT et les documents XQuery. Les fonctions d'extension disponibles sont organisées dans les sections suivantes :

- Fonctions d'extension Java
- Fonctions d'extension .NET
- Scripts MSXSL pour XSLT<sup>1562</sup>

Les deux problèmes principaux considérés pour les descriptions sont : (i) comment sont appelées les fonctions dans les librairies respectives ; et (ii) quelles sont les règles à suivre pour la conversion d'arguments 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.

#### 18.2.2.2.1 Scripts MSXSL pour XSLT

L'élément <msxsl:script> contient des fonctions définies par l'utilisateur et des variables qui peuvent être appelées depuis des expressions XPath dans la feuille de style XSLT. Le <msxsl:script> et un élément de niveau supérieur, c'est à dire, qu'il doit être un élément enfant de <xsl:stylesheet> ou <xsl:transform>.

L'élément <msxsl:script> doit être dans l'espace de nom urn:schemas-microsoft-com:xslt (voir exemple ci-dessus).

#### Langage de script et espace de nom

Le langage de script utilisé dans le bloc est spécifié dans l'attribut language de l'élément <msssl: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 <msssl:script> se trouveront dans l'espace de nom identifié par le préfixe spécifié dans l'attribut implements-prefix. Lorsqu'une fonction est appelée depuis une expression XPath, le nom de la fonction entièrement qualifié doit se trouver dans le même espace de nom que la définition de la fonction.

#### Exemple

Voici un exemple d'une feuille de stype XSLT complète qui utilise une fonction définie dans un élément <msxsl:script>.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:msxsl="urn:schemas-microsoft-com:xslt"
    xmlns:user="http://mycompany.com/mynamespace">
```

```
Annexes
```

```
<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
  11>
  </msxsl:script>
  <xsl:template match="/">
   <html>
     <body>
        <p>
          <b>Total Retail Price =
            $<xsl:value-of select="user:AddMargin(50)"/>
          </b>
          <br/>
          <b>Total Wholesale Price =
            $<xsl:value-of select="50"/>
          </b>
        </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 msxs1: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 msxs1: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.

## 18.3 Données techniques

Cette section contient des informations utiles concernant certains aspects techniques de votre logiciel. Cette information est organisée dans les sections suivantes :

- <u>SE et exigences de mémoire</u>
   <sup>1565</sup>
- Moteurs Altova 1565
- Prise en charge Unicode
   <sup>1566</sup>
- <u>Utilisation Internet</u><sup>1566</sup>

## 18.3.1 SE et exigences de mémoire

#### Système d'exploitation

Les applications logicielles d'Altova sont disponibles pour les plateformes suivantes :

- Windows 7 SP1 avec mise à jour de la plateforme, Windows 8, Windows 10, Windows 11
- Windows Server 2008 R2 SP1 avec mise à jour de la plateforme ou plus récent

#### Mémoire

Étant donné que le logiciel est rédigé en C++, il ne nécessite pas la performance d'un Java Runtime Environment et nécessite généralement moins de mémoire que les applications basées sur Java comparables. Néanmoins, chaque document est chargé entièrement dans la mémoire de manière à ce qu'il puisse être parsé complètement et pour améliorer la vitesse de la consultation et de l'édition. Les exigences de mémoire augmentent avec la taille du document.

Les exigences de mémoire sont aussi affectées par un historique de la fonction Annuler non limité. Si vous coupez/collez sans arrêt de larges sélections dans des documents volumineux, la mémoire disponible peut baisser rapidement.

## 18.3.2 Moteurs Altova

#### Validateur XML

Lors de l'ouverture d'un document XML, l'application utilise son validateur XML intégré pour vérifier la bonne formation, pour valider le document par rapport à un schéma (si spécifié), et de générer des arborescences et infosets. Le validateur XML est aussi utilisé pour fournir une édition intelligente pendant que vous éditez des documents et pour afficher dynamiquement toute erreur de validation qui peut se produire.

Le validateur XML intégré met en place les spécifications de la Final Recommendation of the W3C's XML Schema 1.0 et 1.1. Les nouveaux développements recommandés par le Groupe de travail de Schéma XML de W3C sont incorporés en continu dans le validateur XML, afin que les produits Altova vous donnent un environnement de développement de pointe.

#### Moteurs XSLT et XQuery

Les produits Altova utilisent les moteurs Altova XSLT 1.0, 2.0 et 3.0, et les moteurs Altova XQuery 1.0 et 3.1. Si un de ces moteurs est inclus dans le produit, la documentation concernant le comportement spécifique à la mise en place pour chaque moteur est indiquée dans les annexes de la documentation.

Note: Altova MapForce génère du code utilisant les moteurs XSLT 1.0, 2.0 et XQuery 1.0.

## 18.3.3 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.

## 18.3.4 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.

## 18.4 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 <u>la page des informations juridiques Altova</u> sur le <u>site web Altova</u>.

## **18.4.1** Distribution électronique de logiciel

Ce produit est disponible par le biais de la distribution électronique de logiciel, une méthode de distribution qui fournit les avantages uniques suivants :

- Vous pouvez évaluer gratuitement le logiciel pendant 30 jours avant de vous décider à l'achat. (Note : Altova MobileTogether Designer dispose d'une une licence gratuite.)
- Une fois que vous avez décidé d'acheter le logiciel, vous pouvez passer vos commandes en ligne sur le site web Altova et vous obtiendrez en quelques minutes un produit bénéficiant d'une pleine licence.
- Lorsque vous passez une commande en ligne, vous disposerez toujours de la dernière version de nos logiciels.
- Le pack de produits comprend une aide sur écran qui peut être accédé depuis l'intérieur de l'interface de l'application. La dernière version du manuel d'utilisateur est disponible sous <u>www.altova.com</u> (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.

## 18.4.2 Activation de logiciel et le license metering

En tant que partie intégrante de l'Activation du logiciel Altova, le logiciel peut utiliser votre réseau interne et votre connexion Internet à des fins de transmission des données relatives à la licence au moment de l'installation, de l'enregistrement, de l'utilisation ou de la mise à jour d'un serveur de licence utilisé par Altova et valider l'authenticité des données relatives à la licence pour protéger Altova contre une utilisation sans licence ou illégale du logiciel et pour améliorer le service clientèle. L'activation est basée sur l'échange des données relatives aux licences comme les systèmes d'exploitation, l'adresse IP, la date/heure, la version de logiciel et le nom de l'ordinateur, ainsi que d'autres informations échangées entre votre ordinateur et un serveur de licence Altova.

Votre produit Altova comporte un module intégré de contrôle des licences qui vous aide à éviter toute violation non-intentionnelle du contrat de licence de l'utilisateur final. Votre produit est licencé soit en tant qu'une installation utilisateur simple soit en tant qu'installation multi-utilisateur, et le module de contrôle des licences permet de vous assurer qu'aucune licence outre celles accordées pour le nombre d'utilisateurs sous licence n'utilise l'application simultanément.

Cette technologie de contrôle des licences utilise votre réseau local (LAN) pour communiquer entre les instances de l'application exécutée sur plusieurs ordinateurs.

#### Licence simple

Lorsque l'application est démarrée dans le cadre du processus de contrôle de la licence, le logiciel envoie un bref datagramme de diffusion pour trouver d'autres instances du produit exécuté sur d'autres ordinateurs dans le même segment de réseau. S'il n'obtient pas de réponses, il ouvrira un port pour écouter d'autres instances de l'application.

#### Licence utilisateurs multiples

Si plus d'une seule instance de l'application est utilisée dans le même LAN, ces instances communiqueront brièvement l'une avec l'autre lors du démarrage. Ces instances échangent des codes-clés afin de vous aider à mieux déterminer que le nombre de licences concurrentes achetées n'est pas violé accidentellement. Il s'agit de la même technologie de contrôle des licences généralement utilisée dans l'univers Unix et dans un certain nombre d'outils de développement de bases de données. Elle permet aux clients Altova d'acheter des licences multi-utilisateurs d'utilisation simultanée à des prix raisonnables.

Nous avons également conçu les applications de manière à ce qu'elles envoient des paquets de réseau peu importants et peu nombreux pour ne pas surcharger votre réseau. Les ports TCP/IP (2799) utilisés par votre produit Altova sont officiellement enregistrés auprès de l'IANA (*voir <u>IANA Service Name Registry</u> pour plus de détails*) et notre module le contrôle de licence est testé et éprouvé technologiquement.

Si vous utilisez un pare-feu, vous pourrez éventuellement apercevoir des communications sur le port 2799 entre les ordinateurs qui exécutent les produits Altova. Vous pouvez, bien évidemment, bloquer ce trafic entre les groupes différents dans votre entreprise, du moment que vous pouvez assurer par d'autres moyens que votre contrat de licence n'a pas été violé.

Vous noterez également que, si vous êtes en ligne, votre logiciel Altova contient de nombreuses fonctions utiles ; celles-ci ne concernent pas la technologie de contrôle des licences.

#### Note à propos des certificats

Votre application Altova contacte le serveur de mise sous licence Altova (link.altova.com) via HTTPS. Pour établir cette communication, Altova utilise un certificat SSL enregistré. Si ce certificat est remplacé (par exemple, par votre département IT ou une agence externe), votre application Altova vous avertira que la connexion n'est pas sûre. Vous pourriez utiliser le certificat de remplacement pour lancer votre application Altova, mais vous le ferez à vos propres risques et périls. Si vous voyez un message d'avertissement *Connexion non-sécurisée*, vérifiez l'origine du certificat et consultez votre équipe IT (qui sera en mesure de décider si l'interception et le remplacement du certificat Altova devrait continuer ou pas).

Si votre organisation nécessite d'utiliser son propre certificat (par exemple, pour surveiller la communication de et vers les machines client), nous vous recommandons d'installer le logiciel de gestion de licence gratuit d'Altova, <u>Altova LicenseServer</u>, 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.

## **18.4.3** 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 : <u>https://www.altova.com/privacy</u>

# Index

# A

#### AàZ,

trier le composant, 517

#### abs,

en tant que fonction MapForce (dans lang | math functions), 750

en tant que fonction MapForce (dans xpath2 | numeric functions), 806

#### acos,

en tant que fonction MapForce (dans lang | math functions), 750

#### Actions,

lié à la connexion, 87

#### ActiveX,

integration at application level, 1410 integration at document level, 1413 integration prerequisites, 1406

#### ActiveX controls,

adding to the Visual Studio Toolbox, 1408

#### add,

en tant que fonction MapForce (dans core | math functions), 669

#### ADO,

configurer une connexion, 165

en tant qu'interface de connexion de données, 158

#### ADO.NET,

configurer une connexion, 171

#### age,

en tant que fonction MapForce (dans lang | datetime functions), 726

#### Aide,

À propos de MapForce, 1060 Activation logiciel, 1060 Centre de support, 1060 FAQ sur le web, 1060 Formation MapForce, 1060 Formulaire de commande, 1060 Index, 1060 Inscription, 1060 MapForce sur Internet, 1060 Recherche, 1060 Télécharger les composants et les outils gratuits, 1060 Vérifier les mises à jour, 1060

#### Ant,

configurer les variables d'environnement pour,, 1200 construire du code Java avec, 1087

Construire un projet Eclipse avec, 1084

#### API,

documentation, 1221

#### Application object, 1225

#### archive-log,

en tant que fonction MapForce (dans lang | math functions), 753

#### asin,

en tant que fonction MapForce (dans lang | math functions), 751

#### atan,

en tant que fonction MapForce (dans lang | math functions), 751

#### ATTLIST,

URI d'espace de noms DTD, 121

#### Automatique,

chargement des bibliothèques, 613

#### Automatisé,

en cours de traitement, 950

#### auto-number,

en tant que fonction MapForce (dans core | generator functions), 660

#### avg,

En tant que fonction MapForce (dans core | aggregate functions), 634

## B

#### Barres,

Barres d'outils, 31 Menu, 31 Statut de l'application, 31 **Base de données,** attribuer le Schéam XML dans le champ, 372 écrire fichiers XML dans, 375 Éditeur SQL, 1051 Encodage, 1051 Géneration SQL, 1051 générer des valeurs séquentielles et uniques pour, 280 générer plusieurs fichiers XML depuis, 901 Mode des Résultats, 1051 paramètres d'édition SQL, 1051 Base de données, Polices de texte, 1051 requête. 382 Base de données Access, mise à jour sur la base d'une condition IF, 459 base de données Oracle, se connecter par le biais d'ODBC, 227 se connecter par le biais de JDBC, 225 Base de données Progress OpenEdge, se connecter par le biais d'ODBC, 237 se connecter par le biais de JDBC, 235 Bases de données, ajouter au mappage, 249 comme Ressources globales, 982 et contexte de mappage, 876 exécuter des mappages par rapport à, 276 fusionner les données dans, 298 insérer des données dans plusieurs tables liées, 290 insérer des données dans une table, 286 mettre à jour et insérer les données de table, 298 mettre à jour les données de table, 295 supprimer les données de table, 315 utiliser l'insert bulk, 323 base-uri, En que fonction MapForce (dans xpath2 | accessors library), 776 Batch, automatisation en cours de traitement, 950 BD. TRIER PAR. 355 Bibliothèque, 1217 Chargement automatique de, 613 Bibliothèques personnalisées, add, 612 C#, 612 C++, 612 Java, 612 référence, 612 Binary files, adding to the mapping, 471 overview, 471 reading data from, 473 writing data to, 474 booléenne, en tant que fonction MapForce (dans core | conversion functions), 641 build.xml. activer le zip64mode dans,, 1200

## С

#### C#,

code, 1065 error handling, 1226 générer du code de programme, 1074 integration of MapForce, 1417 intégrer du code généré, 1098 paramètres de génération code, 1201 C++, code, 1065 error handling, 1226 générer du code de programme, 1069 intégrer du code généré, 1100 paramètres de génération code, 1201 capitalize, en tant que fonction MapForce (dans lang | string functions), 760 Caractère de remplissage, dans les fichiers de texte plats, 463 Caractères génériques, Importer un schéma, 134 schéma wrapper, 134 sélections, 134 xs:any/xs:anyAttribute, 134 Catalogues, 993 dans DTD, 994 dans Schéma XML, 994 personnaliser, 998 Variables d'environnement, 1000 CDATA, section, 133 ceiling, en tant que fonction MapForce (dans core | math functions), 669 Champ, clés dans les champs de texte, 451 Champs vides, dans les fichiers CSV, 455 dans les fichiers de texte plats, 463 char-from-code. en tant que fonction MapForce (dans core | string functions), 708 charset-decode, en tant que fonction MapForce (dans lang | string functions), 760

#### charset-encode, Commandes de menu, 1024 en tant que fonction MapForce (dans lang | string functions), Aide, 1060 762 Composant, 1034 Chemins. Connexion, 1036 dans le code généré, 112 Déboguer, 1040 rendre absolu, 112 Éditer, 1028 Chemins de fichier, Fichier, 1025 absolu, 82, 83 Fonction, 1037 cassé, 83 Insérer, 1029 dans des environnements d'exécution, 85 Mode, 1041 dans le code généré, 85 **Outils**, 1043 des bases de données basées sur le fichier, 83 Outils | Clavier, 1045 relatif. 82. 83 Outils | Options, 1048 relatif vs absolu, 85 Outils | Options | Base de données, 1051 réparer les références cassées, 83 Outils | Options | Java, 1050 Cible, 24 Outils | Options | Proxy de réseau, 1056 Class ID, Outils | Personnaliser, 1044 in MapForce integration, 1424 Personnaliser, 1044, 1045 Classes Wrapper, Projet, 1032 dans code généré, 1201 Sortie, 1038 Clé, Windows, 1059 champs dans les fichiers de texte, 451 Commentaire en bloc, 387 trier la clé, 517 Commentaire en ligne, 387 Clé étrangère, Commentaires, dans les mappages de base de données, 260, 270 ajouter au fichier cible, 132 Clé primaire, Composant, dans les mappages de base de données, 260, 270 trier les données, 517 générer pendant l'action d'insertion de la base de données, Composant de cible, 286 Changer l'ordre de traitement de, 889 générer pour un composant de base de données, 280 Composants, Code, Actions de Table de base de données, 1034 intégrer code MapForce, 1095 Actualiser, 1034 SPL, 1203 ajouter au mappage, 78 types intégrés, 1216 Ajouter Entrée dupliquée après, 1034 Code generation, Ajouter Entrée dupliquée avant, 1034 sample, 1241 Ajouter/Supprimer/Éditer des objets de base de données, Code généré, 1034 lancer des exceptions depuis, 547 aligner, 80 code-from-char, Aligner arborescence à droite, 1034 en tant que fonction MapForce (dans core | string Aligner arborescence à gauche, 1034 functions), 708 aperçu, 76 Collation, Base de données, 1029 collation locale, 517 bases, 80 Point de code unicode, 517 Bases de données, 156 trier le composant, 517 changer les paramètres, 80 collation locale, 517 chercher, 80 COM API, commande de menu, 1034 documentation, 1221 Condition IF-Else, 1029

Composants, Constante, 1029 Créer un mappage pour EDI X12 997, 1034 Créer un mappage pour EDI X12 999, 1034 Document XBRL, 1029 Écrire du contenu en tant que section CDATA, 1034 EDI, 1029 Éditer Configuration FlexText, 1034 Éditer une définition de Schéma dans XMLSpy, 1034 Entrée simple, 1029 Exception, 1029 Fichier Excel 2007+, 1029 Fichier Protocol Buffers, 1029 Fichiers de texte, 1029 Filtrer: Nœuds/Lignes, 1029 Fonction de Service web, 1029 Insérer entrée, 1029 Insérer sortie, 1029 items supprimés, 100 Join. 1029 Modifier Élément racine, 1034 paramètres, 80 Propriétés, 1034 référence d'icône, 76 Requête de base de données, 1034 Schéma JSON/Fichier, 1029 Schéma XML, 122 Schéma XML/Fichier, 1029 Sortie simple, 1029 SQL/NoSQL-WHERE/ORDER, 1029 structurel, 76, 120, 121, 156 supprimer, 101 Supprimer doublon, 1034 transformation, 76 Trier: Nœuds/Lignes, 1029 Value-Map, 1029 Variable, 1029 XML. 122 XML et Schéma XML, 122, 126, 128, 132, 133, 134, 137 Composants de structure, base de données, 156 Schéma XML, 121 XML, 121 XML et Schéma XML, 121 concat. (en tant que fonction) exemple d'utilisation, 459 en tant que fonction MapForce (dans core | string

functions), 709

Conditions If-Else, ajouter au mappage, 523 Configurer, fichier mff, 613 Connecteur, consulter l'historique des valeurs traitées, 937 Connexion de base de données, exemples de configuration, 191 Connexion de la base de données, configuration, 158 lancer l'assistant, 160 réutilisation depuis les Ressources Globales, 190 Connexions, annotation, 97 Auto connexion des enfants correspondants, 1036 Connecter les enfants correspondants, 1036 connexions parent manquantes, 87 copier, 87 Copier-tout (Copier les items enfants), 1036 copy-all, 90, 95 correctif, 100 corriger après l'édition du schéma, 100 créer, 87 déplacer, 87, 100 enfants correspondants, 90, 93 entrées obligatoires, 87 garder les après avoir supprimé les composants, 101 menu contextuel, 98 mettre en surbrillance de manière sélective. 87 mixte, 90 modifier. 87 Orienté vers la cible (Standard), 1036 Orienté vers la source (contenu mixte), 1036 orientée vers la cible, 90 orientée vers la source, 90 paramètres, 97 Paramètres pour connecter des enfants correspondants, 1036 Propriétés, 1036 standard, 90 supprimer, 87 types, 90, 97 Voir les info-bulles, 87 Connexions incorrectes, après avoir changé le schéma, 100 dans bases de données, 100 dans fichiers XML, 100 Constantes.

1575

#### Constantes, ajouter, 552 contains, $\square$ en tant que fonction MapForce (dans core | string functions), 710 Contenu mixte. Avec des connexions orientées vers la cible, 91 avec des connexions standard, 91 mappage, 91 contexte de mappage, 874 Contexte de parent, exemple, 880 Contexte de priorité, 884 exécuter des mappages avec, 886 Contrat de licence de l'utilisateur final, 1568, 1570 Conventions, 23 convert-to-utc, en tant que fonction MapForce (dans lang | datetime functions), 726 cos. en tant que fonction MapForce (dans lang | math functions), 751 count, En tant que fonction MapForce (dans core | aggregate functions). 635 count-substring, en tant que fonction MapForce (dans lang | string functions), 764 create-guid, en tant que fonction MapForce (dans lang | generator functions), 747 Créer, régions, 388 CSV. créer des hiérarchies - clés, 451 créer plusieurs lignes, 448 en tant que source de mappage, 446 current, en tant que fonction MapForce (dans xslt | xslt functions préparation pour, 930 library), 836 default-collation, current-date, en tant que fonction MapForce (dans xpath2 | context functions), 781 current-dateTime, en tant que fonction MapForce (dans xpath2 | context 752 functions), 781 current-time, en tant que fonction MapForce (dans xpath2 | context functions), 782

#### Data streaming,

définition, 78

#### date-from-datetime,

en tant que fonction MapForce (dans lang | datetime functions), 727

#### datetime-add,

en tant que fonction MapForce (dans lang | datetime functions), 727

#### datetime-diff,

en tant que fonction MapForce (dans lang | datetime functions), 728

#### datetime-from-date-and-time,

en tant que fonction MapForce (dans lang | datetime functions), 729

#### datetime-from-parts,

en tant que fonction MapForce (dans lang | datetime functions), 730

#### day-from-datetime,

en tant que fonction MapForce (dans lang | datetime functions), 732

#### day-from-duration,

en tant que fonction MapForce (dans lang | datetime functions), 732

#### Débogage,

à propos de, 931 arrêter, 1040 avec des points d'arrêt, 926 démarrer, 1040 étape minimale, 1040 étape par étape, 926 limitations, 926 pas à pas détaillé, 1040 pas à pas principal, 1040 pas à pas sortant, 1040

en tant que fonction MapForce (dans xpath2 | context functions), 782

#### degrees,

en tant que fonction MapForce (dans lang | math functions),

#### Délimiteur,

modifier dans les fichiers CSV, 455

#### distinct-values,

#### distinct-values,

en tant que fonction MapForce (dans core | sequence functions), 681

#### Distribution,

des produits logiciels Altova, 1568

#### des produits logiciels d'Altova, 1568

#### divide,

en tant que fonction MapForce (dans core | math functions), 670

#### divide-integer,

en tant que fonction MapForce (dans lang | math functions), 752

#### Dlimiteur,

modification dans les fichiers de texte plats, 463

#### document,

en tant que fonction MapForce (dans xslt | xslt functions library), 836

#### Documenter,

mappages, 916

#### Document-level,

examples of integration of XMLSpy, 1417

#### données de table,

trier, 517

#### Données XML,

écrire dans les champs de base de données, 372 lecture depuis les champs de base de données, 372

#### Dossier,

mise en page - Requête de base de données, 389

#### Dossiers,

comme Ressources globales, 980

#### DoTransform.bat,

exécuter avec RaptorXML Server, 949

#### Double entrée, 51

DSN Fichier,

configuration, 178

#### DSN Système,

configuration, 178

#### DSN utilisateur,

configuration, 178

#### DTD,

source et cible, 121

#### duration-add,

en tant que fonction MapForce (dans lang | datetime functions), 733

#### duration-from-parts,

en tant que fonction MapForce (dans lang | datetime functions), 733

#### duration-subtract,

en tant que fonction MapForce (dans lang | datetime functions), 735

## Ε

#### Eclipse,

compiler du code Java avec, 1087 générer du code de mappage pour, 1081

#### Éditer,

Annuler, 1028 Couper/ Copier/ Coller/ Supprimer, 1028 Recherche, 1028 Rétablir, 1028 Tout sélectionner, 1028 Trouver précédent, 1028 Trouver suivant, 1028

#### Éditeur SQL,

créer des régions, 388 décommenter du texte, 387 insérer des commentaires, 387 insérer des régions, 388 insérer des signets, 387 marge du signet, 387 supprimer des commentaires, 387 Supprimer des régions, 388 supprimer des signets, 387 utiliser des régions, 388

#### utiliser des signets, 387 égal,

en tant que fonction MapForce (dans core | logical functions), 663

#### Élargir,

régions, 388

#### element-available,

en tant que fonction MapForce (dans xslt  $\mid$  xslt functions library), 837

#### empty,

en tant que fonction MapForce (dans lang | string functions), 765

#### Encodage,

modification dans les fichiers de texte plats, 463 modifier dans les fichiers CSV, 455

#### Entrée,

dupliquer, 80

#### Entrée de mappage,

Fournir plusieurs fichiers en tant que, 893, 896, 898 fournir un nom de fichier personnalisé en tant que, 899

Enumerations, in MapForceControl, 1472 equal-or-greater, en tant que fonction MapForce (dans core | logical functions), 664 equal-or-less. en tant que fonction MapForce (dans core | logical functions), 664 Erreurs. dépannage, 78 mémoire insuffisante, 78 Error handling, general description, 1226 Espaces de noms, déclarer manuellement, 137 personnaliser, 137 Exception de base de données, gestion, 423, 429 journalisation, 440 Exception Mémoire insuffisante, résoudre, 1200 Exceptions. ajout, 547 exemple, 548 lancer quand le nœud est manquant, 548 Exceptions de base de données, journalisation, 435 Exigences de mémoire, 1565 exists. en tant que fonction MapForce (dans core | sequence functions), 683 exp, en tant que fonction MapForce (dans lang | math functions), 753 Expressions régulières, utilisé dans des mappages, 627 Extensions Altova, fonctions graphiques (voir fonctions graphique), 1487 F false, en tant que fonction MapForce (dans xpath2 | boolean functions), 779 Fenêtre Contexte, à propos de, 931, 939

Fenêtre Points d'arrêt,

à propos de, 931, 941 Fenêtre Valeurs, à propos, 937 à propos de, 931 Onglet Contexte, 937 Onglet Historique, 937 Onglet Lié, 937 Fichier, Aperçu d'impression, 1025 comme une touche dans un composant, 80 Compiler sur fichier d'exécution MapForce Server, 1025 Déployer sur FlowForce Server, 1025 en tant que touche sur des composants, 893 Enregistrer, 1025 Enregistrer sous, 1025 Fermer, 1025 Fermer tout, 1025 Fichiers récents, 1025 Générer code, 1025 Générer documentation, 1025 Imprimer, 1025 Nouveau, 1025 Open, 1025 Ouvrir gestionnaire d'identifiant, 1025 Paramètres de mappage, 1025 Paramètres d'impression, 1025 Quitter, 1025 Recharger, 1025 Tout enregistrer, 1025 Valider mappage, 1025 Fichier : (default), en tant que nom de nœud de racine, 893 Fichier : <dynamic>, en tant que nom de nœud de racine, 893 fichier mff, configurer, 613 Fichier plat, mappage, 446 Fichier/String, comme une touche dans un composant, 80 en tant que touche sur des composants, 893 Fichiers CSV, ajouter ou supprimer des champs dans,, 455 définir l'encodage de,, 455 en tant que composant cible, 455 en tant que composant de source, 455 prévisualiser des données provenant de,, 455 Fichiers de projet (.mfp), 115

#### Fichiers texte, ajouter ou supprimer des champs dans,, 463 configurer l'encodage de,, 463 définir la taille du champ fixe, 459 définir le caractère de remplissage, 459 en tant que composant cible, 463 en tant que composant source, 463 mapper des données depuis, 459 prévisualiser les données depuis,, 463 Fichiers XML, comme Ressources globales, 978 générer depuis les enregistrement de base de données, 901 générer depuis une seule source XML, 900 Filtre, objets de base de données sur le mappage, 254 Filtrer. dans bases de données, 359 données des composants, 523 le Navigateur en ligne, 390 objets de base de données. 390 tables de base de données. 523 Filtres, ajouter au mappage, 523 find-substring, en tant que fonction MapForce (dans lang | string functions), 765 Firebird, Se connecter par le biais de JDBC, 192 se connecter par le biais de ODBC, 193 first-items. en tant que fonction MapForce (dans core | sequence functions), 685 Fixe. longueur des fichiers - mappage, 446 floor, en tant que fonction MapForce (dans core | math functions), 670 Flow Force Server, déployer des mappages dansin, 960 déployer des ressources globales dansResources to, 960 déployer des Ressources globales sur, 989 Ressources globales dans, 989 fonction nœud-nom. alternatives d'utilisation, 851 Fonctions, 551 ajouter, 552 ajouter des paramètres, 552 bases, 552

chercher, 552 constantes, 552 Créer une fonction définie par l'utilisateur, 1037 Créer une fonction définie par l'utilisateur depuis la sélection, 1037 description, 552 fonctions dans la fenêtre Bibliothèques, 552 Insérer entrée, 1037 Insérer sortie, 1037 nœud, 560, 562 paramètres, 552 Paramètres de fonction, 1037 supprimer des paramètres :, 552 Supprimer la fonction, 1037 trouver des occurrences dans le mappage actif, 552 type de données d'argument, 552 Fonctions définies par l'utilisateur, ajouter des paramètres, 581 aperçu, 575 appeler, 576 appeler récursivement, 586 avantages, 575 copier-coller, 576 créer, 576 de type complexe, 581 de type simple, 581 exemple, 575 exemples, 586, 588 importer, 576 inline, 576 lookup, 588 modifier, 576 naviguer, 576 ordre du paramètre, 581 paramètres, 581 paramètres de sortie, 576 paramètres d'entrée, 576 recherche récursive, 586 récursif, 586 régulier, 576 structures de type complexe, 581 supprimer, 576 Fonctions d'extension dans les scripts MSXSL, 1562 Fonctions d'extension pour XSLT et XQuery, 1561 Fonctions nœud, créer. 562 Fonctions par défaut et Fonctions de nœud, annotation. 571

Fonctions par défaut et Fonctions de nœud,

appliquer conditionnellement, 565 configuration de la règle, 562 configurer, 562 côté entrée, 560 côté sortie, 560 créer, 562 défauts pour nœuds non connectés, 565 échelle, 571 écraser les règles héritées, 565 filtre, 565 fractionDigits, 571 Icônes de fonction, 560 indices visuels, 562 longueur, 571 maxLength, 571 métadonnées, 571 minLength, 571 precision, 571 règles de blocage, 565 scénarios, 565

totalDigits, 571

#### format-date,

en tant que fonction MapForce (dans core | conversion functions), 641

#### format-dateTime,

en tant que fonction MapForce (dans core | conversion functions), 642

#### format-guid-string,

en tant que fonction MapForce (dans lang | string functions), 766

#### format-number,

en tant que fonction MapForce (dans core | conversion functions), 645

#### format-time,

en tant que fonction MapForce (dans core | conversion functions), 648

#### function-available,

en tant que fonction MapForce (dans xslt  $\mid$  xslt functions library), 837

## G

#### generate-id,

en tant que fonction MapForce (dans xslt | xslt functions library), 838

generate-sequence,

en tant que fonction MapForce (dans core | sequence functions), 686 Générateur de Code, 1065 Génération de Code, 105, 112 Générer, du code depuis le schéma, 1065 Gestionnaire de schéma, aperçu CLI, 149 aperçu de, 139 Commande Aide CLI, 149

Commande de mise à jour CLI, 154 Commande de mise à niveau CLI, 155

- Commande désinstallation CLI. 153
- Commande Info CLI, 150

Commande Initialiser CLI, 150

- Commande Installer CLI, 151 Commande Liste CLI, 152
- Commande Réinitialiser CLI, 153
- comment exécuter, 142
- corriger un schéma, 147
- désinstaller un schéma. 148
- installer un schéma. 147
- mettre à niveau un schéma, 147
- recenser les schémas par statut dans, 145
- réinitialiser, 148
- statut de schémas dans, 145

#### get-fileext,

en tant que fonction MapForce (dans core | file path functions), 656

#### get-folder,

en tant que fonction MapForce (dans core | file path functions), 657

#### greater,

en tant que fonction MapForce (dans core | logical functions), 665

#### group-adjacent,

en tant que fonction MapForce (dans core  $\mid$  sequence functions), 687

#### group-by,

en tant que fonction MapForce (dans core | sequence functions), 689

#### group-ending-with,

en tant que fonction M apForce (dans core | sequence functions), 691

#### group-into-blocks,

en tant que fonction MapForce (dans core | sequence functions), 693

#### group-starting-with,

en tant que fonction MapForce (dans core | sequence functions), 695

#### GUI, 30

# Η

Hiérarchie, depuis des fichiers de texte, 451 hour-from-datetime, en tant que fonction MapForce (dans lang | datetime functions), 736 hour-from-duration, en tant que fonction MapForce (dans lang | datetime functions), 736 HRESULT, and error handling, 1226 HTML, documentation de mappage, 916 integration of MapForce, 1426 visionner la sortie de mappage en tant que, 912 HTML example, of MapForceControl integration, 1424, 1425, 1426

#### IBM DB2,

écrire dans les champs de type XML, 372 lecture depuis les champs de type XML, 372 se connecter par le biais de JDBC, 196 se connecter par le biais de ODBC, 198

**IBM DB2 for i,** se connecter par le biais d'JDBC, 204

IBM DB2 pour i,

se connecter par le biais d'ODBC, 205

IBM Informix,

se connecter par le biais de JDBC, 208

#### lcônes,

dans la fenêtre Messages de Requête de base de données, 394

dans la fenêtre Résultats de Requête de base de données, 394

#### implicit-timezone,

en tant que fonction MapForce (dans xpath2 | context functions), 782

Information de Copyright, 1568

Information juridique, 1568

Informations générales, 1565

#### Informations techniques, 1565

INNER JOIN,

dans les composants Join, 339

#### Insérer,

commentaire en bloc, 387

commentaire en ligne, 387

commentaires, 387 Composant SQL WHERE, 355

régions, 388

signets, 387

#### **Instructions de traitement,** ajouter au fichier cible, 132

Integrating,

MapForce in applications, 1405

#### Intégration,

Avec les produits d'Altova, 28

#### Intégrer,

dans C#, 1098

dans C++, 1100

dans Java, 1096

#### Intégrer code MapForce, 1095

Interface utilisateur, 30

#### Internet usage,

dans les produits Altova, 1566

#### is-not-null,

en tant que fonction MapForce (dans db functions), 723 **is-null**,

#### s nun,

en tant que fonction MapForce (dans db functions), 723

### is-xsi-nil,

en tant que fonction MapForce (dans core | node functions), 675

### ltem,

manquant, 100

#### item-at,

en tant que fonction MapForce (dans core | sequence functions), 697

#### items manquants, 100

#### items-from-till,

en tant que fonction MapForce (dans core | sequence functions), 698

## J

#### Java, 1431

aux classes générées, 1184 code, 1065 Emplacement de la bibliothèque VM, 1050 éviter des exceptions dans le code généré, 1200

#### Java, 1431

générer du code de programme, 1081, 1087 intégrer du code généré, 1096

#### JavaScript,

error handling, 1226

#### JDBC,

configurer une connexion (Windows), 182 en tant qu'interface de connexion de données, 158 gérer des références dans du code généré, 1085 se connecter à Teradata, 242

#### Journalisation,

exceptions de base de données, 435, 440

#### JScript,

code-generation sample, 1241

## Langages de transformation, 25 last.

en tant que fonction MapForce (dans xpath2 | context functions), 782

#### last-items,

en tant que fonction MapForce (dans core | sequence functions), 699

#### leapyear,

en tant que fonction M ap<br/> Force (dans lang  $\mid$  date<br/>time functions), 737

#### left,

en tant que fonction MapForce (dans lang | string functions), 767

#### LEFT OUTER JOIN,

dans les composants Join, 339

#### left-trim,

en tant que fonction MapForce (dans lang | fonctions string), 767

#### less,

en tant que fonction MapForce (dans core | logical functions), 665

#### Licence, 1060, 1570

information à propos de, 1568

#### Licence de produit de logiciel, 1570

#### License metering,

des produits Altova, 1569

#### Lignes,

mapper depuis - fichiers de texte, 451

#### local-name-from-QName,

en tant que fonction MapForce (dans lang | QName functions), 679

#### log10,

en tant que fonction MapForce (dans lang | math functions), 754

#### logical-and,

en tant que fonction MapForce (dans core | logical functions), 666

#### logical-not,

en tant que fonction M apForce (dans core | logical functions), 666

#### logical-or,

en tant que fonction MapForce (dans core | logical functions), 667

#### logical-xor,

en tant que fonction MapForce (dans lang | logical functions), 748

#### lowercase,

en tant que fonction MapForce (dans lang | string functions), 768

## Μ

#### main-mfd-filepath,

en tant que fonction MapForce (dans core | file path functions), 657 MapForce,

aperçu, 23 API, 1221 integration, 1405

#### MapForce API, 1221

MapForce integration,

example of, 1424, 1425, 1426

#### MapForce Server,

compiler des mappages pour, 957 lancer des exceptions depuis, 547 mappages automatisés, 950 Ressources globales dans, 989

#### MapForceCommand,

in MapForceControl, 1453 MapForceCommands, in MapForceControl, 1455

#### MapForceControl, 1456

documentation of, 1405 example of integration at application level, 1424, 1425, 1426 examples of integration at document level, 1417

integration using C#, 1417

integration using HTML, 1426

integration using Visual Basic, 1441

object reference, 1453

#### MapForceControlDocument, 1464 MapForceControlPlaceHolder, 1470 Mappage,

bases, 74 composants, 74 connecteurs, 74 connexions, 74 créer, 74 débogage, 926 Documenter, 916 format fichier plat, 446 notions fondamentales, 74 orienté vers la source - contenu mixte, 91 paramètres, 112 parties, 74 termes, 74 terminologie, 74 valider, 103

#### Mappages,

traitement automatisé, 950

#### MariaDB.

se connecter par ODBC, 210

#### match-pattern,

en tant que fonction MapForce (dans lang | string functions), 769

#### max,

en tant que fonction MapForce (dans core | aggregate functions), 636

en tant que fonction MapForce (dans lang | math functions), 754

#### max-string,

en tant que fonction MapForce (dans core | aggregate functions), 636

#### MERGE.

en tant qu'instruction dans le SQL généré par MapForce, 298

en tant qu'instruction dans SQL généré par MapForce, 305

#### Messages,

fenêtre- Requête de base de données, 394

icônes dans Requête de base de données, 394

#### mfd-filepath.

en tant que fonction MapForce (dans core | file path functions), 657

#### mff,

fichier bibliothèque, 612 fichier mff.xsd, 612

#### Microsoft Access,

se connecter par le biais d'ADO, 165, 212

Microsoft SharePoint Server,

ajouter des fichiers en tant que composants depuis, 78

#### Microsoft SQL Server,

se connecter par le biais d'ODBC, 217

se connecter par le biais d'ADO, 215

#### millisecond-from-datetime,

en tant que fonction MapForce (dans lang | datetime functions), 737

#### millisecond-from-duration,

en tant que fonction MapForce (dans lang | datetime functions), 738

#### min.

en tant que fonction MapForce (dans core | aggregate functions), 637 en tant que fonction MapForce (dans lang | math functions), 755

#### min-string,

en tant que fonction MapForce (dans core | aggregate functions), 638

#### minute-from-datetime,

en tant que fonction MapForce (dans lang | datetime functions), 738

#### minute-from-duration,

en tant que fonction MapForce (dans lang | datetime functions), 739

#### Mise en page,

Navigateur, 389

#### Mixte.

mappage de contenu, 91 mappage orienté vers la source, 91

#### Mode.

Afficher Annotations, 1041 Afficher astuces, 1041 Afficher la Bibliothèque dans l'en-tête Fonction, 1041 Afficher les connecteurs de composant sélectionnés, 1041 Afficher les connexions depuis la source vers la cible, 1041 Afficher Types, 1041 Aperçu, 1041 Barre de statut, 1041 Bibliothèques, 1041 commande de menu, 1041 Fenêtre de projet, 1041 Fenêtres Débogage, 1041

Gérer les Bibliothèques, 1041

Messages, 1041

Options d'affichage XBRL, 1041 Précédent, 1041

#### Suivant, 1041

Zoom, 1041

#### Mode Texte,

© 2016-2022 Altova GmbH

#### Mode Texte, negative, chercher, 109 en tant que fonction MapForce (dans lang | logical functions), 748 coloration syntaxique, 106 node-name. guides de retrait, 106 En que fonction MapForce (dans xpath2 | accessors library), marge pliable, 106 777 marquage du texte, 106 en tant que fonction MapForce (dans core | node functions), marqueurs d'espace blanc, 106 676 marqueurs de fin de ligne, 106 Noms de fichier, numérotation des lignes, 106 fournir en tant que paramètres d'entrée de mappage, 899 pliage de source, 106 Noms de méthode lors de la génération de code, pretty-printing, 106 réserver, 1200 signets, 106 Noms de nœud. word-wrap, 106 mapper des données depuis/vers, 851 zoomer, 106 normalize-space, modèle d'objet, en tant que fonction MapForce (dans core | string aperçu, 1225 functions), 711 modulus, not-equal, en tant que fonction MapForce (dans core | math functions), en tant que fonction MapForce (dans core | logical 671 functions), 668 month-from-datetime, not-exists, en tant que fonction MapForce (dans lang | datetime en tant que fonction MapForce (dans core | sequence functions), 739 functions), 700 month-from-duration, Nouvelles fonctions, 18 en tant que fonction MapForce (dans lang | datetime Version 2019, 21 functions), 740 Version 2020, 20 Moteurs Altova, Version 2021, 19 dans les produits Altova, 1565 Version 2022, 18 MSXML, Version 2023, 18 générer du code pour, 1201 now, msxsl:script, 1562 en tant que fonction MapForce (dans lang | datetime multiply, functions), 740 en tant que fonction MapForce (dans core | math functions), NULL, 672 attribut, 128 MySQL, gérer dans les mapapges de base de données, 327 se connecter par le biais d'ODBC, 223 valeurs, 128 valeurs dans les bases de données, 128 Nulls, N gérer dans les composants de base de données, 280 number, en tant que fonction MapForce (dans core | conversion namespace-uri-form-QName, functions), 649 en tant que fonction MapForce (dans lang | QName numeric, functions), 680

en tant que fonction MapForce (dans lang | logical functions), 749

appliquer des filtres, 390 filtrer, 390 Requête de base de données, 389

Naviguer,

Navigateur,

signets, 387

## 0

Objets de base de données, Package Oracle, ajouter au mappage, 254 procédures stockées et fonctions, 399 filtre, 254 pad-string-left, supprimer du mappage, 254 en tant que fonction MapForce (dans lang | string functions), 769 Objets globaux, pad-string-right, dans SPL, 1207 en tant que fonction MapForce (dans lang | string functions), ODBC. 770 configurer une connexion, 178 Paires Key-value, en tant qu'interface de connexion de données, 158 utiliser dans le mappage, 530 se connecter à MariaDB, 210 Paramètres, se connecter à Teradata, 244 fournir dans le mappage, 477, 482 OLE DB, pour l'opération de service web, 112 en tant qu'interface de connexion de données, 158 pour les fichiers de sortie, 112 OpenJDK, pour les mappages, 112 en tant que Java Virtual Machine, 182 Paramètres de génération de code, Oracle, définir globalement pour le projet entier, 118 écrire dans les champs de type XML, 372 parse-date, lecture depuis les champs de type XML, 372 en tant que fonction MapForce (dans core | conversion ordre de tri, functions), 650 modifier, 517 parse-dateTime, trier le composant, 517 en tant que fonction MapForce (dans core | conversion Orienté vers la source, functions), 651 mappage de contenu mixte, 91 parse-number, Outils. en tant que fonction MapForce (dans core | conversion commande de menu, 1043 functions), 654 parse-time. Configuration active, 1043 en tant que fonction MapForce (dans core | conversion Créer mappage inversé, 1043 functions), 655 Gestionnaire de taxonomies XBRL, 1043 PDF, Options, 1043 documentation de mappage, 916 Personnaliser, 1043 visionner la sortie de mappage en tant que, 912 Ressources globales, 1043 Période d'évaluation, Restaurer les barres d'outils et les fenêtres, 1043 des produits logiciels Altova, 1568 Outils | Options, des produits logiciels d'Altova, 1568 Base de données, 1048 Personnaliser, Débogueur, 1048 Clavier, 1045 Édition, 1048 commandes, 1044 Généralités, 1048 Menu par défaut vs. MapForce Design, 1044 Génération, 1048 menus, 1044 Java. 1048 menus contextuels, 1044 Messages, 1048 ombres de menu. 1044 Proxy de réseau, 1048 raccourcis, 1045 XBRL, 1048 réinitialiser la barre de menu, 1044

P
Personnaliser, se connecter par le biais d'ODBC, 234 supprimer commandes, 1044 pow, pi, en tant que fonction MapForce (dans lang | math functions), 756 en tant que fonction MapForce (dans lang | math functions), Prise en charge Unicode, 755 Pilotes de base de données, dans les produits Altova, 1566 Procédures stockées, Aperçu, 162 Pilotes ODBC, appel depuis un mappage, 396, 399, 402, 405, 409, 414, 419 vérifier la disponibilité de, 178 notes de prise en charge, 396 Plateformes, Processeur XQuery, pour les produits Altova, 1565 dans les produits Altova, 1565 Plug-in MapForce pour Eclipse, **Processeurs XSLT**, dans les produits Altova, 1565 à propos de, 1004, 1007, 1013 Projet, accéder aux menus et aux fonctions communs, 1009 configurer pour génération de code automatique, 1020 ajouter des mappages à, 115 créer de nouveaux mappages, 1015 bases, 115 chercher, 115 créer un projet MapForce/Eclipse, 1013 en cours d'installation, 1005 créer, 115 étendre les fonctions, 1023 dossiers, 118 importer des mappages existants dans un projet Eclipse, fermer, 115 1017 génération de code, 115 passer à la perspective MapForce, 1007 nouveauté, 115 point d'extension, 1023 organisation, 115 travailler avec des mappages et des projets, 1013 ouvrir, 115 Plug-in MapForce pour Visual Studio, paramètres, 118 à propos de, 1001 paramètres de génération de code, 118 activer, 1001 prévisualiser l'image, 115 information sur les menus et les fonctions, 1001 propriétés, 118 travailler avec des mappages et des projets, 1001 supprimer, 115 Plug-in Visual Studio, Projets, 115 exécuter MapForce en tant que, 1001 Ajouter des fichiers au projet, 1032 Point d'interrogation, Ajouter fichier actif au projet, 1032 items manquants, 100 Ajouter un fichier de mappage pour l'opération, 1032 Points d'arrêt, Créer dossier, 1032 à propos de, 931 Créer le mappage pour l'opération, 1032 ajouter, 935 Enregistrer, 1032 supprimer, 935 Fermer, 1032 Poitn de code. Générer code dans, 1032 collation, 517 Générer le Code pour tous le projet, 1032 position, Insérer Service Web, 1032 en tant que fonction MapForce (dans core | sequence Ouvrir le fichier dans XMLSpy, 1032 functions), 700 Ouvrir le mappage, 1032 Position de débogueur, Propriétés, 1032 consulter la valeur actuelle de, 937 Recharger, 1032 positive, Proxy de réseau, en tant que fonction MapForce (dans lang | logical automatique, 1056 functions), 749 configuration, 1056 PostgreSQL, manuel, 1056 se connecter nativement, 187 © 2016-2022 Altova GmbH

## Proxy de réseau,

paramètres, 1056 système, 1056

## Q

#### QName,

en tant que fonction MapForce (dans lang | QName functions), 679

#### QName-as-string,

en tant que fonction MapForce (dans lang | QName functions), 759

#### Quote character,

in CSV files, 455

# R

### radians,

en tant que fonction MapForce (dans lang | math functions), 756

random,

en tant que fonction MapForce (dans lang | math functions),  $756\,$ 

## RaptorXML Server,

exécuter une transformation, 949

#### read-binary-file,

en tant que fonction MapForce (dans lang | file functions), 745

## Recherche,

fichiers dans la fenêtre de projet, 115 items dans les composants de mappage, 80 projets, 115

#### Réduire,

régions, 388

## Référence C#,

aux classes générées, 1168

## Référence C++,

aux classes générées, 1153

## Référence de menu, 1024

## Référence d'icône de composant, 76

Régions,

agrandir, 388 créer, 388 insérer, 388 réduire, 388 supprimer, 388 définir dans des mappages, 270

préserver dans les mappages, 260

## Relations locales,

et procédures stockées, 412, 414, 419

## remove-fileext,

en tant que fonction M ap<br/>Force (dans core  $\mid$  file path functions), 658

## remove-folder,

en tant que fonction MapForce (dans core | file path functions), 658

## remove-timezone,

en tant que fonction MapForce (dans lang | datetime functions), 741

#### repeat-string,

en tant que fonction MapForce (dans lang | string functions), 771

#### replace,

en tant que fonction MapForce (dans lang | string functions), 771

#### replace-fileext,

en tant que fonction M apForce (dans core | file path functions), 659

## replicate-item,

en tant que fonction MapForce (dans core | sequence functions), 703

## replicate-sequence,

en tant que fonction MapForce (dans core | sequence functions), 705

## Requête de base de données,

décommenter le texte, 387 filtrer des tables, 390 générer SQL, 389, 394

régions, 388

## signets, 387

## resolve-filepath,

en tant que fonction MapForce (dans core | file path functions), 659

## resolve-uri,

En tant que fonction MapForce (dans xpath2 | any URI functions), 778

#### Ressources globales,

bases de données comme, 982 configuration, 972 créer, 972 dans des environnements d'exécution, 988 dans FlowForce Server, 989 déployer sur FlowForce Server, 960, 989 dossiers comme, 980 Fichier de définition, 972

#### Ressources globales, données de table - Requête de base de données, 394 Fichiers XML en tant que, 978 Séquence, 873 Introduction à, 971 Serveur SQL, Résultats. se connecter par le biais d'ADO, 165 fenêtre-Requête de base de données, 394 se connecter par le biais d'ADO.NET, 171 se connecter via JDBC, 182 icônes dans Requête de base de données, 394 reversefind-substring, set-empty. en tant que fonction MapForce (dans lang | string functions), en tant que fonction MapForce (dans core | sequence 772 functions), 706 right, set-null, en tant que fonction MapForce (dans lang | string functions), en tant que fonction MapForce (dans db functions), 724 773 set-xsi-nil, right-trim, en tant que fonction MapForce (dans core | node functions), en tant que fonction MapForce (dans lang | string functions), 677 773 Signets, round, insérer, 387 en tant que fonction MapForce (dans core | math functions), marge du signet, 387 672 naviguer, 387 round-half-to-even, supprimer, 387 en tant que fonction MapForce (dans xpath2 | numeric sin, functions), 806 round-precision, 757 en tant que fonction MapForce (dans core | math functions), skip-first-items, 673 RTF. functions), 706 documentation de mappage, 916 Sortie, visionner la sortie de mappage en tant que, 912 aperçu, 103 C#, 1038 C++, 1038 S enregistrer, 103 Scénarios de mappage, 26 Schema. génération de Code, 1065 Java, 1038 Schéma, générer pour un fichier XML, 121 schemanativetype, 1204 Régénérer Sortie, 1038 Scripts dans XSLT/XQuery, Signet précédent, 1038 voir sous Fonctions d'extension, 1561 Signet suivant, 1038 SE. pour les produits Altova, 1565 second-from-datetime, valider. 103 en tant que fonction MapForce (dans lang | datetime functions), 741 second-from-duration, XQuery, 1038 en tant que fonction MapForce (dans lang | datetime XSLT 1.0, 1038 functions), 742

Sélection de thème pour MapForce dans Eclipse, 1009 Sélectionner,

en tant que fonction MapForce (dans lang | math functions),

en tant que fonction MapForce (dans core | sequence

Enregistrer le fichier de sortie, 1038 Enregistrer tous les fichiers de sortie, 1038 Exécuter SQL/NoSQL-Script, 1038 Insérer/Supprimer signet, 1038 Moteur d'exécution Built-In, 1038 Paramètres du Mode Texte, 1038 Supprimer tous les signets, 1038 Texte XML Pretty-Print, 1038 Valider le fichier Sortie, 1038 XSLT 2.0, 1038 XSLT 3.0, 1038

## Sortie de mappage,

777

Sortie de mappage, en tant que fonction MapForce (dans core | conversion functions), 656 Générer plusieurs fichiers en tant que, 893, 898 parser des données depuis, 904 Source. 24 sérialiser des données dans, 904 SPL, 1203 sérialiser les données sur, 907 blocs de code, 1203 string-as-QName, conditions, 1211 en tant que fonction MapForce (dans lang | QName foreach, 1212 functions), 759 objets globaux, 1207 string-compare, sous-routines, 1213 en tant que fonction MapForce (dans lang | string functions), utiliser des fichiers, 1208 774 variables, 1206 string-compare-ignore-case, SQL. 385 en tant que fonction MapForce (dans lang | string functions), charger depuis des scripts, 384 775 écrire des instructions, 385 string-join, exécuter des instructions, 384, 386 en tant que fonction MapForce (dans core | aggregate functions), 639 exporter des instructions en tant que scripts SQL, 386 string-length, générer des instructions, 384, 385 importer des scripts SQL, 386 functions), 712 joindre des données, 339 Stylevision, SQL Server. écrire dans les champs de type XML, 372 lecture depuis les champs de base de données, 372 substitute-missing, lecture depuis les champs de type XML, 372 SQL WHERE, functions), 707 composant - insérer, 355 TRIER PAR, 355 SQL WHERE/ORDER, 678 en tant que composant MapForce, 359 substitute-null, SQLite. changer le chemin de base de données en un chemin absolu substring, dans le code généré, 85 écrire fichiers XML dans, 375 functions), 713 mapper des données dans, 907 substring-after, mapper les données dans, 333 sqrt, functions), 713 en tant que fonction MapForce (dans lang | math functions), substring-before, 757 starts-with, functions), 714 en tant que fonction MapForce (dans core | string subtract, functions), 712 static-node-annotation, 673 en tant que fonction MapForce (dans core | node functions), sum, 677 static-node-name, functions), 640 en tant que fonction MapForce (dans core | node functions), 678 à propos de, 931 String, En que fonction MapForce (dans xpath2 | accessors library), Supprimer,

en tant que fonction MapForce (dans core | string créer des feuilles de style avec, 923 générer une documentation de mappage avec, 918 en tant que fonction MapForce (dans core | sequence substitute-missing-with-xsi-nil, en tant que fonction MapForce (dans core | node functions), en tant que fonction MapForce (dans db functions), 724 en tant que fonction MapForce (dans core | string en tant que fonction MapForce (dans core | string en tant que fonction MapForce (dans core | string en tant que fonction MapForce (dans core | math functions), En tant que fonction MapForce (dans in core | aggregate Superpositions de données, suppression de composant intelligent, 101

#### © 2016-2022 Altova GmbH

commentaire en ligne, 387 commentaires, 387 items manquants, 100 régions, 388 signets, 387

## Supprimer les noms de schéma des noms de table,

en tant qu'option de génération de code, 257

#### Sybase,

se connecter par le biais de JDBC, 240

## system-property,

en tant que fonction MapForce (dans xslt | xslt functions library), 838

## Т

## **Tables de consultation,** utiliser dans le mappage, 530

tan,

en tant que fonction MapForce (dans lang | math functions), 758

## Teradata,

se connecter par JDBC, 242 se connecter par ODBC, 244

## Texte,

fichiers - définir les champs de clé, 451 mapper des fichiers texte, 446

## time-from-datetime,

en tant que fonction MapForce (dans lang | datetime functions), 742

#### timezone,

en tant que fonction MapForce (dans lang | datetime functions), 743

## tokenize,

en tant que fonction MapForce (dans core | string functions), 715

## tokenize-by-length,

en tant que fonction MapForce (dans core | string functions), 717

## tokenize-regexp,

en tant que fonction MapForce (dans core | string functions), 720

## Traitement,

mappages automatisés, 950

Traiter les instructions et les commentaires, mappage, 91

Transaction de base de données,

utilisation dans des mappages, 423 Transactions de base de données, activer pour des procédures stockées, 409 Transformations. RaptorXML Server, 949 translate (dans core | string functions), en tant que fonction MapForce, 721 Trier. dans bases de données, 359 données dans la fenêtre de résultat, 394 icône de colonne dans la fenêtre Résultats. 394 tables Requête de base de données, 389 trier le composant, 517 trier la clé, trier le composant, 517 TRIER PAR, Composant SQL where, 355 true, en tant que fonction MapForce (dans xpath2 | boolean functions), 779 Type complexe, trier, 517 Type simple, trier, 517 Types, intégrés, 1216 Types de connexion, copier tout, 95 enfants correspondants, 93 mixte, 91 Orienté vers la cible avec contenu mixte, 91 orienté vers la cible par rapport à orienté vers la source, 91 orientée vers la cible, 90 orientée vers la source, 91

annulation, 429

## standard avec contenu mixte, 91 **Types dérivés**,

standard, 90

mapper vers/depuis, 126 xsi:type, 126

## U

## UDF,

et contexte de mappage, 877 unary-minus,

#### unary-minus,

en tant que fonction MapForce (dans lang | math functions), 758

#### Unicode,

collation point de code, 517

remplacer des caractères spéciaux, 278

## unparsed-entity-uri,

en tant que fonction MapForce (dans xslt | xslt functions library), 839

#### uppercase,

en tant que fonction MapForce (dans lang | string functions), 776

#### URI,

dans DTD, 121

## URI d'espace de nom,

DTD, 121

## URL,

ajouter des fichiers en tant que composants depuis, 78

Utiliser des noms d'objet relatifs au schéma par défaut, en tant qu'option, 257

## V

Valeurs par défaut, 560 créer, 562 Valider, Mapper le design, 103 sortie de mappage, 103 Value-Map, en tant que composant de mappage, 530 exemples, 533, 536 Variables, ajouter au mappage, 494 basé sur BD, 492 changer l'étendue de, 498 complexe, 492 dans SPL, 1206 exemples d'utilisation, 500, 501, 503 simple, 492 Variables d'environnement, ANT\_OPS, 1200 Visual Basic, error handling, 1226 integration of MapForce, 1441 Visual Studio, adding the MapForce ActiveX Controls to the toolbox, 1408

générer du code de mappage pour, 1069, 1074 générer du code pour, 1201

## Volets,

/XSLT, 36 Mappage, 36 Requête BD, 36 Sortie, 36 Sortie StyleVision, 36 XQuery, 36

## Vue Navigateur,

générer des instructions SQL, 392 options menu contextuel, 392

## W

## WebDAV Server, ajouter des fichiers en tant que composants depuis, 78 weekday,

crucy,

en tant que fonction MapForce (dans lang | datetime functions), 743

## weeknumber,

en tant que fonction MapForce (dans lang | datetime functions), 744

## WHERE,

Composant SQL WHERE, 355

## Windows,

Aperçu, 31 Bibliothèques, 31 Cascade, 1059 Dialogue Fenêtre, 1059 Gérer les Bibliothèques, 31 Mappage, 31 Messages, 35 Mosaïque horizontale/verticale, 1059 Plusieurs fenêtres de mappage, 31 prise en charge pour les produits Altova, 1565 Projet, 31 Thème, 1059 Thème clair, 1059 Thème classique, 1059 Thème sombre, 1059 Word. documentation de mappage, 916 Word 2007+. visionner la sortie de mappage en tant que, 912

## write-binary-file,

## write-binary-file,

en tant que fonction MapForce (dans lang | file functions), 746

# X

## Xerces,

générer du code pour, 1201

## XML,

déclaration, 122 écrire dans le champ de base de données, 375 en tant que cible de mappage, 446 mapper des données depuis CSV vers, 446 paramètres d'encodage, 122 Paramètres de composant, 122

version de schéma, 112

signature numérique, 122

## XMLSpy command table :, 1444

## XQuery,

ajouter des fonctions personnalisées, 599, 600 Fonctions d'extension, 1561 importer modules, 600

xs:any, 134

## xs:anyAttribute, 134

## xsi:nil,

en tant qu'attribut dans l'instance XML, 128

## XSLT,

ajouter des fonctions personnalisées, 593 espace de nom de modèle, 593 Fonctions d'extension, 1561 supprimer des fonctions personnalisées, 593

## Y

year-from-datetime, en tant que fonction MapForce (dans lang | datetime functions), 744

#### year-from-duration,

en tant que fonction MapForce (dans lang | datetime functions), 745

# Ζ

## ZàA,

trier le composant, 517 **zip64mode**, activer dans le fichier build.xml, 1200